

UNIVERSIDADE FEDERAL DE SÃO CARLOS
CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA DE PRODUÇÃO

MATEUS PEREIRA MARTIN

PROBLEMAS DE CORTE GUILHOTINADO E RESTRITOS:
FORMULAÇÕES MATEMÁTICAS E MÉTODOS DE SOLUÇÃO

SÃO CARLOS-SP
2019

MATEUS PEREIRA MARTIN

PROBLEMAS DE CORTE GUILHOTINADO E RESTRITOS:
FORMULAÇÕES MATEMÁTICAS E MÉTODOS DE SOLUÇÃO

Tese de doutorado apresentada ao Programa de Pós-graduação em Engenharia de Produção da Universidade Federal de São Carlos (UFSCar), para obtenção do título de Doutor em Engenharia de Produção.

Orientador: Prof. Dr. Reinaldo Morabito Neto.

Coorientador: Prof. Dr. Pedro Augusto Munari Junior.

Financiamento: CAPES, FAPESP (processo nº 2016/08039-1) e CNPq (processo nº 200745/2018-2).

SÃO CARLOS-SP

2019



Folha de Aprovação

Assinaturas dos membros da comissão examinadora que avaliou e aprovou a Defesa de Tese de Doutorado do candidato Mateus Pereira Martin, realizada em 12/12/2019:

Prof. Dr. Reinaldo Morabito Neto
UFSCar

Prof. Dr. Ernesto Julián Goldberg Birgin
USP

Prof. Dr. Flávio Keidi Miyazawa
UNICAMP

Prof. Dr. Horacio Hideki Yanasse
UNIFESP

Profa. Dra. Maria do Socorro Nogueira Rangel
UNESP

Prof. Dr. Pedro Augusto Munari Junior
UFSCar

Certifico que a defesa realizou-se com a participação à distância do(s) membro(s) Ernesto Julián Goldberg Birgin, Flávio Keidi Miyazawa, Pedro Augusto Munari Junior e, depois das arguições e deliberações realizadas, o(s) participante(s) à distância está(ão) de acordo com o conteúdo do parecer da banca examinadora redigido neste relatório de defesa.

Prof. Dr. Reinaldo Morabito Neto

Aos meus amados pais, Mauro e Nair, dedico.

Agradecimentos

Esta tese de doutorado foi, até o momento, meu maior desafio intelectual. Durante esses quase quatro anos, eu tive a orientação, o apoio e a ajuda de algumas pessoas e instituições. Agora expresso minha gratidão a elas.

Agradeço ao Prof. Dr. Reinaldo Morabito, meu orientador. Pela sugestão de problema que foi objeto de pesquisa, que embora simples de ser enunciado, permite ser abordado de diversas formas, o que é muito estimulante. Pela sua orientação irretocável, de profícuas e divertidas “provocações”, que sempre me levaram a boas reflexões e bons caminhos para pesquisar. Pelo exemplo de pesquisador e pessoa, pelas muitas reuniões, pelas atentas sugestões de melhoria e conselhos, pelo rigor científico, e pela delicadeza de cada “sugiro”/“sugiro fortemente”. Agradeço fortemente.

Agradeço ao Prof. Dr. Pedro Munari, meu coorientador. Pelo rigor científico, pela qualidade de cada sugestão de pesquisa, pelo apego aos importantes detalhes no desenvolvimento dos métodos de solução, pelas excelentes sugestões de melhoria de escrita nos artigos. Pelo exemplo de jovem pesquisador e pessoa, por ser muito solícito e, particularmente, por mostrar caminhos e formas de pensamento que sempre me surpreenderam. Sua presença enriqueceu os resultados de pesquisa desta tese, e também a mim como pessoa.

Agradeço ao Prof. Dr. José Fernando Oliveira e à Profa. Dra. Elsa Silva por me receberem tão bem no Porto, em Portugal, durante o ano de doutorado sanduíche. Pelas suas sugestões de pesquisa, pelo rigor científico, e pelos exemplos de pesquisadores e pessoas que são. Obrigado por me ensinarem que mesmo o menor resultado de pesquisa tem altura não-nula, e contribuí para o avanço da ciência. E, principalmente, por despertarem em mim o desejo de ser um pesquisador.

Agradeço aos Profs. Drs. Antonio Moretti, Ernesto Birgin, Flávio Miyazawa, Horacio Yanasse, José Fernando Oliveira e Socorro Rangel por aceitarem participar dos exames de pré-qualificação, de qualificação e/ou de defesa, e pelas suas atentas e bem-recebidas sugestões de melhoria, que enriqueceram esta tese.

Agradeço muito aos meus pais, Mauro e Nair. Pelo apoio constante e palavras de estímulo. Por acreditarem que a educação é sempre um bom caminho. Por me abraçarem e me trazerem muita alegria nos momentos de ansiedade e dúvidas. Agradeço aos meus sobrinhos Giovana e Samuel, pois foi brincando com eles que me surgiu a ideia de alguns modelos desta tese. Agradeço também a minha família como um todo.

Agradeço ao DEP-UFSCar pelo apoio institucional e, particularmente, aos Profs. Drs. Alceu Alves, Ivete Delai, Pedro Munari, Reinado Morabito, Sérgio da Silva e Vitória Pureza pelas disciplinas do doutorado, e ao Robson pelo apoio na secretaria.

Agradeço aos colegas de grupo de pesquisa pelo apoio, risadas e trocas de experiências. Aos amigos do GPO-DEP-UFSCar, Aldair, Alfredo, Alyne, Aura, Beatriz, Cesar, Gislaine, Jonathan, Juan, Karim, Maria Gabriela, Mateus Pileggi, Pedro Miranda, Pedro Hokama e Tamara. Aos amigos do CEGI-INESC TEC, Cristiane, Eduardo, Flávia, Gonçalo, Luis, Pedro, Sara e Sofia, que fizeram o Porto ser ainda mais lindo. Aos meus queridos amigos de longa data, Bruno, Camilla e Jorge.

Agradeço às agências de fomento pelo apoio financeiro. Agradeço o apoio financeiro da CAPES pela bolsa no Brasil vinculada ao PPGEP-UFSCar, de março/2016 até maio/2016. Agradeço o apoio financeiro da FAPESP (processo número 2016/08039-1) pela bolsa no Brasil de junho/2016 até setembro/2018. Agradeço o apoio financeiro do CNPq (processo número 200745/2018-2) pela bolsa no exterior de outubro/2018 até setembro/2019.

Destaco que esta tese também me foi uma fonte de muita satisfação. Uma etapa de crescimento intelectual e pessoal. Agradeço muito ao meu Deus e Senhor por tudo o que vivi e aprendi nesta tese.

“O correr da vida embrulha tudo, a vida é assim:
Esquenta e esfria, aperta e daí afrouxa,
Sossega e depois desinquieta.
O que ela quer da gente é coragem.”

Guimarães Rosa (1956) em “Grande Sertão: Veredas”.

Resumo

Esta tese de doutorado tem como objeto de pesquisa os Problemas de Corte Guilhotinado e Restritos (PCGR). Os PCGR consistem em produzir itens a partir de um objeto, ao se utilizar operações de corte guilhotinado, i.e., operações que repartam os materiais retangulares em dois sub-objetos retangulares adjacentes sem restrição sobre o número de estágios guilhotinados, e que atendam à quantidade máxima de vezes que os itens podem ser cortados a partir do objeto. Novas formulações matemáticas lineares são propostas ao Problema de Corte Bidimensional Guilhotinado e Restrito (PCBGR). A primeira formulação proposta se baseia na discretização do objeto, que também foi estendida para lidar com padrões 2-estágios e 1-grupo, assim como um algoritmo de decomposição de Benders foi desenvolvido a partir dela. Outras duas formulações são propostas ao PCBGR, baseadas no conceito de combinação sucessiva de cópias de tipos de itens e subpadrões, i.e., na abordagem *bottom-up*; a partir de restrições adicionais propostas, esses modelos estritamente lidam com padrões de corte d -estágios, sendo d um escalar inteiro positivo. Duas extensões do PCBGR são abordadas. A primeira delas é o Problema de Corte Bidimensional Guilhotinado e Restrito de Objeto com Defeitos (PCBGR.D). O modelo linear e algoritmo de Benders, ambos baseados na discretização do objeto, são estendidos para lidar com essa variante, e então um novo método de solução baseado em Programação por Restrições é desenvolvido. A segunda extensão é o Problema de Corte Tridimensional Guilhotinado e Restrito (PCTGR), que é abordado por meio de modelos e por um algoritmo combinatório. As abordagens propostas são avaliadas por experimentos computacionais, usando instâncias da literatura ou geradas aleatoriamente, e comparadas a outras abordagens conhecidas da literatura. Apesar da aplicabilidade desses problemas, a literatura levou mais de três décadas para apresentar formulações matemáticas aos PCGR. Essa tese contribui com novos paradigmas de formulação aos PCGR e métodos efetivos para resolvê-los, que são competitivos com os conhecidos da literatura.

Palavras-chave: Corte guilhotinado. Corte 2-estágios. Corte 1-grupo. Defeitos. Corte Tridimensional. Programação Linear Inteira Mista. Otimização Combinatória.

Abstract

We address the Constrained Guillotine Cutting Problems (CGCP) in this doctoral thesis. The CGCP consist of producing items from objects using guillotine cuts, i.e., cuts that divide the rectangular materials into two rectangular sub-objects without restricting the number of guillotine stages, and with a limitation over the maximum number of copies per item type to be cut from the object. New linear mathematical formulations are proposed for the Constrained Two-dimensional Guillotine Cutting Problem (C2GCP). The first proposed formulation is based on object discretization, which is also extended to deal with 2-staged and 1-group patterns, and a Benders decomposition algorithm is developed from it. Two other formulations are proposed to the C2GCP, based on the concept of successive combination of copies of item types and sub-patterns, i.e., the bottom-up approach; from the proposed additional constraints, these models strictly deal with d -staged cutting patterns, where d is a positive integer scalar. Two extensions of the C2GCP are also addressed in this thesis. The first extension is the Constrained Two-dimensional Guillotine Cutting Problem with Defects (C2GCP-D). The linear model and the Benders decomposition algorithm, both based on object discretization, are extended to deal with this variant, and then a new solution method based on Constraint Programming is developed. The second extension is the Constrained Three-Dimensional Guillotine Cutting Problem (C3GCP), which is addressed by models and by a tree-search algorithm. All these proposed approaches are evaluated through computational experiments, using instances of the literature or randomly generated ones, and compared to another approaches in the literature. Despite the applicability of these problems, the literature took more than three decades to present mathematical formulations to the CGCP. Therefore, this thesis contributes to new formulation paradigms for the CGCP and effective methods to solve them, which are competitive with those known in the literature.

Keywords: Guillotine cutting. 2-staged cutting. 1-group cutting. Defects. Three-dimensional cutting. Mixed Integer Linear Programming. Combinatorial Optimization.

Lista de Ilustrações

1.1	Etapas da abordagem da Pesquisa Operacional.	31
2.1	Tipos básicos de Problemas de Corte e Empacotamento. Fonte: Adaptado de Wäscher, Haußner e Schumann (2007).	39
2.2	Padrão de corte bidimensional guilhotinado (a) e não-guilhotinado (b). . .	40
2.3	Padrão de corte tridimensional guilhotinado (a) e não-guilhotinado (b). . .	40
2.4	Padrão de corte bidimensional 2-estágios não-exato (a) e 3-estágios não-exato (b).	41
2.5	Padrão de corte bidimensional 1-grupo exato (a) e 2-grupos exato (b). . . .	42
2.6	Padrão de corte guilhotinado de objeto com três defeitos.	42
2.7	Exemplo de padrão 2-estágios não-exato segundo Lodi e Monaci (2003). . .	57
2.8	Exemplo de padrão 1-grupo não-exato segundo Yanasse e Morabito (2006). .	59
2.9	Corte guilhotinado. Fonte: Adaptado de Ben Messaoud, Chu e Espinouse (2008).	60
2.10	Representação da tripla (q, j, o) de decisão de corte. Fonte: Adaptado de Furini, Malaguti e Thomopulos (2016).	65
3.1	Ilustração das variáveis x_{kij} , $h_{i'j}$ e $v_{ijj'}$	71
3.2	Ilustrações de alguns segmentos de S_{kij}^h e S_{kij}^v	72
3.3	Ilustração das restrições (3.8d) e (3.8e).	74
3.4	Ilustrações das restrições (3.8f), (3.8g) e (3.8h).	75
3.5	Ilustrações de possíveis padrões de cortes considerados pelo PEV.	78
3.6	Ilustrações de cortes guilhotinados sobre um item alocado para padrões 2-estágios.	81
4.1	Padrões de corte bidimensionais guilhotinados com único defeito.	97
4.2	Exemplo ilustrativo com três áreas defeituosas.	98
4.3	Ilustração da variável x_{kij}	99
4.4	Ilustração das variáveis $h_{i'j}$, $v_{ijj'}$ e $p_{i_1i_2j_1j_2}$	100
4.5	Caracterização das restrições de guilhotina.	101
4.6	Ilustrações de alguns segmentos de S_{kij}^h e S_{kij}^v	102
4.7	Soluções ótimas para duas instâncias da categoria 3.	115

5.1	Exemplo de construções horizontal e vertical.	118
5.2	Compreensão do modelo hierárquico.	119
5.3	Esquemas de enumeração da árvore binária para um exemplo com $\bar{n} = 4$	121
5.4	Representações de padrões de corte na árvore binária proposta para um exemplo com $\bar{n} = 4$ (cada nó cinza contém um único item).	122
5.5	Modelo hierárquico. (a) Qualquer nó $j \in J$ pode conter uma cópia de um tipo de item. Um nó não-folha $j \in J \setminus \bar{J}$ pode ser construído de: (b) e (c) dois subpadrões menores (se $j \in J_A$); (d) e (e) um subpadrão menor e uma cópia de um tipo de item (se $j \in J_A \cup J_B$); ou, (f) e (g) duas cópias de tipos de item.	123
5.6	Padrão de corte da Tabela 5.2.	127
5.7	Compreensão do modelo implícito.	127
5.8	Modelo implícito. Um subpadrão é construído de: (a) e (b) duas cópias de tipos de itens; (c) e (d) uma cópia de tipo de item e um subpadrão menor; or, (e) e (f) dois subpadrões menores.	129
5.9	Padrão de corte da Tabela 5.3.	131
5.10	Exemplos de padrões de corte 2-estágios e 3-estágios.	138
6.1	Exemplos de construções na abordagem <i>bottom-up</i>	157
6.2	Padrão 3-estágios para o PCTGR com cortes na sequência vertical-profundidade-horizontal.	161
6.3	Sequências guilhotinadas para padrões 3-estágios do PCTGR.	163
6.4	Padrão de corte de sequência vertical-horizontal-vertical, cuja altura \bar{h}_j é soma das alturas dos tipos de itens representados em vermelho e em azul.	164
6.5	Desperdícios interno e externo de uma solução parcial T	165
6.6	Estimativa de desperdício externo T_o na sequência H-D-V.	167
6.7	Seis permutações para estimar o desperdício externo T_o	168
A.1	Ilustrações do modelo <i>top-down</i> hierárquico.	200

Lista de Tabelas

2.1	Classificação de alguns problemas conforme a tipologia de Dyckhoff (1990).	37
2.2	Sumário das principais abordagens para os problemas PCBGR, PCBGR_D e PCTGR.	46
3.1	Conjunto de instâncias de <i>benchmark</i> para o PCBGR.	86
3.2	Resultados para as três formulações de PLI para o PCBGR.	88
3.3	Resultados do algoritmo B&BC.	91
3.4	Comparação entre o modelo de Lodi, Martello e Monaci (2002) e Grelha-2-estágios.	92
3.5	Comparação entre o modelo de Yanasse e Morabito (2006) e Grelha-1-grupo.	92
4.1	Instância ilustrativa do PCBGR_D.	97
4.2	Caso irrestrito: resultados para objeto $(L \times W) = (75 \times 75)$ (cada entrada dessa tabela é média sobre 60 instâncias).	111
4.3	Caso irrestrito: resultados para objeto $(L \times W) = (112 \times 50)$ (cada entrada dessa tabela é média sobre 60 instâncias).	112
4.4	Caso restrito: resultados de acordo com o número de tipos de itens (cada entrada dessa tabela é média sobre 360 instâncias).	113
4.5	Caso restrito: resultados de acordo com o parâmetro ρ^s (cada entrada dessa tabela é média sobre 600 instâncias).	113
4.6	Caso restrito: resultados de acordo com o número de defeitos (cada entrada dessa tabela é média sobre 450 instâncias).	114
4.7	Caso restrito: resultados de acordo com a categoria (cada entrada dessa tabela é média sobre 1800 instâncias).	114
5.1	Exemplo ilustrativo para o PCBGR.	127
5.2	Solução ótima do exemplo ilustrativo da Tabela 5.1 para o Modelo Não-linear 1.	127
5.3	Solução ótima do exemplo ilustrativo da Tabela 5.1 ao Modelo Não-linear 2.	131
5.4	Três conjuntos de instâncias para o PCBGR.	141
5.5	Resultados dos modelos para o conjunto de instâncias A.	144
5.6	Resultados dos modelos para o conjunto de instâncias B.	147
5.7	Resultados dos modelos para o conjunto de instâncias C.	149
5.8	Resultados dos modelos para o conjunto de instâncias C para padrões 2-estágios.	150

6.1	Características das instâncias <i>gcut_3d</i>	172
6.2	Resultados das instâncias do conjunto A para padrões não-estagiados no caso irrestrito.	176
6.3	Resultados das instâncias do conjunto A para padrões não-estagiados no caso restrito.	176
6.4	Resultados das instâncias do conjunto A para padrões 3-estágios no caso irrestrito.	177
6.5	Resultados das instâncias do conjunto A para padrões 3-estágios no caso restrito.	177
6.6	Características das instâncias <i>ep3</i>	178
6.7	Resultados das instâncias ponderadas do conjunto B para padrões não-estagiados e 3-estágios.	179
6.8	Resultados das instâncias adaptadas não-ponderadas do conjunto B para padrões não-estagiados e 3-estágios.	180
B.1	Tipos de novas soluções parciais geradas ao combinar horizontalmente duas soluções parciais.	209
B.2	Tipos de novas soluções parciais geradas ao combinar em profundidade duas soluções parciais.	209
B.3	Tipos de novas soluções parciais geradas ao combinar verticalmente duas soluções parciais.	209

Lista de Siglas

PCE	Problemas de corte e empacotamento
PCGR	Problemas de corte guilhotinado e restritos
PCBGR	Problema de corte bidimensional guilhotinado e restrito
PCBGR_D	Problema de corte bidimensional guilhotinado e restrito de objeto com defeitos
PCTGR	Problema de corte tridimensional guilhotinado e restrito
PD	Programação dinâmica
PLI	Programação linear inteira
PLIM	Programação linear inteira mista
PNLI	Programação não-linear inteira
PLIM	Programação não-linear inteira mista
PR	Programação por restrições
B&C-CP	Algoritmo <i>branch-and-cut</i> baseado em PR para o PCBGR_D
B&BC_D	Algoritmo de decomposição de Benders para o PCBGR_D

Sumário

1	Introdução	27
1.1	Objetivos	29
1.2	Justificativas e contribuições	30
1.3	Método de pesquisa	31
1.4	Organização do texto	32
2	Problemas de corte guilhotinado e restritos	35
2.1	Tipologias para os Problemas de Corte e Empacotamento	36
2.1.1	Tipologia de Dyckhoff (1990)	36
2.1.2	Tipologia de Wäscher, Haußner e Schumann (2007)	37
2.2	Problemas de corte bidimensionais e tridimensionais guilhotinados	39
2.2.1	Restrições práticas	42
2.2.2	Definições dos problemas abordados	43
2.3	Revisão da literatura	44
2.3.1	Sumário das principais abordagens	45
2.3.2	Problema de Corte Bidimensional Guilhotinado e Restrito (PCBGR)	47
2.3.3	Problema de Corte Bidimensional Guilhotinado e Restrito de Placa com Defeitos (PCBGR_D)	52
2.3.4	Problema de Corte Tridimensional Guilhotinado e Restrito (PCTGR)	53
2.4	Formulações matemáticas para problemas de corte bidimensionais guilhotinados	55
2.4.1	Formulação de Lodi e Monaci (2003)	56
2.4.2	Formulação de Yanasse e Morabito (2006)	58
2.4.3	Formulação de Ben Messaoud, Chu e Espinouse (2008)	59
2.4.4	Formulação de Furini, Malaguti e Thomopoulos (2016)	65
2.5	Comentários finais do capítulo	67
3	Problema de corte bidimensional guilhotinado e restrito: propostas de formulações matemáticas e de método de solução baseados em grelha	69
3.1	Formulações matemáticas	70
3.1.1	Formulação não-linear baseada na discretização do objeto	70
3.1.2	Formulação linear equivalente baseada na discretização do objeto	75

