

Cheienne Chaves

*Um estudo sobre heurísticas residuais para o
problema de corte de estoque bidimensional*

Dissertação de Mestrado

Cheienne Chaves

Um estudo sobre heurísticas residuais para o problema de corte de estoque bidimensional

Dissertação apresentada como parte dos requisitos para obtenção do título de Mestre em Matemática, junto ao Programa de Pós-Graduação em Matemática, do Instituto de Biociências, Letras e Ciências Exatas da Universidade Estadual Paulista “Júlio de Mesquita Filho”, Câmpus de São José do Rio Preto.

Financiadora: CAPES.

Orientadora: Profa. Dra. Maria do Socorro Nogueira Rangel

São José do Rio Preto

2019

C512e

Chaves, Cheienne

Um estudo sobre heurísticas residuais para o problema de corte de estoque bidimensional / Cheienne Chaves. -- São José do Rio Preto, 2019

88 f. : il., tabs.

Dissertação (mestrado) - Universidade Estadual Paulista (Unesp), Instituto de Biociências Letras e Ciências Exatas, São José do Rio Preto

Orientadora: Socorro Rangel

1. Problema de corte de estoque bidimensional. 2. Heurística Residual. 3. Padrão de Corte 2-estágios. 4. Caso exato. 5. Caso não exato. I. Título.

Sistema de geração automática de fichas catalográficas da Unesp. Biblioteca do Instituto de Biociências Letras e Ciências Exatas, São José do Rio Preto. Dados fornecidos pelo autor(a).

Essa ficha não pode ser modificada.

Cheienne Chaves

Um estudo sobre heurísticas residuais para o problema de corte de estoque bidimensional

Dissertação apresentada como parte dos requisitos para obtenção do título de Mestre em Matemática, junto ao Programa de Pós-Graduação em Matemática, do Instituto de Biociências, Letras e Ciências Exatas da Universidade Estadual Paulista “Júlio de Mesquita Filho”, Câmpus de São José do Rio Preto.

Financiadora: CAPES.

Banca Examinadora

Profa. Dra. Maria do Socorro Nogueira Rangel

São José do Rio Preto

Orientadora

Prof. Dr. Silvio Alexandre de Araujo

UNESP - São José do Rio Preto

Profa. Dra. Kelly Cristina Poldi

UNICAMP - Campinas

São José do Rio Preto,
26 de Fevereiro de 2019.

A mim mesma.
Dedico.

Agradecimentos

À minha orientadora, Socorro, que acreditou em mim durante esse período, me apoiou, me ajudou, me fez evoluir como pessoa, estudante, pesquisadora e que por fim depositou muita confiança nesse trabalho.

A minha mãe Ana e à minha tia Marilu por todo o suporte e ajuda dada nos meus estudos ao longo da vida, por sempre terem oferecido palavras de carinho e apoio, que me deram forças para concluir meu sonho.

Aos amigos que fiz durante esse período, Gino, Junior, Paula, Gislaine, Alessandra, Bruno, Heloísa, Kumon, Otávio, Anna, André, Gustavo que foram essenciais ao meu desenvolvimento e tiveram do meu lado o tempo todo, proporcionando palavras de apoio, abraços, gordices, churrascos e muito estudo. Em especial, ao Eduardo que se tornou uma pessoa muito importante na minha vida, me ajudou literalmente em todos os momentos.

Aos Strangers que me mostraram o verdadeiro significado de família, que aguentaram meus surtos, loucuras, mesmo de longe sempre me apoiaram, sempre me lembraram do meu potencial e mais do que ninguém me faziam sorrir e ter forças pra continuar. Em especial aos meus Strangers que tiveram comigo desde o começo dessa caminhada, Barufinho, Vini e Thom, obrigada por tudo, amo muito todos vocês.

Aos professores que contribuíram diretamente e indiretamente com esse trabalho, professor Silvio, Luci, Andréa, Ranga e Cláudio Buzzi.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

"Welcome to the real world. It sucks.

You're gonna love it."

Friends

Resumo

Neste trabalho apresentamos o problema de corte de estoque (PCE) e algumas de suas características e propriedades. Abordamos com mais detalhes o problema de corte de estoque bidimensional, guilhotinado, 2– estágios, exato e não-exato. Exibimos o modelo matemático do PCE e o método de geração de colunas proposto por Gilmore e Gomory na década de 60. Apresentamos duas heurísticas residuais da literatura e sugerimos mudanças no método da ordenação dos padrões de corte de uma delas. Implementamos todo o algoritmo das heurísticas residuais utilizadas, realizamos testes computacionais usando três conjuntos de instâncias da literatura. As instâncias do Conjunto 1 foram geradas aleatoriamente e as dos Conjuntos 2 e 3 foram baseados em dados reais de uma fábrica de móveis. Os resultados obtidos com as heurísticas residuais foram comparadas com resultados disponíveis na literatura e resultados do Sistema CorteBiFur e do método branch-and-cut disponível no Cplex. Os resultados do estudo computacional indicam que a heurística residual estudada se mostra uma boa alternativa para ser implementada no sistema CorteBiFur.

Palavras Chave: Problema de corte de estoque bidimensional. Heurística Residual. Padrão de corte 2– estágios. Problema exato. Problema não exato.

Abstract

In this research we presented the cutting stock problem (CSP), and some of its features and properties. The focus is the two-dimensional cutting stock problem, considering guillotine cuts and, two-staged cutting patterns and the exact and non-exact cases. We exhibit a mathematical model for the CSP and a method for its solution proposed by Gilmore and Gomory in the 1960's. We present two residual heuristics from the literature and we suggest some changes on the criterion of sorting the cutting patterns in one of them. We implemented all the heuristics used in this research. We performed computational tests with three sets of instances available in the literature. The instances of the Set 1 were randomly generated and the instances of the Sets 2 e 3 were based on real data from a furniture industry. The results obtained with the residuals heuristics were compared with results available in the literature, with results from the Software CorteBiFur and the branch-and-cut method available on Cplex. The results indicated that the residual heuristic we studied its a good choice to be implemented on the CorteBiFur.

Keywords: Two-dimensional cutting stock problem. Residual Heuristic. Two-staged pattern. Exact problem. Non-exact problem.

Lista de Figuras

1.1	Distribuição dos empregos nas indústrias moveleiras.	p. 16
1.2	Distribuição das indústrias moveleiras nos estados brasileiros.	p. 17
2.1	Fluxograma com a classificação do PCE2D.	p. 21
2.2	Padrão de corte guilhotinado e não guilhotinado.	p. 22
2.3	Padrão de corte homogêneo e padrão de corte homogêneo maximal.	p. 23
2.4	Padrão de corte 2–estágios e Padrão de corte 3–estágios.	p. 24
2.5	Padrão de corte exato e Padrão de corte não exato.	p. 24
3.1	Padrões de Corte 1 e 2 usados no Exemplo 2 com a HR1.	p. 52
3.2	Padrões de Corte 3 e 4 usados no Exemplo 2 com a HR1.	p. 52
3.3	Padrão de Corte 5 usado no Exemplo 2 com a HR1.	p. 53
3.4	Padrões de Corte 1 e 2 usados no Exemplo 3 com a HR2.	p. 56
3.5	Padrões de Corte 3 e 4 usados no Exemplo 3 com a HR2.	p. 57

Lista de Tabelas

2.1	Dados do Problema.	p. 39
3.1	Solução inteira para o x_2^1	p. 49
3.2	1° solução inteira para o x_3^1	p. 50
3.3	2° solução inteira para o x_3^1	p. 50
3.4	3° solução inteira para o x_3^1	p. 50
3.5	1° solução inteira para o x_2^2	p. 51
3.6	2° solução inteira para o x_2^2	p. 51
3.7	Resultados obtidos com a heurística residual 1.	p. 52
3.8	Arredondamento de acordo com a HR2.	p. 55
3.9	Resultados obtidos com a heurística residual 2.	p. 56
4.1	Conjunto 1 de instâncias.	p. 59
4.2	Conjunto 2 de instâncias.	p. 59
4.3	Conjunto 3 de instâncias (lote duplo).	p. 60
4.4	Descrição da diversidade das dimensões dos itens para as instâncias do Conjunto 1.	p. 60
4.5	Descrição da diversidade das dimensões dos itens para as instâncias do Conjunto 2.	p. 61
4.6	Descrição da diversidade das dimensões dos itens para as instâncias do Conjunto 3.	p. 61
4.7	Resumo dos Testes de 1 a 4.	p. 62
4.8	Teste 1 : definição da melhor estratégia de ordenação para a HR1 para o caso exato.	p. 65

4.9	Teste 1 : definição da melhor estratégia de ordenação para a HR2 para o caso exato.	p. 66
4.10	Teste 3 : resultados das instâncias do Conjunto 1 para o caso exato.	p. 68
4.11	Teste 4 : resultados das instâncias do Conjunto 2 para o caso exato.	p. 70
4.12	Teste 4 : resultados das instâncias do Conjunto 3 para o caso exato.	p. 72
4.13	Teste 1 : definição da melhor estratégia de ordenação para a HR1 para o caso não exato.	p. 74
4.14	Teste 1 : definição da melhor estratégia de ordenação para a HR2 para o caso não exato.	p. 75
4.15	Teste 3 : resultados das instâncias do Conjunto 1 para o caso não exato.	p. 76
4.16	Teste 4 : resultados das instâncias do Conjunto 2 para o caso não exato.	p. 78
4.17	Teste 4 : resultados das instâncias do Conjunto 3 para o caso não exato.	p. 80

LISTA DE ABREVIATURAS E SIGLAS

ABIMOVEL Associação Brasileira das Indústrias do Mobiliário

FO Função Objetivo

GC Geração de Colunas

IEMI Inteligência de Mercado

PCE Problema de Corte de Estoque

PCE1D Problema de Corte de Estoque Unidimensional

PCE2D Problema de Corte de Estoque Bidimensional

PCI Padrão de Corte Irrestrito

PCR Padrão de Corte Restrito

PM Problema Mestre

r Demanda residual

RAIS Relação Anual de Informações Sociais

SCOOP *Scientific Computation of Optimal Programs*

Sumário

1	INTRODUÇÃO	p. 15
2	O PROBLEMA DE CORTE DE ESTOQUE E MÉTODOS DE SOLUÇÃO	p. 19
2.1	O problema de corte de estoque	p. 19
2.2	Características dos padrões de corte	p. 21
2.2.1	Métodos para geração de um padrão de corte	p. 24
2.2.2	Modelagem matemática e métodos de solução	p. 29
2.2.3	Método simplex	p. 30
2.2.4	Decomposição de Dantzig-Wolfe	p. 34
2.2.5	Enumeração implícita e a geração de colunas	p. 41
2.2.6	Heurísticas construtivas	p. 42
2.2.7	Algoritmo genético	p. 42
3	HEURÍSTICAS RESIDUAIS	p. 44
3.1	O problema residual	p. 44
3.2	Heurísticas residuais por truncamento - Cintra <i>et al</i> (2008)	p. 46
3.3	Heurística residual 1 (HR1 - Poldi e Arenales (2006)	p. 46
3.4	Heurística residual 2 (HR2) - Cui e Zhao (2013)	p. 53
3.5	Contribuição: modificações para a HR2	p. 57
4	MÉTODO COMPUTACIONAL	p. 58
4.1	Descrição das instâncias	p. 58
4.2	Testes computacionais realizados e parâmetros para análise dos resultados	p. 61

4.3	Resultados e discussão: caso exato	p. 64
4.4	Resultados e discussão: caso não exato	p. 74
5	CONCLUSÃO	p. 82
	REFERÊNCIAS	p. 84

Capítulo 1

INTRODUÇÃO

Durante a segunda guerra mundial a pesquisa operacional foi usada para resolver problemas como: controle de artilharia antiaérea, dimensionamento de comboios de frota, escolha do avião para uma missão, e manutenção e inspeção de aviões. Após o término da guerra, em 1947, foi implantado o programa *Scientific Computation of Optimal Programs* (SCOOP) no Pentágono. Um dos membros do grupo era George Dantzig, que durante sua participação no SCOOP, desenvolveu o método simplex para resolver problemas de programação linear. Antes de Dantzig, Leonid Kantorovich fez alguns desenvolvimentos nessa área, mas seus trabalhos ficaram conhecidos fora da antiga União Soviética somente na década de 60, quando eles foram publicados. Nessa mesma década a pesquisa operacional iniciou no Brasil, por meio de um simpósio (ARENALES et al., 1998).

A importância da pesquisa nesse campo se justifica, pois, apesar da Pesquisa Operacional ser recente ao se comparar com outras áreas matemáticas, ela pode ser aplicada a diversas áreas de produção e logísticas, de indústrias e organizações de serviço, como, turismo, transporte, trânsito, bancos, coleta de lixo, bibliotecas, educação, energia, papel e papelão, petróleo, saúde, entre tantos outros. Nesse trabalho abordamos uma indústria em particular, a indústria moveleira.

O setor moveleiro brasileiro ocupa o quinto lugar no ranking mundial, com um faturamento de R\$38 bilhões por ano, de acordo com um levantamento da Associação Brasileira da Indústria Moveleira (ABIMOVEL) (IEMI, 2018). A indústria moveleira está entre os mais importantes segmentos da Indústria de Transformação no País, não só pela importância do valor da sua produção, mas também pela geração de empregos dentro da indústria nacional. De acordo com os dados da Relação Anual de Informações Socioeconômicas (RAIS) (BRASIL, 2017), em 2017 o setor moveleiro juntamente com o de colchões empregava 268,9 mil pessoas em 19,6 mil indústrias, sendo que a grande maioria desses estabelecimentos, aproximadamente

80% possuem até nove funcionários. Conforme (SPEROTTO, 2018) no ano de 2016, somente no estado de São Paulo, a indústria moveleira empregava 53.867 pessoas, o que corresponde a 22% dos empregos nacionais nesse setor, distribuídos em 4.036 indústrias que correspondem a 18,5% das indústrias de móveis no Brasil. As Figuras 1.1 e 1.2 apresentam a distribuição de empregos nos estados brasileiros e a distribuição das indústrias, respectivamente.

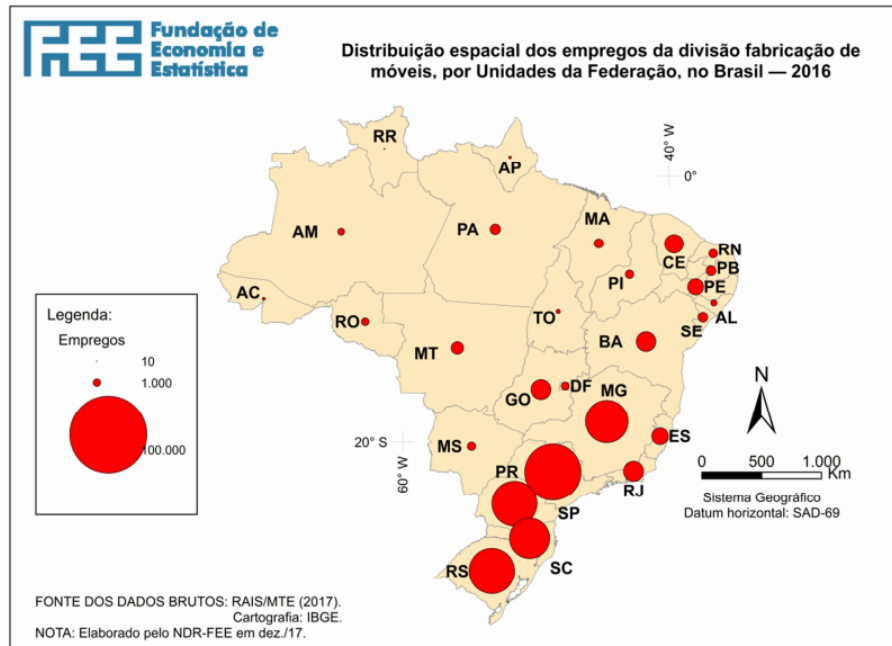


Figura 1.1: Distribuição dos empregos nas indústrias moveleiras.
Fonte: IBGE

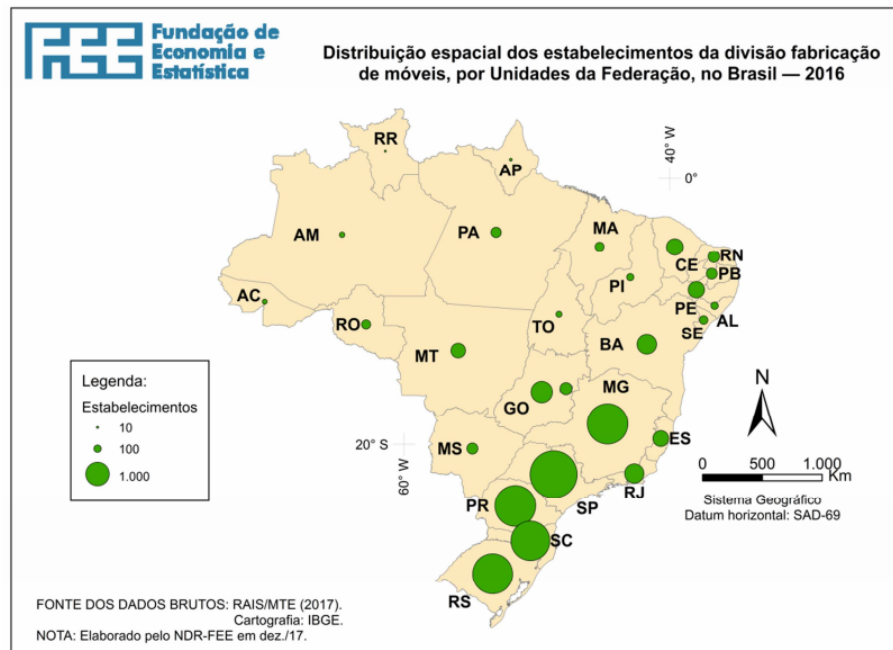


Figura 1.2: Distribuição das indústrias moveleiras nos estados brasileiros.
Fonte: IBGE

Estimativas do Inteligência de Mercado (IEMI) 2011 sugerem que aproximadamente 76% das empresas moveleiras do Brasil fabricam produtos de forma seriada, isto é, móveis padronizados, cujas características físicas não podem ser alteradas pelos consumidores. Assim, aprimorar o processo de produção desses móveis, seja em relação ao tempo gasto, ou à quantidade de matéria prima usada, entre outros, pode acelerar a tomada de decisão e trazer um lucro maior. Nesse trabalho focamos justamente no problema de minimizar os gastos com a matéria prima em fábricas que produzem móveis do tipo retilíneo (FIGUEIREDO; RANGEL, 2008). Nessas fábricas, uma etapa do processo de produção consiste em cortar painéis retangulares de madeira, guardados em estoque, em retângulos menores (itens) que irão compor um móvel. Esse problema é conhecido na literatura como Problema de Corte de Estoque Bidimensional. Uma das maneiras para resolver esse tipo de problema é utilizar softwares prontos, como, por exemplo, o Corte Certo, Otimize Nesting, Optimalon Software e o CorteBiFur (GASQUE; RANGEL, 2016).

Em particular, o sistema CorteBiFur (RANGEL et al., 2016) e (RANGEL, 2016), desenvolvido por Perin e Rangel em 1989, vem sendo aperfeiçoado a quase duas décadas com o objetivo de atender os requisitos da indústria de móveis. O objetivo do presente trabalho é contribuir com o desenvolvimento do sistema CorteBiFur estudando métodos para determinar uma solução inteira. Em particular, fazemos adaptações nas heurísticas residuais encontradas na literatura e

comparamos os resultados com o sistema CorteBiFur e com outros métodos encontrados na literatura.

No Capítulo 2 fazemos uma revisão sobre o Problema de Corte de Estoque, dos diferentes métodos para resolvê-lo, com um enfoque no método simplex com a geração de colunas proposta por Gilmore e Gomory (GILMORE; GOMORY, 1961), (GILMORE; GOMORY, 1963). No capítulo 3 apresentamos a definição da heurística residual, detalhamos duas heurísticas residuais encontradas na literatura, e propomos modificações. No Capítulo 4 exibimos os resultados do estudo computacional realizado para comparar heurísticas residuais apresentadas no Capítulo 3, o Sistema CorteBiFur e outras soluções encontradas na literatura. São discutidos quatro testes diferentes usando três conjuntos de instâncias, considerando duas situações o atendimento exato da demanda por retângulos menores (Caso exato) e a possibilidade de excedê-la (Caso não exato). Por fim, no Capítulo 5 são apresentadas as conclusões e trabalhos futuros.

Capítulo 2

O PROBLEMA DE CORTE DE ESTOQUE E MÉTODOS DE SOLUÇÃO

Nesse capítulo apresentamos uma introdução ao Problema de Corte de Estoque, algumas definições necessárias para resolvê-lo, os algoritmos do método simplex e do método simplex com geração de colunas. Esses algoritmos formam a base das heurísticas residuais discutidas no Capítulo 3.

2.1 O problema de corte de estoque

O Problema de Corte de Estoque (*PCE*) está entre os problemas mais antigos estudados pela Pesquisa Operacional (KANTOROVICH, 1960). Esse problema consiste em cortar objetos grandes, que estão em estoque, em peças menores, chamadas de itens, de acordo com algum critério de otimização. O critério usual é minimizar a quantidade de objetos utilizados. Mas outros critérios podem ser considerados, por exemplo, minimizar o número de padrões de corte (FILHO; MORETTI; PATO, 2017), minimizar os ciclos da serra (TOSCANO; RANGEL; YANASSE, 2015). Esses critérios de otimização podem ser tratados isoladamente (problema mono-objetivo) ou em conjunto (multi-objetivo). Esse problema está presente em diversas áreas, como, na indústria de papel (KALLRATH, 2014), vidro (TSAI; HSIEH; HUANG, 2009), madeira (BOUAINE; LEBBAR; MOHAMED, 2018), barras de metal (VIANNA; POLDI, 2005), barras de aço (TOKUYAMA; UENO, 1985), indústria de colchões (ALTIN et al., 2018), e afins.

O PCE aparece em três tipos principais de versões diferenciados entre si pelo número de

dimensões relevantes ao efetuar os cortes no objeto, podendo ser, de uma, duas ou três dimensões. Se apenas uma dimensão é relevante, o problema é chamado de Problema de Corte de Estoque Unidimensional (PCE1D) e somente o comprimento do objeto é levado em consideração. Para duas dimensões, além do comprimento, a largura também é considerada, e o problema é chamado de Problema de Corte Estoque Bidimensional (PCE2D). O caso tridimensional leva em consideração, além das dimensões citadas, a altura. Existem diversas situações que podem ser consideradas ao se resolver tais problemas, por exemplo, se existe somente um tamanho de objeto em estoque (SCHEITHAUER, 2018), (MUTER; SEZER, 2018), (MARTINOVIC; SCHEITHAUER; CARVALHO, 2018) ou vários (SCHEITHAUER, 2018), (MALAGUTI; DURÁN; TOTH, 2014), (CUI; CUY; YANG, 2014); a forma do objeto e dos itens (retangular, não retangular, regular, irregular) (YANXIN, 2013), (DALALAH; KHRAIS; BATAINEH, 2014), (VALLE et al., 2012); se os itens podem sofrer rotação (Problema não orientado) ou não (Problema orientado). Por esse motivo vários autores categorizaram os diferentes tipos de problemas, sendo as mais utilizadas a de Dyckhoff (DYCKHOFF, 1990) e Wäscher *et al* (WÄSCHER; HAUBNER; SCHUMANN, 2007). A Figura 2.1 ilustra a classificação do PCE2D baseado em (DYCKHOFF, 1990).

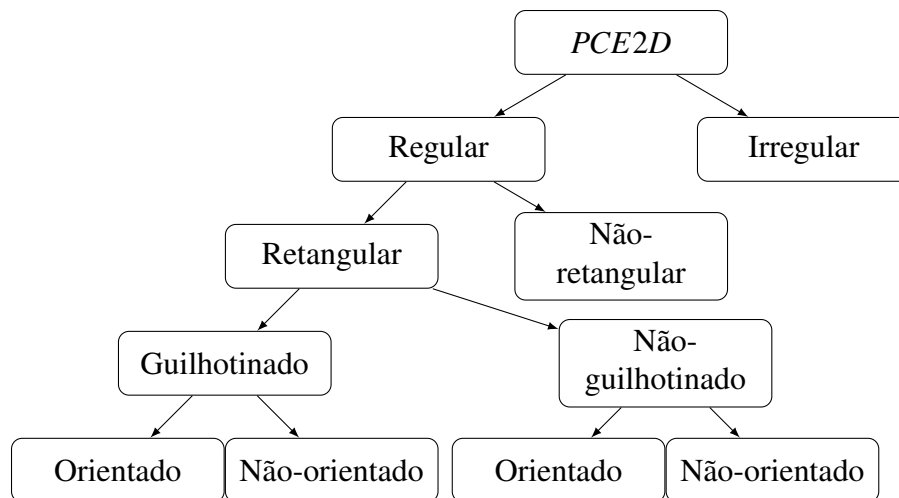


Figura 2.1: Fluxograma com a classificação do PCE2D.

Fonte: Suliman 2006

Definição 1. *O PCE2D consiste em dado um objeto retangular de comprimento (L) e largura (W), queremos minimizar a quantidade de objetos utilizados para obter m itens retangulares de comprimento l_i e largura w_i , atendendo a uma demanda inteira b_i , com $i = 1, \dots, m$*

Para resolver o PCE2D existem três abordagens principais. A primeira é por meio da programação inteira, que é uma abordagem exata, (LODI; MARTELLO; VIGO, 2004), (SILVA; ALVELOS; CARVALHO, 2010) e (MACEDO; ALVES; CARVALHO, 2010) abordaram esse método (Seção 2.2.5). O segundo método é por meio de heurísticas construtivas (CINTRA; WAKABAYASHI, 1998), (SALLAUME et al., 2008), (SULIMAN, 2006) (Seção 2.2.5). E o terceiro é o método da geração de colunas (GILMORE; GOMORY, 1961), (GILMORE; GOMORY, 1963), (GILMORE; GOMORY, 1965), (CINTRA et al., 2007), entre tantos outros (Seção 2.2.1 e 2.2.4). Utilizamos nesse trabalho a geração de colunas em problemas residuais (CUI; ZHAO, 2013), (MALAGUTI; DURÁN; TOTH, 2014) e (VIANNA; POLDI, 2005) (Capítulo 3).

2.2 Características dos padrões de corte

Um importante elemento para a modelagem matemática do PCE2D é a ideia de como os itens são organizados geometricamente no objeto dado, para isso apresentamos as definições a seguir.

Definição 2. *Chama-se **padrão de corte** à combinação geométrica de itens em um dado objeto dispostos dentro do objeto sem sobreposição e sem exceder o tamanho do objeto. O padrão de*

corte é representado por um vetor coluna $A_j \in \mathbb{N}^m$ sendo m o número de tipos de itens, assim, o padrão de corte j possui a_{ij} itens do tipo i .

Definição 3. Chama-se **padrão guilhotinado** o padrão de corte com apenas cortes guilhotinados, que consistem de cortes feitos ortogonalmente de uma extremidade a outra, tanto no sentido horizontal quanto no sentido vertical.

A Figura 2.2 ilustra um padrão com cortes guilhotinados e um padrão com cortes não guilhotinados. A área em cinza representa perda e os retângulos representam os itens demandados.

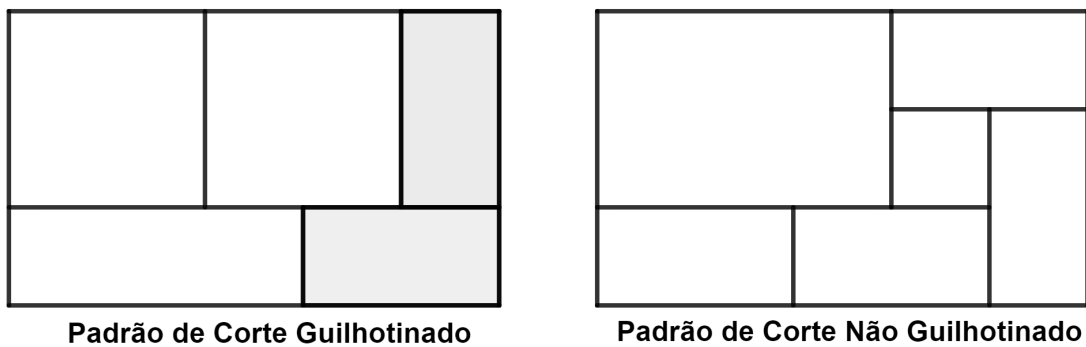


Figura 2.2: Padrão de corte guilhotinado e não guilhotinado.
Fonte: Elaborado pela autora.

Definição 4. Chama-se **padrão de corte restrito** um padrão de corte que para cada item i , a quantidade máxima do item i que pode-se ter no padrão é $r_i, i = 1, \dots, m$, isto é, $a_{ij} \leq r_i$. Caso contrário, o padrão de corte é dito irrestrito.

Definição 5. Chama-se **padrão de corte homogêneo** o padrão de corte que consiste apenas de um tipo de item, ou seja, o vetor $(0, 0, \dots, a_{ij}, \dots, 0)$. Chama-se **padrão de corte homogêneo maximal**, quando o número de vezes que o item aparece no padrão é o maior possível. Para o caso unidimensional irrestrito deve satisfazer, $a_{ij} = \lfloor \frac{L}{l_i} \rfloor$ e para o caso restrito $a_{ij} = \min \{ \lfloor \frac{L}{l_i} \rfloor, r_i \}$, sendo r_i um limitante superior na quantidade de itens do tipo i que podem ser alocados no padrão de corte. Para o caso bidimensional irrestrito, $a_{ij} = \lfloor \frac{L}{l_i} \rfloor \lfloor \frac{W}{w_i} \rfloor$ e para o restrito $a_{ij} = \min \{ \lfloor \frac{L}{l_i} \rfloor \lfloor \frac{W}{w_i} \rfloor, r_i \}$.

A Figura 2.3 apresenta um padrão de corte homogêneo e um padrão de corte homogêneo maximal. A área em cinza representa perda e os retângulos representam os itens demandados.

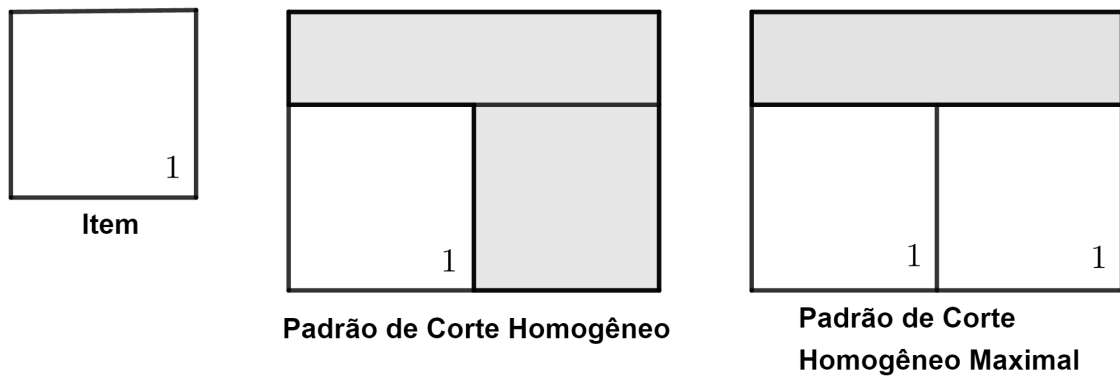


Figura 2.3: Padrão de corte homogêneo e padrão de corte homogêneo maximal.
Fonte: Elaborado pela autora.

Definição 6. Um conjunto (v_1, v_2, \dots, v_n) de vetores de um espaço vetorial V , é uma base de V se: (v_1, v_2, \dots, v_n) é linearmente independente e gera o conjunto V .

Definição 7. Chama-se de **base homogênea**, a base composta somente por padrões de corte homogêneos. Chama-se **base homogênea maximal**, a base composta somente por padrões de corte homogêneo maximal.

O padrão de corte deve atender também aos pré-requisitos de cada indústria, por exemplo, na indústria moveleira, devido ao tipo de serra utilizada, os cortes devem ser guilhotinados; o número de estágios é limitado (Definição 7) e o problema pode ou não permitir a rotação de itens.

Definição 8. Considera-se a **quantidade de estágios do padrão de corte**, o número de vezes que o objeto é rotacionado 90° para fazer um novo corte.

A Figura 2.4 ilustra um padrão 2-estágios e um 4-estágios. A área cinza representa perda e os retângulos itens demandados.

Definição 9. A perda de um padrão de corte é calculada para o caso unidimensional da seguinte forma $p_j = L - \sum_{i=1}^m l_i a_{ij}$, $j = 1, \dots, N$. e para o caso bidimensional $x_{ij} \sum_{i=1}^m l_i \times w_i$.

Podemos observar que quanto mais estágios um padrão de corte tiver, menor será o desperdício do objeto, mas por outro lado demandará uma mão-de-obra maior, demorando mais tempo para ser executado, diminuindo assim a produtividade da máquina. De acordo com (MALAGUTI; DURÁN; TOTH, 2014), no primeiro e segundo estágios não é preciso intervenção para ajustar a serra, assim costuma-se dar preferência aos padrões com no máximo três estágios.

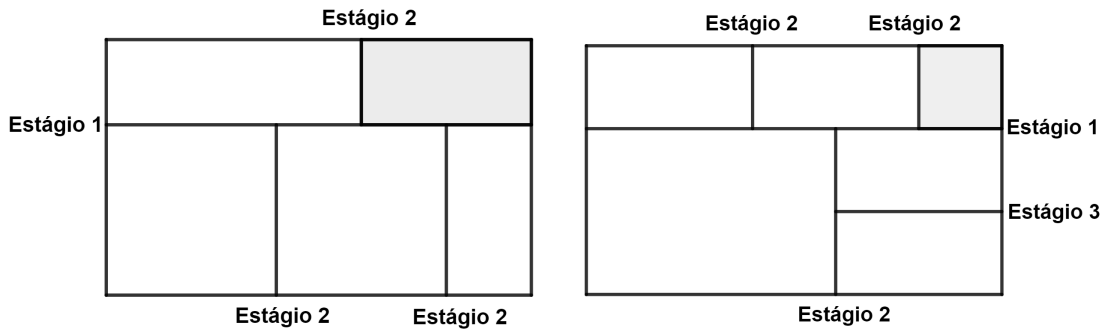


Figura 2.4: Padrão de corte 2—estágios e Padrão de corte 3—estágios.
Fonte: Elaborado pela autora.

Definição 10. *Após realizar um corte 2-estágios, se não for necessário uma apara adicional, ou seja, se todos os itens possuem a mesma largura em cada faixa o padrão é chamado de **padrão exato**, caso necessite da apara, o padrão é denotado por **padrão não exato**.*

A Figura 2.5 ilustra esses dois casos. A área cinza representa perda e os retângulos itens demandados.

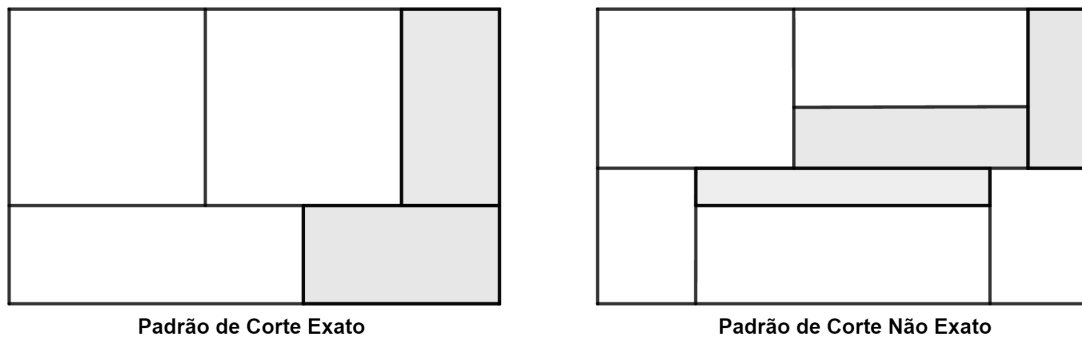


Figura 2.5: Padrão de corte exato e Padrão de corte não exato.
Fonte: Elaborado pela autora.

2.2.1 Métodos para geração de um padrão de corte

Nessa seção descrevemos alguns métodos encontrados na literatura para determinar padrões de unidimensional e bidimensional. No método proposto por Wang que foi desenvolvido para o caso bidimensional (WANG, 1983) inicialmente é feita uma lista com todos os itens disponíveis e os itens são alocados no padrão de forma que minimizem o percentual de perda, conforme um item é utilizado, ele é removido da lista. Os itens escolhidos não podem ultrapassar as

dimensões do objeto e exceder a demanda de cada item. O critério para decidir se o padrão é aceito ou não, é o percentual de perda do padrão.

Outra forma para obter padrões é utilizando o método de de Gilmore e Gomory (GILMORE; GOMORY, 1961) para o caso unidimensional e o método 2–estágios (GILMORE; GOMORY, 1963) e (GILMORE; GOMORY, 1965) para o caso bidimensional. O primeiro estágio consiste em criar faixas e em seguidas usá-las para formar um padrão. A seguir, explicamos como obter padrões de corte para os dois tipos de problemas.

Subproblema para o PCE1D

Associando o problema da mochila¹, com o subproblema do PCE1D, temos que o comprimento do objeto em estoque L é equivalente à capacidade da mochila, o vetor da utilidade dos itens é o π e finalmente o peso é o comprimento de cada item l_i . O subproblema está formulado em (2.1).

$$\begin{aligned} & \text{maximizar} && \pi^t z \\ & \text{sujeito a} && \sum_{i=1}^m l_i z_i \leq L, \\ & && z \in \mathbb{Z}_+^m, \end{aligned} \tag{2.1}$$

sendo m a quantidade de itens, L o comprimento do objeto, l_i o comprimento de cada item, π o vetor com as variáveis duais ótimas associadas ao PMR, e z_i o número de vezes que o item i aparece no padrão.

De acordo com a Definição 4 podemos utilizar outro tipo de padrão de corte, o restrito, que limitará a quantidade de vezes que os itens podem ser utilizados no padrão de corte. O subproblema restrito está apresentado no Modelo (2.2).

$$\begin{aligned} & \text{maximizar} && \pi^t z \\ & \text{sujeito a} && \sum_{i=1}^m l_i z_i \leq L, \\ & && z_i \leq b_i \\ & && z \in \mathbb{Z}_+^m. \end{aligned} \tag{2.2}$$

¹Dada uma capacidade para a mochila quer-se determinar os itens a serem alocados nela de forma que a sua capacidade seja respeitada e o valor total maximizado. Os itens possuem um valor de peso e um de utilidade associado.

Subproblema para o PCE2D

No caso bidimensional, temos duas dimensões relevantes para o corte, o comprimento e a largura. O método proposto em (GILMORE; GOMORY, 1961) e (GILMORE; GOMORY, 1965) analisa uma dimensão por vez para gerar padrões de corte em 2-estágios.

Estágio 1

- Ordene w_i de forma não decrescente: $w_1 \leq w_2 \leq \dots \leq w_m$. Sendo n a quantidade de padrões de corte possíveis;
- Crie w (largura do item) faixas, cujo tamanhos são, $w_i \times L, i = 1 \dots m$ decida quais os melhores itens j para alocar na faixa, de forma que $w_j \leq w_i$. Os itens nessa faixa podem ter como largura no máximo w_i e os melhores itens são determinados resolvendo m problemas da mochila.

Estágio 2

- Decida quais são as melhores faixas, resolvendo um problema da mochila adicional.

Os m problemas da mochila a serem resolvidos no Estágio 1 são dados pelo Modelo (2.3) em que z_{ik} é o número de vezes que o tem k aparece na faixa $i, i = 1, \dots, m$.

$$\begin{aligned} \phi_i = \text{maximizar} \quad & \sum_{k=1}^i \pi_k z_{ik} \\ \text{sujeito a} \quad & \sum_{k=1}^i l_k z_{ik} \leq L \\ & z_{ik} \in \mathbb{Z}^+, i = 1 \dots m. \end{aligned} \tag{2.3}$$

Com as faixas criadas, precisamos saber quais serão utilizadas sem exceder a largura do objeto, para isso basta resolver o Problema da Mochila adicional (2.4), sendo que r_k informa quantas vezes a faixa de largura w_k será utilizada no padrão. Note que a largura da faixa k é igual à largura do item de maior largura incluído na faixa.

$$\begin{aligned} \Phi = \text{maximizar} \quad & \sum_{k=1}^m \phi_k r_k \\ \text{sujeito a} \quad & \sum_{k=1}^m w_k r_k \leq W, r_k \in \mathbb{Z}_+, i = 1 \dots m. \end{aligned} \tag{2.4}$$

Para determinar qual foi o padrão de corte obtido, resolve-se a Equação (2.5).

$$a_{ij} = \sum_{k=1}^m r_k z_{ik}. \quad (2.5)$$

Assim, para obter um padrão de corte bidimensional 2–estágios é necessário resolver $m + 1$ problemas da mochila. Os m primeiros problemas da mochila podem ser resolvidos simultaneamente empregando a programação dinâmica. O Sistema CorteBiFur usa essa estratégia (RANGEL et al., 2016) (RANGEL; CLÓVIS, 1989).

Podemos obter padrões de cortes restritos, para isso são incluídas novas restrições nos $m + 1$ problemas da mochila. Para o problema de construção de faixas temos que adicionar a mesma restrição do caso 1D descrito em (2.2), obtendo o problema (2.6).

$$\begin{aligned} \phi_i = \text{maximizar} \quad & \sum_{k=1}^i \pi_k z_{ik} \\ \text{sujeito a} \quad & \sum_{k=1}^i l_i z_{ik} \leq L \\ & z_{ik} \leq b_i \\ & z_{ik} \in \mathbb{Z}^+, i = 1 \dots m. \end{aligned} \quad (2.6)$$

E para determinar as faixas que serão utilizadas resolvemos o Problema (2.7).

$$\begin{aligned} \Phi = \text{maximizar} \quad & \sum_{k=1}^m \phi_k r_k \\ \text{sujeito a} \quad & \sum_{k=1}^m w_k r_k \leq W \\ & r_k \leq \min \left\{ \frac{b_i}{z_{ik}}, z_{ik} > 0 \right\}, r_k \in \mathbb{Z}^+, i = 1, \dots, m. \end{aligned} \quad (2.7)$$

Nesse trabalho toda vez que for necessário usar o padrão de corte irrestrito (PCI) estamos nos referindo aos subproblemas (2.3) e ao (2.4). E para o problema de corte restrito (PCR), são os subproblemas (2.6) e o (2.7).