3.1.3	Procedimento enumerativo de variáveis	76
3.2	Formulações matemáticas para padrões 2-estágios e 1-grupo	80
3.2.1	Formulações para padrões 2-estágios	80
3.2.2	Formulações para padrões 1-grupo	82
3.3	Algoritmo <i>branch-and-Benders-cut</i> para o PCBGR	83
3.4	Experimentos computacionais	85
3.4.1	Resultados para o PCBGR	87
3.4.2	Resultados para as versões 2-estágios e 1-grupo do PCBGR	90
3.5	Comentários finais do capítulo	93
4	Problema de corte bidimensional guilhotinado e restrito de objeto com defeitos: propostas de formulação matemática e de métodos de solução	95
4.1	Descrição do problema	96
4.2	Formulação matemática	97
4.2.1	Adaptação da discretização dos conjuntos normais	103
4.3	Algoritmo de decomposição de Benders	104
4.4	Algoritmo baseado em Programação por Restrições	106
4.5	Experimentos computacionais	109
4.5.1	Conjunto de instâncias de <i>benchmark</i>	110
4.5.2	Resultados para o caso irrestrito	111
4.5.3	Resultados para o caso restrito	112
4.6	Comentários finais do capítulo	115
5	Problema de corte bidimensional guilhotinado e restrito: propostas de formulações matemáticas com o conceito <i>bottom-up</i>	117
5.1	Formulações matemáticas não-lineares	118
5.1.1	Formulação não-linear 1: abordagem <i>bottom-up</i> hierárquica	119
5.1.2	Formulação não-linear 2: abordagem <i>bottom-up</i> implícita	127
5.2	Formulações matemáticas lineares	131
5.2.1	Formulação linear 1	132
5.2.2	Formulação linear 2	134
5.3	Extensões	136
5.3.1	Problema da Dimensão Aberta Guilhotinado	137
5.3.2	Restrições para padrões <i>d</i> -estágios	137
5.4	Experimentos computacionais	140
5.4.1	Resultados ao conjunto de instâncias A	141
5.4.2	Resultados ao conjunto de instâncias B	145
5.4.3	Resultados ao conjunto de instâncias C gerado aleatoriamente	148
5.5	Pseudocódigos para gerar a alocação dos modelos <i>bottom-up</i>	151
5.6	Comentários finais do capítulo	153

6	Problema de corte tridimensional guilhotinado e restrito: propostas de formulações matemáticas e de método de solução	155
6.1	Formulações matemáticas	156
6.1.1	Formulação não-linear inteira para padrões não-estagiados	156
6.1.2	Formulação linear inteira para padrões 3-estágios	160
6.2	Algoritmo <i>bottom-up</i>	163
6.2.1	Compleitude de soluções parciais	166
6.2.2	Estimativa de desperdício externo T_o	167
6.2.3	Algoritmo <i>bottom-up</i> para padrões 3-estágios	168
6.3	Experimentos computacionais	169
6.3.1	Resultados para as instâncias <i>gcut_3d</i>	171
6.3.2	Resultados para instâncias <i>ep3</i>	178
6.4	Comentários finais do capítulo	181
7	Conclusões	183
7.1	Contribuições desta tese de doutorado	184
7.2	Pesquisas futuras	186
	Referências Bibliográficas	189
	Apêndices	197
A	Problema de corte bidimensional guilhotinado e restrito: propostas de formulações matemáticas com o conceito <i>top-down</i>	199
A.1	Introdução	199
A.2	Formulação matemática <i>top-down</i> hierárquica	199
A.3	Formulação matemática <i>top-down</i> implícita	202
B	Seções adicionais do Capítulo 6 do Problema de corte tridimensional guilhotinado e restrito	205
B.1	Formulação linear para padrões não-estagiados	205
B.2	Pseudocódigo para gerar a alocação do modelo não-estagiado	207
B.3	Enumeração dos estágios guilhotinados ao Algoritmo <i>Bottom-up</i> de padrões 3-estágios	207

Capítulo 1

Introdução

O gerenciamento de operações evoluiu ao longo do último século, conforme pode ser observado pelas escolas de Administração de Produção *Fordista*, *Toyotista*, entre outras (SLACK; CHAMBERS; JOHNSTON, 2009). À medida que a capacidade de ofertar superou os requerimentos de demanda, diversas organizações comerciais e industriais buscaram formas mais eficientes para gerenciar seus processos. Isso foi motivado pelo fato que nesse cenário os preços de venda dos produtos e serviços passaram a ser estipulados pelos mercados. Para obter margens de lucro satisfatórias, as organizações se direcionaram à redução de custos operacionais, mesmo diante de estratégias de diferenciação. O conceito de recursos econômicos escassos é inerente a um ambiente competitivo.

Essa reestruturação das organizações destacou inicialmente os princípios de qualidade de produto e processo, e das relações entre o humano e o trabalho. Em seguida, os processos padronizados e custos operacionais bem estabelecidos permitiram análises quantitativas sobre diversos processos de tomada de decisão organizacionais como, por exemplo, localização de instalações, o que produzir e comprar, quanto produzir e estocar, distribuição de produtos, entre outros. Operações eficientes consomem níveis mínimos de recursos, que em contextos de escala de produção ou recursos custosos, podem fornecer vantagem ou paridade competitiva à organização.

Nesse contexto, uma prática comum em diversos segmentos industriais, como os de papel, vidro, aço e móveis, é a armazenagem de grandes quantidades de matéria-prima (objetos) a ser cortada para fabricar peças (itens). Essa prática é uma resposta organizacional a possíveis cenários de: (i) incerteza na demanda dos itens e, assim, estocar objetos é política mais segura; (ii) possíveis economias de escalas na compra e estoque de objetos; (iii) riscos de indisponibilidade de objetos em compras futuras; e, (iv) custos de transporte, armazenagem e manuseio dos objetos podem ser menores do que os custos dos itens, quando estes são comparados ao longo do processo de manufatura. A forma como esses objetos devem ser cortados para produzir os itens demandados pelos consumidores é o processo decisório dessa operação, e pode ser determinada por meio dos Problemas de Corte e Empacotamento (PCE). Esses clássicos problemas em Pesquisa

Operacional são processos-chave nessas cadeias produtivas, de modo que minimizar os desperdícios de corte ou número de objetos é uma estratégia usual.

A primeira abordagem de operações de corte por meio de modelagem matemática foi desenvolvida por Kantorovich (1960), com primeiro relato em 1939. Em seguida, os trabalhos de Gilmore e Gomory (1961, 1963, 1965, 1966) contribuíram para estender a aplicação a problemas reais. A relevância da solução desses problemas advém da sua contribuição para a melhor utilização de recursos escassos em operações de corte e empacotamento, contribuindo na redução de custos. E também são relevantes para os compromissos de responsabilidade ambiental ao evitar uso desnecessário de matérias-primas.

As operações de corte estão relacionadas às operações de empacotamento. Entende-se por operação de corte o processo de cortar um objeto para obter itens menores, enquanto que por operação de empacotamento o processo de empacotar itens menores em um objeto maior. Essas operações estão presentes em ambientes de manufatura como o corte de bobinas e barras, chapas de metal, madeira e vidro, no corte de blocos de aço e de espuma, e logísticos como alocação e arranjos de carregamento de produtos a paletes e de contêineres a navios portuários. Assim, para um problema de corte existe um problema de empacotamento análogo, e vice-versa. A diferenciação desses problemas ocorre, em parte, na heterogeneidade dos itens, e nos equipamentos e dispositivos de corte e de empacotamento.

A aplicabilidade desses problemas conduz a extensões conforme as particularidades de cada ambiente industrial e logístico. Por exemplo, no corte de chapas apenas duas dimensões são relevantes aos objetos e itens; assim, tem-se os problemas de corte bidimensionais. Nesse contexto, as características do sistema produtivo e dos equipamentos de corte podem restringir que os cortes ocorram de forma ortogonal, e de borda à borda dos objetos (caso guilhotinado) ou não (caso não-guilhotinado). Um tipo de item pode ter sua produção limitada a certo número de vezes a partir do objeto (caso restrito) ou não (caso irrestrito). No caso guilhotinado, o número de rotações de 90° graus da serra de corte pode ser limitado a um número inteiro positivo d (caso d -estagiado) ou não (caso não-estagiado). Pode-se ainda requerer que o padrão de corte seja constituído de tiras horizontais e verticais (padrões 1-grupo). Por “padrão de corte”, entende-se o arranjo geométrico de itens no objeto, sem sobreposição dos itens, e respeitando os limites físicos (dimensões) do objeto. Por outro lado, no corte de blocos retangulares existem três dimensões relevantes aos objetos e itens; assim, tem-se os problemas de corte tridimensionais. Os casos discutidos anteriormente (caso guilhotinado/não-guilhotinado, caso restrito/irrestrito, caso não-estagiado/ d -estagiado) também são válidos aos problemas tridimensionais. Esses tipos de padrões de corte bidimensionais e tridimensionais são discutidos e ilustrados no Capítulo 2.

É importante destacar que a simplicidade de enunciação desses problemas pode omitir sua complexidade computacional. De forma geral, os PCE são *NP-Difíceis*,

isto é, a maioria desses problemas não podem ser resolvidos por algoritmos polinomiais, sob hipótese de $\mathcal{P} \neq \mathcal{NP}$ (GAREY; JOHNSON, 1990). Os PCE bidimensionais e tridimensionais são *NP-Difíceis* (SCHEITHAUER, 1992; FAYARD; HIGI; ZISSIMOPOULOS, 1998; LODI; MARTELLO; MONACI, 2002).

1.1 Objetivos

Esta tese de doutorado tem como objeto de pesquisa os Problemas de Corte Guilhotinado e Restritos (PCGR). Esse tipo de padrão supõe cortes do caso ortogonal guilhotinado e restrito. Trata-se de corte de itens retangulares menores (i.e., retângulos ou blocos retangulares) a partir de um objeto retangular maior (i.e., chapa retangular ou bloco retangular). O foco será dado aos padrões de corte não-estagiados; no entanto, o caso *d*-estagiado também é estudado. Os PCGR serão analisados como modelos do Problema de Alocação – ou Problema de Arranjo ou de Colocação –, do inglês, *Placement Problem*. Segundo a tipologia de Wäscher, Haubner e Schumann (2007) para Problemas de Corte e Empacotamento (PCE), os PCGR são classificados como *Two/Three-dimensional Rectangular Single Large Object Placement Problem*.

Os objetivos dessa tese de doutorado envolvem a análise e formulação de modelos de Otimização Matemática, e métodos de solução para os PCGR, que sejam competitivos com os conhecidos da literatura. Esses objetivos são apresentados a seguir:

- Estudar e analisar as formulações existentes de Programação Linear Inteira Mista (PLIM) aos PCGR;
- Propor formulações compactas e/ou pseudopolinomiais alternativas de PLIM para os PCGR, para lidar com conjuntos de poucos tipos de itens e um único objeto;
- Desenvolver métodos de solução baseados em decomposição como estratégia de solução, tais como decomposição de Benders, capazes de tratar com problemas realistas.

Nota-se que em uma formulação compacta, seus números de variáveis e/ou de restrições são diretamente definidas de acordo com os parâmetros da instância considerada. Por outro lado, em uma formulação pseudopolinomial, seus números de variáveis e/ou de restrições são definidos de acordo com algum tipo de pré-processamento nos parâmetros da instância considerada. Os objetivos auxiliares que visam aprimorar a qualidade das soluções obtidas e que tratam das análises a serem realizadas são apresentados a seguir:

- Propor formulações e métodos de solução para o Problema de Corte Bidimensional Guilhotinado e Restrito (PCBGR), assim como para suas versões de padrões 2-estágios e 1-grupo;

- Propor modelos e métodos de solução para contemplar o caso de objeto com defeitos, isto é, para o Problema de Corte Bidimensional Guilhotinado e Restrito de Objeto com Defeitos (PCBGR_D);
- Propor modelos e métodos de solução para o caso tridimensional, isto é, para o Problema de Corte Tridimensional Guilhotinado e Restrito (PCTGR), de corte de blocos retangulares menores a partir de um bloco retangular maior;
- Analisar essas abordagens por meio de experimentos computacionais utilizando instâncias da literatura, a fim de obter recomendações nesses processos decisórios.
- Verificar a influência da limitação sobre o número de itens nos padrões de corte e do número de estágios dos padrões de corte e sua relação com os desperdícios de corte.

Ressalta-se a relevância do desenvolvimento de modelos para problemas de otimização. Um modelo contribui para caracterização e compreensão do problema. Também pode motivar o desenvolvimento de novos métodos de solução, baseados em técnicas de decomposição (por exemplo, relaxação Lagrangeana, decomposição de Benders ou de Dantzig-Wolfe), métodos meta-heurísticos, ou mesmo métodos híbridos, que possibilitem a resolução de grandes instâncias do problema.

1.2 Justificativas e contribuições

As contribuições dessa pesquisa envolvem a proposição de novas formas efetivas de resolução de problemas de corte guilhotinado e restritos, competitivos com aqueles reportados na literatura.

Sob perspectiva prática, os PCGR podem ser verificados nas operações de corte das indústrias moveleiras, metalúrgica, de vidros, de granito e de aço. Em geral, os equipamentos de corte desses sistemas exigem cortes guilhotinados. O caso restrito e não-estagiado é relevante em contextos de demandas reduzidas para alguns tipos de itens, e de custos de matéria-prima significativos no preço de venda dos produtos finais. Soluções efetivas para esse problema são relevantes à redução de custos desses sistemas produtivos por processar menores volumes de matéria-prima, e indiretamente por reduzir custos de energia, mão-de-obra, entre outros. Também são relevantes para evitar o uso desnecessário de recursos naturais sob perspectiva de responsabilidade ambiental, além da redução da produção de resíduos e descartes industriais e também de lixo.

Sob perspectiva científica, poucos trabalhos abordaram os PCGR por meio de formulações matemáticas. Podem ser citados os trabalhos de Christofides e Whitlock (1977), Christofides e Hadjiconstantinou (1995), Hifi (1997) e Cung, Hifi e Cun, Le (2000) como abordagens exatas. Hifi (1997) destacou que o PCBGR é fortemente *NP-Difícil*, o que justifica abordagens heurísticas, como em Álvarez-Valdés, Parajón e Tamarit (2002b) e Morabito e Pureza (2010). Ben Messaoud, Chu e Espinouse (2008) alegaram ser os

primeiros autores a propor uma formulação de PLIM ao PCBGR. Mais recentemente, Furini, Malaguti e Thomopoulos (2016) propuseram outra formulação para o problema ao estender a formulação de Dyckhoff (1981), que trata o caso unidimensional. Em outras palavras, apesar da aplicabilidade do problema, a literatura levou mais de três décadas para apresentar uma formulação com inequações lineares a esse problema, considerando como trabalho seminal ao PCBGR o trabalho de Christofides e Whitlock (1977) até a formulação de Ben Messaoud, Chu e Espinouse (2008). Por outro lado, os problemas de corte bidimensional d -estagiados ($d \leq 3$) e guilhotinados irrestritos podem ser resolvidos por métodos efetivos de solução (GILMORE; GOMORY, 1965; BEASLEY, 1985a; CINTRA et al., 2008; SILVA; ALVELOS; Valério de Carvalho, 2010; MACEDO; ALVES; Valério de Carvalho, 2010).

O PCBGR.D e o PCTGR foram muito pouco abordados pela literatura, conforme discutido na Seção 2.3. Em outras palavras, nem objetos heterogêneos devido a regiões defeituosas nem o corte guilhotinado tridimensional tem recebido destaque, apesar da potencial aplicabilidade deles em alguns sistemas produtivos, como na produção de móveis customizados de madeira com nós de crescimento nos objetos, e no corte de blocos aço e de espuma.

1.3 Método de pesquisa

O ramo de conhecimento da Pesquisa Operacional propõe abordagens quantitativas para auxiliar o processo de tomada de decisões. Nesse contexto, esta tese de doutorado pode ser caracterizada como uma pesquisa quantitativa axiomática. Este tipo de pesquisa é orientada à proposição de um modelo científico a partir de um modelo conceitual (BERTRAND; FRANSOO, 2002). Os esforços para melhorar a qualidade das soluções está associado a uma pesquisa axiomática normativa. Esta tese apresenta algumas etapas reconhecidas em estudos de Pesquisa Operacional. A Figura 1.1 ilustra as etapas da abordagem discutidas em Arenales et al. (2007). O foco dessa pesquisa está nas etapas de modelagem, desenvolvimento de métodos de solução e de validação dos modelos propostos aos problemas estudados. Arenales et al. (2007) destacaram as etapas desse processo iterativo: modelagem do problema real, dedução e análise do modelo matemático, interpretação e inferência das conclusões do modelo e a avaliação das decisões a serem tomadas.

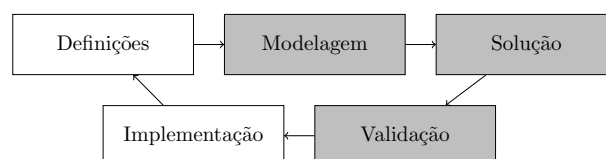


Figura 1.1: Etapas da abordagem da Pesquisa Operacional.

Destaca-se que, apesar dos dois paradigmas de modelagem ao PCBGR,

propostos nos Capítulos 3 e 5, inicialmente obterem modelos não-lineares, o foco desta tese de doutorado são formulações de Programação Linear Inteira Mista, cujos *solvers* de propósito geral atuais são capazes de lidar com grandes números de variáveis e restrições, inclusive com certificados de otimalidade.

1.4 Organização do texto

O texto está organizado como segue. No Capítulo 2, apresentam-se as duas tipologias dos Problemas de Corte e Empacotamento, baseados nos trabalhos de Dyckhoff (1990) e Wäscher, Haußner e Schumann (2007), para compreensão desses problemas. Os principais tipos de padrões bidimensionais guilhotinados também são discutidos e ilustrados, assim como algumas restrições práticas desses problemas. Também define-se o PCBGR e são discutidos a revisão bibliográfica desse problema, de problemas correlatos e seus métodos de solução. Também são discutidos o Problema de Corte Bidimensional Guilhotinado e Restrito de Objeto com Defeitos (PCBGR_D) e o Problema de Corte Tridimensional Guilhotinado e Restrito (PCTGR). As duas formulações de PLIM da literatura para o PCBGR também são apresentadas, e brevemente discutidas.

No Capítulo 3, apresenta-se uma nova formulação pseudopolinomial de Programação Não-Linear Inteira (PNLI) ao PCBGR baseada em grelha, isto é, na discretização do objeto. Essa formulação é uma extensão da formulação de Beasley (1985b), que trata o caso não-guilhotinado restrito. Também desenvolve-se uma formulação de Programação Linear Inteira (PLI) equivalente ao modelo não-linear. Como estratégia de solução, um algoritmo de decomposição de Benders é proposto a partir da formulação linear. Adicionalmente, novas formulações para os casos 2-estágios e 1-grupo são propostas por meio de adaptações na formulação linear desenvolvida. Experimentos computacionais com instâncias de *benchmark* são realizados para avaliar o desempenho das formulações lineares, em comparação às formulações da literatura (LODI; MONACI, 2003; YANASSE; MORABITO, 2006; BEN MESSAOUD; CHU; ESPINOUSE, 2008; FURINI; MALAGUTI; THOMOPULOS, 2016).

No Capítulo 4, aborda-se o PCBGR_D, uma variante do PCBGR cujo objeto a ser cortado apresenta algumas regiões defeituosas, sobre as quais os objetos não podem ser cortados. Dado que as regiões defeituosas estão posicionadas e fixadas no objeto, nesse capítulo estendem-se as abordagens baseadas na discretização do objeto do Capítulo 3. Particularmente, o uso dos conjuntos normais de discretização (CHRISTOFIDES; WHITLOCK, 1977) no caso com defeitos são discutidos, no contexto da formulação linear e algoritmo de decomposição de Benders, originalmente propostos para o PCBGR. Adicionalmente, um algoritmo baseado em Programação por Restrições também é desenvolvido como estratégia de solução alternativa. Tais abordagens são analisadas nos casos restrito e irrestrito, a partir de experimentos computacionais, com as instâncias propostas por Afsharian, Niknejad e Wäscher (2014).

No Capítulo 5 são apresentadas duas novas formulações de Programação Não-Linear Inteira Mista (PNLIM), uma compacta e uma pseudopolinomial, para o PCBGR. Também são apresentadas suas formulações de PLIM equivalentes. Tais formulações são baseadas na abordagem de combinações horizontais e verticais sucessivas dos itens demandados (e subpadrões), isto é, a abordagem *bottom-up*, originalmente proposta por Wang (1983). Adicionalmente, restrições que limitam o corte a padrões d -estágios, com d é número inteiro positivo, são propostas às formulações. Experimentos computacionais são realizados para avaliar o desempenho das formulações lineares, em comparação às formulações de Lodi e Monaci (2003) e Furini, Malaguti e Thomopoulos (2016), com instâncias de *benchmark* e instâncias geradas com o gerador de instâncias de Silva, Oliveira e Wäscher (2014).

No Capítulo 6, aborda-se o PCTGR, motivado por uma aplicação industrial de corte de aço. O PCTGR pode ser entendido como a versão tridimensional do PCBGR. Inicialmente, estende-se a formulação compacta do Capítulo 5 para o caso tridimensional. Motivado pela aplicação industrial que considera poucos estágios guilhotinados, uma formulação ao PCTGR que lida estritamente com padrões 3-estágios é proposta. Como método de solução, o algoritmo *bottom-up* de Wang (1983), originalmente proposto para o caso bidimensional, é estendido para os casos tridimensionais não-estagiado e 3-estágios. Tais abordagens são analisadas nos casos restrito e irrestrito, a partir de experimentos computacionais, com as instâncias propostas por De Queiroz et al. (2012) e Egeblad e Pisinger (2009), e comparadas a essas duas abordagens.

Por fim, no Capítulo 7 as conclusões desta tese são apresentadas, e oportunidades para pesquisas futuras são discutidas.

Destaca-se que o PCBGR_D e o PCTGR, ambos entendidos como extensões do PCBGR nesta tese de doutorado, têm complicantes particulares. Até onde esse autor tem conhecimento, as duas únicas formulações de PLIM da literatura para o PCBGR (BEN MESSAOUD; CHU; ESPINOUSE, 2008; FURINI; MALAGUTI; THOMOPULOS, 2016) não são diretamente estendíveis ao PCBGR_D, isto é, à presença de defeitos no objeto. Além disso, essas mesmas formulações no contexto do PCTGR iriam exigir números excessivos de variáveis e de restrições, o que tende a inviabilizar essas possíveis abordagens no contexto de *solvers* de propósito geral para instâncias realistas do problema.