Suliman (SULIMAN, 2006) propôs um método similar, sendo que na primeira etapa é criada a faixa que produz o mínimo de perda em relação a largura, na segunda etapa, determina-se a melhor combinação de faixas para formar um bom padrão de corte. Cui e Zhao (CUI; ZHAO, 2013) também desenvolveram um método de duas fases para gerar padrões 2–estágios guilhotinado e ainda apresentam uma extensão do método para quando é permitida a rotação dos itens. Yanasse e Katsurayama (YANASSE; KATSURAYAMA, 2005) apresentam um método em três

etapas, sendo que na primeira etapa são criadas as faixas, na segunda são criadas faixas para serem combinadas com as faixas criadas na primeira etapa e na última fase a melhor combinação delas é determinada. Outros trabalhos que utilizam as duas fases são (MALAGUTI; DURÁN; TOTH, 2014) e (VIANNA; ARENALES; GRAMANI, 2003).

Modelos matemáticos também podem ser utilizados para obter padrões de corte, Yanasse e Morabito (YANASSE; MORABITO, 2008), revisam modelos lineares e não lineares para gerar padrões de corte 2–estágios guilhotinado, podendo ser exato ou não exato, restrito ou irrestrito e apresentam novos modelos não lineares e suas linearizações baseados em adaptações ou extensões de modelos para gerar padrões de corte bidimensionais restritos 1–grupo². Lodi e Monaci (LODI; MONACI, 2003) apresentam modelos lineares para gerar um padrão de corte guilhotinado 2–estágios restrito, não exato e orientado. O primeiro modelo consiste em criar faixas com as larguras de cada item e a variável de decisão do modelo que é binária informa se o item será alocado ou não na faixa. O segundo modelo possui uma variável inteira para determinar quantas vezes cada item é alocado e uma variável binária informa se a faixa é utilizada no padrão de corte. Ainda é possível adaptar esses modelos para considerar o caso exato, com rotação de itens e irrestrito.

Outro modo de obter os padrões de corte é por meio de grafos. Christofides e Whitlock (CHRISTOFIDES; WHITLOCK, 1977) usam uma estrutura de árvore para gerar padrões de cortes guilhotinados e restritos, sendo que cada arco representa um possível corte no objeto (nó raiz) e os nós representam os retângulos gerados por esse corte. Morabito *et al* (MORABITO; ARENALES; ARCARO, 1992) apresentam um método parecido para obter padrões guilhotinados, não estagiados e irrestritos. Morabito e Arenales (MORABITO; ARENALES, 1996) estenderam o método para o caso k -estágios e restritos, utilizando a abordagem de Grafo E-OU, no qual os arcos do tipo OU são as possibilidades de cortes que podem ser feitos no objeto, e os arcos do tipo E , são os arcos bifurcados que representam o resultado de um corte guilhotinado no objeto.

Malaguti *et al* (MALAGUTI; DURÁN; TOTH, 2014) apresentam diversos métodos para resolver o PCE2D com vários tamanhos de objetos em estoque, usando uma heurística residual como subproblema em algum deles. Para gerar padrões de corte é empregado um método, chamado de *Strip heuristic algorithm*, que cria várias faixas para cada tamanho de objeto diferente considerando a largura dos itens disponíveis. A melhor faixa é escolhida e a demanda dos itens é atualizada, esse processo continua até que a melhor faixa encontrada possua uma perda abaixo

²Um padrão n -grupo, é um padrão de corte guilhotinado em 2–estágios no qual as faixas resultantes do primeiro estágio são divididas em grupos de forma que cada grupo de faixas é cortado simultaneamente no segundo estágio. Se o padrão contiver apenas um grupo de faixas temos um padrão tabuleiro ou 1–grupo.

de um valor pré determinado.

Em seguida, as faixas são alocadas nos objetos e todos os padrões que tem uma perda de área menor que um valor dado são selecionadas para a solução final, os demais padrões são descartados. Depois, um procedimento de preenchimento de padrões é chamado para tentar alocar os itens restantes nos padrões aceitos, e se mesmo assim ainda existirem itens que não tiveram sua demanda atendida, é resolvido um problema residual final.

2.2.2 Modelagem matemática e métodos de solução

Considere que:

- m representa a quantidade de itens;
- b representa o vetor da demanda dos objetos, $i = 1, \dots, m$;
- n representa a quantidade de padrões de cortes possíveis;
- c_j o custo do padrão de corte j , $j = 1, \dots, n$;
- A_j representa o padrão de corte j , $j = 1, \dots, n$.

O PCE2D pode ser representado por (2.8) se a demanda é atendida exatamente (caso exato) ou por (2.9) se é permitido o excesso de itens (caso não exato).

$$\begin{aligned} & \text{minimizar} && \sum_{j=1}^n c_j x_j \\ & \text{sujeito a} && \sum_{j=1}^n A_j x_j = b, \\ & && x_j \geq 0, \quad x_j \in \mathbb{N}^n, \quad j = 1, \dots, n, \end{aligned} \tag{2.8}$$

$$\begin{aligned} & \text{minimizar} && \sum_{j=1}^n c_j x_j \\ & \text{sujeito a} && \sum_{j=1}^n A_j x_j \geq b, \\ & && x_j \geq 0, \quad x_j \in \mathbb{N}^n, \quad j = 1, \dots, n, \end{aligned} \tag{2.9}$$

sendo que x_j representa o número de objetos cortados de acordo o padrão de corte j . Podemos ainda simplificar a notação do Modelo (2.8), reescrevendo-o na forma matricial, cada coluna da matriz $A \in \mathbb{N}^{m \times n}$ representará um padrão de corte.

$$\begin{aligned}
& \text{minimizar} && \sum_{j=1}^n c_j x_j \\
& \text{sujeito a} && Ax = b, \\
& && x \geq 0, x \in \mathbb{N}^n
\end{aligned} \tag{2.10}$$

Em Silva *et al* (SILVA; ALVELOS; CARVALHO, 2010) foi desenvolvido um modelo de programação inteira para a resolução do PCE2D guilhotinado com padrões de 2 e 3 estágios, considerando a geração de padrões de corte restrito e irrestrito, com rotação de itens, vários tamanhos de objetos em estoque e a função objetivo pode ser adaptada para minimizar o desperdício. O modelo informa quantas vezes um item i foi utilizado em uma placa j , para isso cria-se uma faixa com a largura do item e depois é feito mais um corte para obtê-lo, esse objeto é chamado $j = 0$, e os demais valores de j representam, as placas residuais, que são os objetos obtidos após o corte de um item. A função objetivo minimiza os cortes realizados nas placas do tipo $j = 0$. Em Macedo *et al* (MACEDO; ALVES; CARVALHO, 2010) é apresentada uma extensão do modelo proposto por (CARVALHO, 1999), o modelo original é voltado para o caso unidimensional e a extensão para o caso bidimensional. O modelo cria padrões de cortes guilhotinados 2-estágios e é um modelo de fluxo em arcos, os autores ainda apresentam métodos para calcular limitantes inferiores e planos de cortes. Em Sávic *et al* (SAVIC; KRATICA; FILIPOVI, 2013) são apresentados dois modelos não lineares com e sem rotação de itens para o caso 2D. Nas Subseções 2.4.1 e 2.4.2 são apresentadas estratégias para resolver o problema de corte de estoque bidimensional usando o método proposto por Gilmore e Gomory (GILMORE; GOMORY, 1961), (GILMORE; GOMORY, 1963).

2.2.3 Método simplex

Considere a relaxação linear do Problema (2.10), ou seja, substitua a condição de integridade das variáveis por $x \in \mathbb{R}_+^n$. Seja A a matriz que contém todos os padrões de corte n . Determina-se uma partição de A , tal que $A = (B, N)$, com $B \in \mathbb{R}^{m \times n}$ e N contém as demais $(m \times (n - m))$ colunas. A matriz B é formada por m colunas linearmente independentes, por isso, forma uma base e é chamada de matriz básica, e N é a matriz não básica. Da mesma forma é feito uma partição no vetor, $x = (x_B, x_N)$, com x_B sendo o vetor das variáveis não básicas e x_N das variáveis básicas, e no vetor $c^t = (c_B, c_N)^t$, com c_B sendo o vetor de custo das variáveis básicas e o c_N o vetor de custo das variáveis não básicas.

Uma solução básica factível \hat{x} qualquer pode ser escrita da seguinte forma: $\hat{x}_B = B^{-1}b \geq 0$ e $\hat{x}_N = 0$

A solução geral utilizando a partição básica e a função objetivo $f(x)$ podem ser determinadas da seguinte forma,

$$x = (x_B, x_N)^t, \text{ com } x_B = B^{-1}b - B^{-1}Nx_N.$$

$$f(x) = c^t x = (c_B, c_N)^t (x_B, x_N) = c_B^t x_B + c_N^t x_N.$$

Como

$$x_B = B^{-1}b - B^{-1}Nx_N, \text{ temos,}$$

$$f(x) = c_B^t (B^{-1}b - B^{-1}Nx_N) + c_N^t x_N$$

$$= c_B^t B^{-1}b + (c_N^t - c_N^t B^{-1}N)x_N$$

(2.11)

O termo $c_B^t B^{-1}b$ corresponde ao valor da função objetivo em \hat{x} . Logo,

$$f(\hat{x}) = c_B^t B^{-1}b.$$

Assim, temos que

$$f(\hat{x}) = c_B^t \hat{x}_B + c_N^t \hat{x}_N = c_B^t (B^{-1}b) + c_N^t (0) = c_B^t (B^{-1}b).$$

Para facilitar os cálculos definamos o vetor π , de acordo com (2.12) e o chamemos de **multiplicador simplex** ou **variável dual**.

$$\pi^t = c_B^t B^{-1}, \tag{2.12}$$

Substituindo o multiplicador simplex na equação (2.11), obtemos

$$f(x) = f(\hat{x}) + (c_N^t - \pi^t N)x_N.$$

Como

$$\begin{aligned} c_N^t - \pi^t N &= (c_{N_1}, c_{N_2}, \dots, c_{N_{n-m}}) - \pi^t (a_{N_1}, a_{N_2}, \dots, a_{N_{n-m}}) \\ &= (c_{N_1} - \pi^t a_{N_1}, c_{N_2} - \pi^t a_{N_2}, \dots, c_{N_{n-m}} - \pi^t a_{N_{n-m}}) \\ e \ x_N &= (x_{N_1}, x_{N_2}, \dots, x_{N_{n-m}}). \end{aligned}$$

Obtemos

$$f(x) = f(\hat{x}) + \hat{c}_{N_1} x_{N_1} + \hat{c}_{N_2} x_{N_2} + \dots + \hat{c}_{N_{n-m}} x_{N_{n-m}} \quad (2.13)$$

Os coeficientes $\hat{c}_{N_j} = (c_{N_j} - \pi^t a_{N_j})$ das variáveis não básicas são chamados de **custos relativos** ou reduzidos.

Para determinar se uma solução é ótima aplica-se o teste chamado de **condição de otimalidade** que consiste em, dada a partição básica $A = (B \ N)$ e uma solução básica factível $\hat{x} = B^{-1}b \geq 0$ e $\pi^t = c_B^t B^{-1}$, verificar se $c_N - \pi^t a_{n_j} \geq 0, j = 1, \dots, n - m$. Se todos os custos relativos são maiores ou iguais a zero, então a solução básica é ótima. Caso contrário, devemos encontrar uma solução factível, para isso utiliza-se a **estratégia simplex**. Dada uma solução que não satisfaça a condição de otimalidade, ou seja, existe um k tal que, $\hat{c}_{N_k} = c_{N_k} - \pi^t a_{n_k} < 0$. A estratégia consiste em determinar uma solução factível alterando o valor da k -ésima variável não básica.

$$\begin{cases} x_{N_k} = \varepsilon \geq 0, \\ x_{N_j} = 0, j = 1, \dots, n - m, j \neq k \end{cases} \quad (2.14)$$

Substituindo em (2.14) temos que agora a função objetivo será:

$$f(x) = f(\hat{x}) + \hat{c}_{N_1} 0 + \dots + \hat{c}_{N_k} \varepsilon + \dots + \hat{c}_{N_{n-m}} 0 = f(\hat{x}) + \hat{c}_{N_k} \varepsilon < f(\hat{x}).$$

Sendo k a k -ésima variável não básica. Assim, obtemos que o valor da função objetivo irá decrescer quando ε aumentar. Usando $y = B^{-1}a_{N_k}$, as variáveis básicas são modificadas da seguinte forma,

$$\begin{aligned}
x_B &= B^{-1}b - B^{-1}Nx_N \\
&= \hat{x}_B - B^{-1}a_{N_k}\varepsilon \\
&= \hat{x}_B - y\varepsilon,
\end{aligned}$$

Chamaremos de **direção simplex** o vetor $y = B^{-1}a_{n_k}$. Com isso, $x_{B_i} = \hat{x}_{B_i} - y_i\varepsilon, i = 1, \dots, m$. Como a função objetivo decresce quando ε cresce, queremos determinar o maior valor para ε que mantêm a solução perturbada factível. Para isso, faz-se, $\hat{\varepsilon} = \frac{\hat{x}_{B_l}}{y_l}$ com $y_i > 0$. Porém se $y_i \leq 0$, para todo $i = 1, \dots, m$ o problema é ilimitado e o método termina. Supondo que a l -ésima variável básica sairá da base, então $x_{B_l} = \hat{x}_{B_l} - y_l\hat{\varepsilon} = \hat{x}_{B_l} - y_l\left(\frac{\hat{x}_{B_l}}{y_l}\right) = 0$. A nova solução será $(x_{B_1} \dots x_{B_l} \dots x_{B_m} \mid 0 \dots x_{N_k} \dots 0)$, com $x_{B_l} = 0$ e $x_{N_k} = \hat{\varepsilon}$. Dessa forma, as matrizes básicas e não básicas são atualizadas e somente uma das colunas de cada uma é alterada.

$$B = (a_{B_1} \dots a_{B_l} \dots a_{B_m}) \rightarrow B^* = (a_{B_1} \dots a_{N_k} \dots a_{B_m})N = (a_{N_1} \dots a_{N_k} \dots a_{N_{n-m}}) \rightarrow N^* = (a_{N_1} \dots a_{B_l} \dots a_{N_{n-m}}).$$

Nesse momento, a primeira iteração do método Simplex termina. O processo é repetido até a solução ótima ser obtida ou ser identificado que o problema é ilimitado. O Algoritmo 1 detalha os passos do método simplex. Maiores detalhes podem ser obtidos em (ARENALES et al., 1998).

Algoritmo método simplex

Algoritmo 1: ALGORITMO DO MÉTODO SIMPLEX

Entrada: A, b, c

- 1 Determinar uma solução factível tal que: $A = (B N), x = (x_B, x_N)$ e $c = (c_b, c_N)$;
 - 2 Determinar uma solução básica factível: $\hat{x} = B^{-1}b \geq 0, \bar{x}_N = 0$;
 - 3 Determinar a direção simplex (cálculo da solução básica dual): $\pi^t = c_b^t B^{-1}0$;
 - 4 Determinar os custos reduzidos: $\hat{c}_{N_j} = c_{N_j} - \pi^t a_{N_j}, j = 1, \dots, n - m$;
 - 5 Determinar a variável a entrar na base: $c_{N_k}^* = \min \hat{c}_{N_j}, j = 1, \dots, n - m$.
 - 6 **enquanto** $\hat{c}_{N_j} < 0$ **faça**
 - 7 Determine a direção simplex: $y = B^{-1}a_{N_k}$
 - 8 Determine o passo: $\varepsilon = \frac{x_{B_l}}{y_l} = \min \frac{\hat{x}_{B_l}}{y_i}, y_i > 0, i = 1, \dots, m$; Se $y \leq 0$, pare. O problema é ilimitado. **se** $y \leq 0$ **então**
 - 9 PARE. O problema é ilimitado.
 - 10 **fim**
 - 11 Atualize a base: $a_{B_l} \leftrightarrow a_{N_k}$;
 - 12 Determine: $\hat{x}_B = B^{-1}b, \pi = c_b^t B^{-1}$,
 - 13 $\hat{c}_{N_j} = c_{N_j} - \pi^t a_{N_j}, j = 1, \dots, n - m$.
 - 14 Escolha o menor $c_{N_k}^* = \min \hat{c}_{N_j}, j = 1, \dots, n - m$.
 - 15 **fim**
- Saída:** \hat{x}
-

2.2.4 Decomposição de Dantzig-Wolfe

O Método Simplex com Geração de Colunas

O Princípio de Decomposição de Dantzig-Wolfe foi desenvolvido por Dantzig e P. Wolfe em 1961. Ele é baseado na versão revisada do método simplex e tem uma aplicabilidade eficiente em problemas da programação linear de grande porte. Os problemas da programação linear de grande porte usualmente possuem alguma estrutura especial, e isso deve ser explorado a fim de desenvolver métodos computacionais mais eficientes (LUENBERGER; YE, 1984). Nesses casos a matriz dos coeficientes tem estrutura bloco-angular e possui um ou mais blocos com restrições independentes, chamadas de restrições com estrutura especial, e algumas restrições com variáveis em comum, chamadas de restrições de acoplamento. De acordo com o Problema Original (2.15), a restrição de acoplamento é a primeira, enquanto que as demais são as restrições com estrutura especial. Essa decomposição visa facilitar a resolução do problema original,

reduzindo assim a dimensão da base e gerando um problema equivalente ao original porém com grande redução no número de restrições. Além disso, analisando o Modelo (2.15), nota-se que qualquer problema da programação linear pode obter essa forma, basta fazer $m = 1$, o modelo gerado está representado em (2.16).

$$\begin{aligned}
 \text{minimizar } f &= c_1x_1 + c_2x_2 + \dots + c_mx_n \\
 \text{sujeito a } &= A_1x_1 + A_2x_2 + \dots + A_mx_n = b_0 \\
 &B_1x_1 = b_1 \\
 &B_2x_2 = b_2 \\
 &\quad \quad \quad \ddots \\
 &B_mx_m = b_m \\
 &x_i \geq 0 \quad i = 1, \dots, m.
 \end{aligned} \tag{2.15}$$

$$\begin{aligned}
 \text{minimizar } f &= c^t x \\
 \text{sujeito a } &Ax = b \\
 &B_i x = b_i \\
 &x \geq 0.
 \end{aligned} \tag{2.16}$$

A ideia do método é decompor o problema original em dois novos problemas mais simples de serem resolvidos. O primeiro chamado de Problema Mestre que é semelhante ao original mas possui um número reduzido de restrições, as de acoplamento, mas com um número de colunas maior. E o segundo problema que é um subproblema originado das restrições com estrutura especial.

Para simplificar, utilizaremos $m = 1$, ou seja, o Modelo (2.16) que tem a estrutura de bloco-angular. Iremos supor também que o conjunto convexo (poliedro), $X = \{x | Bx = b, x \geq 0\}$ é limitado. É possível estender essas hipóteses ($m = 1$ e o conjunto convexo ser limitado) e as modificações necessárias podem ser encontradas em (LASDON, 1970). Substituindo $Bx = b$ por X no Modelo (2.16), obtemos o Modelo (2.17).

$$\begin{aligned}
 \text{minimizar } f &= c^t x \\
 \text{sujeito a } &Ax = b \\
 &x \in X.
 \end{aligned} \tag{2.17}$$

Podemos reescrever as variáveis do Modelo (2.17) como uma combinação convexa dos pontos extremos de X de acordo com a Definição 9 dada a seguir. Fazendo $x \in X$, como combinação convexa dos p pontos extremos x^1, x^2, \dots, x^p , obtemos, a equação (2.18) e a restrição

(2.19) chamada de restrição de convexidade.