Capítulo 2

Problemas de corte guilhotinado e restritos

Neste capítulo discutem-se duas tipologias dos Problemas de Corte e Empacotamento (PCE) na Seção 2.1, assim como distintos padrões de corte bidimensionais e tridimensionais na Seção 2.2. Discute-se a revisão da literatura de trabalhos relacionados ao objeto de pesquisa dessa tese na Seção 2.3. Por fim, na Seção 2.4 são apresentadas formulações de Programação Linear Inteira Mista (PLIM) ao PCBGR não-estagiado, 2-estágios e 1-grupo.

Objetivos do capítulo:

- Apresentar as tipologias dos PCE;
- Discutir a bibliografia do PCBGR, PCBGR.D e PCTGR;
- Apresentar formulações de PLIM para o PCBGR existentes na literatura.

2.1 Tipologias para os Problemas de Corte e Empacotamento

Os trabalhos seminiais de Gilmore e Gomory na década de 1960 contribuíram para aumentar o interesse da comunidade acadêmica nos Problemas de Corte e Empacotamento (PCE). Esse fenômeno foi confirmado pelo aumento de publicações no tema nas décadas seguintes, e inclusive por diversos *surveys* como Dyckhoff et al. (1985), Haessler e Sweeney (1991), Oliveira e Wäscher (2007), entre outros. Destaca-se também o *EURO Special Interest Group on Cutting and Packing* (ESICUP) ligado à Associação Europeia de Sociedades de Pesquisa Operacional (EURO).

Para contribuir na compreensão desses problemas, Dyckhoff (1990) propôs uma tipologia para classificá-los segundo alguns critérios. Quase duas décadas depois, Wäscher, Haubner e Schumann (2007) aperfeiçoaram os critérios de Dyckhoff (1990) ao propor uma nova tipologia. Ambas classificações são apresentadas a seguir.

2.1.1 Tipologia de Dyckhoff (1990)

Dyckhoff (1990) propôs categorizar os PCE por meio de quatro critérios. Esses critérios e seus domínios são apresentados a seguir.

1. Dimensão:

- (1) Uma dimensão;
- (2) Duas dimensões;
- (3) Três dimensões;
- (N) N-dimensões, com $N > 3$.

2. Tipos de Atribuição:

- (B) Todos os objetos e uma seleção dos itens;
- (V) Uma seleção dos objetos e todos os itens.

3. Características dos objetos:

- (O) Um objeto;
- (I) Diversos objetos de formas idênticas;
- (D) Diversos objetos de formas diferentes.

4. Características dos itens:

- (F) Poucos itens de formas diferentes;
- (M) Muitos itens de muitas formas diferentes;
- (R) Muitos itens com poucas formas diferentes (não-congruentes);

(C) Formas congruentes.

Segundo essa tipologia, o primeiro critério condiz ao número mínimo de dimensões geométricas necessárias para descrever o padrão de corte, isto é, a *dimensão*. O segundo critério trata sobre a relação dos conjuntos de objetos e itens, isto é, o *tipo de atribuição*. Esse critério analisa se todos os objetos são utilizados para cortar/empacotar uma parte dos itens (B), ou se apenas uma parte dos objetos será utilizada para obter todos os itens (V). O terceiro e quarto critérios são, respectivamente, sobre as *características dos objetos* e *características dos itens*. Esses critérios analisam sobretudo a heterogeneidade desses conjuntos.

A combinação dos domínios de cada critério resulta em 96 tipos de problemas. Um tipo de problema é descrito pela quádrupla “dimensão/tipo de atribuição/característica dos objetos/característica dos itens”. A Tabela 2.1 ilustra a classificação de alguns problemas clássicos de corte e empacotamento, conforme essa tipologia. Um símbolo faltante para um critério significa que todos os respectivos domínios são possíveis. Seguindo as definições dessa tipologia, o PCBGR) poderia ser classificado como 2/B/O, enquanto o PCTGR como 3/B/O, com os requerimentos adicionais de corte guilhotinado e produção restrita.

Tabela 2.1: Classificação de alguns problemas conforme a tipologia de Dyckhoff (1990).

Problema	Quádrupla
Problema da mochila	1/B/O
Problema de <i>bin packing</i> unidimensional	1/V/I/M
Problema de corte de estoque unidimensional	1/V/I/R
Problema de corte de estoque bidimensional	2/V/I/R
Problema de carregamento de contêineres	3/V/I ou 3/B/O

2.1.2 Tipologia de Wäscher, Haußner e Schumann (2007)

Wäscher, Haußner e Schumann (2007) indicaram que a tipologia de Dyckhoff (1990) “inicialmente forneceu um excelente instrumento de organização e categorização da literatura existente e nova. No entanto, com o passar dos anos algumas deficiências dessa tipologia ficaram evidentes”. Essas deficiências estão relacionadas a possibilidade de um mesmo problema-padrão possuir duas quádruplas, ou um problema apresentar uma quádrupla não consensual. Por exemplo, enquanto Dyckhoff (1990) classificou o Problema da dimensão aberta (do inglês, *Strip Packing Problem*) como 2/V/D/M, Wäscher, Haußner e Schumann (2007) julgou que a quádrupla 2/V/O/M seria mais adequada.

A tipologia de Wäscher, Haußner e Schumann (2007) é baseada na tipologia de Dyckhoff (1990), e considera cinco critérios para compreender esses problemas, que são indicados a seguir:

1. Tipos de atribuição:
 - (a) Maximizar a saída (do inglês, *output maximization*): os objetos disponíveis não são suficientes à alocação de todos os itens;
 - (b) Minimizar a entrada (do inglês, *input minimization*): os objetos disponíveis são suficientes à alocação dos itens.
2. Variedade dos itens:
 - (a) Idênticos;
 - (b) Pouco heterogêneos;
 - (c) Muito heterogêneos.
3. Dimensão:
 - (a) 1-, 2-, 3-dimensional;
 - (b) n -dimensional.
4. Variedade dos objetos:
 - (a) Um único objeto: com dimensões fixas ou variáveis;
 - (b) Vários objetos: idênticos, pouco heterogêneos ou muito heterogêneos.
5. Forma dos itens:
 - (a) Regulares;
 - (b) Irregulares.

A partir desses critérios, Wäscher, Haußner e Schumann (2007) classificaram a literatura dos PCE como tipos de problemas básicos, intermediários e refinados. Os problemas básicos são avaliados pela combinação dos critérios de tipos de atribuição e variedade dos itens. A seguir, a Figura 2.1 ilustra essa proposta e destaca o tipo de problema proposto para estudo nesta tese. Os tipos de problemas intermediários são avaliados segundo a variedade dos objetos, a partir dos problemas básicos. Os problemas refinados são definidos pelos critérios de dimensão e forma dos itens, a partir dos problemas intermediários.

O Problema de Alocação (do inglês, *Placement Problem*) é o problema-padrão a ser estudado nesta tese, no contexto de padrões bidimensionais e tridimensionais de corte guilhotinado e restritos. Esses problemas têm a seguinte classificação, respectivamente, *Two-dimensional Rectangular Single Large Object Placement Problem* e *Three-dimensional Rectangular Single Large Object Placement Problem*. Seguindo as definições dessa tipologia, os problemas de alocação lidam com a alocação de (alguns) itens pouco heterogêneos (isto é, de poucas cópias por tipo de item) para um conjunto

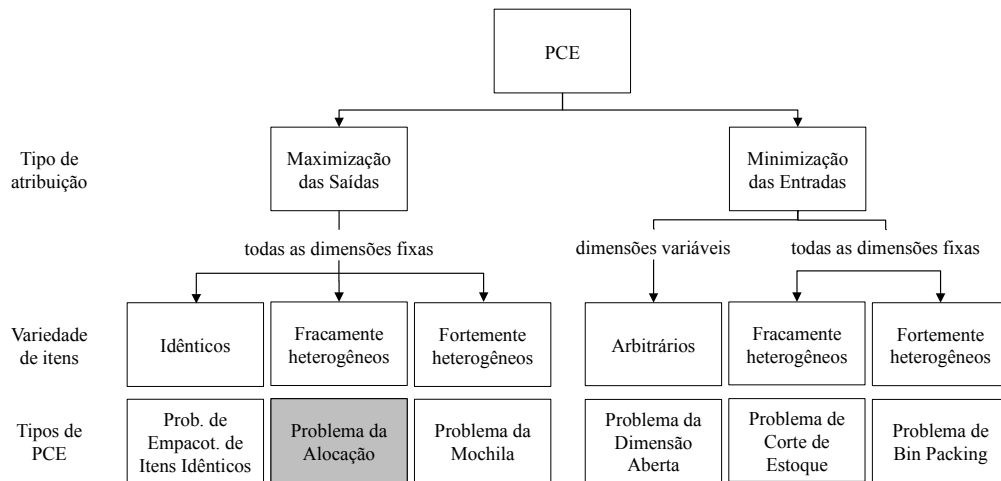


Figura 2.1: Tipos básicos de Problemas de Corte e Empacotamento. Fonte: Adaptado de Wäscher, Haußner e Schumann (2007).

restrito de objetos maiores, buscando maximizar algum critério, como o valor dos itens. Eles são problemas de maximização de saídas, assim nem todos os itens serão cortados do(s) objeto(s).

2.2 Problemas de corte bidimensionais e tridimensionais guilhotinados

A análise das tipologias da seção anterior indica que um problema de corte ou de empacotamento requer dois conjuntos de dados: o conjunto dos materiais a ser cortado ou preenchido, chamado de *conjunto dos objetos*, e o conjunto de materiais demandados a ser obtido por corte ou armazenamento, chamado de *conjunto dos itens*.

Essa tese de doutorado considera o problema de corte de retângulos menores (itens) a partir de um retângulo maior (objeto) no caso bidimensional, e de corte de blocos retangulares menores (itens) a partir de um bloco retangular maior (objeto) no caso tridimensional. Assim, no contexto bidimensional, apenas duas dimensões são necessárias para descrever esses padrões de corte, visto que considera-se que os itens e o objeto têm a mesma espessura.. A seguir, alguns tipos de padrões bidimensionais são discutidos. Nota-se que esses padrões também podem ser considerados no contexto tridimensional, partindo do princípio que os conjuntos de itens e de objetos são formados por blocos retangulares, em vez de retângulos.

O sistema produtivo e os equipamentos de corte podem impor outras restrições aos padrões de corte. Por exemplo, na indústria moveleira, papelreira e de vidros é usual que os cortes ocorram de forma ortogonal, isto é, que os itens tenham suas bordas paralelas às bordas do objeto. Além disso, os cortes devem ocorrer de borda à borda do objeto. Esse tipo de padrão é chamado de corte guilhotinado. Se essa restrição não é

imposta, o padrão é chamado de não-guilhotinado. A ortogonalidade é sempre considerada nesta tese, seja para cortes guilhotinados ou não-guilhotinados, a não ser que seja explicitado o contrário. A Figura 2.2 ilustra esses tipos de padrão de corte; os desperdícios são representados pelo retângulo hachurado. A Figura 2.3 ilustra dois padrões de corte tridimensionais, um guilhotinado e outro não-guilhotinado.

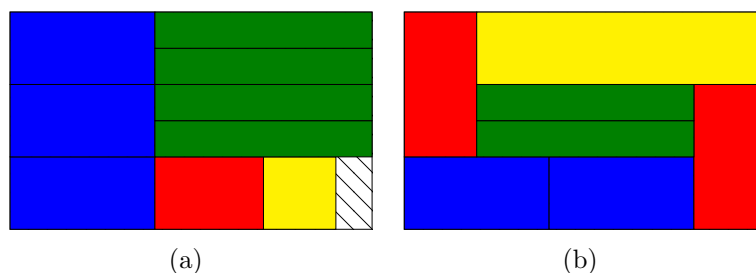


Figura 2.2: Padrão de corte bidimensional guilhotinado (a) e não-guilhotinado (b).

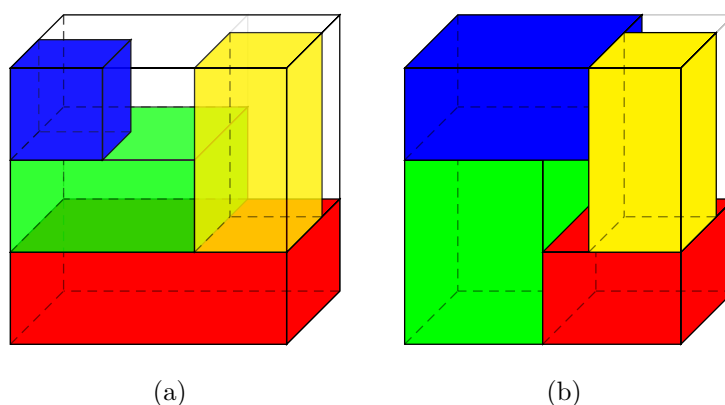


Figura 2.3: Padrão de corte tridimensional guilhotinado (a) e não-guilhotinado (b).

Uma sub-classe dos padrões de corte guilhotinados são os padrões d -estagiados. O número de estágios de um padrão de corte está associado ao número de rotações de 90° realizados pela serra de corte para obter os itens. Assim, pode-se limitar o arranjo dos itens nos objetos a d -estágios, por exemplo, dois ou três estágios. Por outro lado, quando não se limita o número máximo de estágios guilhotinados, tem-se um padrão não-estagiado — por padrão não-estagiado entende-se um padrão infinitamente estagiado.

Os padrões 2-estágios realizam cortes horizontais (ou verticais) produzindo tiras; em seguida, realizam cortes verticais (ou horizontais) para obter os itens. Se as tiras apresentam itens de mesma altura dentro delas, tem-se os padrões 2-estágios exatos, caso contrário tem-se o caso não-exato. Os padrões 3-estágios primeiramente produzem tiras e, em seguida, pilhas, que podem apresentar diversos itens. Se a largura dos itens nas pilhas for distinta, tem-se o caso não-exato, caso contrário tem-se o caso exato. A Figura 2.4 ilustra esses tipos de padrões. Lodi, Martello e Monaci (2002), Puchinger e Raidl (2007), Silva, Alvelos e Valério de Carvalho (2010) e Furini et al. (2012) propuseram formulações

matemáticas para esses tipos de padrões. Alguns autores não distinguem a operação de separação dos itens e dos desperdícios em padrões d -estagiados. Dessa forma, um padrão 2-estágios não-exato é considerado um padrão 3-estágios; um padrão 3-estágios não-exato seria um padrão 4-estágios, e assim por diante.

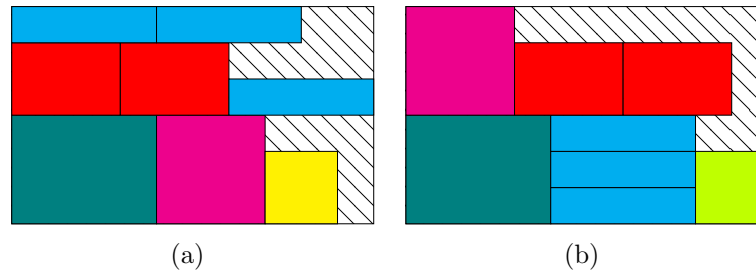


Figura 2.4: Padrão de corte bidimensional 2-estágios não-exato (a) e 3-estágios não-exato (b).

Os padrões de corte do tipo tabuleiro são casos especiais dos padrões anteriores. Um padrão 1-grupo é um caso especial do padrão 2-estágios em que os itens em cada tira têm as mesmas larguras, de forma que elas podem ser empilhadas e cortadas em uma única operação. Um padrão p -grupo, com $p \geq 2$, apresenta p padrões 1-grupo. A Figura 2.5 ilustra esses tipos de padrões. Esses padrões também admitem os casos exato e não-exato. O caso não-exato exige uma operação adicional para separar os itens dos desperdícios de corte, afinal permite que itens de comprimentos e larguras diferentes sejam alocados às tiras horizontais e verticais. Yanasse e Morabito (2006) e Yanasse e Morabito (2008) propuseram formulações matemáticas para esses tipos de padrões.

O Problema de Corte Bidimensional Guilhotinado Não-Estagiado é o caso mais geral dentre os problemas de corte bidimensionais guilhotinados. Esse tipo de padrão oferece soluções com desperdícios de corte pelo menos iguais ou inferiores aos padrões d -estagiados. Em outras palavras, enquanto esses últimos padrões podem ser produzidos em tempos menores pelas máquinas de corte, em detrimento da utilização de material, os padrões não-estagiados buscam o melhor uso dos recursos de matéria-prima. O PCBGR é uma extensão desse problema em que, pelo menos para algum tipo de item, o número máximo de vezes que ele pode aparecer no padrão é estritamente inferior ao limitante geométrico. Por limitante geométrico, entende-se o produto entre as divisões inteiras de cada dimensão do objeto pela correspondente dimensão do tipo de item. O PCBGR será enunciado na Seção 2.2.2, assim como os demais problemas a serem aqui estudados. Segundo Yoon, Ahn e Kang (2013), o PCBGR é mais difícil de resolver que o problema de corte bidimensional guilhotinado irrestrito ou que problemas estagiados restritos; dado que, em contraponto ao PCBGR, esses dois últimos problemas apresentam formulações de Programação Dinâmica (PD), que conseguem rapidamente resolver instâncias desafiadoras (CINTRA et al., 2008).

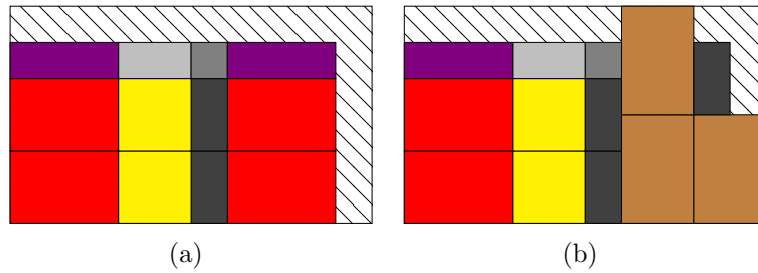


Figura 2.5: Padrão de corte bidimensional 1-grupo exato (a) e 2-grupos exato (b).

2.2.1 Restrições práticas

Outras restrições são relevantes no contexto de aplicações práticas de operações de corte bidimensionais ou tridimensionais. Essas restrições podem ser negligenciadas, ou contornadas por algumas estratégias. Algumas dessas restrições são discutidas a seguir.

Restrições de objeto

Uma restrição comum nos casos ortogonais é limitação de rotação dos itens, devido à anisotropia do objeto a ser cortado. Por exemplo, o corte de madeira requer o caso de orientação fixa, pois a madeira apresenta fibras de crescimento, de forma que a rotação dos itens poderia afetar o comportamento estrutural da peça. Por tanto, se o item de dimensões (w, l) é considerado diferente do item (l, w) , tem-se o caso de orientação fixa, caso contrário a rotação é permitida. Nota-se que, quando a rotação dos itens é permitida, esse caso pode ser facilmente lidado ao introduzir novos tipos de itens (w, l) na instância para cada tipo de item original (l, w) , considerando agregadas suas demandas.

Devido a irregularidades no objeto ou características estéticas que os itens devam apresentar, algumas regiões do objeto podem não ser aptas a produzir: (i) todos os itens requeridos, ou (ii) alguma seleção deles. Esse caso é conhecido como objetos defeituosos na literatura, e normalmente requer métodos específicos para sua resolução (VIANNA; ARENALES, 2006; AFSHARIAN; NIKNEJAD; WÄSCHER, 2014). A Figura 2.6 ilustra um padrão bidimensional guilhotinado em objeto com três defeitos, destacados pelos retângulos de tabuleiro. Um defeito, em geral, é representado pelo menor conjunto de retângulos, em que ele possa ser inscrito.

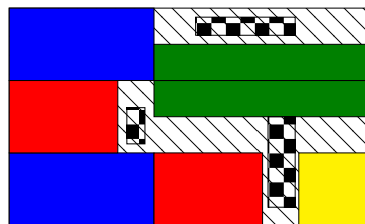


Figura 2.6: Padrão de corte guilhotinado de objeto com três defeitos.

Neste contexto, Gilmore e Gomory (1965) discutiram que irregularidades na

espessura do objeto podem alterar o valor obtido pela produção de certo item. Em outras palavras, esse valor seria dado pelas suas dimensões (considerando o caso não-ponderado), e também pela sua posição no objeto.

Distâncias mínimas entre cortes é outra restrição prática no contexto de objetos frágeis, como vidros. Em geral, esse requerimento não é abordado pela literatura.

Restrições de itens

As restrições dos itens normalmente envolvem: (i) a limitação sobre as quantidades possíveis de serem cortadas (isto é, o caso restrito); e (ii) a consideração de diferentes valorizações na função-objetivo, proporcionais à área dos itens ou não.

Restrições de tecnologia de corte

Restrições sobre o arranjo geométrico podem também considerar a espessura da corte, de acordo com a serra utilizada, que provoca imperfeições sobre as peças. Possivelmente, operações de reparos são necessárias sobre elas. No entanto, essa característica pode ser facilmente contornada ao adicionar essas dimensões de corte sobre as dimensões dos objetos e itens sob análise.

Nota-se que o requerimento de padrões guilhotinados é uma restrição da tecnologia de corte considerada.