Definição 11. Dado um conjunto $X \subseteq \mathbb{R}^n$ é uma combinação convexa de pontos de X se existir um conjunto de pontos $\{x^j\}^P \in X$ tal que $x = \sum_{j=1}^P \lambda^j x^j$, $\sum_{j=1}^P \lambda^j = 1$, $\lambda^j \geq 0$. O envoltório convexo de X , denotado por $\text{Conv}(X)$, é o conjunto de todos os pontos que são combinações convexas de pontos de X (WOLSEY, 1998).

$$x = \sum_{j=1}^P \lambda^j x^j, \quad (2.18)$$

$$\sum_{j=1}^P \lambda^j x^j = 1, \lambda^j \geq 0, j = 1, \dots, P. \quad (2.19)$$

Para obter uma solução para o Problema Original (2.15), devemos determinar dentre todas as soluções de X que são factíveis pra as restrições $B_i x = b_i$ e $x \geq 0$ do Modelo (2.16), satisfazem a restrição $Ax = b$ e minimizam z . Substituindo as restrições (2.18) e (2.19) no Problema (2.17) obtemos, a restrição (2.20) e a função objetivo (2.21)

$$Ax = b \Leftrightarrow A\left(\sum_{j=1}^P \lambda^j x^j\right) = b \Leftrightarrow \sum_{j=1}^P (Ax^j) \lambda^j = b. \quad (2.20)$$

$$f = c^t x \Leftrightarrow f = c^t \left(\sum_{j=1}^P \lambda^j x^j\right) = b \Leftrightarrow f = \sum_{j=1}^P (c^t x^j) \lambda^j. \quad (2.21)$$

Assim o novo Problema (2.22) obtido em relação a variável λ^j é equivalente ao Problema Original (2.15). Esse problema é chamado de Problema Mestre. Como o número de pontos extremos de um poliedro é muito grande em um problema de grande porte, não é conveniente enumerar todos eles, pois isso também provocaria milhares de colunas. Em resumo, para resolver o problema original sem necessitar enumerar todas as colunas *a priori*, resolvemos o problema mestre método simplex. É utilizado o método descrito para gerar novas colunas, essa coluna fará parte da base se o seu custo reduzido for negativo, caso contrário o procedimento termina.

$$\begin{aligned}
\text{minimizar } f & \quad \sum_{j=1}^p (c^p \lambda^j) x_j \\
\text{sujeito a} & \quad \sum_{j=1}^p (Ax^j) \lambda^j = b \\
& \quad \sum_{j=1}^p \lambda^j = 1 \\
& \quad \lambda^j \geq 0, j = 1, \dots, t.
\end{aligned} \tag{2.22}$$

Por meio dessa decomposição foi demonstrado por Vance (VANACE, 1998) em 1988 que a partir do primeiro modelo disponível na literatura para o Problema de Corte de Estoque Unidimensional, o modelo atribuído a Kantorovich (2.23) é possível obter o modelo de Gilmore e Gomory 2.8 mostrando a equivalência entre ambos.

$$\begin{aligned}
\text{minimizar } f & \quad \sum_{j=1}^k \mu_j \\
\text{sujeito a} & \quad \sum_{j=1}^k x_{ij} \geq b_i, i = 1, \dots, m \\
& \quad \sum_{i=1}^m l_i x_{ij} \geq L \mu_j, j = 1, \dots, k \\
& \quad x_{ij} \in \mathbb{Z}_+^n, \mu_j \in \{0, 1\}, i = 1, \dots, m, j = 1, \dots, k.
\end{aligned} \tag{2.23}$$

k é um limitante superior para o número de objetos necessários para atender a demanda b , x_{ij} representa a quantidade de vezes que o item i é cortado de acordo com o padrão j , L e l representam respectivamente o comprimento do objeto e dos itens e μ_j é a variável binária que vale 1 se o padrão de corte j é utilizado, e caso contrário será zero.

Método simplex com geração de colunas para o PCE

Gilmore e Gomory propuseram nos anos de 1961 (GILMORE; GOMORY, 1961) e 1963 (GILMORE; GOMORY, 1963) respectivamente o método de geração de colunas para o problema de corte de estoque unidimensional e bidimensional. A técnica consiste em relaxar o Problema (2.10), obtendo o Problema Mestre Restrito (2.24), considerando que a matriz B é composta por um subconjunto m de padrões de cortes que formam uma base factível para o problema.

$$\begin{aligned}
\text{minimizar} & \quad \sum_{j=1}^n c_j x_j \\
\text{sujeito a} & \quad Bx = b, \\
& \quad x \geq 0, x \in \mathbb{R}^n
\end{aligned} \tag{2.24}$$

Para resolver o Modelo (2.24) utiliza-se o método simplex com geração de colunas, de acordo com o Algoritmo 1. Determina-se x_B e as variáveis duais π^t , linhas 2, 3 e 4 do Algoritmo 1. Geramos uma nova coluna associada a variável x_k com o menor custo relativo para entrar na base, pois o critério da função objetivo é minimizar. Dessa forma, a coluna procurada é a A_{N_k} , mas para simplificarmos a notação usaremos A_k , ela deve satisfazer $\hat{c}_k = c_k - \pi^t A_k = \min\{c_j - \pi^t A_j\}$, ou seja, devemos resolver o subproblema (2.25).

$$\text{minimizar } g = c_j - \pi^t A_j \quad (2.25)$$

$$\text{sujeito a } A_j. \quad (2.26)$$

A_j é um padrão de corte factível. O subproblema (2.25) também é chamado de pricing (CHVATAL, 1983). No caso do PCE-1D (2.25) corresponde ao problema da mochila. Como estamos considerando como critério de otimização minimizar a quantidade de objetos utilizados, netão, $c_j = 1, j = 1, \dots, (n - m)$ e podemos reescrever a função objetivo do Modelo (2.25) como, $z = \min(1 - \pi^t A_j) = 1 - \min(-\pi^t A_j) = 1 - \max(\pi^t A_j)$, e reformulando o subproblema, temos o Problema (2.27).

$$\text{maximizar } g = \pi^t A_j \quad (2.27)$$

$$\text{sujeito a } A_j. \quad (2.28)$$

Suponha que a k -ésima coluna foi gerada e que ela possui como custo relativo, $c_k = 1 - z$. Se $c_k = 1 - z \geq 0$ a solução básica é ótima, caso contrário, essa coluna entra na base melhorando a solução do problema. Para decidir a coluna que sairá da base deve-se seguir os passos das Linhas 8 – 15 do Algoritmo 1.

Os subproblemas resolvidos para o caso unidimensional e bidimensional são diferentes, apesar dos seus problemas mestres não serem. O subproblema para o caso unidimensional é o apresentado em (2.1) e para o caso bidimensional são os problemas (2.3) e o (2.4).

Utilizamos a instância descrita no Exemplo 1, para ilustrar o método simplex com a GC. Essa instância é a P1 – 15, disponível em (FIGUEIREDO; RANGEL, 2008). Aqui resolveremos somente uma iteração da problema. Nos Exemplos 2 e 3 do capítulo 3, é apresentada a resolução completa dessa instância.

Exemplo 1:

Resolva o problema PCE2D considerando os dados apresentados na Tabela 2.1.

2750	Comprimento do objeto
1850	Largura do objeto
(647, 1062, 710)	Comprimento dos itens
(453, 530, 535)	Largura dos itens
(960, 320, 320)	Demanda dos itens

Tabela 2.1: Dados do Problema.

Iniciamos o problema mestre restrito (2.29) com a matriz identidade 3×3 .

$$\begin{aligned}
 & \text{minimizar} && x_1 + && x_2 + && x_3 \\
 & \text{sujeito a} && x_1 && && & = && 960 \\
 & && && x_2 && & = && 320 \\
 & && && && x_3 & = && 320 \\
 & && && && & & & x_i \in \mathbb{R}_+^3, \quad i = 1, 2, 3
 \end{aligned} \tag{2.29}$$

Obtendo como solução $x^1 = (960, 320, 320)^t$ e o valor da função objetivo igual a 1600.

De acordo com o Algoritmo Simplex devemos determinar as variáveis duais, para isso fazemos:

$$\pi = (1, 1, 1) \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \text{ logo } \pi = (1, 1, 1).$$

Agora, criamos os $m = 3$ primeiros problemas da mochila (2.30), (2.31) e (2.32) para gerar as faixas.

$$\begin{aligned}
 \phi_1 = & \text{maximizar} && r_1 \\
 & \text{sujeito a} && 647r_1 \leq 2750, \\
 & && r_1 \in \mathbb{Z}^+.
 \end{aligned} \tag{2.30}$$

$$\begin{aligned}
 \phi_2 = & \text{maximizar} && r_1 + r_2 \\
 & \text{sujeito a} && 647r_1 + 1062r_2 \leq 2750 \\
 & && r_1, r_2 \in \mathbb{Z}^+.
 \end{aligned} \tag{2.31}$$

$$\begin{aligned}
\phi_2 = \text{maximizar} & & r_1 + r_2 + r_3 \\
\text{sujeito a} & & 647r_1 + 1062r_2 + 710r_3 \leq 2750 \\
& & r_1, r_2, r_3 \in Z^+.
\end{aligned} \tag{2.32}$$

Obtendo as respectivas soluções,

$$r^1 = (4, 0, 0) \text{ e } \phi_1 = 4.$$

$$r^2 = (4, 0, 0) \text{ e } \phi_2 = 4.$$

$$r^3 = (4, 0, 0) \text{ e } \phi_3 = 4.$$

E por último criaremos a mochila $m + 1$ (2.33),

$$\begin{aligned}
\Phi = \text{maximizar} & & 4z_1 + 4z_2 + 4z_3 \\
\text{sujeito a} & & 453z_1 + 530z_2 + 535z_3 \leq 1850 \\
& & z_1, z_2, z_3 \in Z^+.
\end{aligned} \tag{2.33}$$

Que gera a solução, $(4, 0, 0)$. E para recuperar a coluna obtida, fazemos:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 4 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 16 \\ 0 \\ 0 \end{bmatrix}$$

Assim a nova coluna gerada é $(16, 0, 0)^t$. O próximo passo no algoritmo é calcular o custo mínimo reduzido para verificar se a coluna gerada entrará na base. Custo Reduzido = $1 - (1, 1, 1) \times (16, 0, 0)^t = -15 < 0$. Logo a coluna entrará na base, por isso devemos determinar a coluna a deixar a base. Para isso calculamos a direção simplex,

$$y = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \times (16, 0, 0)^t.$$

Então, $y = (16, 0, 0)$. Como o único elemento de $y > 0$ é o primeiro, não precisamos determinar o mínimo, pois de acordo com o algoritmo o y_i deve ser maior que zero. Dessa forma a primeira coluna deixará a base e será substituída pela coluna encontrada, $(16, 0, 0)^t$. Dessa forma a nova base obtida é,

$$B = \begin{bmatrix} 16 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

É necessário iniciar outra iteração, para verificar se a nova base é ótima. No Exemplo 2 e 3 do Capítulo 3 completamos a solução desse problema.

Utilizar o método simplex com geração de colunas não garante que a solução será ótima e nem que será inteira, pois para utilizar o método simplex o problema deve estar relaxado, assim a solução obtida pode ser contínua. Para determinar uma solução inteira factível a partir da solução contínua pode-se utilizar um método *branch-and-price* (BELOV; SCHEITHAUER, 2006), (MALAGUTI; DURÁN; TOTH, 2014), heurísticas de arredondamento (WÄSCHER; GAU, 1996), (POLDI; ARENALES, 2006), (CUI; ZHAO, 2013) ou no caso exato apenas utilizar a função teto. O problema em utilizar a função teto é que apesar de continuar obtendo uma solução factível, ela pode provocar uma produção de itens muito acima da demanda, o que pode gerar custos de estoque de itens e dificuldades no gerenciamento dos itens excedentes. O método *branch-and-price* discutido na Seção 2.4.3 possui uma convergência lenta e necessita de muito tempo computacional e memória, por isso o interesse no desenvolvimento de heurísticas de arredondamento, em particular nas heurísticas residuais. No Capítulo 3, abordamos as heurísticas de arredondamento.

2.2.5 Enumeração implícita e a geração de colunas

O método chamado de *branch-and-price*, é uma junção do método *branch-and-bound* (WOLSEY, 1998) em que cada nó é aplicado o método de geração de colunas. Pode-se também adicionar inequações válidas em cada nó para eliminar soluções não inteiras (planos de corte), fazendo com que haja uma convergência mais rápida do método, nesse caso temos o método *branch-and-price-and-cut* (BCP). Em Belov e Scheithauer (BELOV; SCHEITHAUER, 2006) é proposto um algoritmo BCP para resolver o problema de corte de estoque dimensional guilhotinado com padrão 2—estágios. Foi utilizada a geração de colunas proposta por Gilmore e Gomory e os planos de cortes por Chvatal (CHVATAL, 1973), (SALLAUME et al., 2008). Malaguti *et al* (MALAGUTI; DURÁN; TOTH, 2014) apresentam um *branch-and-price* truncado. O algoritmo inicia com uma solução encontrada pelo método GRASP iterativo, em seguida é feito a geração de colunas, se a solução não for inteira, é utilizado o método *diving* no próximo nó da árvore. A cada iteração um limitante inferior é associado a uma variável, então atualiza-se a demanda residual e resolve-se o problema residual encontrado. O método continua até a primeira solução inteira ser encontrada.

2.2.6 Heurísticas construtivas

O algoritmo heurístico baseado em ideias construtivas são considerados algoritmos de repetição exaustiva. A cada iteração é gerado um padrão de corte, e este é utilizado o máximo de vezes sem que a demanda seja excedida. Esses algoritmos podem ser tanto utilizados para resolver o PCE (CINTRA; WAKABAYASHI, 1998), (SALLAUME et al., 2008) ou como sub rotina para aprimorar alguma heurística (CINTRA et al., 2008).

Os algoritmos diferem pela estratégia de construção do padrão. Em Cintra (CINTRA, 2014) são apresentadas algumas heurísticas construtivas clássicas, o algoritmo *Next Fit* (NF) consiste em alocar um item, atualizar o comprimento disponível do objeto e ir alocando os itens até que a dimensão do objeto não seja excedida. No momento que determinado item não couber mais em um objeto, ele é alocado em um novo objeto e os itens seguintes serão alocados nesse novo objeto, o processo continua até toda a demanda ser atendida. O algoritmo *First Fit* (FF) tem a mesma premissa, porém o item pode ser alocado em qualquer objeto, de preferência, no objeto que apresenta o menor índice de área utilizada na qual ele caiba. O algoritmo *First Fit Decreasing* (FFD) consiste em alocar o maior item em um padrão o máximo de vezes possível, ou seja, até sua demanda ser atendida ou ele não caber mais. Em seguida, o mesmo é feito para o segundo maior item e assim por diante até todos os itens serem examinados. O padrão de corte é utilizado o máximo de vezes sem que exceda a demanda. A demanda é atualizada e o processo recomeça até a demanda ser atendida por completo.

2.2.7 Algoritmo genético

As heurísticas baseadas no algoritmo genético podem ser descritas da seguinte forma (HEIS; CONSTANTINO; ARAUJO, 2010). Dado um indivíduo que equivale a uma solução factível para o problema de corte de estoque, o conjunto de padrões de corte associados são considerados como um cromossomo, e os genes representam cada um dos padrões. O método consiste em construir um conjunto de soluções factíveis de forma aleatória (indivíduos da população) e, em seguida, escolher também de forma aleatória os genes para compor a uma nova solução factível. Se a solução encontrada não atender a demanda de algum item, então é chamada uma heurística de construção para torná-la factível (mutação). Para cada indivíduo gerado (solução factível) é calculado o seu valor utilidade, caso essa solução construída seja melhor que alguma já existente na população, então esta substituirá alguém na população. Dessa forma, a cada iteração é gerada uma solução até que o número de gerações chegue ao máximo. Heis *et al* (HEIS; CONSTANTINO; ARAUJO, 2010) usaram esse método para resolver o PCE1D com vários

tamanhos de objetos em estoque, cada tipo de objeto em quantidade limitada. Em Pál (PÁL, 2006) é apresentado duas heurísticas para o caso bidimensional com cortes guilhotinados e com a rotação de itens permitida. Outras heurísticas podem ser encontradas em (ONWUBOLU; MUTINGI, 2003) e (RESENDE, 2013).

Capítulo 3

HEURÍSTICAS RESIDUAIS

Nesse capítulo é feita uma revisão sobre a Heurística Residual, apresentamos algumas heurísticas residuais encontradas na literatura e propomos modificações em uma delas.

3.1 O problema residual

Dado o Problema de Corte de Estoque para o caso exato 3.1 e para o caso não exato 3.2.

$$\begin{aligned}
 & \text{minimizar} && \sum_{j=1}^n x_j \\
 & \text{sujeito a} && Ax = b, \\
 & && x \geq 0, \quad x \in \mathbb{N}^n.
 \end{aligned} \tag{3.1}$$

$$\begin{aligned}
 & \text{minimizar} && \sum_{j=1}^n x_j \\
 & \text{sujeito a} && Ax \geq b, \\
 & && x \geq 0, \quad x \in \mathbb{N}^n.
 \end{aligned} \tag{3.2}$$

Considere a solução da relaxação linear do Problema (3.1) e (3.2). Se essa solução é fracionária, a partir dela queremos determinar uma solução inteira aproximada, y , tal que $Ay \leq b$. Essa solução não excede a demanda, e pode ser que uma parte da demanda não seja atendida. A demanda não atendida associada a solução y é chamada de demanda residual, r , e definida como $r = b - Ay$.

Se no Problema (3.1), fizermos, $b = r$, obtemos o Problema (3.3), chamado de Problema Residual. Uma heurística residual consiste em resolver de forma iterativa o Problema (3.3) com a atualização da demanda residual a cada iteração, enquanto $r > 0$. Se em alguma iteração $y = 0$,

resolve-se o Problema Residual Final. Nesse caso, faltam poucos itens com demanda residual diferente de zero, então determina-se um método para gerar padrões de corte que incluam os itens que faltam para atender exatamente a demanda. No Algoritmo 2 é apresentada a estrutura geral da Heurística Residual, em que B é uma base factível inicial para o sistema $Ax = b$ (por exemplo, composta por padrões de cortes homogêneos).

$$\begin{aligned} & \text{minimizar} && \sum_{j=1}^n x_j \\ & \text{sujeito a} && Ax \geq r, \\ & && x \geq 0, \quad x \in \mathbb{N}^n. \end{aligned} \tag{3.3}$$

Algoritmo 2: ESTRUTURA GERAL DA HEURÍSTICA RESIDUAL

Entrada: $r = b, B$

1 **enquanto** $r_i > 0, i = 1 \dots m$ **faça**

2 Resolva o Problema (3.3) relaxado por geração de colunas e com B como base inicial.

3 Seja x^k a solução encontrada. **se** $x^k \in N$ **então**

4 **PARE**

5 **senão**

6 Determine uma solução inteira aproximada, y^k .

7 **fim**

8 **fim**

9 **se** $y^k == 0N$ **ou o Problema (3.3) for infactível** **então**

10 Resolva o Problema Residual Final.

11 **senão**

12 $r^{k+1} \leftarrow r^k - By^k,$

13 $k \leftarrow k + 1.$

14 **fim**

15 **fim**

16 **fim**

Saída: y

Nas próximas seções descrevemos diversas alternativas encontradas na literatura para determinar uma solução aproximada y_k para resolver o problema residual final (linhas 7 e 11 do Algoritmo 2).

3.2 Heurísticas residuais por truncamento - Cintra *et al* (2008)

A heurística residual por truncamento estabelece um critério para obter uma solução inteira (linha 7 do Algoritmo 2). Para isso utiliza-se um arredondamento trivial como o apresentado em 3.4. Se o arredondamento implicar $y = 0$, aplica-se uma heurística construtiva para atender a demanda residual final.

$$y = (\lfloor x_1 \rfloor, \lfloor x_2 \rfloor, \dots, \lfloor x_n \rfloor) \quad (3.4)$$

Cintra *et al* (CINTRA et al., 2008) apresentam uma heurística residual por truncamento, que consiste em resolver o problema (3.2) relaxado pelo método simplex com geração de colunas por meio da programação dinâmica. Para obter a solução inteira, utilizam o truncamento (3.4), e enquanto a demanda não for atendida, resolvem o problema residual obtido. Se em alguma iteração ocorrer que $y = 0$ usam heurísticas construtivas para atender a demanda residual final, uma delas é a M-HFF que é a *Hybrid First Fit* (HFF) modificada para considerar as demandas dos itens (CINTRA et al., 2007) ou o método *First Fit Decreasing Height using Rotations* (FFDHR) se a rotação de itens for permitida.

3.3 Heurística residual 1 (HR1 - Poldi e Arenales (2006))

Poldi e Arenales (POLDI; ARENALES, 2006) desenvolveram uma heurística residual para o PCE unidimensional com apenas um objeto em estoque, e para o caso com vários objetos em estoque (POLDI; ARENALES, 2009), e Vianna e Poldi (VIANNA; POLDI, 2005) a modificaram para resolver o caso bidimensional, com aplicação na indústria de esquadrias metálicas. Na heurística original os padrões de corte são gerados pelo método de Gilmore e Gomory (descrito na seção 2.2.1), enquanto que na adaptação é utilizada a abordagem Grafo E/OU. Aqui, apresentamos a versão para o caso bidimensional.

Na heurística residual proposta por (POLDI; ARENALES, 2006), a matriz A do Problema (3.3) é iniciada com padrões homogêneos maximais. O método simplex com geração de colunas é usado para determinar a solução relaxada do problema mestre restrito (\bar{x}). Se \bar{x} é fracionária, uma solução inteira é obtida aplicando a função teto na primeira coordenada do vetor e igualando a zero as demais, ou seja, $y^1 = (\lceil \bar{x}_1 \rceil, 0, 0, \dots, 0)$. Se essa solução excede a demanda, diminui-se uma unidade de y_1 e o processo é repetido considerando o segundo padrão, isto é: $y^1 = (y_1, \lceil \bar{x}_2 \rceil, 0, \dots, 0)$. Quando todas as frequências são verificadas, o processo é interrompido.

A demanda residual é atualizada. Para encontrar a solução do problema, as soluções de cada iteração são combinadas. O detalhamento da heurística residual é apresentado no Algoritmo 3.

Estratégias para Ordenação de Padrões de Corte

Quatro propostas da heurística residual são apresentadas e se diferenciam pela estratégia utilizada para ordenação dos padrões (linha 7 do Algoritmo 3). As estratégias propostas para ordenação dos padrões estão descritas abaixo.

1. Ordenação natural;
2. Ordene x^k tal que, $x_1^k \geq x_2^k \geq \dots \geq x_m^k$;
3. Seja p_j a perda no padrão de corte $j, j = 1, \dots, N.$, calculada de acordo com a Definição (9).
Ordene x^k tal que, $p_1 \geq p_2 \geq \dots \geq p_m$;
4. Seja f_j a parte fracionária de x_j . Ordene x_j tal que, $f_1 \geq f_2 \geq \dots \geq f_m$.
Sendo $f_i = x_i - \lfloor x_i \rfloor$.

Ordenações para as frequências dos padrões de corte para a HR1.