2.2.2 Definições dos problemas abordados

O PCBGR propõe padrões bidimensionais ortogonais guilhotinados não-estagiados e restritos, conforme discutido nas seções anteriores. Esta seção define esse problema no contexto do Problema da Alocação, seguindo a tipologia de Wäscher, Haußner e Schumann (2007), para corroborar à comparação das formulações presentes na literatura (BEN MESSAOUD; CHU; ESPINOUSE, 2008; FURINI; MALAGUTI; THOMOPULOS, 2016). Neste contexto, a seção também define o PCBGR.D e o PCTGR. Note que o Problema da Alocação é um problema de maximização das saídas, em que a seleção mais valiosa do conjunto dos itens capaz de ser obtida de um único objeto deve ser encontrada. As Definições 1, 2 e 3 enunciam o PCBGR, PCBGR.D e PCTGR, respectivamente.

Definição 1 *O Problema de Corte Bidimensional Guilhotinado e Restrito (PCBGR) considera um objeto retangular de comprimento L e largura W , e um conjunto $K = \{1, \dots, m\}$ de tipos de itens retangulares demandados, de comprimento l_k , largura w_k , valor v_k , requeridas em até u_k cópias, com $k \in K$. O objetivo é selecionar e cortar uma seleção de cópias dos tipos de itens $k \in K$, que maximiza a soma total do valor v_k dos itens cortados, a partir do objeto. Os requerimentos são: (i) cortes guilhotinados não-estagiados; (ii) a produção não superior ao limitante u_k para cada tipo de item $k \in K$;*

e (iii) a não-sobreposição de quaisquer dois pares de itens, assim como que os itens não excedam o tamanho do objeto.

Definição 2 O Problema de Corte Bidimensional Guilhotinado e Restrito de Placas com Defeitos (PCBGR_D) considera um objeto retangular de comprimento L e largura W , com δ regiões defeituosas representadas por retângulos, e um conjunto $K = \{1, \dots, m\}$ de tipos de itens retangulares demandados, de comprimento L_k , largura W_k , valor V_k , requeridas em até U_k cópias, com $k \in K$. Cada região defeituosa $d \in \{1, \dots, \delta\}$ é caracterizada pelo seu comprimento l_d e largura w_d , com canto inferior-esquerdo fixado na posição (ox_d, oy_d) , que considera o objeto como um sistema de coordenadas Cartesianas, cujo canto inferior-esquerdo é a origem de tal sistema. O objetivo é selecionar e cortar uma seleção de cópias dos tipos de itens $k \in K$, que maximiza a soma total do valor V_k dos itens cortados, a partir do objeto. Os requerimentos são: (i) cortes guilhotinados não-estagiados; (ii) a produção não superior ao limitante U_k para cada tipo de item $k \in K$; (iii) a não-sobreposição de quaisquer dois pares de itens, assim como que os itens não excedam o tamanho do objeto; e, (iv) que os itens não sejam cortados a partir de regiões defeituosas.

Definição 3 O Problema de Corte Tridimensional Guilhotinado e Restrito (PCTGR) considera um bloco retangular (objeto) de comprimento L , largura W e altura H , e um conjunto $K = \{1, \dots, m\}$ de tipos de blocos retangulares demandados (itens), de comprimento l_k , largura w_k , altura h_k , valor v_k , requeridas em até u_k cópias, com $k \in K$. O objetivo é selecionar e cortar uma seleção de cópias dos tipos de itens $k \in K$, que maximiza a soma total do valor v_k dos itens cortados, a partir do objeto. Os requerimentos são: (i) cortes guilhotinados não-estagiados; (ii) a produção não superior ao limitante u_k para cada tipo de item $k \in K$; e, (iii) a não-sobreposição de quaisquer dois pares de itens, assim como que os itens não excedam o tamanho do objeto.

Note que as Definições 1 e 2 distinguem a notação dos itens, pois optou-se por denotar os comprimentos e as larguras dos defeitos com letras minúsculas no PCBGR_D. Particularmente, no contexto da tipologia de Wäscher, Haußner e Schumann (2007), o Problema da Alocação e o Problema da Mochila diferem pela heterogeneidade dos itens no contexto de maximização de saídas, independentemente da dimensão considerada.

2.3 Revisão da literatura

Discute-se nessa seção a revisão bibliográfica dos problemas PCBGR, PCBGR_D e PCTGR, assim como problemas relacionados, destacando as principais estratégias de resolução registradas na literatura dos PCE. Ela é dividida em quatro partes. Inicialmente, um sumário das principais abordagens é apresentado na Seção 2.3.1. A revisão do

PCBGR é discutida na Seção 2.3.2. Nas Seções 2.3.3 e 2.3.4 são discutidas, respectivamente, a revisão dos problemas PCBGR_D e PCTGR.

2.3.1 Sumário das principais abordagens

As principais abordagens do PCBGR, PCBGR_D, PCTGR, e de algumas abordagens relacionadas, são sumarizadas, a seguir, na Tabela 2.2. Para informações de outros problemas de corte e empacotamento amplamente estudados pela literatura, o leitor é direcionado ao livro-texto de Scheithauer (2018). Nessa tabela, os trabalhos foram agrupados por método de solução, e apresentados de forma cronológica.

Pode-se notar que a literatura tem abordado o PCBGR, principalmente, por meio de representações em árvores, utilizando alguma técnica de busca em árvore, aliado ao uso de limitantes inferiores e superiores para reduzir o espaço de busca. Os algoritmos combinatórios de Wang (1983) e Oliveira e Ferreira (1990) são algoritmos de busca em árvore. Destaca-se também algoritmos recursivos e baseados em Programação Dinâmica. Por outro lado, os problemas PCBGR_D e PCTGR, conforme discutido nas seções anteriores, têm sido menos abordados.

Tabela 2.2: Sumário das principais abordagens para os problemas PCBGR, PCBGR_D e PCTGR.

Autores	Problema			Abordagem		Extensão	Descrição
	PCBGR	PCBGR_D	PCTGR	Exata	Heurística		
Christofides e Whitlock (1977)	✓			✓			Busca em árvore e PD
Wang (1983)	✓			✓			Algoritmos combinatórios
Oliveira e Ferreira (1990)	✓				✓		Algoritmos combinatórios
Morabito, Arenales e Arcaro (1992)	✓				✓	Irrestrito	Grafo E/OU e heurísticas
Viswanathan e Bagchi (1993)	✓			✓			Busca em árvore e PD
Christofides e Hadjiconstantinou (1995)	✓			✓			Busca em árvore e relax. de espaço de estados em PD
Hifi (1997)	✓			✓			Busca em árvore e PD
Cung, Hifi e Cun, Le (2000)	✓			✓			Algoritmo <i>branch-and-bound</i>
Álvarez-Valdés, Parajón e Tamarit (2002b)	✓				✓		Meta-heurísticas GRASP e Busca Tabu
Álvarez-Valdés, Parajón e Tamarit (2002a)	✓				✓	Múltiplos objetos	Meta-heurísticas e Geração de Colunas (GC)
Hifi (2004a)	✓				✓		Busca em árvore com PD e <i>hill climbing</i>
Chen (2008)	✓				✓		Algoritmo recursivo <i>top-down</i>
Morabito e Pureza (2010)	✓				✓		Grafo E/OU e relax. de espaço de estados
Dolatabadi, Lodi e Monaci (2012)	✓			✓			Algoritmo recursivo <i>bottom-up</i>
Yoon, Ahn e Kang (2013)	✓			✓			Algoritmo <i>branch-and-bound</i>
Wei e Lim (2015)	✓				✓		Algoritmo recursivo bottom-up e <i>top-down</i>
Velasco e Uchoa (2019)	✓			✓	✓	Múltiplos objetos	PD com relax. de espaço de estados, GRASP e GC
Lodi e Monaci (2003)	✓			✓		2-estágios	Formulação de PLIM
Silva, Alvelos e Valério de Carvalho (2010)	✓			✓		2 e 3-estágios	Formulação de PLIM
Yanasse e Morabito (2006)	✓			✓		1-grupo	Formulação de PLIM
Yanasse e Morabito (2008)	✓			✓		2- e 3-grupos	Formulação de PLIM
Ben Messaoud, Chu e Espinouse (2008)	✓			✓			Formulação de PLIM
Furini, Malaguti e Thomopulos (2016)	✓			✓		Um e vários objetos	Formulação de PLIM
Scheithauer e Terno (1988)		✓		✓		Irrestrito	PD
Afsharian, Niknejad e Wäscher (2014)		✓			✓	Irrestrito	PD
Pisinger e Sigurd (2007)	✓	✓		✓			Decomposições e PR
Amossen e Pisinger (2010)			✓	✓		Minimizar núm. de objetos	Decomposições e PR
De Queiroz et al. (2012)			✓	✓		Irrestrito e Múltiplos objetos	PD e GC

2.3.2 Problema de Corte Bidimensional Guilhotinado e Restrito (PCBGR)

Nesta seção discute-se a revisão bibliográfica das abordagens que lidaram com o PCBGR e de abordagens relacionadas. Os trabalhos que abordaram alguma versão do Problema de Corte de Estoque Bidimensional e, que de alguma forma abordaram o PCBGR, por exemplo, no contexto de um algoritmo de geração de colunas, também são discutidos. Nota-se que a literatura apresenta duas formulações de PLIM para o PCBGR; elas são os trabalhos de Ben Messaoud, Chu e Espinouse (2008) e Furini, Malaguti e Thomopulos (2016). Essas formulações são discutidas na Seção 2.4, assim como as formulações de Lodi e Monaci (2003) para padrões 2-estágios e de Yanasse e Morabito (2006) para padrões 1-grupo.

Em relação à classe de Problemas de Corte Bidimensionais, Lodi, Martello e Monaci (2002) apresentaram um *survey* com modelos matemáticos, limitantes inferiores, algoritmos de aproximação, heurísticas e meta-heurísticas e também abordagens enumerativas. Em relação ao PCBGR, a maior parte dos trabalhos envolve abordagens heurísticas, diante da complexidade de resolução do problema e pelos altos tempos computacionais dos métodos exatos (YOON; AHN; KANG, 2013).

Christofides e Whitlock (1977) desenvolveram a primeira abordagem para o PCBGR. Esses autores propuseram um método exato baseado em um algoritmo de busca em profundidade (*depth-first search*) ao desenvolver uma árvore que enumera todos os possíveis padrões de corte, no qual os ramos correspondem aos cortes guilhotinados e os nós aos itens. Para restringir a busca, limitantes baseados num problema de transporte e num problema da mochila bidimensional são considerados. Esse tipo de abordagem é conhecida na literatura como abordagem *top-down*. Para reduzir a busca, eles consideraram efeitos de simetria e ordem da sequência de cortes para evitar a análise de padrões constituídos pelos mesmos itens. Assim como Herz (1972), eles apresentaram o conceito de padrões normais ou canônicos, que consideram cortes apenas em posições que representam combinações lineares inteiras positivas dos comprimentos e larguras dos tipos de itens. Christofides e Hadjiconstantinou (1995) aprimoraram esse método ao considerar limitantes provenientes de uma relaxação do espaço de estados de uma formulação baseada em Programação Dinâmica e um procedimento do tipo de sub-gradiente. As formulações matemáticas propostas no Apêndice A são inspiradas nesse conceito.

Wang (1983) propôs outra abordagem para esse problema baseada em dois algoritmos combinatórios. Esses algoritmos propõem padrões de cortes por combinações sucessivas de construções horizontais e verticais, ou seja, retângulos são gerados pelas combinações sucessivas dos itens demandados. Para reduzir a quantidade de combinações avaliadas e medir o limite de erro em relação à solução ótima, dois indicadores são propostos. Esse tipo de abordagem é conhecida na literatura como abordagem *bottom-up*. Esses algoritmos foram revisitados e aprimorados por Vasko (1989), Oliveira e Ferreira

(1990), Viswanathan e Bagchi (1993), Parada, Gómes e Diego (1995) e Vasko e Bartkowski (2009). As formulações matemáticas propostas no Capítulo 5 são inspiradas nesse conceito.

Oliveira e Ferreira (1990) refinaram o algoritmo de Wang (1983) e obtiveram soluções de qualidade similar, com tempos de processamento significativamente reduzidos. Eles aperfeiçoaram o critério de aspiração, isto é, o critério de aceitação dos itens ao avaliar cada solução parcial obtida, que se maior que certo limiar são descartados. Eles se basearam na formulação de Programação Dinâmica (PD) de Gilmore e Gomory (1966) para o caso irrestrito nesse processo.

Viswanathan e Bagchi (1993) propuseram um método exato de busca *best-first*, baseados no método de Wang (1983), no qual se busca construir padrões guilhotinados no objeto de corte a partir de arranjos horizontais e verticais (sub-retângulos). Quando esses sub-retângulos não violam as restrições de demanda das peças nem o tamanho do objeto de corte, o algoritmo os coloca no canto inferior esquerdo do objeto de corte e explora a área não ocupada nesse objeto em busca de bons limitantes superiores, ao utilizar equações de funções recursivas.

Parada, Gómes e Diego (1995) consideraram as melhorias de Oliveira e Ferreira (1990) e propuseram gerar retângulos por meio de uma pesquisa informada em uma representação do problema por meio de grafos E/OU aditivo, no algoritmo AAO*. Parada et al. (1998) propuseram um algoritmo baseado em *Simulated Annealing* (SA) para o PCBGR, representando retângulos em uma árvore binária. A formulação proposta envolve o mapeamento dos padrões de corte como uma árvore binária, para facilitar a geração aleatória de soluções vizinhas.

Parada et al. (2000) desenvolveram um estudo comparativo das abordagens de Wang (1983), Oliveira e Ferreira (1990), Parada, Gómes e Diego (1995) e do algoritmo genético (AG) de Parada, Muñoz e Alvarenga (1995). Após estudos computacionais com 1000 instâncias geradas aleatoriamente, esses autores destacaram que o SA e o AG foram os únicos métodos capazes de resolver todas as instâncias, e rapidamente com desperdício inferior a 20%. O SA se destaca com menores tempos de processamento, enquanto o AG por obter soluções de melhor qualidade. Em relação ao algoritmo de Wang, eles analisaram que ele é capaz de obter soluções ótimas em cenários combinatórios reduzidos, enquanto que as abordagens de Oliveira e Ferreira (1990) e Parada, Gómes e Diego (1995) se destacam em cenários combinatórios moderados.

Morabito e Arenales (1996) propuseram uma abordagem baseada em busca de grafo E/OU, em que os nós indicam as peças e os arcos indicam os cortes guilhotinados. Assim, um padrão de corte é compreendido como um caminho completo no grafo. Os autores lidaram com a limitação sobre o número máximo de estágios e essa abordagem pode ser estendida para lidar com problemas de dimensões maiores. Para encontrar soluções foi considerado uma busca híbrida baseada em heurísticas. Em Morabito, Arenales e Arcaro (1992) foi proposta uma estratégia de solução semelhante para o problema

não-guilhotinado irrestrito, e em Morabito e Arenales (1994) um problema tridimensional, isto é, o Problema de Carregamento de Contêineres (do inglês, *Container Loading Problem*) foi abordado com essa técnica.

Hifi (1997) aprimorou a abordagem exata de Viswanathan e Bagchi (1993) ao introduzir mochilas unidimensionais canalizadas e técnicas de Programação Dinâmica para obter melhores limitantes inferiores e superiores. Cung, Hifi e Cun, Le (2000) aprimorou a abordagem de Hifi (1997). Esses autores desenvolveram um algoritmo *branch-and-bound* que utiliza um limitante inferior de melhor qualidade no nó raiz, definem limitantes superior mais refinados aos nós da árvore e utiliza novas estratégias para eliminação de simetrias. Yoon, Ahn e Kang (2013) também propuseram um algoritmo *branch-and-bound* em busca *best-first*. Os autores desenvolveram métodos para eliminar padrões duplicados e refinar os limitantes superiores; além disso, eles propuseram uma estratégia para podar mais de um nó por vez.

Fayard, Higi e Zissimopoulos (1998) abordaram os problemas de corte bidimensionais guilhotinados ponderados e não-ponderados, restritos e irrestritos, isto é, as quatro versões desse problema. Por instância não-ponderada, entende-se que o valor de cada um de seus tipos de itens é sua respectiva área; caso contrário, tem-se uma instância ponderada. Eles propuseram uma heurística baseada em mochilas unidimensionais, e resolvidas por técnicas de Programação Dinâmica, para combinar tiras horizontais e verticais.

Álvarez-Valdés, Parajón e Tamarit (2002b) desenvolveram um algoritmo refinado de Busca Tabu para problemas de grande porte (isto é, as instâncias APT de itens demandados bem pequenos em relação ao objeto), que obtém soluções de boa qualidade em tempo computacional moderado. Os autores também desenvolveram um algoritmo GRASP capaz de obter soluções de boa qualidade rapidamente para problemas restritos e irrestritos. Inicialmente, duas heurísticas construtivas foram propostas para auxiliar na construção de padrões de corte. Elas se baseiam em limitantes obtidos ao resolver mochilas unidimensionais. As instâncias propostas por esses autores são consideradas instâncias de grande porte na literatura, e são consideradas nos experimentos computacionais do Capítulo 5. Álvarez-Valdés, Parajón e Tamarit (2002a) consideraram o PCBGR com múltiplos objetos, isto é, o problema de corte de estoque. O algoritmo de geração de colunas foi considerado, e subproblemas foram resolvidos por meio da técnica de Programação Dinâmica de Beasley (1985a), e métodos heurísticos propostos em Álvarez-Valdés, Parajón e Tamarit (2002b). Para obter soluções inteiras, a partir da solução ótima do problema-mestre restrito relaxado, foram propostos procedimentos de arredondamento para cima, truncamento na árvore *branch-and-bound* e solução de um problema residual. Os experimentos computacionais do Capítulo 5 consideram as instâncias APT.

Hifi (2004a) propôs uma abordagem híbrida que combina um procedimento construtivo e estratégias *hill-climbing* com procedimentos baseados em programação dinâmica. O autor utiliza uma série de mochilas unidimensionais para contribuir com limi-

tantes inferiores. Hifi, M'Hallah e Saadi (2009) desenvolveram métodos para o PCBGR-duplo, que apresenta restrições sobre a quantidade mínima e máxima de vezes que cada item pode aparecer no padrão. Segundo esses autores, trata-se da única abordagem para esse problema. Eles propuseram abordagens exatas e aproximadas. O método aproximado envolve duas fases: na primeira, o PCBGR é resolvido por meio de uma heurística construtiva; a segunda fase, envolve eliminar a possível não factibilidade da solução do PCBGR para o PCBGR-duplo. O método exato é baseado num algoritmo *branch-and-bound*, cujos limitantes inferiores são obtidos da primeira fase e limitantes superiores são obtidos a partir de problemas da mochila relaxados.

Pisinger e Sigurd (2007) propuseram um método para resolver o Problema de Empacotamento Bidimensional. Eles propuseram uma abordagem que combina Programação por Restrições (PR) e um algoritmo *branch-and-cut* para resolvê-lo. A geração de colunas partiu de um problema de cobertura de conjuntos, dado que cada item apresenta única cópia em problemas de empacotamento. A técnica de Programação por Restrições permitiu lidar com cortes guilhotinados e objetos irregulares. Os resultados computacionais mostraram que os limitantes inferiores da geração de colunas eram apertados, de forma que o método foi capaz de resolver instâncias de grande porte à otimalidade.

Puchinger e Raidl (2007) também geraram colunas, mas para o Problema de Empacotamento em três estágios. Eles propuseram formulações compactas para os casos restrito e irrestrito. Eles utilizaram o método de geração rápida de colunas, em que o Subproblema (SP) é resolvido, em sequência quando necessário, por uma heurística gulosa, um algoritmo genético, um método exato simplificado, e apenas quando a coluna não foi gerada, o modelo de *pricing* exato é utilizado. Eles também consideraram desigualdades duais ótimas para estabilizar a geração de colunas no modelo de partição.

Chen (2008) propôs um algoritmo recursivo para o PCBGR. Esse algoritmo divide o objeto em uma sequência de blocos retangulares menores. Para um bloco sob análise, ele seleciona um item e o aloca no canto inferior-esquerdo do bloco, e encontra a direção de corte para gerar outros dois blocos menores. Limitantes superiores baseados no problema irrestrito são considerados para reduzir o espaço de soluções. Os resultados computacionais foram satisfatórios quando comparados aos de Álvarez-Valdés, Parajón e Tamarit (2002b) e Hifi (2004a).

Morabito e Pureza (2010) propuseram uma abordagem heurística baseada na proposta de Christofides e Hadjiconstantinou (1995), com busca em grafo E/OU. Em vez de realizar a busca em árvores, os autores propuseram aplicar uma heurística de “factibilização” nas soluções infactíveis do procedimento de otimização do tipo subgradiente. Embora esse método não garanta a otimalidade, ele gera limitantes inferiores de boa qualidade, e no caso de obtenção de soluções ótimas, ele gera um certificado de otimalidade.

Silva, Alvelos e Valério de Carvalho (2010) propuseram uma formulação compacta para o problema de corte de estoque bidimensional e restrito, limitado a 2

e 3 estágios. Esses autores estenderam a abordagem de Dyckhoff (1981) para o caso unidimensional e, assim, as variáveis representam a quantidade de vezes em que cada corte possível no objeto é realizado. Os estudos computacionais realizados mostraram que essa abordagem foi superior às propostas de Lodi e Monaci (2003), que propuseram dois modelos para o caso 2-estágios, e de Puchinger e Raidl (2007), no contexto de muitas cópias por tipo de itens.

Cui e Chen (2012) propuseram uma heurística gulosa ao problema. O algoritmo envolve uma função de recursão para considerar um conjunto de padrões de corte e usa uma técnica para descartar ramos não promissores. Os autores destacam o possível uso desse algoritmo numa estratégia de procedimento heurístico sequencial (*Sequential Heuristic Procedure*) para a versão corte de estoque (Múltiplos objetos) do PCBGR, isto é, de propor padrões de corte até a exaustão da demanda.