Algoritmo 3: HEURÍSTICA RESIDUAL HR1

Entrada: $r = b$, $A = B$, $k = 1$

1 **enquanto** $r_i > 0, i = 1 \dots m$ **faça**

2 Resolva o Problema 3.1 relaxado por geração de colunas e tem B como base inicial.

3 Seja x^k a solução encontrada.

4 **se** $x^k \in N$ **então**

5 PARE

6 **senão**

7 Defina uma ordenação para os padrões de corte associados a x^k . **para**

$i = 1 \dots m$ e $x_i \in \mathbb{R}$ **faça**

8 $y_i^k = \lceil x_i^k \rceil$

9 **para** $j > i$ **faça**

10 $y_j^k = 0$;

11 **fim**

12 **enquanto** $By^k > r$ **faça**

13 $y_i^k = y_i^k - 1$

14 **fim**

15 **fim**

16 **fim**

17 **fim**

18 $r^{k+1} \leftarrow r^k - By^k$,

19 $k \leftarrow k + 1$

20 **fim**

Saída: y

Para ilustrar a Heurística Residual 1 resolvemos a instância descrito no Exemplo 1. Os resultados são descritos no Exemplo 2.

Exemplo 2.

Considere a instância do PCE2D descrita no Exemplo 1. Aplicamos a heurística residual 1, considerando que os padrões tem a ordenação natural.

Iteração 1

Iniciamos a resolução do problema com a Base Homogênea (3.5), e assim, obtemos o Problema (3.6).

$$B_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.5)$$

$$\begin{aligned} \text{minimizar} \quad & x_1 + x_2 + x_3 \\ \text{sujeito a} \quad & x_1 = 0 \\ & x_2 = 2 \\ & x_3 = 8 \\ & x_i \in \mathbb{R}_+^3, \quad i = 1, 2, 3 \end{aligned} \quad (3.6)$$

O problema 3.6 foi resolvido pelo método simplex com geração de colunas, obtendo a base B_2 e a solução ótima x^2 , exibidos em 3.7.

$$B_2 = \begin{bmatrix} 16 & 0 & 0 \\ 0 & 6 & 3 \\ 0 & 0 & 6 \end{bmatrix}, \quad x^2 = \begin{bmatrix} 60 \\ 26,7 \\ 53,3 \end{bmatrix}. \quad (3.7)$$

Como a solução não é inteira, precisamos aplicar a heurística de acordo com as linhas 7 – 17 do Algoritmo 1. Iniciamos analisando a variável x_1^1 , mas como ela já é inteira, podemos fixar $y_1 = 60$. Verificamos agora a variável x_2^1 , e obtemos os resultados exibidos na Tabela 3.1. A segunda linha representa o arredondamento do vetor solução, a terceira linha o quanto da demanda ainda falta ser atendida nessa iteração. Podemos notar que essa solução não excede a demanda, então analisamos a próxima frequência que é a variável x_3 , os resultados são exibidos na Tabela 3.2, com esse valor para o y_1^1 , temos que a demanda do segundo e terceiro item seria excedida, logo diminuimos uma unidade do y_3^1 e obtemos os resultados da Tabela 3.3, que continua excedendo a demanda do item dois. Então, novamente diminuimos uma unidade do y_3^1 , os resultados são apresentados na Tabela 3.4. Como a demanda não foi excedida e todas as frequências foram analisadas, atualizamos a demanda residual de acordo com 3.8.

	1	2	3
y^1	60	27	0
$B_2 y^1$	960	162	0
Demanda	960	320	320

Tabela 3.1: Solução inteira para o x_2^1 .

	1	2	3
y^1	60	27	54
B_2y^1	960	324	324
Demanda	960	320	320

Tabela 3.2: 1° solução inteira para o x_3^1 .

	1	2	3
y^1	60	27	53
B_2y^1	960	321	318
Demanda	960	320	320

Tabela 3.3: 2° solução inteira para o x_3^1 .

	1	2	3
y^1	60	27	52
B_2y^1	960	318	312
Demanda	960	320	320

Tabela 3.4: 3° solução inteira para o x_3^1 .

$$r^2 = \begin{bmatrix} 960 \\ 320 \\ 320 \end{bmatrix} - \begin{bmatrix} 16 & 0 & 0 \\ 0 & 6 & 3 \\ 0 & 0 & 6 \end{bmatrix} \times \begin{bmatrix} 60 \\ 27 \\ 52 \end{bmatrix} = \begin{bmatrix} 0 \\ 2 \\ 8 \end{bmatrix}. \quad (3.8)$$

Iteração 2

Analogamente a primeira iteração, resolvemos o Problema Residual (3.9). Obtendo como base ótima e solução as matrizes representadas em (3.10)

$$\begin{aligned} & \text{minimizar} && x_1 + && x_2 + && x_3 \\ & \text{sujeito a} && x_1 && && = 0 \\ & && && x_2 && = 2 \\ & && && && x_3 = 8 \\ & && && && x_i \in \mathbb{R}_+^3, \quad i = 1, 2, 3 \end{aligned} \quad (3.9)$$

$$B_3 = \begin{bmatrix} 0 & 0 & 1 \\ 2 & 0 & 0 \\ 6 & 6 & 0 \end{bmatrix}, \quad x^3 = \begin{bmatrix} 1 \\ 0,33333333... \\ 0 \end{bmatrix}. \quad (3.10)$$

Aplicamos novamente o arredondamento para a variável x_2 , os resultados são exibidos na Tabela 3.5. Esse valor para y^2 excederia a demanda do terceiro item, então diminuimos uma unidade do y^2 , obtendo os resultados da Tabela 3.6, que não excede a demanda. Como somente o x_2 era fracionário, atualizamos a demanda conforme (3.11) e iniciamos a terceira iteração.

	1	2	3
y^2	1	1	0
B_3y^1	0	2	12
Demanda	0	2	8

Tabela 3.5: 1° solução inteira para o x_2^2 .

	1	2	3
y^2	1	0	0
B_3y^1	0	2	6
Demanda	0	2	8

Tabela 3.6: 2° solução inteira para o x_2^2 .

$$r = \begin{bmatrix} 0 \\ 2 \\ 8 \end{bmatrix} - \begin{bmatrix} 0 & 0 & 1 \\ 2 & 6 & 0 \\ 6 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix}. \quad (3.11)$$

Iteração 3

Resolvemos o Problema Residual 3.12. Obtendo como base ótima e solução as matrizes representadas em (3.13)

$$\begin{aligned} \text{minimizar} \quad & x_1 + x_2 + x_3 \\ \text{sujeito a} \quad & x_1 = 0 \\ & x_2 = 0 \\ & x_3 = 2 \\ & x_i \in \mathbb{R}_+^3, \quad i = 1, 2, 3 \end{aligned} \quad (3.12)$$

$$B_4 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 2 & 0 & 0 \end{bmatrix}, \quad x^4 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}. \quad (3.13)$$

Como a solução encontrada é inteira, temos que ela é ótima. Atualizando a demanda residual, obtemos $(0, 0, 0)^t$. Como todos os itens foram produzidos, recuperamos a solução. A Tabela 3.7 contém os padrões de corte (PC) que foram utilizados e a frequência de cada um, o valor da função objetivo é o somatório de todas as frequências que é igual a 141. E nas Figuras 3.1, 3.2 e 3.3 segue uma ilustração dos padrões de corte obtidos.

	PC1	PC2	PC3	PC3	PC3	Σ frequências
Item 1	16	0	0	0	0	
Item 2	0	6	3	2	0	
Item 3	0	0	6	6	2	
Frequência	60	27	52	1	1	141

Tabela 3.7: Resultados obtidos com a heurística residual 1.

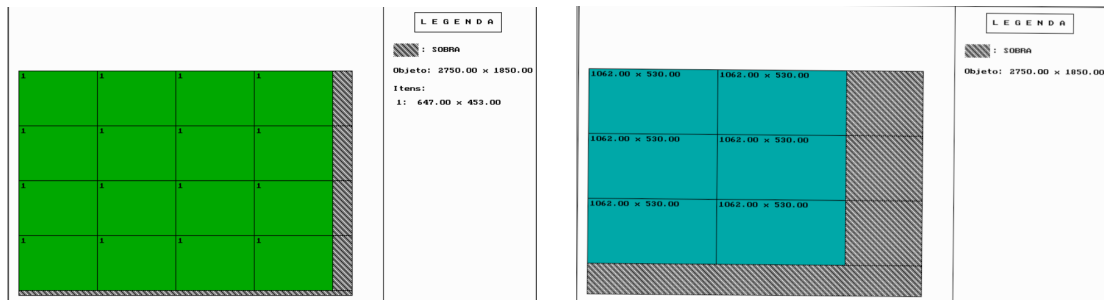


Figura 3.1: Padrões de Corte 1 e 2 usados no Exemplo 2 com a HR1.
Fonte: Sistema CorteBifur.

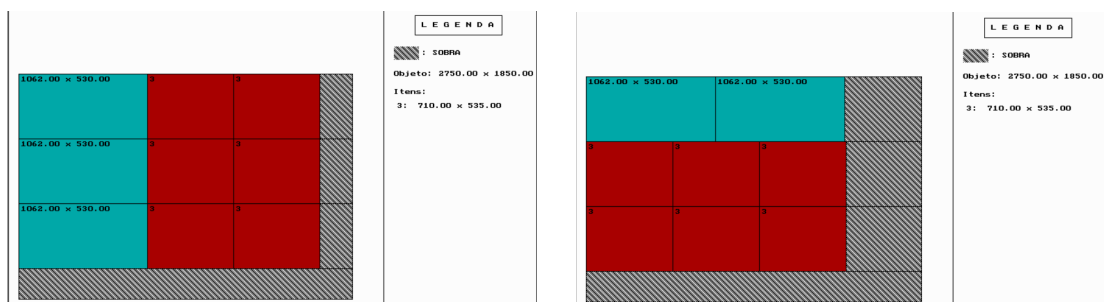


Figura 3.2: Padrões de Corte 3 e 4 usados no Exemplo 2 com a HR1.
Fonte: Sistema CorteBifur.

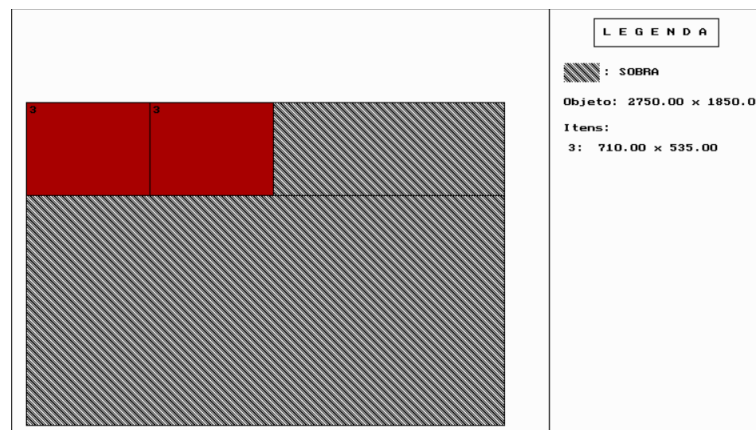


Figura 3.3: Padrão de Corte 5 usado no Exemplo 2 com a HR1.
Fonte: Sistema CorteBifur.

3.4 Heurística residual 2 (HR2) - Cui e Zhao (2013)

Cui e Zhao (CUI; ZHAO, 2013) propuseram uma heurística residual que se diferencia da HR1 apenas no método em determinar a solução inteira. Dada a solução fracionária x^k , faça $y^k = \lfloor x^k \rfloor$ e atualize a demanda residual. Se for possível aumentar alguma das frequências sem exceder a demanda, a frequência é aumentada e a demanda residual é atualizada novamente. O critério utilizado para ordenar os padrões de corte é a ordenação por ordem não crescente da área usada, calculada de acordo com o critério de ordenação 3 da seção 3.3.

No Algoritmo 4, segue uma descrição da heurística. Cui e Zhao desenvolveram um algoritmo especial para gerar padrões de corte para a HR2. Neste trabalho, implementamos a heurística HR2 gerando padrões de corte com o método de Gilmore e Gomory. No Exemplo 3 ilustramos a HR2.

Algoritmo 4: HEURÍSTICA RESIDUAL HR2 PROPOSTA POR CUI E ZHAO (2013)

Entrada: $r = b, A = B, k = 1$

1 **enquanto** $r_i > 0, i = 1 \dots m$ **faça**

2 Se $k = 1$, resolva o Problema (3.1) não exato por geração de colunas e tem B como base inicial.

3 Se $k > 1$, resolva o Problema 3.2 exato.

4 Seja x^k a solução encontrada.

5 **se** $x^k \in N$ **então**

6 PARE

7 **senão**

8 **para** $i = 1 \dots m$ **faça**

9 $y_i^k = \lceil x_i^k \rceil$

10 **fim**

11 $r^k \leftarrow r^k - By^k$

12 **fim**

13 **fim**

14 Ordene os padrões de corte por ordem não crescente da área utilizada.

15 **para** $j = 1, \dots, m$ **faça**

16 $\Delta = \min\{\lfloor \frac{r_i}{a_{ij}} \rfloor, a_{ij} > 0\}$ **se** $\Delta > 0$ **então**

17 $y_j^k = y_j^k + \Delta$ $r^k \leftarrow r^k - By^k$

18 **fim**

19 **fim**

20 $k \leftarrow k + 1$

21 **fim**

Saída: y

Exemplo 3.

Considere a instância do Exemplo 1. Aplicamos a Heurística Residual (HR2).

Primeira Iteração

Temos que a base e a solução ótima serão as mesmas do Exemplo 2 para o problema inicial relaxado. De acordo com (3.7) e (3.6), respectivamente.

Como a solução não é inteira, precisamos aplicar a heurística de arredondamento de acordo com as linhas 7 – 10 do Algoritmo 4, os resultados estão apresentados na Tabela 3.8, sendo

que na segunda linha são os valores da solução após serem arredondados, e na terceira linha a demanda residual atualizada. No próximo passo verificamos se é possível aumentar alguma variável sem exceder a demanda. O critério de ordenação das frequências é por ordem não crescente de área utilizada. Assim o vetor será analisado da seguinte forma: x_2, x_3 .

	1	2	3
y^1	60	26	53
$r_1 - B_2 y^1$	0	5	2

Tabela 3.8: Arredondamento de acordo com a HR2.

Continuando com a aplicação da heurística de acordo com as linhas 14 – 20. Desejamos saber se é possível aumentar a frequência dos padrões de corte sem ultrapassar a demanda residual. Para isso determinamos se alguma das variáveis pode receber um incremento, de acordo com a linha 15 do Algoritmo 4. O cálculo para o x_2 está ilustrado em 3.14, e temos que não é possível aumentar a sua frequência, o x_3 está apresentado em 3.15 e também não é possível aumentá-lo, logo passamos para a segunda iteração.

$$\Delta = \lfloor \min\left\{\frac{5}{6}\right\} \rfloor = 0 \quad (3.14)$$

$$\Delta = \lfloor \min\left\{\frac{3}{5}, \frac{2}{6}\right\} \rfloor = 0 \quad (3.15)$$

Segunda Iteração

Começamos novamente utilizando a matriz identidade como a base inicial, e resolvemos o Problema (3.16)

$$\begin{array}{rcl}
 \text{minimizar} & x_1 + & x_2 + & x_3 \\
 \text{sujeito a} & x_1 & & = 0 \\
 & & x_2 & = 5 \\
 & & & x_3 = 2 \\
 & & & x_i \in \mathbb{R}_+^3, \quad i = 1, 2, 3
 \end{array} \quad (3.16)$$

Após a geração de colunas, a base e a solução ótima encontradas foram

$$B_3 = \begin{bmatrix} 0 & 1 & 0 \\ 5 & 0 & 0 \\ 2 & 0 & 1 \end{bmatrix}, x_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

Como a solução encontrada já é inteira, o arredondamento não precisa ser aplicado. Atualizamos a demanda residual e obtemos $(0, 0, 0)^t$. Assim, a demanda foi satisfeita e agora basta encontrar a solução geral do problema, fazemos da mesma forma que foi descrito na Heurística 1. O valor da função objetivo foi de 140, os resultados encontrados estão descritos na Tabela 3.9 e nas Figuras 3.4 e 3.5, é ilustrado os padrões de cortes obtidos.

	PC1	PC2	PC3	PC4	Σ frequências
Item 1	16	0	0	0	
Item 2	0	6	3	5	
Item 3	0	0	6	2	
Frequência	60	26	53	1	140

Tabela 3.9: Resultados obtidos com a heurística residual 2.

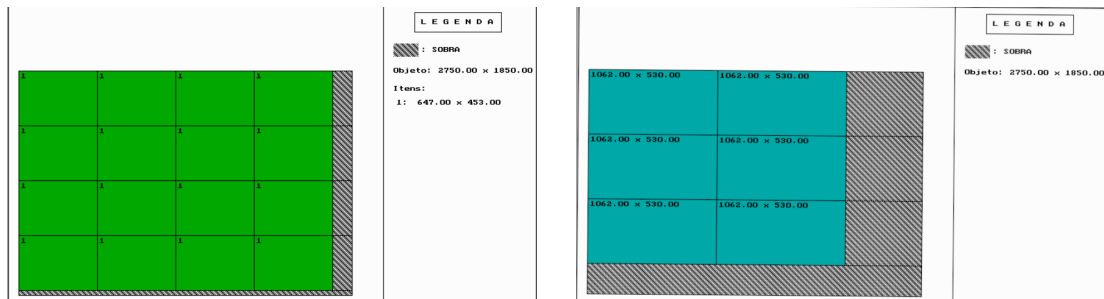


Figura 3.4: Padrões de Corte 1 e 2 usados no Exemplo 3 com a HR2.

Fonte: Sistema CorteBifur.

Comparando o Exemplo 2 e 3, temos que a função objetivo do Exemplo 2 foi igual a 141 e usou cinco padrões de corte diferentes, enquanto que o Exemplo 3, a função objetivo foi de 140 e utilizou somente três padrões de corte.

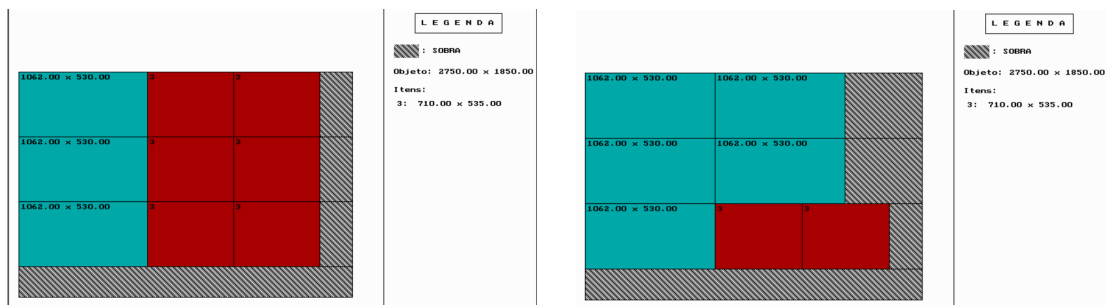


Figura 3.5: Padrões de Corte 3 e 4 usados no Exemplo 3 com a HR2.
Fonte: Sistema CorteBifur.

3.5 Contribuição: modificações para a HR2

A heurística residual proposta por Cui e Zhao, HR2, foi testada somente com a ordenação por ordem não crescente da área utilizada por cada padrão que é equivalente a ordenar por ordem não decrescente de perda (linha 13 do Algoritmo 4). Então, além dessa, fizemos testes com a ordenação natural, ordem não crescente das frequências utilizadas e ordem não crescente da parte fracionária das frequências, descritas abaixo. Tais mudanças são chamadas de HR2M.

1. Ordenação natural x_1^k, x_2^k, \dots, x_m ;
2. Ordem não crescente das frequências: $x_1^k \geq x_2^k \geq \dots \geq x_m$;
3. Ordem não crescente da parte fracionária: $f_1 \geq f_2 \geq \dots \geq f_m$;
4. Ordem não crescente da área: $a_1 \geq a_2 \geq \dots \geq a_m$.

Ordenações para as frequências dos padrões de corte para a HR2.

Capítulo 4

MÉTODO COMPUTACIONAL

Nesse capítulo apresentamos os resultados computacionais dos testes realizados com as heurísticas residuais HR1 e HR2 e um estudo comparativo entre elas e outros métodos de solução para o PCE2D (o sistema CorteBiFur, CPLEX e resultados encontrados na literatura). Os testes foram realizados em um equipamento com Windows 10 PRO, processador Intel(R), Core(TM) i7 260 CPU 3.4GHz. Os métodos foram implementados em Julia (THE...), (BEZANSON et al., 2014), (CASTELUCCI, 2017) com o auxílio do pacote JuMP, e os problemas de otimização envolvidos foram resolvidos pelo CPLEX versão 12.7 (IBM...).