Furini et al. (2012) abordaram o problema de corte de estoque bidimensional guilhotinado irrestrito para múltiplos objetos em cortes 2-estágios, em um esquema de geração de colunas. Para a solução do SP gerador de padrões estagiados, os autores propuseram uma formulação de PLIM e um procedimento heurístico baseado em Programação Dinâmica. Abordagens heurísticas foram consideradas para obtenção de soluções inteiras. Os resultados computacionais mostraram que essa abordagem é competitiva à proposta de Cintra et al. (2008).

Dolatabadi, Lodi e Monaci (2012) desenvolveram um procedimento recursivo capaz de encontrar soluções ótimas para o PCBGR. Esse procedimento requer um limitante inferior. Esses autores propuseram dois algoritmos baseados nesse procedimento. O Algoritmo 1 itera esse procedimento recursivo variando esse limitante inferior. O Algoritmo 2 é um *Branch-and-Cut* que se baseia na solução de um problema de Programação Linear Inteira.

Wei e Lim (2015) desenvolveram um algoritmo heurístico que combina as abordagens *top-down* e *bottom-up* para o PCBGR. Sob perspectiva *bottom-up*, o algoritmo primeiro constrói alguns subpadrões, isto é, combinações de retângulos demandados. Sob perspectiva *top-down*, o algoritmo aloca, a cada passo, um bloco ao canto inferior-esquerdo de um espaço livre do objeto, dividindo-o em objetos menores. Os autores alegaram que essa abordagem baseada em blocos reduz o número de passos para empacotar o objeto. Os experimentos computacionais consideraram alguns conjuntos de instâncias, e mostraram que a abordagem é capaz de obter soluções ótimas para instâncias de tamanho moderado, e superar outras abordagens da literatura para instâncias maiores.

Velasco e Uchoa (2019) propuseram um algoritmo baseado em programação inteira, que obtém (provadamente segundo os autores) os pesos ótimos para o algoritmo do tipo sub-gradiente proposto por Christofides e Hadjiconstantinou (1995), de relaxação de espaços de estados para obtenção de limitantes superiores ao PCBGR. Esse algoritmo é estendido para obter pesos bidimensionais aos itens. Os autores obtiveram soluções primais por meio de uma meta-heurística GRASP. Essa abordagem também lida com o

PCBGR de Múltiplos objetos, associando esses algoritmos ao método de Geração de Colunas (CG), pelo arredondamento da solução fracionária de CG para uma solução inteira, e resolução do problema residual, definido pelos itens com demanda ainda não totalmente atendida. Os experimentos computacionais consideraram a rotação ou orientação fixa de itens, e a abordagem foi capaz de obter soluções de alta qualidade, na maioria dos casos ótimas, para as diversas instâncias consideradas.

Poucas abordagens na literatura lidaram diretamente com padrões de corte do tipo tabuleiro (p -grupos). Para padrões 1-grupo, destaca-se os modelos de Programação Não-linear Inteira (PNLI) de Morabito e Arenales (2000) e de Programação Linear Inteira Mista (PLIM) de Yanasse e Morabito (2006). Enquanto que para padrões 2-grupos e 3-grupos, destaca-se os modelos lineares de Yanasse e Morabito (2008). Yanasse e Katsurayama (2005, 2008) propuseram algoritmos enumerativos para esses tipos de padrões de corte.

Recentemente, Russo et al. (2019) apresentaram um *survey* especificamente para o PCBGR. Esses autores discutiram as abordagens ao problema classificadas entre: formulações de PD, algoritmos de busca em árvores (*top-down* e *bottom-up*) e algoritmos heurísticos e meta-heurísticos. O leitor é direcionado a esse trabalho para uma revisão e categorização dos limitantes superiores ao problema. Particularmente, os autores também discutiram a falta de acurácia de algumas abordagens da literatura devido à restrição sobre o número máximo de cópias a ser produzido, isto é, algumas formas de construção de padrões de corte com garantia de otimalidade para o caso irrestrito são, na verdade, heurísticas para o caso restrito. Por exemplo, segundo Russo et al. (2019), as abordagens de Dolatabadi, Lodi e Monaci (2012) podem conduzir a solução não-ótimas, apesar desses autores alegaram que sua abordagem é exata.

2.3.3 Problema de Corte Bidimensional Guilhotinado e Restrito de Placa com Defeitos (PCBGR_D)

O problema de corte bidimensional de único objeto com defeitos, em sua forma geral, considera um objeto heterogêneo, sobre o qual as peças devem ser obtidas, sendo que alguns tipos de itens não devem ser cortados a partir de algumas regiões do objeto por restrições de qualidade. Materiais com defeitos aparecem em diversos contextos produtivos como, por exemplo, na indústria de móveis customizados no corte de objetos de madeira com imperfeições de nós de crescimento, na indústria de vidros em imperfeições por enxofre, e nas indústrias de granito e mármore com trincas nas pedras, na indústria de couro devido à imperfeições no tecido, entre outras. Apesar de sua aplicabilidade, poucos estudos na literatura abordaram esse tipo de problema. Embora o trabalho seminal de Gilmore e Gomory (1965) já tivesse indicado o possível uso de diferentes valores aos tipos de item na função-objetivo, dependendo de suas posições no objeto heterogêneo, além de suas dimensões. Durak e Aksu (2017) discutiram algumas

aplicações de corte e empacotamento que consideram materiais com defeitos em contextos unidimensionais. As principais abordagens no contexto bidimensional são baseadas em técnicas de Programação Dinâmica (PD) para o problema irrestrito, isto é, em que não se limita o número máximo de cópias por tipo de item.

Hahn (1968) e Scheithauer e Terno (1988) propuseram abordagens para padrões guilhotinados 3-estágios baseados na formulação de PD de Gilmore e Gomory (1961), ao considerar cada estágio como uma função de mochila unidimensional, e os defeitos são analisados no último estágio. Os padrões de Hahn (1968) são um caso especial dos padrões 3-estágios, pois sua abordagem limita que um único tipo de item seja obtido nos cortes do último estágio.

Carnieri, Mendoza e Luppold (1993) desenvolveram uma heurística baseada na formulação de PD de Gilmore e Gomory (1965), para objetos com um único defeito. Para o caso de objetos com múltiplas regiões com defeitos, Vianna e Arenales (2006) propuseram um procedimento de enumeração implícita com estratégia de busca *Hill-Climbing*, em um espaço de soluções representado por um grafo E/OU. Neidlein et al. (2008) desenvolveram um limitante superior para estes algoritmos.

Mais recentemente, Afsharian, Niknejad e Wäscher (2014) desenvolveram uma heurística efetiva para o caso não-estagiado, que permite lidar com diversos defeitos, ao estender as formulações de PD (GILMORE; GOMORY, 1965; BEASLEY, 1985a). Neidlein, Scholz e Wäscher (2016) propuseram um gerador de instâncias para o caso irrestrito com defeitos.

Todas as abordagens mencionadas previamente trataram de problemas sem limitar o número máximo de vezes que um tipo de item pode aparecer no padrão (caso irrestrito). De fato, abordagens de PD podem ser intratáveis no caso restrito, devido ao enorme número de espaços de estados necessários para representar, além das posições de corte, a quantidade limitada de itens a ser produzida. Segundo o conhecimento desse autor, Pisinger e Sigurd (2007) desenvolveram a única abordagem da literatura que poderia abordar o caso bidimensional guilhotinado não-estagiado e restrito, a partir de um algoritmo baseado em Programação por Restrições (PR). No entanto, os autores não forneceram experimentos computacionais associados a objetos com defeitos. Esse problema é abordado no Capítulo 4 por meio de Otimização Matemática e um outro algoritmo baseado em PR.

2.3.4 Problema de Corte Tridimensional Guilhotinado e Restrito (PCTGR)

Os problemas de corte tridimensionais guilhotinados têm sido pouco abordados pela literatura (BORTFELDT; WÄSCHER, 2013). No entanto, problemas como o PCTGR tem potencial aplicabilidade, como no corte de aço, de espumas para colchões e de pedras de granito. Outro exemplo foi indicado por Gilmore e Gomory (1965) sobre

o corte de blocos de grafite para anodos. Particularmente, o corte guilhotinado de aço é um processo industrial importante em empresas de soluções de aço que cortam blocos estocados de dezenas de toneladas para produzir itens de até poucas dezenas de gramas. Esse processo considera o custo e qualidade dos objetos estocados, as demandas por itens, e os tempos de produção e entrega. Por exemplo, uma operação de corte em um objeto de poucos metros em cada dimensão para gerar dois sub-objetos pode demorar mais do que uma hora. Essas decisões mostram os objetivos conflitantes entre a busca de maiores taxas de utilização e a busca de menores tempos de produção. No Capítulo 6 ambos os casos são considerados ao propor abordagens para padrões não-estagiados e de 3-estágios.

O Problema de Carregamento de Contêineres (PCC) (do inglês, *Container Loading Problem*) é um problema relacionado ao PCTGR, que pode considerar restrições adicionais no contexto de sistemas logísticos, como a distribuição de peso, as prioridades de carregamento, empilhamento, posicionamento e estabilidade (BORTFELDT; WÄSCHER, 2013). O PCC pertence a classe dos problemas *NP-Hard* (SCHEITHAUER, 1992), e é abordado principalmente por alguns tipos de heurísticas, recentemente comparadas por Zhao et al. (2016). Apesar de padrões de empacotamento guilhotinados não serem desejáveis, e devam ser evitados devido a questões estabilidade de carga, há abordagens de solução como *wall-building* e *layer-building* (TOFFOLO et al., 2017) que geram cortes guilhotinados como efeito colateral. Recentemente, Silva, Toffolo e Wauters (2019) propuseram um estudo comparativo dos principais métodos exatos para o PCC, como as formulações matemáticas de posições relativas de Chen, Lee e Shen (1995) e de objeto discretizado de Junqueira, Morabito e Sato Yamashita (2012b) e de Junqueira, Morabito e Sato Yamashita (2012a).

Os problemas de corte tridimensionais guilhotinados têm sido menos abordados do que os problemas bidimensionais guilhotinados pela literatura, possivelmente devido ao menor número de aplicações industriais. Um problema de empacotamento tridimensional muito estudado é o Problema de Carregamento de Contêineres (PCC), do inglês, *Container Loading Problem*. Esse problema, em geral, não requer a restrição de corte guilhotinado visando maiores taxas de utilização do contentor, e também por questões de estabilidade de carga. Por outro lado, ele considera restrições adicionais como, por exemplo, distribuição de carregamento, prioridade de carregamento, restrições de empilhamento e posicionamento, entre outras (BORTFELDT; WÄSCHER, 2013).

Há poucas abordagens relacionadas ao PCTGR, que são principalmente baseadas em técnicas de PD. Morabito e Arenales (1994) generalizaram ao caso tridimensional a formulação de PD de Gilmore e Gomory (1965)'s PD para padrões bidimensionais de 2-estágios, ao considerar camadas e pilhas. Eles também desenvolveram uma abordagem de grafo E/OU com limitantes melhorados e heurísticas para reduzir o espaço de soluções. Hifi (2002) propôs um algoritmo baseado em uma versão melhorada da formulação de PD de Gilmore e Gomory (1965) e alguns algoritmos aproximativos. Em seguida, Hifi (2004b) adaptou ao caso tridimensional a formulação de PD de Gilmore e

Gomory (1966) e desenvolveu um algoritmo de busca em profundidade. De Queiroz et al. (2012) abordaram o PCTGR ao estender as formulações de PD de Beasley (1985b) para padrões não-estagiados e estagiados. Eles também abordaram as versões de Corte de Estoque e de Dimensão Aberta do PCTGR ao adaptar seus algoritmos no contexto de soluções heurísticas. Nota-se novamente que algoritmos de PD geralmente consideram padrões irrestritos (isto é, padrões que não limitam o número máximo de cópias por tipo de item); essas abordagens podem ser intratáveis no caso restrito devido ao grande número de espaços de estados (isto é, a maldição da dimensão).

Martello et al. (2007) abordaram o problema de ortogonalmente empacotar um conjunto de itens no número mínimo de objetos. Os autores desenvolveram uma abordagem *branch-and-bound* exata que considera um algoritmo baseado em PR para preencher cada objeto por meio de padrões não-guilhotinados. Egeblad e Pisinger (2009) desenvolveram um algoritmo baseado em *Simulated Annealing* para as versões bidimensional e tridimensional do 2D_R_CG_SLOPP sem considerar a restrição de guilhotina. Essas abordagens consideram empacotamentos de retângulos/blocos retangulares a partir de pares/triplas para representar os padrões de corte. Amossen e Pisinger (2010) propuseram algoritmos para o *Bin Packing Problem* de qualquer dimensão. Eles consideraram padrões guilhotinados e não-guilhotinados. Mais recentemente, Liu et al. (2014) abordaram uma versão do PCBGR que garante suporte total e restrições de orientação ao propor um algoritmo heurístico de busca em árvore. Esse algoritmo é dividido em passos em que os itens são agrupados em tiras, e então as tiras são agrupadas em camadas; diversas funções de mochila unidimensional são resolvidas nesses passos.

Dentre todas essas abordagens, apenas Morabito e Arenales (1994) destacaram que suas abordagens eram capazes de lidar com o caso restrito, e de forma heurística.

2.4 Formulações matemáticas para problemas de corte bidimensionais guilhotinados

Essa seção discute as abordagens de Programação Linear Inteira Mista (PLIM) de Lodi e Monaci (2003) para padrões 2-estágios na Seção 2.4.1, e de Yanasse e Morabito (2006) para padrões 1-grupo na Seção 2.4.2. As formulações de PLIM de Ben Messaoud, Chu e Espinouse (2008) e Furini, Malaguti e Thomopoulos (2016) para o PCBGR são discutidas, respectivamente, nas Seções 2.4.3 e 2.4.4. Para mais detalhes, os artigos originais são indicados ao leitor. As formulações propostas nos Capítulos 3 e 5 são comparadas a essas formulações em seus experimentos computacionais.

A seguir, os parâmetros do PCBGR são caracterizados:

- L e W são, respectivamente, o comprimento e a largura do objeto original;
- m é o número de tipos de itens distintos demandados;

- K é o conjunto dos tipos de itens, com $k \in \{1, \dots, m\}$;
- l_k e w_k são, respectivamente, o comprimento e a largura do tipo de item k ;
- v_k e u_k são, respectivamente, o valor da peça e o número máximo de cópias do tipo de item k a ser cortado;
- n é o número total de itens demandados $(\sum_{k=1}^m u_k)$.

Buscou-se padronizar os índices e termos utilizados nas formulações das próximas seções. As notações distintas das previamente apresentadas, quando necessárias, são destacadas.

2.4.1 Formulação de Lodi e Monaci (2003)

As abordagens clássicas para padrões 2-estágios consideram que o objeto original é dividido em tiras horizontais ou verticais. Suponha o caso de tiras horizontais de comprimento L , cuja largura de cada tira (ou prateleira) é a largura do item mais largo obtido a partir dela. Lodi e Monaci (2003) propuseram dois modelos de PLIM para problemas de corte bidimensionais guilhotinados 2-estágios, assim como desigualdades válidas e análises comparativas entre elas. Essas formulações partem da premissa que para qualquer solução ótima 2-estágios não-exata, sempre existe uma solução equivalente, cujo primeiro item de cada prateleira é sempre o mais largo, conforme ilustrado pela Figura 2.4(a); tal item é chamado de “inicializador” da tira.

Optou-se por utilizar o “modelo 1” desses autores (pelo seu melhor desempenho), que considera cada item como distinto, mesmo diante de dimensões idênticas. Além disso, os itens são considerados ordenados de tal forma que $w_1 \geq w_2 \geq \dots \geq w_n$. O modelo considera que n possíveis tiras podem ser “inicializadas”, sendo que uma tira k sempre é inicializada pelo item k , com $k \in \{1, \dots, n\}$. Assim, os cortes dos itens demandados em tiras é descrita pela seguinte variável de decisão:

$$x_{jk} = \begin{cases} 1, & \text{se o item } j \text{ é cortado a partir da tira } k, j \geq k, \\ 0, & \text{caso contrário.} \end{cases}$$

A seguir, o Modelo (2.1) descreve a formulação de Lodi e Monaci (2003) para padrões 2-estágios não-exato.

$$\text{Max} \quad \sum_{j=1}^n v_j \sum_{k=1}^j x_{jk}, \quad (2.1a)$$

$$\text{s.a } \sum_{k=1}^j x_{jk} \leq 1, \quad j \in \{1, \dots, n\}, \quad (2.1b)$$

$$\sum_{j=k+1}^n w_j x_{jk} \leq (L - l_k) x_{kk}, \quad k \in \{1, \dots, n-1\}, \quad (2.1c)$$

$$\sum_{k=1}^n w_k x_{kk} \leq W, \quad (2.1d)$$

$$x_{jk} \in \{0, 1\}, \quad k \in \{1, \dots, n\}, j \in \{k, \dots, n\}. \quad (2.1e)$$

A função-objetivo (2.1a) maximiza a soma total do valor dos itens cortados. As restrições (2.1b) garantem que cada item é cortado no máximo uma vez, e a partir de uma tira cuja largura é ao menos igual à largura do item (dado que eles foram ordenados). As restrições (2.1c) asseguram que o comprimento de cada tira é respeitado. A restrição (2.1d) assegura que a largura do objeto é respeitada. As restrição (2.1e) tratam do domínio da variável de decisão.

A Figura 2.7, a seguir, ilustra um padrão 2-estágios não-exato, destacando as tiras horizontais dessa formulação. Assim, a variável x_{kk} significa: (i) a tira k existe; e, (ii) o item k é o primeiro item cortado nessa tira. Nota-se que para obter padrões de tiras verticais, uma opção simples é trocar os parâmetros de entrada referentes às dimensões horizontais e verticais. Outra alternativa é reescrever as restrições (2.1c) e (2.1d) para o caso vertical, com ordenamento prévio segundo os comprimentos dos itens (em vez de suas larguras).

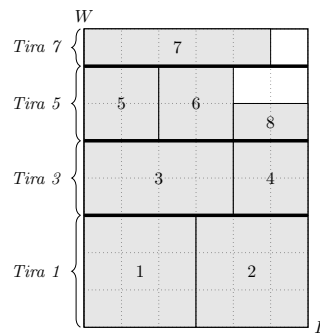


Figura 2.7: Exemplo de padrão 2-estágios não-exato segundo Lodi e Monaci (2003).

No contexto de *solvers* de propósito geral, o Modelo (2.1) tem melhor desempenho quando o número de itens n pequeno. Por outro lado, o modelo de Silva, Alvelos e Valério de Carvalho (2010) para padrões 2-estágios tem melhor desempenho com o número de cópias por tipo de item é grande, o que em geral resulta em um número de itens n maior.

2.4.2 Formulação de Yanasse e Morabito (2006)

Yanasse e Morabito (2006) propuseram formulações para padrões 1-grupo exatos e não-exatos. Por conveniência, a seguir, a formulação não-linear inteira desses autores é apresentada; em seguida, as adaptações para obtenção de um modelo linear equivalente por eles propostas são indicadas. Sejam os seguintes parâmetros e variáveis:

- I e J são, respectivamente, os números de diferentes comprimentos l_k e larguras w_k ;
- o parâmetro $v_{kij} = \begin{cases} v_k, & \text{se } l_k \leq l_i \text{ e } w_k \leq w_j, \\ 0, & \text{caso contrário.} \end{cases}$
- a variável λ_i representa o número de vezes que o comprimento l_i é cortado ao longo do objeto;
- a variável μ_j representa o número de vezes que a largura w_j é cortada ao longo do objeto;
- a variável a_{kij} representa o número de retângulos $l_i \times w_j$ contendo uma peça do tipo k .

A seguir, o Modelo (2.2) descreve a formulação não-linear de Yanasse e Morabito (2006) para padrões 1-grupo não-exatos.

$$\text{Max} \quad \sum_{k=1}^m \sum_{i=1}^I \sum_{j=1}^J v_{kij} a_{kij}, \quad (2.2a)$$

$$\text{s.a.} \quad \sum_{i=1}^I l_i \lambda_i \leq L, \quad (2.2b)$$

$$\sum_{j=1}^J w_j \mu_j \leq W, \quad (2.2c)$$

$$\sum_{k=1}^m a_{kij} \leq \lambda_i \mu_j, \quad i \in \{1, \dots, I\}, j \in \{1, \dots, J\}, \quad (2.2d)$$

$$\sum_{i=1}^I \sum_{j=1}^J a_{kij} \leq u_k, \quad k \in \{1, \dots, m\}, \quad (2.2e)$$

$$\lambda_i, \mu_j, a_{kij} \in \mathbb{Z}_+, \quad k \in \{1, \dots, m\}, i \in \{1, \dots, I\}, j \in \{1, \dots, J\}. \quad (2.2f)$$

A função-objetivo (2.2a) maximiza o valor total dos itens cortados. As restrições (2.2b) e (2.2c) asseguram que as dimensões do objeto são respeitadas. As restrições (2.2d) limitam a quantidade de itens alocados a retângulos $l_i \times w_j$ para a quantidade de retângulos com essas dimensões ($\lambda_i \mu_j$). As restrições (2.2e) restringem as alocações ao caso restrito. As restrições (2.2f) tratam do domínio das variáveis.

O Modelo (2.2) é não-linear devido ao produto de variáveis inteiras na restrição (2.2d). Esses autores propuseram reescrever uma dessas variáveis como uma expansão binária. No caso, $\lambda_i = \sum_{s=1}^{s_i} 2^{s-1} \beta_{is}$, onde $s_i = \lceil \log_2(L/l_i) + 1 \rceil$ representa o número necessário de *bits* para a expansão binária, e a variável binária β_{is} indica se o *bit* s é utilizado. Dessa forma, a restrição (2.2d) é reescrita como:

$$\sum_{k=1}^m a_{kij} \leq \sum_{s=1}^{s_i} 2^{s-1} \beta_{is} \mu_j, \quad i \in \{1, \dots, I\}, j \in \{1, \dots, J\},$$

que pode ser substituída pelas seguintes restrições lineares:

$$\begin{aligned} \sum_{k=1}^m a_{kij} &\leq \sum_{s=1}^{s_i} 2^{s-1} f_{ijs}, & i \in \{1, \dots, I\}, j \in \{1, \dots, J\}, \\ \mu_j - M(1 - \beta_{is}) &\leq f_{ijs} \leq \mu_j, & i \in \{1, \dots, I\}, j \in \{1, \dots, J\}, s \in \{1, \dots, s_i\}, \\ 0 &\leq f_{ijs} \leq M\beta_{is}, & i \in \{1, \dots, I\}, j \in \{1, \dots, J\}, s \in \{1, \dots, s_i\}. \end{aligned}$$

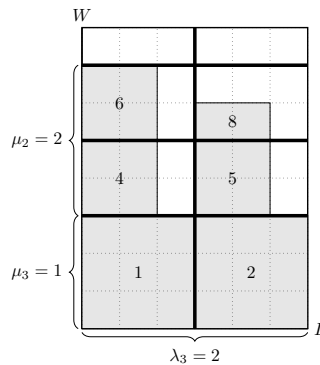


Figura 2.8: Exemplo de padrão 1-grupo não-exato segundo Yanasse e Morabito (2006).

onde M é um número grande (por exemplo, $\lfloor W/w_{min} \rfloor$). Nota-se que $f_{ijs} = \mu_j$, se $\beta_{is} = 1$, e $f_{ijs} = 0$, caso contrário. A Figura 2.8, a seguir, ilustra um padrão 1-grupo não-exato, destacando as expressões dessa formulação, e supondo a mesma instância da Figura 2.7. Os índices das variáveis λ_i e μ_j representam os distintos tamanhos horizontais e verticais, respectivamente. Ressalta-se que para obtenção de padrões 1-grupo exatos, somente é necessário redefinir o parâmetro v_{kij} como v_k , se $l_k = l_i$ e $w_k = w_j$, e 0, caso contrário.

2.4.3 Formulação de Ben Messaoud, Chu e Espinouse (2008)

Ben Messaoud, Chu e Espinouse (2008) propuseram uma formulação binária no contexto do Problema Guilhotinado da Dimensão Aberta (do inglês, *Guillotine Strip Packing Problem* – GSPP). O *Strip Packing Problem* aloca itens retangulares a um objeto retangular que apresenta uma dimensão ilimitada, isto é, uma tira ilimitada. O objetivo

é alocar todos os itens, de forma a minimizar o tamanho dessa dimensão ilimitada, diante de cortes guilhotinados não-estagiados. O problema é restrito, pois existe um número finito de itens a serem alocados.

Esses autores provaram um teorema para caracterizar um padrão guilhotinado. Seja um padrão de corte composto por n itens, de forma que cada item k , com $k \in \{1, \dots, n\}$, é alocado em (α_k, β_k) . Sem perda de generalidade, assume-se que os itens são renumerados de forma que $\alpha_1 \leq \alpha_2 \leq \dots \leq \alpha_n$. Uma permutação $\sigma : \{1, 2, \dots, n\} \mapsto \{1, 2, \dots, n\}$ é definida tal que $\beta_{\sigma(1)} \leq \beta_{\sigma(2)} \leq \dots \leq \beta_{\sigma(n)}$. Para quaisquer inteiros i_1, i_2, j_1 e j_2 , tal que $1 \leq i_1 < i_2 \leq n$ e $1 \leq j_1 < j_2 \leq n$, o seguinte conjunto de itens é definido: $E(i_1, i_2, j_1, j_2) = \{i_1, \dots, i_2\} \cap \{\sigma(j_1), \dots, \sigma(j_2)\}$, que representa o conjunto de itens cujos cantos inferiores-esquerdo estão entre as áreas delimitadas pelos eixos verticais que passam por α_{i_1} e α_{i_2} , e pelos eixos horizontais $\beta_{\sigma(j_1)}$ e $\beta_{\sigma(j_2)}$. O exemplo desses autores para um padrão de 4 itens é ilustrado, a seguir, na Figura 2.9(a). Nesse exemplo, os itens são renumerados de forma que $\alpha_1 \leq \alpha_2 \leq \alpha_3 \leq \alpha_4$. A permutação $\beta_{\sigma(1)} \leq \beta_{\sigma(2)} \leq \beta_{\sigma(3)} \leq \beta_{\sigma(4)}$ é definida com: $\sigma(1) = 3$, $\sigma(2) = 1$, $\sigma(3) = 2$ e $\sigma(4) = 4$. Assim, $E(1, 4, 1, 3) = \{1, 2, 3, 4\} \cap \{\sigma(1), \sigma(2), \sigma(3)\} = \{1, 2, 3, 4\} \cap \{3, 1, 2\} = \{1, 2, 3\}$, que são os itens cujos cantos inferiores-esquerdo estão entre os eixos verticais α_1 e α_4 , e os eixos horizontais $\beta_{\sigma(1)}$ e $\beta_{\sigma(3)}$.

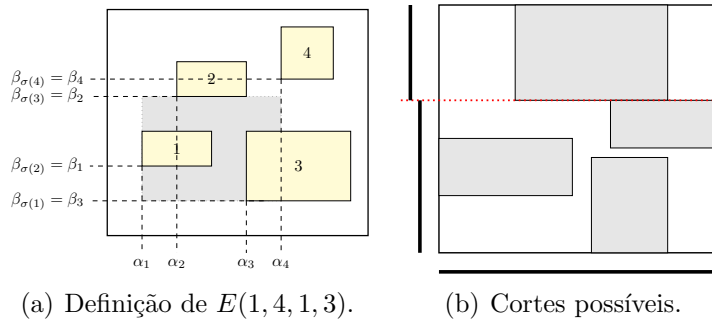


Figura 2.9: Corte guilhotinado. Fonte: Adaptado de Ben Messaoud, Chu e Espinouse (2008).

Esses autores afirmaram que um padrão é guilhotinado, se e somente se, para quaisquer inteiros i_1, i_2, j_1 e j_2 , tal que $1 \leq i_1 < i_2 \leq n$ e $1 \leq j_1 < j_2 \leq n$, ao menos uma das seguintes condições é satisfeita:

1. $E(i_1, i_2, j_1, j_2)$ contém no máximo um elemento (item);
2. $\bigcup_{k \in E(i_1, i_2, j_1, j_2)} (\alpha_k, \alpha_k + l_k)$ é constituído de pelo menos dois intervalos disjuntos, isto é,

$$\bigcup_{k \in E(i_1, i_2, j_1, j_2)} (\alpha_k, \alpha_k + l_k) \neq \left(\min_{k \in E(i_1, i_2, j_1, j_2)} \alpha_k, \min_{k \in E(i_1, i_2, j_1, j_2)} (\alpha_k + l_k) \right);$$
3. $\bigcup_{k \in E(i_1, i_2, j_1, j_2)} (\beta_k, \beta_k + w_k)$ é constituído de pelo menos dois intervalos disjuntos, isto

é,

$$\bigcup_{k \in E(i_1, i_2, j_1, j_2)} (\beta_k, \beta_k + w_k) \neq \left(\min_{k \in E(i_1, i_2, j_1, j_2)} \beta_k, \min_{k \in E(i_1, i_2, j_1, j_2)} (\beta_k + w_k) \right).$$

A Figura 2.9(b) ilustra as projeções dos segmentos horizontais e verticais dos itens de outro padrão de 4 itens, e o conceito de intervalos disjuntos. Enquanto as projeções horizontais mostram que nenhum corte horizontal é possível, as projeções verticais indicam dois intervalos disjuntos e, assim, um corte vertical é possível. Além desse teorema, os autores propuseram um algoritmo polinomial para verificar a condição de padrão guilhotinado. Em seguida, eles desenvolveram uma formulação de programação binária baseada nesse teorema, em que cada item é considerado distinto, mesmo diante de dimensões idênticas.

Para contribuir com a análise dos experimentos computacionais do Capítulo 3, esse modelo foi adaptado para o PCBGR, de forma que as adaptações propostas são aqui destacadas. Por exemplo, os itens de mesmas dimensões não são considerados distintos como na proposta original, logo $k \in \{1, \dots, m\}$. Dado que o PCBGR é problema de maximização de saídas, uma seleção dos itens será considerada para corte, em contrapartida ao GSPP que corta todos os itens (minimização de entradas). Assim, optou-se por considerar como parâmetro n um limitante superior ao número de itens que cabem no objeto, em vez de $\sum_{k \in K} u_k$. Esse limitante pode ser objeto ao resolver uma mochila unidimensional, cujo peso dos itens é dado por suas áreas. A variável x_i , $i \in \{1, \dots, n\}$, representa a posição horizontal do início do i -ésimo item horizontal. A variável y_j , $j \in \{1, \dots, n\}$, representa a posição vertical do início do j -ésimo item vertical. A seguinte variável de decisão caracteriza a alocação dos itens à tira:

$$z_{ijk} = \begin{cases} 1, & \text{se o tipo de item } k \text{ é alocado como } (i, j), \\ 0, & \text{caso contrário,} \end{cases}$$

em que “alocado como (i, j) ” significa que uma cópia do item de tipo $k \in K$ representa o i -ésimo item alocado na posição horizontal, e j -ésimo item alocado na posição vertical. Em outras palavras, eles não representam posições em um objeto discretizado. Os índices $i, i', i_1, i_2, j, j', j_1, j_2$ e k variam no conjunto de itens a serem alocados, isto é, no conjunto $\{1, \dots, n\}$.

As seguintes variáveis intermediárias são necessárias:

$$t_{ijj'} = \begin{cases} 1, & \text{se o item alocado como } (i, j), \text{ caso exista, não exceder horizontalmente } x_{j'}, \\ & \text{com } i' > i, \\ 0, & \text{caso contrário.} \end{cases}$$

$$v_{ijj'} = \begin{cases} 1, & \text{se o item alocado como } (i, j), \text{ caso exista, não exceder verticalmente } y_{j'}, \\ & \text{com } j' > j, \\ 0, & \text{caso contrário.} \end{cases}$$

As seguintes variáveis são usadas para representar as restrições guilhotinadas:

$$p_{i_1 i' j_1 j_2} = \begin{cases} 1, & \text{se nenhum item alocado entre } (i_1, j_1) \text{ e } (i' - 1, j_2) \text{ excede } x_{j'} \text{ com } i' > i_1, \\ & \text{logo, um corte vertical em } x_{j'} \text{ não intercepta nenhum item alocado entre} \\ & (i_1, j_1) \text{ e } (i' - 1, j_2), \\ 0, & \text{caso contrário,} \end{cases}$$

$$q_{i_1 i_2 j_1 j'} = \begin{cases} 1, & \text{se nenhum item alocado entre } (i_1, j_1) \text{ e } (i_2, j' - 1) \text{ excede } y_{j'} \text{ com } j' > j_1, \\ & \text{logo, um corte horizontal em } y_{j'} \text{ não cruza nenhum item alocado entre} \\ & (i_1, j_1) \text{ e } (i_2, j' - 1), \\ 0, & \text{caso contrário,} \end{cases}$$

$$d_{i_1 i_2 j_1 j_2} = \begin{cases} 1, & \text{se existe no máximo um item alocado entre } (i_1, j_1) \text{ e } (i_2, j_2), \\ 0, & \text{caso contrário.} \end{cases}$$

O Modelo (2.3), a seguir, é a reformulação do modelo de Ben Messaoud, Chu e Espinouse (2008) para o PCBGR.

$$\mathbf{Max} \quad \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n v_k z_{ijk}, \quad (2.3a)$$

$$\mathbf{s.a} \quad 0 \leq x_1 \leq x_2 \leq \dots \leq x_n, \quad (2.3b)$$

$$0 \leq y_1 \leq y_2 \leq \dots \leq y_n, \quad (2.3c)$$

$$\sum_{i=1}^n \sum_{j=1}^n z_{ijk} \leq u_k, \quad \forall k, \quad (2.3d)$$

$$\sum_{i=1}^n \sum_{k=1}^n z_{ijk} \leq 1, \quad \forall j, \quad (2.3e)$$

$$\sum_{j=1}^n \sum_{k=1}^n z_{ijk} \leq 1, \quad \forall i, \quad (2.3f)$$

$$x_{i'} - x_i - \sum_{k=1}^n l_j z_{ijk} \geq (t_{ij i'} - 1)L, \quad \forall i, \forall j, \forall i' > i, \quad (2.3g)$$

$$x_{i'} - x_i - \sum_{k=1}^n l_j z_{ijk} \leq t_{ij i'} L, \quad \forall i, \forall j, \forall i' > i, \quad (2.3h)$$

$$y_{j'} - y_j - \sum_{k=1}^n w_k z_{ijk} \geq (v_{ij j'} - 1)W, \quad \forall i, \forall j, \forall j' > j, \quad (2.3i)$$

$$y_{j'} - y_j - \sum_{k=1}^n w_k z_{ijk} \leq v_{ij j'} W, \quad \forall i, \forall j, \forall j' > j, \quad (2.3j)$$

$$(1 - d_{i_1 i_2 j_1 j_2})n \geq \sum_{i=i_1}^{i_1} \sum_{j=j_1}^{j_2} \sum_{k=1}^n z_{ijk} - 1, \quad \forall i_1, \forall j_1, \forall i_2 > i_1, \forall j_2 > j_1, \quad (2.3k)$$

$$p_{i_1 i' j_1 j_2} \leq \sum_{j=j_1}^{j_2} \sum_{k=1}^n z_{i' j k}, \quad \forall i_1, \forall j_1, \forall i' > i_1, \forall j_2 > j_1, \quad (2.3l)$$

$$p_{i_1 i' j_1 j_2} \leq \sum_{i=i_1}^{i'-1} \sum_{j=j_1}^{j_2} \sum_{k=1}^n z_{ijk}, \quad \forall i_1, \forall j_1, \forall i' > i_1, \forall j_2 > j_1, \quad (2.3m)$$

$$(i' - i_1)(j_2 - j_1 + 1)p_{i_1 i' j_1 j_2} \leq \sum_{i=i_1}^{i'-1} \sum_{j=j_1}^{j_2} t_{ij i'}, \quad \forall i_1, \forall j_1, \forall i' > i_1, \forall j_2 > j_1, \quad (2.3n)$$

$$q_{i_1 i_2 j_1 j'} \leq \sum_{i=i_1}^{i_2} \sum_{k=1}^n z_{i j' k}, \quad \forall i_1, \forall j_1, \forall i_2 > i_1, \forall j' > j_1, \quad (2.3o)$$

$$q_{i_1 i_2 j_1 j'} \leq \sum_{i=i_1}^{i_2} \sum_{j=j_1}^{j'-1} \sum_{k=1}^n z_{ijk}, \quad \forall i_1, \forall j_1, \forall i_2 > i_1, \forall j' > j_1, \quad (2.3p)$$

$$(j' - j_1)(i_2 - i_1 + 1)q_{i_1 i_2 j_1 j'} \leq \sum_{i=i_1}^{i_2} \sum_{j=j_1}^{j'-1} v_{ij j'}, \quad \forall i_1, \forall j_1, \forall i_2 > i_1, \forall j' > j_1, \quad (2.3q)$$

$$\begin{aligned} d_{i_1 i_2 j_1 j_2} + \sum_{i'=i_1+1}^{i_2} p_{i_1 i' j_1 j_2} \\ + \sum_{j'=j_1+1}^{j_2} q_{i_1 i_2 j_1 j'} \geq 1, \quad \forall i_1, \forall j_1, \forall i_2 > i_1, \forall j_2 > j_1, \end{aligned} \quad (2.3r)$$

$$L \geq x_i + \sum_{j=1}^n \sum_{k=1}^n l_j z_{ijk}, \quad \forall i, \quad (2.3s)$$

$$W \geq y_j + \sum_{i=1}^n \sum_{k=1}^n w_k z_{ijk}, \quad \forall j, \quad (2.3t)$$

$$z_{ijk} \in \{0, 1\}, \quad \forall i, \forall j, \forall k, \quad (2.3u)$$

$$t_{ijj'} \in \{0, 1\}, \quad \forall i, \forall j, \forall i' > i, \quad (2.3v)$$

$$v_{ijj'} \in \{0, 1\}, \quad \forall i, \forall j, \forall j' > j, \quad (2.3w)$$

$$d_{i_1 i_2 j_1 j_2}, p_{i_1 i_2 j_1 j_2}, q_{i_1 i_2 j_1 j_2} \in \{0, 1\}, \quad \forall i_1, \forall j_1, \forall i_2 > i_1, \forall j_2 > j_1. \quad (2.3x)$$

A função-objetivo (2.3a) maximiza o valor dos itens alocados ao objeto. As restrições (2.3b) e (2.3c) garantem a ordinalidade às posições horizontais e verticais, respectivamente. As restrições (2.3d) garantem o caso restrito, enquanto as restrições (2.3e) à (2.3f) garantem que uma posição é ocupada no máximo uma vez. As restrições (2.3g) e (2.3h) asseguram a definição da variável $t_{ijj'}$. Por simetria, as restrições (2.3i) e (2.3j) asseguram a definição da variável $v_{ijj'}$. As restrições (2.3k) garantem a definição da variável $d_{i_1 i_2 j_1 j_2}$. As restrições (2.3l) à (2.3n) garantem a definição da variável $p_{i_1 i_2 j_1 j_2}$. Por simetria, as restrições (2.3o) à (2.3q) garantem a definição da variável $q_{i_1 i_2 j_1 j_2}$. As restrições (2.3r) asseguram que pelo menos um dos critérios do teorema proposto pelos autores é garantido para cada subpadrão, de forma que o padrão final será guilhotinado. As restrições (2.3s) e (2.3t) garantem os limites do objeto aos itens. As restrições (2.3u) à (2.3x) tratam do domínio de variáveis. Note que esse modelo não requer que as variáveis x_i e y_j sejam inteiras.

As adaptações realizadas para escrever esse modelo como uma mochila foram:

- O parâmetro n é um limitante superior ao número de itens que cabem no objeto, em vez de $\sum_{k \in K} u_k$;
- A função-objetivo original minimiza a altura W da tira, e aqui o valor total dos itens alocados é maximizado;
- As restrições (2.3d) à (2.3f) originais apresentam sinais de igualdade. O sinal de menor ou igual aqui proposto evita a infactibilidade, dado que nem todos os itens ou posições necessariamente serão alocados ao objeto. Particularmente, as restrições (2.3d) foram agregadas por tipo de item, o que tende a contribuir na redução do número de variáveis do modelo;
- As restrições (2.3i) e (2.3j) originais consideram um limitante superior \bar{W} para o tamanho da tira, em vez da dimensão W aqui proposta;
- As restrições (2.3t) originais consideram o valor da função-objetivo, e não a dimensão fechada (W) do objeto.

Os experimentos computacionais realizados pelos autores foram limitados a cenários de 5 itens pelos requisitos de memória computacional e relaxação linear fraca dessa formulação, dado que os números de variáveis e restrições da formulação, embora polinomiais, são da ordem de $O(n^4)$.

2.4.4 Formulação de Furini, Malaguti e Thomopulos (2016)

Furini, Malaguti e Thomopulos (2016) desenvolveram uma formulação de Programação Linear Inteira (PLI) ao PCBGR. Esses autores estenderam a abordagem “one-cut” de Dyckhoff (1981) ao caso guilhotinado não-estagiado. Anteriormente, Silva, Alvelos e Valério de Carvalho (2010) propuseram uma abordagem similar para cortes de 2-estágios e 3-estágios.

Furini, Malaguti e Thomopulos (2016) propuseram os conceitos de *corte* e *chapa*, sendo que uma chapa pode representar o objeto original ou qualquer chapa residual obtida após a realização de um corte. Essa formulação propõe que cada decisão de corte seja representada por uma tripla (q, j, o) , sendo que a posição q indica a distância do canto inferior-esquerdo da chapa j realizada com orientação o , conforme ilustrado na Figura 2.10. A formulação exige, sem perda de generalidade, que os parâmetros de entrada sejam inteiros positivos. Considere os seguintes conjuntos:

J : é o conjunto de chapas retangulares; cada chapa j tem dimensões (l_j, w_j) ;

$\bar{J} \subset J$: é o conjunto de chapas retangulares com dimensões iguais a um dos itens;

O : é o conjunto de possíveis orientações para um corte (horizontal e vertical);

$Q(j, o)$: é o conjunto de posições em que se pode cortar a chapa j com orientação $o \in O$.