4.1 Descrição das instâncias

Foram usadas 33 instâncias divididas em três conjuntos. As instâncias do primeiro conjunto (Conjunto 1), apresentadas na Tabela 4.1, foram retiradas da OR-LIBRARY (IBM,) e adaptadas por Cintra que gerou demandas para os itens no intervalo $[1 - 100]$. O segundo e o terceiro conjuntos (Conjuntos 2 e 3) são descritos nas Tabelas 4.2 e 4.3, são baseados em dados reais de uma Fábrica de Móveis, e foram retirados de (FIGUEIREDO; RANGEL, 2008) e (FIGUEIREDO; RANGEL, 2006). As instâncias na Tabela 4.2 estão associadas a três tipos de móveis $P1, P2, P3$ produzidos com objetos de diferentes espessuras. O nome $P1 - 03$, representa a instância contendo itens cortados a partir de objetos com espessura de 3mm. A Tabela 4.3 ilustra uma característica da empresa em criar lotes duplos, esse lote é chamado de heterogêneo se nele existem dois produtos diferentes, por exemplo, a instância $P1, P2 - 03$ contém itens associados aos móveis $P1$ e $P2$ que devem ser obtidos a partir de objetos com espessura de 3mm. Nas Tabelas 4.1, 4.2 e 4.3, m representa a quantidade de itens de cada instância, (L, W) o comprimento e a largura dos objetos, e nas colunas associadas a l_i, w_i e b_i é exibido o intervalo associado ao comprimento, largura e demanda, respectivamente.

Instâncias	m	(L, W)	l_i	w_i	b_i
<i>Gcutd1</i>	10	(250, 250)	[66, 167]	[86, 184]	[13, 90]
<i>Gcutd2</i>	20	(250, 250)	[18, 168]	[68, 186]	[2, 96]
<i>Gcutd3</i>	30	(250, 250)	[63, 176]	[71, 186]	[2, 100]
<i>Gcutd4</i>	50	(250, 250)	[62, 184]	[63, 179]	[7, 96]
<i>Gcutd5</i>	10	(500, 500)	[132, 364]	[145, 356]	[27, 94]
<i>Gcutd6</i>	20	(500, 500)	[132, 355]	[134, 372]	[6, 93]
<i>Gcutd7</i>	30	(500, 500)	[129, 365]	[147, 374]	[6, 97]
<i>Gcutd8</i>	50	(500, 500)	[131, 362]	[127, 374]	[3, 97]
<i>Gcutd9</i>	10	(1000, 1000)	[292, 673]	[341, 688]	[18, 95]
<i>Gcut10</i>	20	(1000, 1000)	[269, 730]	[260, 742]	[2, 99]
<i>Gcutd11</i>	30	(1000, 1000)	[266, 674]	[274, 745]	[1, 99]
<i>Gcut12d</i>	32	(3000, 3000)	[365, 970]	[116, 1890]	[4, 97]

Tabela 4.1: Conjunto 1 de instâncias.

Instâncias	m	(L, W)	l_i	w_i	b_i
<i>P1 – 03</i>	3	(2750, 1830)	[388, 445]	[213, 383]	[600, 1200]
<i>P1 – 12</i>	2	(2750, 1830)	[370, 390]	[110, 110]	[900, 1800]
<i>P1 – 15</i>	2	(2750, 1830)	[450, 600]	[132, 440]	[600, 900]
<i>P2 – 03</i>	3	(2750, 1830)	[647, 1062]	[453, 535]	[320, 960]
<i>P2 – 09</i>	3	(2750, 1830)	[295, 630]	[50, 50]	[320, 480]
<i>P2 – 12</i>	6	(2750, 1830)	[440, 635]	[50, 180]	[160, 960]
<i>P2 – 15</i>	4	(2750, 1830)	[700, 970]	[75, 570]	[160, 480]
<i>P3 – 03</i>	8	(2750, 1830)	[454, 2500]	[215, 565]	[40, 240]
<i>P3 – 12</i>	2	(2750, 1830)	[454, 635]	[180, 180]	[160, 320]
<i>P3 – 15</i>	7	(2750, 1830)	[499, 1050]	[212, 535]	[80, 200]
<i>P3 – 20</i>	9	(2750, 1830)	[430, 2500]	[40, 60]	[80, 991040]
<i>P3 – 20</i>	3	(2750, 1830)	[430, 1050]	[60, 60]	[80, 160]

Tabela 4.2: Conjunto 2 de instâncias.

Nas Tabelas 4.4, 4.5 e 4.6 apresentamos uma descrição das características das instâncias dos conjuntos 1, 2 e 3, considerando a diversidade das dimensões dos itens. A diversidade de uma instância é calculada como a razão entre o número de dimensões distintas (comprimento/largura) das instâncias desse conjunto e o número total de itens. Podemos observar que as instâncias do Conjunto 1 possuem uma alta diversidade, entre 70% e 100% relativos ao comprimento e entre 87% e 100% relativos à largura. Assim, segundo a tipologia de Wäscher (WÄSCHER; HAUBNER; SCHUMANN, 2007) as instâncias do Conjunto 1 têm um sortimento fortemente heterogêneo. Podemos notar que para as instâncias do Conjunto 2 apesar, de algumas delas possuírem uma diversidade baixa em relação à largura como a *P3 – 20*, 22,22%, a diversidade do comprimento é bem alta. Das 13 instâncias incluídas nesse conjunto somente

Instâncias	m	(L, W)	l_i	w_i	b_i
$P1, P2 - 03$	6	(2750, 1830)	[388, 1062]	[213, 535]	[320, 1200]
$P1, P2 - 12$	8	(2750, 1830)	[390, 635]	[50, 180]	[160, 1800]
$P1, P2 - 15$	6	(2750, 1830)	[450, 970]	[75, 570]	[160, 900]
$P1, P3 - 03$	11	(2750, 1830)	[380, 2500]	[213, 535]	[40, 1200]
$P1, P3 - 12$	4	(2750, 1830)	[370, 365]	[110, 180]	[160, 1800]
$P1, P3 - 15$	9	(2750, 1830)	[450, 1049]	[132, 535]	[80, 900]
$P2, P3 - 03$	11	(2750, 1830)	[454, 2500]	[215, 565]	[40, 960]
$P2, P3 - 12$	8	(2750, 1830)	[440, 635]	[50, 180]	[160, 960]
$P2, P3 - 15$	11	(2750, 1830)	[454, 1049]	[75, 570]	[80, 480]

Tabela 4.3: Conjunto 3 de instâncias (lote duplo).

quatro delas não apresentam a diversidade igual à 100%, assim esse conjunto também é heterogêneo. Para o conjunto 3, apresentado na Tabela 4.6, notamos que a diversidade é bem menor do que a diversidade das instâncias dos lotes simples, porém podemos perceber que a heterogeneidade é de média para alta, pois apenas a instância $P2, P3 - 12$, possui diversidade menor que 50%. A heterogeneidade é média se o valor estiver entre 50% – 70% e alta a partir de 70%. De um modo geral, todas as instâncias utilizadas nessa dissertação possuem itens heterogêneos.

Instâncias	m	$\dim l_i / \dim \text{total } l_i$	Diversidade	$\dim w_i / \dim \text{total } w_i$	Diversidade
$Gcutd1$	10	9/10	90%	10/10	100%
$Gcutd2$	20	18/20	90%	19/20	95%
$Gcutd3$	30	30/30	100%	26/30	87%
$Gcutd4$	50	39/50	78%	44/50	88%
$Gcutd5$	10	10/10	100%	10/10	100%
$Gcutd6$	20	19/20	95%	19/20	95%
$Gcutd7$	30	27/30	90%	28/30	93%
$Gcutd8$	50	46/50	92%	46/50	92%
$Gcutd9$	10	10/10	100%	10/10	100%
$Gcutd10$	20	20/20	100%	20/20	100%
$Gcutd11$	30	30/30	100%	30/30	100%
$Gcutd12$	32	50/50	100%	50/50	96%

Tabela 4.4: Descrição da diversidade das dimensões dos itens para as instâncias do Conjunto 1.

Instâncias	m	$\dim l_i/\dim \text{total } l_i$	Diversidade	$\dim l_i/\dim \text{total } l_i$	Diversidade
$P1 - 03$	3	3/3	100%	3/3	100%
$P1 - 12$	2	2/2	100%	1/2	50%
$P1 - 15$	2	2/2	100%	2/2	100%
$P2 - 03$	3	3/3	100%	3/3	100%
$P2 - 09$	3	3/3	100%	3/3	33,33%
$P2 - 12$	6	4/6	66,67%	1/3	66,67%
$P2 - 15$	4	2/4	50%	4/6	100%
$P3 - 03$	8	8/8	100%	4/4	62,5%
$P3 - 12$	2	2/2	100%	5/8	50%
$P3 - 15$	7	6/7	85,71%	1/2	85,71%
$P3 - 20$	9	8/9	88,89%	6/7	22,22%
$P3 - 20$	3	3/3	100%	1/3	33,33%

Tabela 4.5: Descrição da diversidade das dimensões dos itens para as instâncias do Conjunto 2.

Instâncias	m	$\dim l_i/\dim \text{total } l_i$	Diversidade	$\dim l_i/\dim \text{total } l_i$	Diversidade
$P1, P2 - 03$	6	(2750, 1830)	100%	[213, 535]	100%
$P1, P2 - 12$	8	(2750, 1830)	62,5%	[50, 180]	62,5%
$P1, P2 - 15$	6	(2750, 1830)	66,67%	[75, 570]	100%
$P1, P3 - 03$	11	(2750, 1830)	100%	[213, 535]	72,72%
$P1, P3 - 12$	4	(2750, 1830)	100%	[110, 180]	50%
$P1, P3 - 15$	9	(2750, 1830)	88,88%	[132, 535]	88,88%
$P2, P3 - 03$	11	(2750, 1830)	91%	[215, 565]	72,72%
$P2, P3 - 12$	8	(2750, 1830)	37,5%	[50, 180]	50%
$P2, P3 - 15$	11	(2750, 1830)	63,64%	[75, 570]	81,81%

Tabela 4.6: Descrição da diversidade das dimensões dos itens para as instâncias do Conjunto 3.

4.2 Testes computacionais realizados e parâmetros para análise dos resultados

No estudo computacional foram realizados quatro testes (Teste 1, Teste 2, Teste 3 e Teste 4). Os Teste 1 e 2 foram propostos para comparar as Heurísticas HR1 e HR2 e determinar a mais eficiente para as instâncias testadas, enquanto que os Testes 3 e 4, são para comparar a melhor heurística residual com dados da literatura e outros métodos de resolução. Em todos os métodos implementados foram gerados padrões de corte usando o método 2– estágios de Gilmore e Gomory descrito na seção 2.2.1. Os resultados são apresentados considerando dois casos: Caso exato (Seção 4.3) e Caso Não Exato (Seção 4.4). No caso exato as heurísticas residuais foram implementadas para sempre resolverem o problema exato. No caso não exato, para a HR1 todas as iterações são resolvidas utilizando o problema não exato. Para a HR2 somente

na primeira iteração resolvemos o problema não exato, enquanto que nas demais iterações o problema a ser resolvido é exato. Na Tabela 4.7 apresentamos um resumo dos testes realizados.

Teste 1	Determinar a melhor ordenação dos padrões de corte para a HR1 e a HR2.
Teste 2	Determinar a melhor heurística residual entre as determinadas no Teste 1.
Teste 3	Comparar a heurística residual determinada no Teste 2, para as instâncias do Conjunto 1, com o CorteBiFur, ZIP, HCP e resultados da literatura.
Teste 4	Comparar a heurística residual determinada no Teste 2, para as instâncias do Conjunto 2 e 3, com o CorteBiFur, ZIP e aHCP.

Tabela 4.7: Resumo dos Testes de 1 a 4.

Teste 1 - Melhor Estratégia de Ordenação

O objetivo deste teste é determinar qual é o melhor critério de ordenação dos padrões (linha 7 do Algoritmo 3 e linha 3 do Algoritmo 4) em cada uma das heurísticas estudadas HR1 e HR2. As ordenações avaliadas foram:

- FNa: ordenação natural;
- FAr: ordem não crescente da área utilizada;
- FFc: ordem não crescente das frequências;
- FFr: ordem não crescente da parte fracionária das frequências.

Teste 2 - Comparação entre as melhores estratégias para a HR1 e HR2

O objetivo do Teste 2 é considerando a melhor estratégia de ordenação para a HR1 e para a HR2, definida no Teste 1, determinar a melhor heurística residual (MHR).

Teste 3 - Comparação da melhor heurística residual com outros métodos para as instâncias do Conjunto 1

O objetivo do Teste 3 é comparar a melhor estratégia definida do Teste 2 com outros métodos e resultados da literatura. Esse teste foi realizado apenas para as instâncias do Conjunto 1, por não dispormos de resultados para as instâncias dos demais conjuntos. Para o caso exato consideramos a MHR, o Sistema CorteBiFur, o Z_{IP} e a implementação própria da Heurística de Costa e Poldi (HCP). Para o caso não exato são consideradas a MHR, o Sistema CorteBiFur, o Z_{IP} e os resultados apresentados em (CUI; ZHAO, 2013), (CINTRA et al., 2008) e (COSTA; POLDI, 2016).

Os algoritmos de Cintra (CINTRA et al., 2008) e Cui e Zhao (CUI; ZHAO, 2013) estão descritos no Capítulo 3. O método de Costa e Poldi (COSTA; POLDI, 2016) e o sistema CorteBiFur também constroem uma solução inteira factível a partir da solução da relaxação linear do Problema (3.1) resolvido pelo método simplex com geração de colunas. Para obter a solução inteira, Costa e Poldi (POLDI; ARENALES, 2006) utilizam o seguinte critério: se a parte fracionária de x_i^* é maior ou igual a 0,5, então $x_i = \lceil x_i \rceil$, caso contrário $x_i = \lfloor x_i \rfloor$. Se a solução obtida com esse critério não satisfazer a demanda, retomam a solução fracionária, arredondam para cima e utilizam o Método BRURED (WÄSCHER; GAU, 1996). O Sistema CorteBiFur no caso exato, utiliza a função piso (ver expressão (3.4)), que gera uma solução que na maioria das vezes não atende toda a demanda. No caso não exato é aplicado a função teto. O método Z_{IP} consiste em utilizar todas as colunas obtidas com HR2-FFr para compor a matriz A do Problema e resolver o problema inteiro pelo método *Branch-and-Cut* incluído no software. Abaixo apresentamos um resumo do Teste 3.

Teste 3 (Caso exato)	Teste 3 (Caso não exato)
<ul style="list-style-type: none"> • MHR; • Implementação própria do método de Costa e Poldi (HCP) (COSTA; POLDI, 2016); • Sistema CorteBiFur; • Z_{IP}. 	<ul style="list-style-type: none"> • MHR; • Sistema CorteBiFur; • Z_{IP}; • Cintra <i>et al</i> (caso irrestrito) (CINTRA et al., 2008); • Cui e Zhao (CUI; ZHAO, 2013); • Costa e Poldi (COSTA; POLDI, 2016).

Resumo dos métodos utilizados para as instâncias do Conjunto 1 para o caso exato e não exato.

Teste 4 - Comparação da melhor heurística residual com outros métodos para as instâncias dos Conjuntos 2 e 3

No Teste 4 fazemos uma aplicação do Teste 3 para às instâncias dos Conjuntos 2 e 3. Para essas instâncias não há resultados na literatura. Por isso foi necessária fazer a implementação do método de Costa e Poldi (COSTA; POLDI, 2016). Os métodos comparados são:

- MHR;
- Implementação própria do método de Costa e Poldi (HCP) (COSTA; POLDI, 2016);
- Z_{IP} ;
- Sistema CorteBiFur.

Parâmetros para análise

Para analisar os resultados dos Testes 1, 2, 3 e 4 foi usado o valor da função objetivo (Z_{lb}) e o gap de integrabilidade (gap), calculado de acordo com a expressão (4.1)

$$gap = \frac{f(x) - Z_{lb}}{f(x)}, \quad (4.1)$$

sendo $f(x)$ o valor da função objetivo obtido com o método testado e Z_{lb} o teto da solução ótima do problema mestre restrito relaxado.

Os resultados dos Testes 3 e 4 foram analisados considerando além do valor da função objetivo (F) e do gap (G), a quantidade de padrões gerados (P), e o tempo de resolução para cada instância (T).

4.3 Resultados e discussão: caso exato

Teste 1

Comparamos as heurísticas com diferentes tipos de ordenação do vetor de solução (linha 7 do algoritmo 3 e linha 13 do algoritmo 4). Na Tabela 4.8, apresentamos os resultados relativos à HR1 e na Tabela 4.9 em relação à HR2. A primeira coluna exibe o nome das instâncias, na segunda, quarta, sexta e oitava são exibidas o valor da função objetivo considerando as

diferentes formas de ordenação, sendo elas a ordenação natural (FNa), por área utilizada (FAr), ordem não crescente das frequências (FNc) e ordem não crescente das partes fracionárias das frequências (FFr), e nas colunas três, cinco, sete e nove o gap (G) associado.

Instâncias	Heurística 1							
	FNa	G	FAr	G	FNc	G	FFr	G
<i>Gcutd1</i>	296	0,34	296	0,34	295	0	295	0
<i>Gcutd2</i>	346	0,29	346	0,29	346	0,29	345	0
<i>Gcutd3</i>	343	0,29	345	0,87	343	0,29	343	0
<i>Gcutd4</i>	847	0,24	847	0,24	846	0,12	845	0
<i>Gcutd5</i>	208	0,48	250	17,2	208	0,48	208	0,48
<i>Gcutd6</i>	376	0,27	376	0,27	377	0,53	377	0,53
<i>Gcutd7</i>	602	0,33	602	0,33	602	0,33	602	0,33
<i>Gcutd8</i>	723	0,41	723	0,41	721	0,14	721	0,14
<i>Gcutd9</i>	135	0	135	0	135	0	136	0,73
<i>Gcutd10</i>	316	0,32	317	0,63	316	0,32	317	0,63
<i>Gcutd11</i>	352	0,85	350	0,29	350	0,29	351	0,57
<i>Gcutd12</i>	677	0,29	677	0,29	675	0	677	0,29

Tabela 4.8: Teste 1 : definição da melhor estratégia de ordenação para a HR1 para o caso exato.

Para determinar a melhor ordenação para a HR1, analisamos quantas soluções ótimas cada uma teve e para quantas instância o gap foi menor ou igual que as demais ordenações. A HR1 com a ordenação FNc e FFr encontrou o maior número de soluções ótimas (3) e com a ordenação FNc o método obteve a melhor quantidade de melhores gaps (9) enquanto que com a ordenação FFr o método obteve 3 soluções ótimas. Logo, a ordenação FFr foi escolhida como a melhor para a HR1.

De forma similar, analisamos os resultados para a HR2, de acordo com a Tabela 4.9.

Instâncias	Heurística 2							
	FNa	G	FAr	G	FNc	G	FFr	G
<i>Gcutd1</i>	295	0	295	0	295	0	295	0
<i>Gcutd2</i>	346	0,29	347	0,58	346	0,29	346	0,29
<i>Gcutd3</i>	344	0,58	347	1,44	342	0	343	0,29
<i>Gcutd4</i>	850	0,59	865	1,85	847	0,24	847	0,24
<i>Gcutd5</i>	208	0,48	250	17,2	208	0,48	208	0,48
<i>Gcutd6</i>	377	0,53	379	1,05	376	0,27	375	0
<i>Gcutd7</i>	601	0,17	601	0,17	601	0,17	601	0,17
<i>Gcutd8</i>	722	0,28	724	0,55	723	0,41	721	0,14
<i>Gcutd9</i>	135	0	138	2,17	135	0	135	0
<i>Gcutd10</i>	316	0,32	316	0,32	316	0,32	316	0,32
<i>Gcutd11</i>	351	0,57	351	0,57	350	0,28	349	0
<i>Gcutd12</i>	676	0,15	675	0	676	0,15	677	0,29

Tabela 4.9: Teste 1 : definição da melhor estratégia de ordenação para a HR2 para o caso exato.

Temos que a ordenação FFr encontrou o maior número de soluções ótimas (4) e também obteve a melhor quantidade de melhores gaps (10). Dessa forma, a ordenação FFr foi escolhida como a melhor para a HR2 nesse conjunto de instâncias.

Teste 2

Comparamos a HR1-FNc com a HR2-FFr que foram as melhores ordenações definidas no Teste 1. A quantidade de soluções ótimas que cada uma obteve e quantas vezes o gap foi melhor ou igual que a outra foram. A HR1-FNc obteve três soluções ótimas e nove instâncias tiveram gap menor ou igual a HR2-FFr. A HR2-FFr obteve quatro soluções ótimas e dez gaps melhores. Como a HR2-FFr teve um melhor desempenho em ambos os critérios, ela foi escolhida como a melhor heurística.

Teste 3

Estabelecida a melhor heurística, usamos o primeiro conjunto de instâncias para realizar as comparações entre a HR2-FFr, o Sistema CorteBiFur, o Z_{IP} e a HCP. Na Tabela 4.10 apresentamos os resultados encontrados, sendo que a primeira colunas representa o nome das instâncias do Conjunto 1, a segunda o teto do valor da função objetivo ao resolver o problema mestre restrito relaxado (Z_{lb}), as colunas com a letra F , representam a função objetivo de cada método,

G é o gap, P é o total de colunas utilizadas e T o tempo em segundos de resolução de cada instância.