A chapa original é representada como $j = 0$. Para uma chapa j e orientação h/v , tem-se que $Q(j, h) \subseteq \{1, \dots, w_j - 1\}$ e $Q(j, v) \subseteq \{1, \dots, l_j - 1\}$. Sejam as seguintes variáveis:

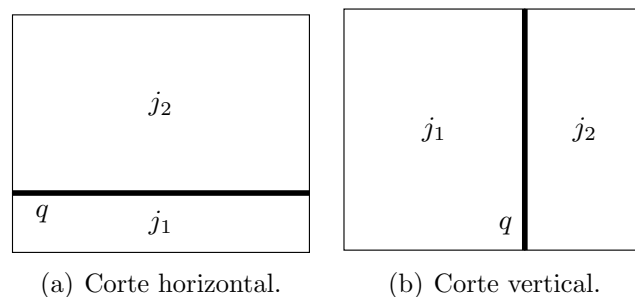


Figura 2.10: Representação da tripla (q, j, o) de decisão de corte. Fonte: Adaptado de Furini, Malaguti e Thomopulos (2016).

x_{qj}^o : número de vezes que a chapa j é cortada na posição q com orientação o ;

y_j : número de chapas j que representam o item demandado j , com $j \in \bar{J}$;

Um parâmetro é necessário para indicar quais chapas são obtidas ao se cortar uma chapa maior. Assim, o parâmetro a_{kj}^o tem valor 1 se a chapa j é obtida ao cortar a chapa k na posição q com orientação o , e 0, caso contrário. Assim, de acordo com a Figura 2.10, uma chapa (original ou intermediária) k , cortada na posição q , gera as chapas j_1 e j_2 . Nota-se que no artigo desses autores, há um pequeno equívoco na descrição dos índices k e j para esse parâmetro; eles estão trocados, o que deixaria o modelo infactível. O Modelo (2.4), a seguir, descreve a formulação desses autores, já corrigida.

$$\text{Max} \quad \sum_{j \in \bar{J}} v_j y_j, \quad (2.4a)$$

$$\text{s.a.} \quad \sum_{k \in J} \sum_{o \in O} \sum_{q \in Q(k,o)} a_{kj}^o x_{qk}^o - \sum_{o \in O} \sum_{q \in Q(j,o)} x_{qj}^o - y_j \geq 0, \quad j \in \bar{J}, j \neq 0, \quad (2.4b)$$

$$\sum_{k \in J} \sum_{o \in O} \sum_{q \in Q(k,o)} a_{kj}^o x_{qk}^o - \sum_{o \in O} \sum_{q \in Q(j,o)} x_{qj}^o \geq 0, \quad j \in J \setminus \bar{J}, \quad (2.4c)$$

$$\sum_{o \in O} \sum_{q \in Q(0,o)} x_{q0}^o + y_0 \leq 1, \quad (2.4d)$$

$$y_j \leq u_j, \quad \forall j \in \bar{J}, \quad (2.4e)$$

$$x_{qj}^o \geq 0 \text{ e inteiro}, \quad j \in J, o \in O, q \in Q(q, o), \quad (2.4f)$$

$$y_j \geq 0 \text{ e inteiro}, \quad j \in \bar{J}. \quad (2.4g)$$

A função-objetivo (2.4a) maximiza o valor total de itens armazenados como itens finais, sendo que v_j representa o valor de cada chapa do tipo j . As restrições (2.4b) impõem que o número de chapas j cortadas para obter chapas menores e armazenadas como itens demandados deve ser inferior ao número de chapas j obtidas pelo corte de chapas maiores k . As restrições (2.4c) são similares às anteriores e tratam do balanço de chapas que não representam itens demandados. A restrição (2.4d) assegura que a chapa original será cortada uma única vez ou será armazenada. Por “chapa armazenada”, entende-se que a chapa não será cortada pois pode representar uma cópia de um tipo de item; nota-se que essa formulação também pode abordar o corte de múltiplos objetos. As restrições (2.4e) garantem o caso restrito, em que u_j denota o número máximo de tipos de itens j presentes no padrão. As restrições (2.4f) e (2.4g) tratam do domínio das variáveis.

O Modelo (2.4) é pseudopolinomial, visto que requer um pré-processamento

para enumerar as variáveis x_{qj}^o , assim como o conjunto de chapas J e o parâmetro a_{kjq}^o . Em outras palavras, essa formulação exige pré-processamento dos dados do cenário a ser analisado, e não é representada (completamente) pelos parâmetros originais do problema, como as dimensões do objeto e dos itens demandados. Sem perda de generalidade, os autores propuseram um procedimento visando obter uma menor cardinalidade nessa enumeração, assim como discretização normal (CHRISTOFIDES; WHITLOCK, 1977; TRICK, 2003). Esses autores também desenvolveram um procedimento de *pricing* de variáveis para lidar com problemas de porte maior. Os autores também apresentaram as adaptações necessárias para que essa formulação lide com o *Strip Packing Problem* e com a versão de Corte de Estoque do PCBGR.

2.5 Comentários finais do capítulo

Nesse capítulo foram discutidos o PCBGR, o PCBGR.D e o PCTGR, assim como as tipologias dos Problemas de Corte e Empacotamento. Alguns tipos de padrões bidimensionais guilhotinados, que também podem ser considerados no caso tridimensional, foram caracterizados. Além disso, uma revisão bibliográfica para esses problemas foi realizada, inclusive com abordagens de padrões bidimensionais mais restritos como 2-estágios e 1-grupo, e com destaques para as formulações de PLIM. O objeto de pesquisa desta tese é o desenvolvimento de formulações de PLIM e de métodos de solução para os problemas discutidos.

Nota-se que as formulações de Lodi e Monaci (2003), Yanasse e Morabito (2006), Ben Messaoud, Chu e Espinouse (2008) e Furini, Malaguti e Thomopoulos (2016) podem facilmente lidar com o caso de rotação de itens, ao adicionar um novo tipo de item $w \times l$ para cada tipo de item $l \times w$ original da instância, considerando agregadas suas cópias quando no caso restrito. Por outro lado, segundo o conhecimento desse autor, a extensão das formulações de Ben Messaoud, Chu e Espinouse (2008) e de Furini, Malaguti e Thomopoulos (2016) para o caso d -estagiado não é direta, assim como a extensão dessas quatro formulações para o caso de placas com defeitos.

Capítulo 3

Problema de corte bidimensional guilhotinado e restrito: propostas de formulações matemáticas e de método de solução baseados em grelha

Neste capítulo discutem-se novas formulações matemáticas e um novo método de solução ao PCBGR, baseados no conceito de discretização do objeto. Uma formulação pseudopolinomial de PNLI e sua formulação linear equivalente são propostas na Seção 3.1. Formulações que lidam estritamente com padrões 2-estágios e 1-grupo são propostas na Seção 3.2. Um algoritmo de decomposição de Benders para o problema é proposto na Seção 3.3. Na Seção 3.4 discutem-se os experimentos computacionais das abordagens propostas.

Objetivos do capítulo:

- Propor uma formulação de PNLI e sua formulação equivalente de PLI ao PCBGR, ambas baseadas em grelha;
- Propor novas formulações de PLI para os problemas 2-estágios e 1-grupo;
- Desenvolver um algoritmo decomposição de Benders, a partir formulação de PLI proposta.

Parte dos resultados deste capítulo foram recentemente publicados:

★ Martin, M.; Birgin, E. G.; Lobato, R. D.; Morabito, R.; Munari, P. *Models for the two-dimensional rectangular single large placement problem with guillotine cuts and constrained pattern*. International Transactions in Operational Research, v. 27, n. 2, p. 767–793, 2020. DOI: <https://doi.org/10.1111/itor.12703>.

3.1 Formulações matemáticas

Esta seção é dividida em três partes. Na Seção 3.1.1 propõe-se uma formulação pseudopolinomial de PNLI que estende o modelo de PLI de Beasley (1985b) ao adicionar conjuntos de variáveis e de restrições associadas a cortes horizontais e verticais, que são utilizados para gerar os padrões guilhotinados. Em seguida, na Seção 3.1.2 desenvolve-se uma formulação de PLI equivalente ao modelo não-linear proposto. Na Seção 3.1.3 discute-se um Procedimento Enumerativo de Variáveis (PEV), que se baseia na técnica de discretização de conjuntos normais (CHRISTOFIDES; WHITLOCK, 1977) para reduzir os números de variáveis e restrições do modelo linear. O leitor é direcionado ao início da Seção 2.4 para a descrição dos parâmetros do PCBGR.

3.1.1 Formulação não-linear baseada na discretização do objeto

Beasley (1985b) propôs uma formulação que aloca itens (de acordo com seus cantos inferiores-esquerdos) a pontos de um objeto discretizado, usando restrições para evitar a sobreposição entre pares de itens alocados. Optou-se por denotar a ideia de objeto discretizado como “grelha”, isto é, trata-se de uma malha reticulada. De acordo com essa abordagem, sem perda de generalidade, as dimensões do objeto e dos itens são inteiros positivos. Seja o conjunto R a discretização completa do objeto, e o conjunto R_k os pontos possíveis para alocação do canto inferior-esquerdo de um tipo de item k , conforme os conjuntos das Expressões (3.1) e (3.2).

$$R = \{(i, j) \in \mathbb{Z}^2 \mid 0 \leq i \leq L, 0 \leq j \leq W\}. \quad (3.1)$$

$$R_k = \{(i, j) \in R \mid i \leq L - l_k, j \leq W - w_k\}. \quad (3.2)$$

A variável de decisão relacionada a alocação de um tipo de item k a uma posição (i, j) do objeto discretizado é definida na Equação (3.3). Além dessa família de variáveis, a formulação proposta ao PCBGR também possui variáveis de decisão associadas aos cortes horizontais ($h_{ii'j}$) e verticais ($v_{ijj'}$), respectivamente, definidas nas Equações (3.4) e (3.5). A Figura 3.1 ilustra essas variáveis.

$$x_{kij} = \begin{cases} 1, & \text{se um tipo de item } k \text{ é alocado à posição } (i, j), \\ 0, & \text{caso contrário,} \end{cases} \quad k \in K, (i, j) \in R_k. \quad (3.3)$$

$$h_{ii'j} = \begin{cases} 1, & \text{se um corte horizontal é realizado de} \\ & (i, j) \text{ para } (i', j), \\ 0, & \text{caso contrário,} \end{cases} \quad (i, j) \in R, (i', j) \in R, i < i'. \quad (3.4)$$

$$v_{ijj'} = \begin{cases} 1, & \text{se um corte vertical é realizado de} \\ & (i, j) \text{ para } (i, j'), \\ 0, & \text{caso contrário,} \end{cases} \quad (i, j) \in R, (i, j') \in R, j < j'. \quad (3.5)$$

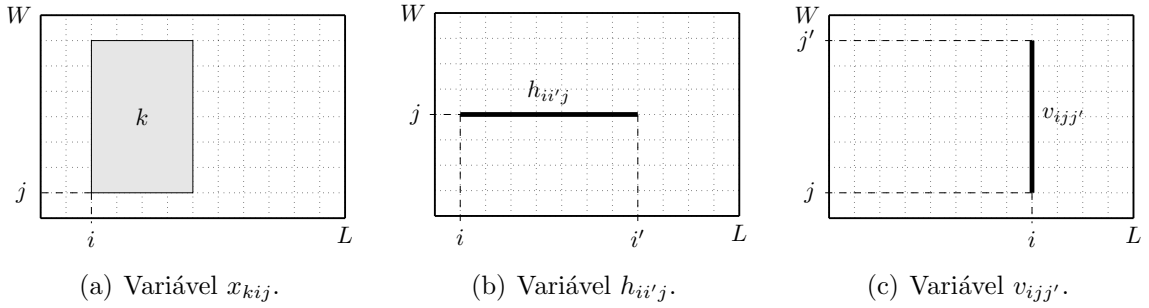


Figura 3.1: Ilustração das variáveis x_{kij} , $h_{ii'j}$ e $v_{ijj'}$.

Seja S_{kij}^h o conjunto de todos os segmentos horizontais (i', i'', j) que cruzam um item alocado representado pela variável x_{kij} , como definido pela Expressão (3.6) e ilustrado na Figura 3.2(a). Similarmente, seja S_{kij}^v o conjunto correspondente aos segmentos verticais, que é definido na Expressão (3.7) e ilustrado na Figura 3.2(b).

$$S_{kij}^h = \{(i', i'', j) \in \mathbb{Z}^3 \mid j < j' < j + w_k, 0 \leq i' < i + l_k, \max\{i, i'\} < i'' \leq L\}, k \in K, (i, j) \in R_k. \quad (3.6)$$

$$S_{kij}^v = \{(i', j', j'') \in \mathbb{Z}^3 \mid i < i' < i + l_k, 0 \leq j' < j + w_k, \max\{j, j'\} < j'' \leq W\}, k \in K, (i, j) \in R_k. \quad (3.7)$$

Para evitar a sobreposição entre qualquer par de itens alocados, tem-se o seguinte parâmetro $f_{kij'j'}$, para cada $k \in K$, $(i, j) \in R_k$ e $(i', j') \in R$, com $i' < L$ e $j' < W$:

$$f_{kij'j'} = \begin{cases} 1, & \text{se } 0 \leq i \leq i' \leq i + l_k - 1 < L \text{ e } 0 \leq j \leq j' \leq j + w_k - 1 < W, \\ 0, & \text{caso contrário.} \end{cases}$$

Nota-se que os conjuntos S_{kij}^h e S_{kij}^v e o parâmetro $f_{kij'j'}$ podem ser ge-

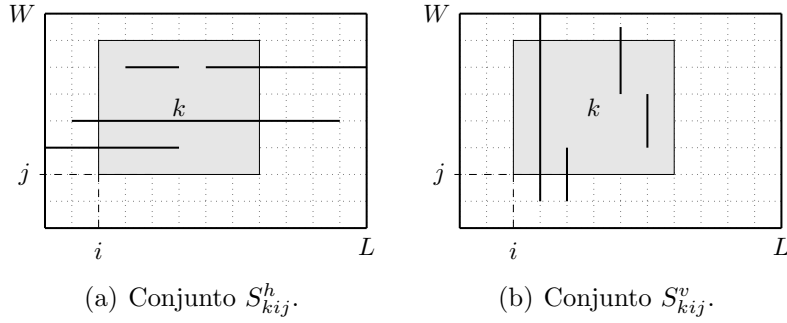


Figura 3.2: Ilustrações de alguns segmentos de S_{kij}^h e S_{kij}^v .

rados durante a montagem do modelo em linguagens algébricas e/ou de programação, isto é, não é necessário dedicar tempo específico para seus processamentos nem memória computacional para seus armazenamentos.

No Modelo (3.8) define-se a formulação pseudopolinomial de PNLI para o PCBGR. Considerando as variáveis de decisão do modelo, destaca-se que: (i) a função objetivo e as restrições que garantem a não-sobreposição de itens e o caso restrito são baseados nas variáveis de alocação x_{kij} apenas, como em Beasley (1985b); (ii) os cortes $h_{i'j'}$ e $v_{i'j'}$ são usados para criar um quadro de guilhotina no objeto; (iii) as variáveis de alocação e de corte são usadas conjuntamente para evitar cortes sobre itens alocados, e para limitar as alocações aos cantos cortados; com as demais restrições, elas garantem a restrição de guilhotina. As variáveis v_{00W} , v_{L0W} , h_{0L0} e h_{0LW} são fixas em um, isto é, as bordas do objeto são necessariamente consideradas cortadas no modelo.

$$\mathbf{Max} \sum_{k \in K} \sum_{(i,j) \in R_k} v_k x_{kij}, \quad (3.8a)$$

s.a

$$\sum_{k \in K} \sum_{(i,j) \in R_k} f_{kij'j'} x_{kij} \leq 1, \quad (i', j') \in R, i' < L, j' < W, \quad (3.8b)$$

$$\sum_{(i,j) \in R_k} x_{kij} \leq u_k, \quad k \in K, \quad (3.8c)$$

$$h_{i_1 i_2 j} \leq \sum_{\substack{0 \leq i'_1, i''_1 \leq i_1; \\ i_2 \leq i'_2, i''_2 \leq L; \\ 0 \leq j_1 < j < j_2 \leq W}} \sum_{\substack{0 \leq j'_1, j''_1 \leq j_1; \\ j_2 \leq j'_2, j''_2 \leq W}} h_{i'_1 i'_2 j_1} h_{i''_1 i''_2 j_2} v_{i_1 j'_1 j'_2} v_{i_2 j''_1 j''_2}, \quad (i_1, j) \in R, (i_2, j) \in R,$$

$$i_1 < i_2, 0 < j < W, \quad (3.8d)$$

$$v_{ij_1j_2} \leq \sum_{\substack{0 \leq j'_1, j''_1 \leq j_1; \\ j_2 \leq j'_2, j''_2 \leq W; \\ 0 \leq i_1 < i < i_2 \leq L}} \sum_{\substack{0 \leq i'_1, i''_1 \leq i_1; \\ i_2 \leq i'_2, i''_2 \leq L}} h_{i'_1 i'_2 j_1} h_{i''_1 i''_2 j_2} v_{i_1 j'_1 j'_2} v_{i_2 j''_1 j''_2}, \quad (i, j_1) \in R, (i, j_2) \in R,$$

$$0 < i < L, j_1 < j_2, \quad (3.8e)$$

$$\sum_{(i', i'', j') \in S_{kij}^h} h_{i' i'' j'} + \sum_{(i', j', j'') \in S_{kij}^v} v_{i' j' j''} \leq (1 - x_{kij}) M_{kij}, \quad k \in K, (i, j) \in R_k, \quad (3.8f)$$

$$x_{kij} \leq \sum_{\substack{0 \leq i' \leq i; \\ i + l_k \leq i'' \leq L}} h_{i' i'' j}, \quad k \in K, (i, j) \in R_k, \quad (3.8g)$$

$$x_{kij} \leq \sum_{\substack{0 \leq j' \leq j; \\ j + w_k \leq j'' \leq W}} v_{i j' j''}, \quad k \in K, (i, j) \in R_k. \quad (3.8h)$$

$$x_{kij} \in \{0, 1\}, \quad k \in K, (i, j) \in R_k, \quad (3.8i)$$

$$h_{ii'j} \in \{0, 1\}, \quad (i, j) \in R, (i', j) \in R, i < i', \quad (3.8j)$$

$$v_{ijj'} \in \{0, 1\}, \quad (i, j) \in R, (i, j') \in R, j < j'. \quad (3.8k)$$

A função objetivo (3.8a) consiste em maximizar o valor total dos itens alocados. As restrições (3.8b) evitam a sobreposição entre pares de itens alocados. As restrições (3.8c) limitam o padrão de corte ao caso restrito. Observa-se que o modelo $\{\mathbf{Max} (3.8a), \mathbf{s.a} (3.8b), (3.8c), (3.8i)\}$ é exatamente a formulação de PLI proposta por Beasley (1985b) para o caso não-guilhotinado restrito. Assim, as demais variáveis de corte e restrições são incluídas para satisfazer a restrição de guilhotina.

O Modelo (3.8) é não-linear devido as restrições (3.8d) e (3.8e), que apresentam somas de produtos de quatro variáveis binárias em seus respectivos *right-hand-side* (rhs). Essas restrições criam um quadro guilhotinável no objeto. O corte de guilhotina requer a existência de um *retângulo ativo* no objeto, sobre o qual tal corte de borda-a-borda pode ser realizado. Assim, um corte horizontal ou vertical pode existir somente se houver um retângulo ativo que suporte as dimensões de tal corte. A Figura 3.3 ilustra essas restrições considerando um retângulo (i_1, i_2, j_1, j_2) que está ativo porque suas bordas estão cortadas e, assim, permite cortes guilhotinados dentro dele. Assim, se o rhs de uma restrição (3.8d) é pelo menos um, a variável $h_{ii'j}$ correspondente pode assumir valor zero ou um; mas se o rhs dessa restrição é zero, então $h_{ii'j} = 0$. A mesma análise é válida para as restrições (3.8e).

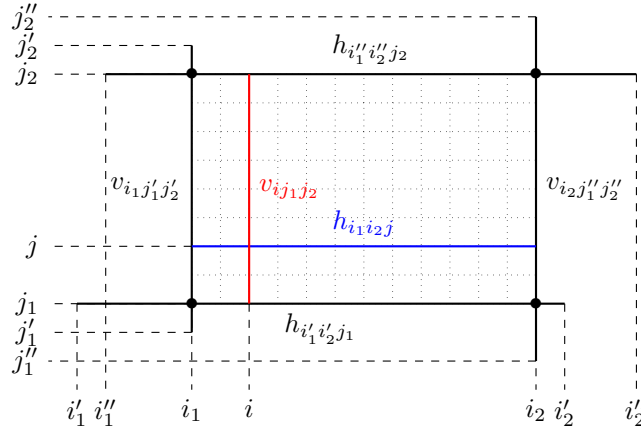


Figura 3.3: Ilustração das restrições (3.8d) e (3.8e).

As restrições disjuntivas (3.8f) proíbem cortes sobre itens alocados (isto é, quando $x_{kij} = 1$), com M_{kij} é um parâmetro suficientemente grande que pode ser definido como a soma da cardinalidade dos conjuntos S_{kij}^h e S_{kij}^v , para cada $k \in K$, $(i, j) \in R_k$. A Figura 3.4(a) ilustra essas restrições.

As restrições (3.8g) e (3.8h) limitam os itens alocados aos cantos (vértices) formados por cortes horizontais e verticais, como ilustrado na Figura 3.4(b). Essas restrições, conjuntamente com as demais, forçam a geração do quadro de guilhotina, e assim padrões não-guilhotinados são eliminados do espaço de solução. As restrições (3.8i) até (3.8k) tratam do domínio das variáveis de decisão.

Nos experimentos computacionais preliminares foi considerado um efetivo *software* de Programação por Restrições para resolver o Modelo (3.8) até a otimalidade, isto é, o IBM CPLEX Optimization Studio v.12.8 (CPOptimizer). Apesar de encontrar soluções para instâncias de *benchmark* da literatura consideradas nos experimentos da Seção 3.4, ele dificilmente encontrava as soluções ótimas e/ou dificilmente certificava a otimalidade das soluções. Note que os números de variáveis e de restrições desse modelo crescem rapidamente com as dimensões do objeto. Como pesquisas futuras, indica-se explorar o uso de outros pacotes de *software* de PNLI, como ANTIGONE (MISENER; FLOUDAS, 2013; MISENER; FLOUDAS, 2014), Baron (TAWARMALANI; SAHINIDIS, 2005) ou SCIP (VIGERSKE et al., 2012), como *solvers* de propósito geral para este modelo. A experiência com o CPOptimizer motivou o desenvolvimento de uma versão linear ao Modelo (3.8). Como primeira tentativa, cada possível retângulo do rhs das restrições (3.8d) e (3.8e) foram representados como variáveis binárias, que foram introduzidas no modelo e relacionadas aos cortes gerados, mas o número de variáveis foi muito grande e inviabilizou a abordagem pelo estouro de memória. Adicionalmente, diferentes tipos de relaxações sobre essas restrições foram consideradas, mas elas conduziam ao caso não-guilhotinado. Por fim, uma formulação de PLI foi obtida, que é descrita na próxima seção, e parece ser uma alternativa mais promissora.