I	$[Z_{lb}]$	HR2-FFr			CorteBi			Z_{IP}			HCP						
		F	G%	T	F	G%	T	F	G%	T	F	G%	T				
<i>Gcutd1</i>	295	295	0	24	11,6399	294*	-	24	0,08	295	0	34	0,0593	295	0	12	9,1163
<i>Gcutd2</i>	345	346	0,29	82	15,5284	343*	-	94	0,01	346	0,29	82	8,1624	350	1,43	40	11,6563
<i>Gcutd3</i>	342	343	0,58	141	27,2990	341*	-	148	0,02	343	0,29	141	0,1180	356	3,93	62	19,6900
<i>Gcutd4</i>	845	847	0,24	231	60,4858	971*	-	240	0,06	846	0,12	231	0,152	863	15,25	72	30,6549
<i>Gcutd5</i>	207	208	0,48	32	11,5674	207*	-	24	$<10^{-2}$	208	0,48	32	0,0579	208	0,48	16	8,7223
<i>Gcutd6</i>	375	375	0	65	14,8959	374*	-	62	0,01	375	0	65	0,0977	381	1,57	23	10,2089
<i>Gcutd7</i>	600	601	0,17	105	22,8798	599*	-	115	0,03	600	0	105	0,0910	604	0,66	43	14,9781
<i>Gcutd8</i>	720	721	0,14	204	60,1752	**	-	-	-	720	0	204	0,137	733	1,77	85	35,0084
<i>Gcutd9</i>	135	135	0	59	11,8091	133*	-	26	0,01	135	0	59	0,0651	137	1,46	13	9,0988
<i>Gcutd10</i>	315	316	0,32	70	14,4689	314*	-	56	0,01	316	0,07	70	0,0666	317	0,94	49	10,4514
<i>Gcutd11</i>	349	349	0	113	24,2963	203*	-	70	0,02	349	0	113	0,0744	356	1,96	40	4,83
<i>Gcutd12</i>	675	677	0,29	190	52,5530	747*	-	197	0,06	675	0	190	0,112	690	2,17	97	38,1155

Tabela 4.10: Teste 3 : resultados das instâncias do Conjunto 1 para o caso exato.

*a demanda não foi atendida por completo.

**o sistema CorteBiFur não conseguiu resolver devido a um erro no código.

Analisando a Tabela 4.10 temos que:

- em relação a quantidade de soluções ótimas temos que a HR2-FFr obteve 4/12. O Sistema CorteBiFur não obteve solução factível para as instâncias desse conjunto e para a instância *Gcutd18* os sistema apresentou um erro no método simplex. A solução obtida não atendeu a demanda por completo devido ao critério de arredondamento utilizado pelo software (função piso). O Z_{IP} obteve 7/12, e em todas as instâncias que o Z_{IP} obteve resultados melhores que HR2-FFr foi com diferença de apenas uma unidade no valor de F. A HCP obteve uma solução ótima.
- em relação ao gap, a HR2-FFr obteve 6/12 dos melhores ou iguais resultados. O Corte-BiFur nenhum, devido ao fato de não ter atendido as demandas. O Z_{IP} obteve 12/12, e a HCP 2/12.
- a média aritmética de padrões de corte gerados por cada método é de 101 para a HR2-FFr, e por consequência o mesmo para o Z_{IP} , de 86 e 43 para o Sistema CorteBiFur e a HCP, respectivamente. Esses dois últimos métodos possuem menos colunas geradas, pois resolvem só uma iteração do problema, ou seja, não possuem o problema residual. Além disso, a HCP inicia com a base homogênea maximal enquanto que o CorteBiFur e a HR2-FFr inicia com a base homogênea.
- em relação ao tempo de solução, o método que possui os melhores tempos é o Sistema CorteBiFur. A programação dinâmica é usada na geração dos padrões de corte. O Z_{IP} também obteve um tempo menor que a HR2-FFr, pois não são gerados novos padrões de corte. O tempo da HR2-FFr se justifica pela quantidade de padrões criados, podemos observar que na instância *Gcut4* que foi a que mais teve padrões de corte o tempo computacional foi muito maior que as demais instâncias.

Na Tabela 4.11 temos os resultados para as instâncias do Conjunto 2 ao comparar a heurística residual HR2–FFr, o Sistema CorteBiFur, o Z_{IP} e a HCP.

I	[Z _{lb}]	HR2-FFr			CorteBi			Z _{IP}			HCP						
		F	G%	T	F	G%	T	F	G%	T	F	G%	T				
P1-03	73	74	1,35	15	11,0178	73	0	15	< 10 ⁻²	74	1,35	15	0,0587	73	0	7	8,3759
P1-12	25	26	3,85	7	10,7326	24*	-	4	0,02	26	3,85	7	0,0552	26	3,85	3	7,9991
P1-15	47	47	0	7	10,7355	47*	-	4	< 10 ⁻²	47	0	7	0,0548	48	2,08	3	7,99
P2-03	140	140	0	10	10,8453	140*	-	7	< 10 ⁻²	140	0	10	0,520	141	0,71	5	8,2296
P2-09	6	7	14,29	12	10,8552	6*	-	8	0,01	7	14,29	12	0,0597	7	14,29	6	7,9998
P2-12	44	46	4,35	32	11,9235	44*	-	17	0,02	46	4,35	32	0,825	47	6,38	14	8,2610
P2-15	75	76	1,32	24	10,9884	75*	-	15	< 10 ⁻²	76	1,32	24	0,0647	75	0	11	8,4004
P3-03	134	134	0	29	11,8732	134*	-	19	< 10 ⁻²	135	0,74	29	0,0826	135	0,74	15	8,6905
P3-12	10	11	9,09	7	10,9835	9*	-	4	0,01	11	9,09	7	0,0565	10	0	3	7,9775
P3-15	57	60	5	44	12,5122	56*	-	16	0,01	65	1,72	44	0,120	59	3,39	19	8,6124
P3-20	29	30	3,33	48	11,9438	29	0	21	5,69	31	6,45	48	0,0768	32	9,37	19	8,5514
P3-25	4	4	0	16	10,4968	4*	-	4	0,01	7	0	16	0,0542	5	20,00	6	8,0232

Tabela 4.11: Teste 4 : resultados das instâncias do Conjunto 2 para o caso exato.

*a demanda não foi atendida por completo.

Para as instâncias do Conjunto 2, Tabela 4.11 temos que:

- a HR2-FFr obteve 4/12 soluções ótimas, e entre as instâncias com gap maior que zero a diferença entre o valor da função objetivo foi de três unidades para a P3-15, duas unidades para a P2 – 12, e para as demais instâncias apenas uma unidade. O sistema CorteBiFur conseguiu soluções factíveis para duas instâncias, a P1 – 03 e a P3 – 20 e em ambas obteve a solução ótima. O Z_{IP} obteve 3/12 soluções ótimas e a HCP obteve 2/12. Para esse conjunto de instâncias a HR2-FFr obteve o melhor desempenho.
- entre os gaps a HR2-FFr também obteve um desempenho computacional melhor, sendo que em 8/12 instâncias teve um gap melhor ou igual ao demais métodos, o CorteBiFur 2/12, que apesar de usar um método simples de arredondamento foi o único que conseguiu a solução ótima para as instâncias P–03 e P3 – 20, o Z_{IP} teve 7/12 e a HCP 5/12.
- a média aritmética de padrões de corte gerados por cada método é de 27 para HR2-FFr, e de 13 e 9 para o Sistema CorteBiFur e a HCP, respectivamente.
- novamente o CorteBiFur tem os melhores resultados em relação ao tempo computacional sendo que em 5/12 instâncias o tempo foi menor que 10^{-2} segundos. A HR2-FFr teve o maior tempo, pois o Z_{IP} não gera novas colunas e a HCP só possui uma iteração.

Na Tabela 4.12 temos os resultados para as instâncias do Conjunto 3 ao comparar a heurística residual HR2-FFr, o Sistema CorteBiFur, o Z_{IP} e a HCP.

I	$\lceil Z_{lb} \rceil$	HR2-FFr			CorteBi			ZIP			HCP						
		F	G%	P	T	F	G%	P	T	F	G%	P	T				
P1,P2-03	197	197	0	34	12,2253	197	0	19	0,01	197	0	34	0,0765	198	0,50	15	8,4215
P1,P2-12	53	55	3,77	36	12,1588	32*	-	16	0,01	55	3,77	36	0,0861	57	5,23	21	9,0857
P1,P2-15	67	70	4,29	25	11,5376	112*	-	15	0,02	69	7,02	25	0,0614	69	7,02	14	8,7096
P1,P3-03	125	127	1,57	61	14,1254	198*	-	43	0,02	127	1,57	61	16,2590	128	2,90	36	10,4681
P1,P3-12	33	34	2,94	19	11,3853	32*	-	12	0,01	37	10,81	19	0,940	34	2,94	11	8,6616
P1,P3-15	62	63	1,59	37	11,6541	96*	-	44	0,01	63	1,59	37	0,1620	66	6,45	24	9,3948
P2,P3-03	270	272	2,00	43	12,7028	270*	-	30	0,01	272	2,00	43	0,142	272	2,00	22	9,1252
P2,P3-12	53	55	3,64	34	12,0688	54*	-	22	0,02	55	3,64	34	0,1041	57	7,02	20	8,3801
P2,P3-15	75	75	0	30	11,3584	75*	-	22	0,01	78	3,85	30	0,1131	75	0	19	8,7105

Tabela 4.12: Teste 4 : resultados das instâncias do Conjunto 3 para o caso exato.

* a demanda não foi atendida por completo.

Analisando a Tabela 4.12, temos que:

- a HR2-FFr obteve duas soluções ótimas entre nove, enquanto os demais métodos obtiveram apenas uma solução ótima. As instâncias desse conjunto comparadas as instâncias anteriores tiveram menos soluções ótimas.
- em relação ao gap, a HR2-FFr possui oito soluções melhores ou iguais que os demais métodos, o Sistema CorteBiFur uma, o Z_{IP} sete e a HCP quatro.
- a HR2-FFr obteve uma média maior de padrões gerados para as instâncias desse conjunto 35, o Sistema CorteBiFur obteve 25 e a HCP 20.
- o tempo foi aproximadamente de 12,13 segundos para a HR2-FFr, o CorteBiFur teve tempos somente até 0,02 segundos, o Z_{IP} apenas para a instância P1-P3 demorou mais de um segundo e a HCP levou aproximadamente nove segundos em cada instância.

Nas instâncias do Conjunto 1 o Z_{IP} com um conjunto restrito de colunas foi o melhor método considerando que resolveu de forma ótima 58,33% das instâncias. Para as instâncias do Conjunto 2, o melhor método foi a HR2-FFr, que obteve otimalidade em 33,33% das instâncias. Para as instâncias do Conjunto 3, a HR2-FFr também apresentou o melhor desempenho e resolveu de forma ótima 22,22% das instâncias. Para o caso exato, notamos que as instâncias do Conjunto 3 foram as mais difíceis de obter solução ótima. Esses resultados podem ser justificados, pois as instâncias do Conjunto 2 e 3, de acordo com as Tabelas 4.5 e 4.6 se mostram mais diversificadas em relação a largura e comprimento dos itens, e ainda possuem um intervalo de demanda muito maior que as instâncias do Conjunto 1 (Tabelas 4.1, 4.2 e 4.3).

Apesar dos resultados para as instâncias do Conjunto 1 terem sido melhores do ponto de otimalidade, foi necessário gerar mais padrões de corte. Para o Conjunto 1 a HR2-FFr, teve uma média de 101 padrões de corte para cada instância. A HR2-FFr para os Conjuntos 2 e 3 obteve, respectivamente, uma média de 35 e 27 padrões de corte. Uma das características do Conjunto 1 que pode influenciar nesses resultados é que de acordo com a Tabela 4.1 esse conjunto possui demandas mais baixas, todas entre o intervalo de [1 – 100]. Consequentemente, como o Conjunto 1 teve o maior número de colunas geradas o seu tempo computacional também foi o maior.

De uma forma geral, os melhores gaps foram encontrados pelo método Z_{IP} no Conjunto 1. A HR2-FFr teve os melhores gaps para os Conjuntos 2 e 3. O método que menos gerou padrões de corte foi a HCP. E o Sistema CorteBiFur foi o método mais rápido.

4.4 Resultados e discussão: caso não exato

Teste 1

Os testes computacionais dessa seção são análogos aos da seção anterior, porém agora consideramos o caso não exato. Na Tabela 4.13 apresentamos os resultados para cada ordenação, usando a HR1.

Instâncias	Heurística 1							
	FNa	G	FAr	G	FNc	G	FFr	G
<i>Gcutd1</i>	296	0,34	296	0,34	295	0	295	0
<i>Gcutd2</i>	434	20,51	345	0	346	0,29	346	0,29
<i>Gcutd3</i>	344	0,58	345	0,87	344	0,58	344	0,58
<i>Gcutd4</i>	872	3,10	846	0,12	847	0,24	846	0,12
<i>Gcutd5</i>	208	0,48	208	0,48	208	0,48	208	0,48
<i>Gcutd6</i>	376	0,27	378	0,79	375	0	377	0,53
<i>Gcutd7</i>	602	0,33	604	0,66	602	0,33	601	0,17
<i>Gcutd8</i>	722	0,28	721	0,14	720	0	722	0,28
<i>Gcutd9</i>	135	0	136	0,73	135	0	135	0
<i>Gcutd10</i>	317	0,63	317	0,63	317	0,63	317	0,63
<i>Gcutd11</i>	357	2,24	350	0,29	356	1,96	351	0,57
<i>Gcutd12</i>	676	0,15	678	0,44	677	0,29	678	0,79

Tabela 4.13: Teste 1 : definição da melhor estratégia de ordenação para a HR1 para o caso não exato.

Com a ordenação FNc a HR1 obteve o maior número de soluções ótimas (4) e o maior número de melhores gaps 7/12. Logo, a ordenação FNc foi considerada a melhor ordenação para a HR1 para as instâncias do Conjunto 1.

Na Tabela 4.14 apresentamos os resultados para cada ordenação, usando a HR2. Com as ordenações FFc e a FFr a HR2 obteve as melhores quantidades de soluções ótimas (4). Em relação ao gap a ordenação FFr obteve 11 gaps melhores ou iguais às demais ordenações. A ordenação FFr foi escolhida como a melhor para a HR2 para as instâncias do Conjunto 1.

Instâncias	Heurística 1							
	FNa	G	FAr	G	FNc	G	FFr	G
<i>Gcutd1</i>	295	0	295	0	295	0	295	0
<i>Gcutd2</i>	348	0,86	348	0,86	348	0,86	348	0,86
<i>Gcutd3</i>	343	0,29	352	2,84	346	1,16	343	0,29
<i>Gcutd4</i>	848	0,35	848	0,35	846	0,12	846	0,12
<i>Gcutd5</i>	208	0,48	208	0,48	208	0,48	208	0,48
<i>Gcutd6</i>	377	0,53	376	0,27	377	0,53	370	0,27
<i>Gcutd7</i>	601	0,17	603	0,50	600	0	600	0
<i>Gcutd8</i>	722	0,28	728	1,10	720	0	720	0
<i>Gcutd9</i>	136	0,73	136	0,73	135	0	135	0
<i>Gcutd10</i>	316	0,32	315	0	316	0,32	316	0,32
<i>Gcutd11</i>	350	0,29	364	4,12	350	0,29	350	0,29
<i>Gcutd12</i>	677	0,29	678	0,44	676	0,15	676	0,15

Tabela 4.14: Teste 1 : definição da melhor estratégia de ordenação para a HR2 para o caso não exato.

Teste 2

Comparando a melhor ordenação para a HR1 (FFc) (colunas 6 e 7 da Tabela 4.13) com a melhor ordenação para a HR2 (FFr) (colunas 8 e 9 da Tabela 4.14), temos que HR1-FFc obteve quatro soluções ótimas, a mesma quantidade que a HR2-FFr. Os gaps da HR2-FFr foram nove vezes melhores ou iguais que a HR1-FFc. A HR1-FFc obteve sete melhores ou iguais gaps a HR2-FFr. Dessa forma, a HR2-FFc foi escolhida como a melhor heurística residual para as instâncias do Conjunto 1 (CUI; ZHAO, 2013), (CINTRA et al., 2008) e (COSTA; POLDI, 2016)

Teste 3

Na Tabela 4.15, apresentamos os resultados para as instâncias do Conjunto 1 entre os métodos: HR2-FFr, CorteBiFur, Z_{IP} e os resultados da literatura.

I	Z _{ib}			HR2-FFr			CorteBi			Z _{IP}			Cui e Zhao			Cintra et al			HCP				
	F	G%	T	F	G%	T	F	G%	T	F	G%	T	F	G%	T	F	G%	T	F	G%	T		
Gcutd1	295	0	19,2070	295f	0	19	0,02	295	0	32	0,0719	295	0	21	0,03	295	0	11	0,29	0	11	0,29	
Gcutd2	348	0,86	14,0881	346	0,29	52	0,02	346	0,29	82	8,1624	350	1,43	40	11,6563	345	0	173	0,45	346	0,29	46	1,85
Gcutd3	342	0,29	27,6337	355	3,66	149	0,03	343	0,29	126	0,0832	342	0	—	—	343	0,29	534	1,31	553	1,45	7	0,74
Gcutd4	846	0,12	40,0395	869	2,76	281	0,06	845	0	160	0,0759	846	0,12	—	—	845	0	1506	5,99	863	15,25	72	30,6549
Gcutd5	207	0,48	12,5463	208	0,48	14	0,01	208	0,48	30	0,0722	207	0	—	—	207	0	21	0,03	208	0,48	7	0,13
Gcutd6	375	0,53	14,4754	380	1,32	51	0,01	375	0	66	0,0707	375	0	—	—	375	0	86	0,16	376	0,27	30	1,19
Gcutd7	600	0	19,8649	606	0,82	96	0,02	600	0	9	0,0711	600	0	—	—	600	0	357	0,71	609	0,33	57	5,04
Gcutd8	720	0	54,5882	739	2,44	310	0,07	720	0	214	0,6630	721	0,14	—	—	720	0	693	3,5	935	0,96	33	6,6
Gcutd9	135	0	12,2082	138	2,17	19	0,01	136	0,73	33	0,0649	135	0	—	—	135	0	57	0,08	137	1,46	13	0,34
Gcutd10	315	0,32	14,8061	379	1,05	44	0,02	315	0	71	0,0755	315	0	—	—	315	0	122	0,21	378	0,94	29	1,14
Gcutd11	349	0,29	24,5734	354	1,13	117	0,04	349	0	134	0,0836	349	0	—	—	349	0	289	0,079	352	0,85	53	10,06
Gcutd12	675	0,15	55,4583	687	1,75	245	0,06	675	0	207	0,727	676	0,15	—	—	676	0,15	1167	5,28	895	1,01	32	9,52

Tabela 4.15: Teste 3 : resultados das instâncias do Conjunto 1 para o caso não exato.

- o Sistema CorteBiFur e a heurística HCP obtiveram uma solução ótima cada, a HR2-FFr obteve quatro, o método Z_{IP} e o algoritmo original de Cui e Zhao obtiveram 9 e o algoritmo de Cintra *et al* obteve 10 soluções ótimas. A HR2-FFr e o método de Cui e Zhao se diferenciam na ordenação dos padrões de cortes e no método gerar coluna. A HR2-FFr utiliza a geração de padrões de Gilmore e Gomory enquanto que Cui e Zhao desenvolveram um método próprio, o que provavelmente tornou o seu algoritmo mais eficiente. A HR2-FFr conseguiu encontrar otimalidade em cada uma das instâncias que o método de Cui e Zhao não conseguiu (*Gcutd8*). Para a instância *Gcutd12* somente o Z_{IP} conseguiu a otimalidade.
- em relação ao gap o método de Cintra *et al* e de Cui e Zhao são os melhores, seguido pelo Z_{IP} e depois pela HR2-FFr.
- a HR2-FFr teve uma média de 100 padrões de cortes por instância, o sistema CorteBiFur 116, o método de Cintra *et al* 419 e a HCP 30. A média de padrões de cortes utilizados pelo algoritmo de Cintra *et al* é muito mais alta que os demais métodos usados do Teste 3, então esse alto índice de padrões de cortes gerados contribuiu para determinar um método mais eficiente. Cui e Zhao não disponibilizaram tais resultados.
- em relação ao tempo computacional, o método de Cintra *et al*, e o CorteBiFur tiveram tempos de resolução mais baixos que os demais métodos, pois eles utilizam a programação dinâmica para resolver os problemas da mochila. Comparando o tempo computacional que a HR2-FFr levou para resolver cada instância, percebemos que para três delas o tempo foi mais elevado, *Gcutd14*, *Gcutd8* e a *Gcutd12*. Cui e Zhao não disponibilizaram o tempo computacional para resolver cada instância.

Teste 4

Na Tabela 4.16, apresentamos as instâncias do Conjunto 2 entre os métodos: HR2-FFr, CorteBiFur e o Z_{IP} .

I	$[Z_{lb}]$	HR2-FFr						CorteBi						Z_{JP}						HCP					
		F	G%	P	T	F	G%	P	T	F	G%	P	T	F	G%	P	T	F	G%	P	T				
P1-03	73	74	1,35	14	11,7050	73	0	5	$<10^{-2}$	73	0	14	0,0562	73	0	14	0,0562	73	0	5	8,1051				
P1-12	25	26	3,85	9	11,3507	26	3,85	2	0,01	26	3,85	9	0,0595	26	3,85	9	0,0595	26	3,85	1	8,234				
P1-15	47	47	0	7	11,2969	48	2,08	2	$<10^{-2}$	47	0	7	0,575	48	2,08	1	8,0578	48	2,08	1	8,0578				
P2-03	140	140	0	9	11,4001	141	0,71	4	$<10^{-2}$	140	0	9	0,0673	141	0,71	2	8,0880	141	0,71	2	8,0880				
P2-09	6	7	14,29	12	11,4943	7	14,29	5	$<10^{-2}$	7	14,29	12	0,0613	7	14,29	3	8,0802	7	14,29	3	8,0802				
P2-12	44	46	4,35	32	12,1219	44	0	11	0,02	45	2,22	32	0,0753	47	6,38	8	8,4295	47	6,38	8	8,4295				
P2-15	75	76	1,32	19	11,5706	76	1,32	11	0,02	75	0	19	0,0572	75	0	7	8,2257	75	0	7	8,2257				
P3-03	134	136	1,47	29	11,9919	135	0,75	11	$<10^{-2}$	134	0	29	0,0699	135	0,75	7	8,3486	135	0,75	7	8,3486				
P3-12	10	10	0	5	11,3711	10	0	2	10^{-2}	10	0	5	0,605	10	0	1	8,5258	10	0	1	8,5258				
P3-15	57	58	1,72	42	12,6072	58	1,72	17	0,01	57	0	42	0,0675	59	3,39	12	8,9470	59	3,39	12	8,9470				
P3-20	29	31	6,45	28	12,2549	32	9,37	12	6,12	30	3,33	28	0,619	32	9,37	9	8,5286	32	9,37	9	8,5286				
P3-25	4	4	0	16	11,6799	5	1,25	4	0,01	4	0	16	0,0577	5	1,25	3	8,1716	5	1,25	3	8,1716				

Tabela 4.16: Teste 4 : resultados das instâncias do Conjunto 2 para o caso não exato.

Analisando os resultados apresentados na Tabela 4.16 temos que:

- a HR2-FFr obteve quatro soluções ótimas entre doze, o CorteBiFur três, o Z_{IP} oito e a HCP quatro. Entre as seis instâncias que nem o método CorteBiFur e nem a HR2-FFr encontraram o valor ótimo, em quatro delas os métodos obtiveram o mesmo resultado e nos outros dois o valor da função objetivo da HR2-FFr foi menor. Para algumas instâncias nenhum método conseguiu obter o mesmo valor de Z_{lb} , sendo elas a $P1 - 12$, $P2 - 09$ e a $P3 - 20$.
- em relação ao gap o CPLEX obteve nove resultados melhores ou iguais que os demais métodos, a HR2-FFr seis, o CorteBiFur cinco e a HCP quatro. Analisando em particular somente a HR2-FFr e o CorteBiFur temos que os gaps da HR2-FFr foram melhores em apenas uma instância.
- a HR2-FFr teve uma média de 17 padrões de cortes gerados por instância, o CorteBiFur sete e a HCP cinco. Apesar da HR2-FFr estar gerando mais padrões de cortes que o CorteBiFur, percebemos dos itens anteriores que a diferença entre as soluções desses dois métodos foram pequenas.
- em relação ao tempo computacional, a HR2-FFr teve um tempo quase constante de 11 segundos por instância, o CorteBiFur resolveu seis instâncias com um tempo menor que 10^{-2} , e a HCP para todas as instâncias teve o tempo em média de oito segundos.

Na Tabela 4.17 apresentamos os resultados das instâncias do Conjunto 3, utilizando a HR2-FFr, CorteBiFur, Z_{IP} e a HCP.

I	[Z _{lb}]			HR2-FFr			CorteBi			Z _{IP}			HCP			
	F	G%	T	F	G%	T	F	G%	T	F	G%	T	F	G%	T	
P1,P2-03	199	1,01	34	13,4609	199	1,01	13	0,01	197	0	34	0,0771	198	0,50	9	9,2069
P1,P2-12	55	3,64	54	16,94148	70	24,29	16	0,01	54	1,85	36	0,0716	57	7,02	12	9,4519
P1,P2-15	70	4,29	33	12,2467	115	44,35	19	0,01	68	1,47	33	0,0601	69	2,90	7	10,9345
P1,P3-03	127	1,57	58	14,0835	202	49,35	36	0,01	126	0,79	58	0,172	128	2,34	24	10,7615
P1,P3-12	34	2,94	21	11,7856	34	2,94	8	0,01	33	0	21	0,0606	34	2,94	6	8,8535
P1,P3-15	63	1,59	32	13,0787	98	36,73	33	0,01	62	0	32	0,0908	66	6,06	17	10,4666
P2,P3-03	272	0,73	41	13,0312	272	0,73	20	0,01	270	0	41	0,0833	272	0,73	8	9,0506
P2,P3-12	56	5,36	48	12,6214	56	5,36	14	0,01	54	1,85	48	0,0777	57	7,02	11	9,0644
P2,P3-15	76	1,32	19	11,4680	76	1,32	11	0,01	75	0	19	0,0574	75	0	6	8,7277

Tabela 4.17: Teste 4 : resultados das instâncias do Conjunto 3 para o caso não exato.

Analisando os resultados da Tabela 4.17, temos que:

- a HR2-FFr e o CorteBiFur não conseguiram determinar soluções ótimas, a HCP obteve uma solução ótima e o Z_{IP} cinco soluções ótimas de nove instâncias.
- o gap do Z_{IP} foi melhor para todas as instâncias. Comparando somente o gap da HR2-FFr com o do CorteBiFur temos que o gap de HR2-FFr teve nove vezes gaps melhores ou iguais ao do CorteBiFur.
- a HR2-FFr teve uma média de 38 padrões de cortes por instância, o CorteBiFur a metade desse valor, 19, e a HCP 11.
- o tempo computacional da HR2-FFr variou entre onze e dezesseis segundos, sendo que a instância $P1, P2 - 15$ foi a que teve o maior tempo de resolução. O CorteBiFur conseguiu resolver todas as instâncias com 0,01 segundos. A HCP teve um intervalo de resolução de oito e 11 segundos.

Em resumo, para o caso não exato, o método de Cintra *et al* foi o que obteve o maior número de soluções ótimas para as instâncias do Conjunto 1, o equivalente a 83,33%, mas também obteve um número muito maior de padrões de cortes gerados, em duas instâncias o método gerou mais de mil padrões de corte ($Gcutd4, Gcutd5$). O segundo melhor foi o método de Cui e Zhao e o Z_{IP} que resolveram 75% das instâncias de forma ótima. A HR2-FFr resolveu otimamente 33,33%. Para as instâncias do Conjunto 2, o Z_{IP} resolveu 66,67% das instâncias de forma ótima, a HR2-FFr 33,33%, o CorteBiFur e a HCP resolveram 25%. Para as instâncias do Conjunto 3, o Z_{IP} resolveu 55,56% das instâncias de forma ótima, enquanto a HR2-FFr e o Sistema CorteBiFur não encontraram soluções ótimas. Assim, notamos que as instâncias do conjunto 3 foram as mais difíceis de serem resolvidas. Comparando o CorteBiFur e a HR2-FFr, a HR2-FFr obteve o melhor desempenho para todos os conjuntos. Para as instâncias do Conjunto 1 os gaps encontrados não ultrapassam 1%. Para o Conjunto 2, as instâncias que tiveram o gap maior que zero o valor da função objetivo foi no máximo duas unidades maiores que o Z_{lb} . E para o Conjunto 3, a diferença máxima entre a função objetivo e o Z_{lb} foi de três unidades.

Capítulo 5

CONCLUSÃO

Nesta dissertação estudamos o problema de corte de estoque bidimensional voltado para a indústria moveleira, discutimos conceitos sobre esse problema e sobre padrões de corte bidimensionais 2-estágios. O método utilizado neste trabalho para resolver o PCE2D foi por meio de heurísticas residuais que utilizam o método simplex com geração de colunas. Testamos duas heurísticas disponíveis na literatura (POLDI; ARENALES, 2006) e (CUI; ZHAO, 2013) com diferentes ordenações para os padrões de corte. Foram usados três conjuntos de instâncias, divididos em quatro testes. Todos os testes foram feitos considerando o caso exato e o caso não exato.

Para ambos os casos (exato e não exato) a heurística residual 2 (HR2) com a ordenação que propomos, por ordem não crescente da parte fracionária das frequências (FFr), foi a que obteve os melhores resultados. Para o caso exato, a HR2-FFr foi o melhor método para as instâncias dos Conjuntos 2 e 3 e o segundo melhor para o Conjunto 1. Para o caso não exato, os métodos disponíveis na literatura (CUI; ZHAO, 2013) e (CINTRA et al., 2008) se mostraram mais eficientes para o Conjunto 1. Para as instâncias dos Conjuntos 2 e 3 o CPLEX obteve os melhores resultados seguidos pela HR2-FFr. O Conjunto 3 foi o mais difícil de obter soluções ótimas, tanto para o caso exato quanto para o caso não exato. Em particular, no caso não exato nem a HR2-FFr e nem o CorteBiFur obtiveram soluções ótimas.

A contribuição deste trabalho é continuar com o aprimoramento do Sistema CorteBiFur propondo heurísticas para obter soluções inteiras factíveis. A HR2-FFr se mostrou uma boa alternativa e forneceu soluções melhores que o CorteBiFur em todos os testes, mas principalmente para as instâncias do caso exato, pois o CorteBiFur usa como método de arredondamento a função piso.

As colunas geradas pela HR2-FFr mostram-se eficientes, pois o CPLEX utilizou somente essas colunas e conseguiu obter resultados melhores para a função objetivo, principalmente para o caso não exato. Uma das propostas de continuidade desse trabalho é de estudar os padrões de corte gerados e terminar algum critério de escolha para os padrões aceitos. Além disso, implementar funções na HR2-FFr que o Sistema CorteBiFur já possui, como a rotação de itens e a inclusão da espessura da serra. Para melhorar o tempo computacional, a programação dinâmica pode ser utilizada para resolver os m problemas da mochila necessários na geração de padrões de corte simultaneamente.

REFERÊNCIAS

- ALTIN, S. et al. Three dimensional cutting stock problem in mattress production: a case study. p. 949–960, 01 2018.
- ARENALES, M. et al. *Pesquisa Operacional*. Rio de Janeiro: Elsevier, 1998. 723 p. (Série do livro).
- 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, p. 85–106, 05 2006.
- BEZANSON, J. et al. Julia language communication. p. 1–229, 03 2014.
- BOUAINE, A.; LEBBAR, M.; MOHAMED, A. H. Minimization of the wood wastes for an industry of furnishing: A two dimensional cutting stock problem. *MANAGEMENT AND PRODUCTION ENGINEERING REVIEW*, v. 9, p. 42–51, 06 2018.
- BRASIL. Ministério do trabalho e emprego (mte). relação de informações sociais (rais). v. 12, p. 288–298, 2017.
- CARVALHO, J. Exact solution of bin-packing problems using column generation and branch-and-bound. *Annals of Operations Research - Annals OR*, v. 86, p. 629–659, 01 1999.
- CASTELUCCI, P. B. Julia e jump: novas ferramentas para programação matemática. *Pesquisa Operacional para o Desenvolvimento*, v. 9, p. 48–61, 07 2017.
- CHRISTOFIDES, N.; WHITLOCK, C. An algorithm for two-dimensional cutting problems. *Pesquisa Operacional*, v. 25, p. 30–44, 01 1977.
- CHVATAL, V. Edmonds polytopes and a hierarchy of combinatorial problems. v. 4, p. 305–337, 04 1973.
- CHVATAL, V. *Linear Programming*. [S.l.]: W. H. Freeman, 1983. 478 p.
- CINTRA, G. Algoritmos para problemas de corte de guilhotina bidimensional. 2014.
- CINTRA, G. et al. A note on the approximability of cutting stock problems. *European Journal of Operational Research*, v. 183, p. 1328–1332, 12 2007.
- CINTRA, G.; WAKABAYASHI, Y. Um algoritmo híbrido para o problema de corte unidimensional. *XXX Simpósio Brasileiro de Pesquisa Operacional*, p. 61–84, 09 1998.
- CINTRA, G. F. et al. Algorithms for two-dimensional cutting stock and strip packing problems using dynamic programming and column generation. *European Journal of Operational Research*, v. 191, p. 61–85, 11 2008.

- COSTA, L. L. S.; POLDI, K. C. Um estudo sobre o problema de corte de estoque bidimensional 2-estágios. *Dissertação*, 2016.
- CUI, Y.; CUY, Y. P.; YANG, L. Heuristic for the two-dimensional arbitrary stock-size cutting stock problem. *Computers e Industrial Engineering*, v. 78, p. 195–204, 12 2014.
- CUI, Y.; ZHAO, Z. Heuristic for the rectangular two-dimensional single stock size cutting stock problem with two-staged patterns. *European Journal of Operational Research*, v. 231, p. 288–298, 12 2013.
- DALALAH, D.; KHRAIS, S.; BATAINEH, K. Waste minimization in irregular stock cutting. *Journal of Manufacturing Systems*, v. 33, p. 27–40, 01 2014.
- DYCKHOFF, H. A typology of cutting and packing problems. *European Journal of Operational Research*, v. 44, p. 145–159, 01 1990.
- FIGUEIREDO, A. G.; RANGEL, S. Análise de produtividade de padrões de corte na indústria de móveis. *Dissertação*, v. 26, 05 2006.
- FIGUEIREDO, A. G.; RANGEL, S. O problema de corte de estoque em indústrias de móveis de pequeno e médio portes. *Pesquisa Operacional*, v. 28, p. 451–472, 09 2008.
- FILHO, A.; MORETTI, A.; PATO, M. A comparative study of exact methods for the bi-objective integer one-dimensional cutting stock problem. *Journal of the Operational Research Society*, v. 69, p. 1–18, 02 2017.
- GASQUE, D. H. F.; RANGEL, S. Migração do software cortebifur para sistema web. 2016.
- GILMORE, P.; GOMORY, R. A linear programming approach to the cutting-stock problem. *Operations Research*, v. 9, p. 849–859, 1961.
- GILMORE, P.; GOMORY, R. A linear programming approach to the cutting-stock problem-part ii. *Operations Research*, v. 11, p. 863–888, 1963.
- GILMORE, P.; GOMORY, R. Multistage cutting stock problem of two and more dimensions. *Operations Research*, v. 13, p. 94–120, 1965.
- HEIS, A.; CONSTANTINO, A. A.; ARAUJO, S. A. Um algoritmo genético para o problema de corte unidimensional inteiro. *Revista Eletrônica Técnico-Científica do IFSC*, v. 1, p. 1–5, 2010.
- IBM. *OR-Library*. Disponível em: <<http://people.brunel.ac.uk/~mastjjb/jeb/info.html>>.
- IBM Ilog CPLEX Optimization Studio. Disponível em: <<https://www.ibm.com/br-pt/products/ilog-cplex-optimization-studio>>.
- IEMI. Estimativas de inteligência de mercado brasil móveis. v. 45, 2018.
- KALLRATH, J. e. a. Solving real-world cutting stock-problems in the paper industry: mathematical approaches, experience and challenges. *European Journal of Operational Research*, v. 238, p. 374–389, 10 2014.
- KANTOROVICH, L. V. Mathematical methods of organizing and planning production. *Management Science*, v. 6, p. 366–422, 07 1960.

- LASDON, L. S. *Optimization theory for large systems*. Chelmsford: Courier Corporation, 1970. 523 p.
- LODI, A.; MARTELLO, S.; VIGO, D. Models and bounds for two-dimensional level packing problems. *Journal of Combinatorial Optimization*, v. 8, p. 363–379, 2004.
- LODI, A.; MONACI, M. M. Integer programming models for 2-staged two-dimensional knapsack problems. v. 94, p. 257–278, 2003.
- LUENBERGER, D. G.; YE, Y. *Linear and Nonlinear Programming*. Nova Iorque: Springer, 1984. 511 p.
- MACEDO, R.; ALVES, C.; CARVALHO, J. Arc-flow model for the two-dimensional guillotine cutting stock problem. *Computers e OR*, v. 37, p. 991–1001, 06 2010.
- MALAGUTI, E.; DURÁN, R.; TOTH, P. Approaches to real world two-dimensional cutting problems. *Omega*, v. 47, p. 99–115, 09 2014.
- MARTINOVIC, J.; SCHEITHAUER, G.; CARVALHO, J. M. V. d. A comparative study of the arcflow model and the one-cut model for one-dimensional cutting stock problem. *European Journal of Operational Research*, v. 266, p. 458–471, 04 2018.
- MORABITO, R.; ARENALES, M.; ARCARO, V. An and-or-graph approach for two-dimensional cutting problems. *European Journal of Operational Research*, v. 58, p. 263–271, 02 1992.
- MORABITO, R.; ARENALES, M. N. Staged and constrained two-dimensional guillotine cutting problems: An and/or-graph approach. *European Journal of Operational Research*, v. 94, p. 548–560, 11 1996.
- MUTER, I.; SEZER, Z. Algorithms for the one-dimensional two-stage cutting stock problem. *European Journal of Operational Research*, v. 271, p. 20–32, 11 2018.
- ONWUBOLU, G.; MUTINGI, M. A genetic algorithm approach for the cutting stock problem. *Journal of Intelligent Manufacturing*, v. 14, p. 209–218, 04 2003.
- PÁL, L. A genetic algorithm for the two-dimensional single large object placement problem *. 05 2006.
- POLDI, K. C.; ARENALES, M. N. Heurísticas para o problema de corte de estoque unidimensional inteiro. *Operations Research*, v. 26, p. 473–492, 09 2006.
- POLDI, K. C.; ARENALES, M. N. Heuristics for the one-dimensional cutting stock problem with limited multiple stock lengths. *Computers and Operations Research*, v. 36, p. 3680–3691, 09 2009.
- RANGEL, S. et al. Cortebifur. Disponível em: <<http://www.dcce.ibilce.unesp.br/cortebifur/login.jsf>>, 2016.
- RANGEL, S.; CLÓVIS, P. O problema de corte bidimensional. 1989.
- RANGEL, S. e. a. Cortebifur - corte bidimensional retangular em dois estágios. *Manual do programa de computador em fase de registro - Unesp-SJRP, departamento de matemática aplicada*, 2016.

- RESENDE, M. G. C. Heuristics for the one-dimensional cutting stock problem with limited multiple stock lengths. *Simpósio Brasileiro de Pesquisa Operacional*, p. 2074–2081, 07 2013.
- SALLAUME, S. et al. One dimensional cutting stock problem with redevelopment of the surplus material. *International Conference on Engineering Optimization*, 06 2008.
- SAVIC, A.; KRATICA, J.; FILIPOVI, V. A new nonlinear model for the two-dimensional rectangle packing problem. *Publications de l Institut Mathematique*, v. 93, p. 95–107, 01 2013.
- SCHEITHAUER, G. Introduction to cutting and packing optimization. v. 263, 01 2018.
- SILVA, E.; ALVELOS, F.; CARVALHO, J. An integer programming model for two- and three-stage two-dimensional cutting stock problems. *European Journal of Operational Research*, v. 205, p. 699–708, 09 2010.
- SPEROTTO, F. Q. Setor moveleiro brasileiro e gaúcho: características, configuração e perspectiva. v. 45, p. 43–60, 2018.
- SULIMAN, S. A sequential heuristic procedure for the two-dimensional cutting-stock problem. *International Journal of Production Economics*, v. 99, p. 177–185, 02 2006.
- THE Julia programming language. Disponível em: <<https://julialang.org>>.
- TOKUYAMA, H.; UENO, N. The cutting stock problem for large sections in the iron and steel industries. *European Journal of Operational Research*, v. 22, p. 280–292, 02 1985.
- TOSCANO, A.; RANGEL, S.; YANASSE, H. A heuristic approach to minimize the number of saw cycles in small-scale furniture factories. *Annals of Operations Research*, v. 258, 08 2015.
- TSAI, J. F.; HSIEH, P. L.; HUANG, Y. H. An optimization algorithm for cutting stock problems in the industries. *Computers e Industrial Engineering*, v. 57, p. 913–919, 10 2009.
- VALLE, A. M. D. et al. Heuristics for two-dimensional knapsack and cutting stock problems with items of irregular shape. *Expert Systems with Applications*, v. 57, p. 919–919, 11 2012.
- VANCE, P. H. Branch-and-price algorithms for the one-dimensional cutting stock problem. *Computational Optimization and Applications*, v. 9, p. 221–228, 03 1998.
- VIANNA, A. C. G.; ARENALES, M. N.; GRAMANI, M. C. N. Two-stage and constrained two-dimensional guillotine cutting problems. [s.l: s.n.]. p. 1679–1699, 2003.
- VIANNA, A. C. G.; POLDI, K. C. O problema de corte de estoque bidimensional aplicado a uma indústria de esquadrias metálicas. *Simpósio Brasileiro de Pesquisa Operacional*, v. 37, p. 1679–1699, 05 2005.
- WANG, P. Y. Two algorithms for constrained two-dimensional cutting stock problems. operations. *Operations Research*, v. 31, p. 573–586, 06 1983.
- WÄSCHER, G.; GAU, T. Heuristics for the integer one-dimensional cutting stock problem: A computational study. *OR Spectrum*, v. 18, p. 131–144, 09 1996.
- WÄSCHER, G.; HAUBNER, H.; SCHUMANN, H. An improved typology of cutting and packing problems. *European Journal of Operational Research*, v. 183, p. 1109–1130, 12 2007.

WOLSEY, L. A. *Integer Programming*. Nova Iorque: CWiley-Interscience, 1998. 288 p.

YANASSE, H.; KATSURAYAMA, D. M. Checkerboard patterns: Proposals for its generation. *International Transactions in Operational Research*, v. 12, p. 21 – 45, 01 2005.

YANASSE, H.; MORABITO, R. A note on linear models for two-group and three-group two-dimensional guillotine cutting problems. *International Journal of Production Research - INT J PROD RES*, v. 46, p. 6189–6206, 10 2008.

YANXIN, X. A heuristic based on pso for irregular cutting stock problem. *IFAC Proceedings Volumes*, v. 46, p. 473–477, 07 2013.