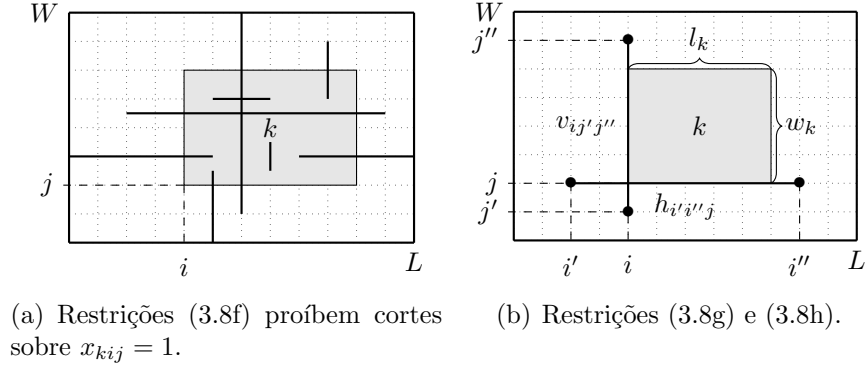


Figura 3.4: Ilustrações das restrições (3.8f), (3.8g) e (3.8h).

3.1.2 Formulação linear equivalente baseada na discretização do objeto

Para obter uma formulação de PLI equivalente ao Modelo (3.8), criou-se um novo conjunto de variáveis que representam os retângulos ativos e inativos (conforme explicado a seguir), que garantem os cortes guilhotinados. Para cada $(i_1, j_1), (i_2, j_2) \in R$, com $i_1 < i_2$ e $j_1 < j_2$, a variável binária $p_{i_1 i_2 j_1 j_2}$ representa a (sub-)área do objeto original, como definido na Expressão (3.9), que pode assumir valor unitário se suas bordas são cortadas.

$$p_{i_1 i_2 j_1 j_2} = \begin{cases} 1, & \text{se o retângulo de canto} \\ & \text{inferior-esquerdo } (i_1, j_1) \\ & \text{e superior-direito } (i_2, j_2) \\ & \text{está ativo,} \\ 0, & \text{caso contrário.} \end{cases} \quad (i_1, j_1) \in R, (i_2, j_2) \in R, i_1 < i_2, j_1 < j_2. \quad (3.9)$$

As restrições não-lineares (3.8d) e (3.8e) do Modelo (3.8) podem ser substituídas, sem perda de generalidade, pelas restrições lineares (3.10).

$$p_{i_1 i_2 j_1 j_2} \leq \sum_{i'_1 \leq i_1, i_2 \leq i'_2} h_{i'_1 i'_2 j_1}, \quad (i_1, j_1) \in R, (i_2, j_2) \in R, i_1 < i_2 \text{ e } j_1 < j_2, \quad (3.10a)$$

$$p_{i_1 i_2 j_1 j_2} \leq \sum_{i'_1 \leq i_1, i_2 \leq i'_2} h_{i'_1 i'_2 j_2}, \quad (i_1, j_1) \in R, (i_2, j_2) \in R, i_1 < i_2 \text{ e } j_1 < j_2, \quad (3.10b)$$

$$p_{i_1 i_2 j_1 j_2} \leq \sum_{j'_1 \leq j_1, j_2 \leq j'_2} v_{i_1 j'_1 j'_2}, \quad (i_1, j_1) \in R, (i_2, j_2) \in R, i_1 < i_2 \text{ e } j_1 < j_2, \quad (3.10c)$$

$$p_{i_1 i_2 j_1 j_2} \leq \sum_{j'_1 \leq j_1, j_2 \leq j'_2} v_{i_2 j'_1 j'_2}, \quad (i_1, j_1) \in R, (i_2, j_2) \in R, i_1 < i_2 \text{ e } j_1 < j_2, \quad (3.10d)$$

$$h_{i_1 i_2 j} \leq \sum_{j_1 < j_2 \leq \lfloor (j_1 + j_2)/2 \rfloor} p_{i_1 i_2 j_1 j_2}, \quad (i_1, j) \in R, (i_2, j) \in R, i_1 < i_2, \quad (3.10e)$$

$$v_{i j_1 j_2} \leq \sum_{i_1 < i_2 \leq \lfloor (i_1 + i_2)/2 \rfloor} p_{i_1 i_2 j_1 j_2}, \quad (i, j_1) \in R, (i, j_2) \in R, j_1 < j_2, \quad (3.10f)$$

$$p_{i_1 i_2 j_1 j_2} \in \{0, 1\}, \quad (i_1, j_1) \in R, (i_2, j_2) \in R, i_1 < i_2, j_1 < j_2. \quad (3.10g)$$

As restrições (3.10a) até (3.10d) impõem que a variável $p_{i_1 i_2 j_1 j_2}$ pode assumir o valor de 1 se o rhs de cada uma das quatro restrições correspondentes é pelo menos 1, isto é, se as bordas do retângulo, (i_1, i_2, j_1, j_2) estão cortadas; caso contrário, $p_{i_1 i_2 j_1 j_2}$ assume o valor de zero.

As restrições (3.10e) e (3.10f) permitem cortes guilhotinados apenas sobre retângulos ativos. Elas permitem que uma variável associada a um corte horizontal ou vertical assumam o valor de 1 apenas se a correspondente restrição tem um rhs maior ou igual a 1. Note que as restrições também impõem que os cortes horizontais e verticais são realizados até a metade do retângulo ativo, com $\lfloor x \rfloor$ é o maior inteiro menor ou igual a x . Essa consideração é sem perda de generalidade dado que o objeto é homogêneo (isto é, qualquer item pode ser obtido de qualquer região do objeto), e contribui para evitar alguns padrões de corte simétricos no espaço de soluções (CHRISTOFIDES; WHITLOCK, 1977) – a mesma consideração pode ser feita nos cortes guilhotinados do Modelo (3.8). As restrições (3.10g) tratam do domínio das variáveis de decisão. Considerando todas essas restrições, tem-se o Modelo (3.11), que é uma formulação pseudopolinomial de PLI para o PCBGR.

$$\mathbf{Max} \quad (3.8a), \quad (3.11a)$$

$$\mathbf{s.a} \quad (3.8b) - (3.8c), (3.8f) - (3.8k), (3.10a) - (3.10g). \quad (3.11b)$$

Similarmente ao Modelo não-linear (3.8), as quatro bordas do objeto original estão cortadas no Modelo (3.11). Assim, as variáveis correspondentes são fixadas ao valor unitário. Também a variável p_{LOW} pode ser fixada no valor unitário, porque o retângulo original (objeto) está ativo. Nota-se que desigualdades válidas poderiam ser desenvolvidas para associar as variáveis de decisão x_{kij} com $p_{i_1 i_2 j_1 j_2}$ e, assim, melhorar a relaxação linear do modelo. No entanto, dado que o número de restrições do modelo é significativamente alto (como destacado na Seção 3.4 de experimentos computacionais), optou-se por considerar essa relação implicitamente por meio do algoritmo proposto na próxima seção.

3.1.3 Procedimento enumerativo de variáveis

A literatura dos PCE tipicamente recorre a discretização dos conjuntos normais em vez de discretização completa — como nas Seções 3.1.1 e 3.1.2 — para decisões

relacionadas a alocações e cortes guilhotinados (CHRISTOFIDES; WHITLOCK, 1977; SCHEITHAUER, 2018). Para evitar padrões simétricos, essa técnica considera apenas as combinações lineares inteiras positivas representando combinações dos tipos de itens demandados, e define padrões de corte usando itens alocados ao canto inferior-esquerdo apenas. Para o PCBGR, o domínio das variáveis x_{kij} pode ser modificado, sem perda de generalidade segundo a literatura, de acordo com a redefinição do conjunto R , conforme a Equação (3.12).

$$R = \{(i, j) \in \mathbb{Z}^2 \mid i \in \bar{X}, j \in \bar{Y}\}, \quad (3.12)$$

sendo que:

$$\bar{X} = \left\{ q_x \mid q_x = \sum_{k \in K} n_k l_k, 0 \leq q_x \leq L - \min_{k \in K} \{l_k\}, n_k \in \mathbb{N}, n_k \leq u_k \right\} \cup \{L\} \text{ e}$$

$$\bar{Y} = \left\{ q_y \mid q_y = \sum_{k \in K} n_k w_k, 0 \leq q_y \leq W - \min_{k \in K} \{w_k\}, n_k \in \mathbb{N}, n_k \leq u_k \right\} \cup \{W\}$$

representam a discretização dos conjuntos normais para cortes verticais e horizontais, respectivamente.

O Procedimento Enumerativo de Variáveis (PEV) proposto foca em reduzir não apenas o número das variáveis x_{kij} , mas também os números de variáveis $h_{ii'j}$, $v_{ijj'}$ e $p_{i_1 i_2 j_1 j_2}$, no Modelo (3.11). Para tanto, sem perda de generalidade, o procedimento se baseia na discretização dos conjuntos normais, e algumas propriedades do problema. Como saídas, o PEV determina que os conjuntos \mathcal{X} , \mathcal{H} , \mathcal{V} e \mathcal{P} correspondam aos conjuntos reduzidos dos índices das variáveis x_{kij} , $h_{ii'j}$, $v_{ijj'}$ e $p_{i_1 i_2 j_1 j_2}$, respectivamente. O procedimento é definido de acordo com as seguintes considerações

1. o corte horizontal (i_1, i_2, j) deve pertencer a \mathcal{H} , de acordo com o retângulo (i_1, i_2, j_1, j_2) , se cada uma das seguintes condições são satisfeitas:
 - o corte é realizado até a metade do retângulo ($j_1 < j \leq \lfloor (j_1 + j_2)/2 \rfloor$);
 - no sub-retângulo inferior (i_1, i_2, j_1, j) cabe ao menos um item;
 - no sub-retângulo superior (i_1, i_2, j, j_2) cabe ao menos um item;
 - o valor $j - j_1$ pertence a \bar{Y} , isto é, cada corte horizontal gerará sempre um sub-retângulo inferior (ou subpadrão) com largura igual a uma combinação linear das larguras dos tipos de itens; e,
 - o valor $j_2 - j$ pertence a \bar{Y} ou $j_2 = W$, isto é, cada corte horizontal gerará sempre: (i) um sub-retângulo superior (ou subpadrão) com largura igual a

combinação linear das larguras dos tipos de itens, ou (ii) permitirá a existência de padrões que conduzam a cortes de aparas (do inglês, *trim cuts*);

2. o corte vertical (i, j_1, j_2) deve pertence a \mathcal{V} , de acordo com o retângulo (i_1, i_2, j_1, j_2) , se cada uma das seguintes condições são satisfeitas:

- o corte é realizado até a metade do retângulo ($i_1 < i \leq \lfloor (i_1 + i_2)/2 \rfloor$);
- no sub-retângulo esquerdo (i_1, i, j_1, j_2) cabe ao menos um item;
- no sub-retângulo direito (i, i_2, j_1, j_2) cabe ao menos um item;
- o valor $i - i_1$ pertence a \bar{X} , isto é, cada corte vertical gerará sempre um sub-retângulo esquerdo (ou subpadrão) com comprimento igual a uma combinação linear dos comprimento dos tipos de itens; e,
- o valor $i_2 - i$ pertence a \bar{X} ou $i_2 = L$, isto é, cada corte vertical gerará sempre: (i) um sub-retângulo esquerdo (ou subpadrão) com comprimento igual a combinação linear dos comprimentos dos tipos de itens, ou (ii) permitirá a existência de padrões que conduzam a cortes de aparas (do inglês, *trim cuts*);

3. o domínio (i_1, i_2, j_1, j_2) deve pertencer a \mathcal{P} se a área $(i_2 - i_1) \times (j_2 - j_1)$ é capaz de conter ao menos dois itens.

O PEV proposto é descrito no Algoritmo 3.1.1. A Figura 3.5 mostra dois exemplos ilustrativos de padrões de cortes possíveis considerados pelo PEV, e destaca que: (i) se $i_2 < L$ e $j_2 < W$, a formulação apenas precisa considerar retângulos $p_{i_1 i_2 j_1 j_2}$ com comprimento $(i_2 - i_1)$ e largura $(j_2 - j_1)$ pertencentes a \bar{X} e \bar{Y} , respectivamente; e (ii) se $i_2 = L$ ou $j_2 = W$, permite-se situações que conduzam a cortes de aparas nos retângulos superiores e direitos.

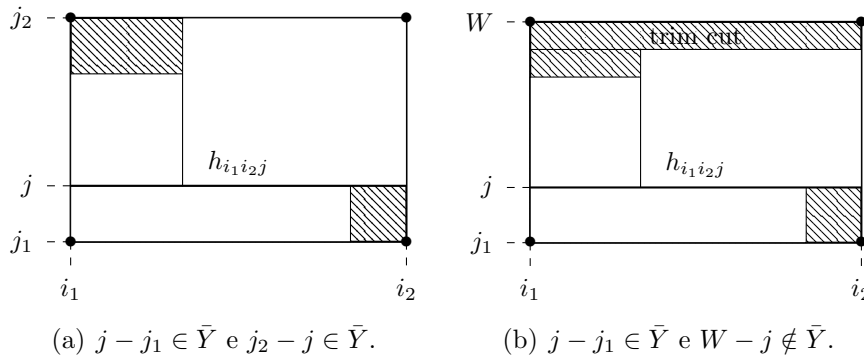


Figura 3.5: Ilustrações de possíveis padrões de cortes considerados pelo PEV.

Destaca-se que o Modelo (3.11) pode ser declarado sem perda de generalidade usando os domínios \mathcal{X} , \mathcal{H} , \mathcal{V} e \mathcal{P} para definir todas as suas variáveis e restrições. As melhorias discutidas nessa seção são utilizadas nos experimentos computacionais da Seção 3.4.

Algoritmo 3.1.1: Procedimento Enumerativo de Variáveis para um objeto.

Entrada: $L, W, K = \{1, \dots, m\}$, (l_k, w_k, v_k) para todo $k \in K$, e os conjuntos de discretização \bar{X} e \bar{Y} .

1 $\mathcal{X} = \emptyset$, $\mathcal{H} = \{(0, L, 0), (0, L, W)\}$, $\mathcal{V} = \{(0, 0, W), (L, 0, W)\}$, $\mathcal{P} = \{(0, L, 0, W)\}$,
auxP = $\{(0, L, 0, W)\}$

2 **para** $k \in K$ **faça**

3 **para** $i \in \bar{X}$ **faça**

4 **para** $j \in \bar{Y}$ **faça**

5 **se** $\{(i + l_k \leq L) \text{ e } (j + w_k \leq W)\}$ **então**

6 $\mathcal{X} = \mathcal{X} \cup (k, i, j)$

7 **repita**

8 selecione um retângulo $(i_1, i_2, j_1, j_2) \in \text{auxP}$ e remova-o de *auxP*

 // cortes horizontais

9 **para** $j \in \bar{Y} \mid j_1 < j \leq \lfloor (j_1 + j_2)/2 \rfloor$ **faça**

10 **se** em cada um dos retângulos (i_1, i_2, j_1, j) e (i_1, i_2, j, j_2) cabe ao menos um item, e $j - j_1 \in \bar{Y}$ e $(j_2 - j \in \bar{Y} \text{ ou } j_2 = W)$ **então**

11 $\mathcal{H} = \mathcal{H} \cup (i_1, i_2, j)$

12 **se** retângulo $(i_1, i_2, j_1, j) \notin \mathcal{P}$ e nele cabe ao menos dois itens **então**

13 $\mathcal{P} = \mathcal{P} \cup (i_1, i_2, j_1, j)$, *auxP* = *auxP* $\cup (i_1, i_2, j_1, j)$

14 **se** retângulo $(i_1, i_2, j, j_2) \notin \mathcal{P}$ e nele cabe ao menos dois itens **então**

15 $\mathcal{P} = \mathcal{P} \cup (i_1, i_2, j, j_2)$, *auxP* = *auxP* $\cup (i_1, i_2, j, j_2)$

 // cortes verticais

16 **para** $i \in \bar{X} \mid i_1 < i \leq \lfloor (i_1 + i_2)/2 \rfloor$ **faça**

17 **se** em cada um dos retângulos (i_1, i, j_1, j_2) e (i, i_2, j_1, j_2) cabe ao menos um item, e $i - i_1 \in \bar{X}$ e $(i_2 - i \in \bar{X} \text{ ou } i_2 = L)$ **então**

18 $\mathcal{V} = \mathcal{V} \cup (i, j_1, j_2)$

19 **se** retângulo $(i_1, i, j_1, j_2) \notin \mathcal{P}$ e nele cabe ao menos dois itens **então**

20 $\mathcal{P} = \mathcal{P} \cup (i_1, i, j_1, j_2)$, *auxP* = *auxP* $\cup (i_1, i, j_1, j_2)$

21 **se** retângulo $(i, i_2, j_1, j_2) \notin \mathcal{P}$ e nele cabe ao menos dois itens **então**

22 $\mathcal{P} = \mathcal{P} \cup (i, i_2, j_1, j_2)$, *auxP* = *auxP* $\cup (i, i_2, j_1, j_2)$

23 **até** *auxP* estiver vazio;

Saída: Domínios \mathcal{X} , \mathcal{H} , \mathcal{V} e \mathcal{P} .

3.2 Formulações matemáticas para padrões 2-estágios e 1-grupo

Esta seção propõe formulações de PLI para duas variantes do PCBGR, que estritamente consideram casos especiais de padrões guilhotinados, conhecidos como padrões 2-estágios e 1-grupo. Essas formulações são obtidas do Modelo (3.11) ao remover o conjunto de variáveis $p_{i_1 i_2 j_1 j_2}$, que se tornam desnecessárias, e modificar as variáveis de cortes, que contribuem na simplificação de alguns conjuntos de restrições.

3.2.1 Formulações para padrões 2-estágios

Os padrões guilhotinados 2-estágios são formados por tiras horizontais ou verticais no objeto, a partir das quais os itens são cortados. Considerando o caso de tiras horizontais, a variável de corte horizontal $h_{i'j}$ pode ser redefinida, conforme (3.13), porque nesse contexto os cortes horizontais são sempre de borda-a-borda do objeto original.

$$h_j = \begin{cases} 1, & \text{se um corte horizontal é realizado de } (0, j) \text{ até } (L, j), \\ 0, & \text{caso contrário,} \end{cases} \quad 0 \leq j < W. \quad (3.13)$$

Um item alocado não pode cruzar um corte horizontal. Assim, como os itens alocados dentro de uma tira não devem se sobrepor entre si, os cortes verticais não são mais necessários. Tem-se no Modelo (3.14) é uma formulação pseudopolinomial de PLI com padrões 2-estágios não-exatos, obtida a partir do Modelo (3.11). Os conjuntos de variáveis $v_{ijj'}$ e $p_{i_1 i_2 j_1 j_2}$ foram eliminados, e as variáveis x_{kij} são mantidas conforme definido na Seção 3.1.

Max (3.8a),

s.a

(3.8b), (3.8c), (3.8i),

$$\sum_{j < j' < j + w_k} h_{j'} \leq (1 - x_{kij}) M_{kij}, \quad k \in K, (i, j) \in R_k, \quad (3.14a)$$

$$x_{kij} \leq h_j, \quad k \in K, (i, j) \in R_k, \quad (3.14b)$$

$$h_j \in \{0, 1\}, \quad 0 \leq j < W. \quad (3.14c)$$

As restrições (3.14a) garantem que nenhum corte horizontal cruze um item alocado, sendo que o parâmetro M_{kij} pode ser definido como $w_k - 1$. As restrições (3.14b) limitam as alocações sobre cortes horizontais. As restrições (3.14c) tratam do domínio da variável h_j . As demais restrições estão conforme definidas previamente. Similarmente aos modelos da Seção 3.1, a borda inferior do objeto é considerada cortada (isto é, $h_0 = 1$).

Para um padrão 2-estágios formado por tiras verticais, pode-se realizar uma

análise semelhante, mas convertendo as variáveis $v_{i'j}$ em v_i , e as variáveis $h_{ijj'}$ permanecem como na formulação original. Outra possibilidade é trocar os comprimentos e as larguras de cada tipo de item e objeto original, antes de utilizar o Modelo (3.15) para resolver a instância em questão.

O Modelo (3.14) pode ser reformulado para usar um número reduzido de variáveis e restrições, o que pode contribuir para seu desempenho médio em *solvers* de propósito geral. Após essa redução, a formulação resultante é dada pelo Modelo (3.15).

Max (3.8a),

s.a

(3.8b), (3.8c), (3.8i), (3.14c),

$$\sum_{k \in K} \sum_{0 \leq i \leq L - l_k} \sum_{j' < j < j' + w_k} x_{kij'} \leq (1 - h_j)M_j^1, \quad 0 < j < W, \quad (3.15a)$$

$$\sum_{k \in K} \sum_{0 \leq i \leq L - l_k} x_{kij} \leq M_j^2 h_j, \quad 0 < j < W, \quad (3.15b)$$

As restrições (3.15a) garantem que nenhum item alocado cruze um corte horizontal. Elas geram o mesmo comportamento que as restrições (3.14a) do Modelo (3.14), que garantem que nenhum corte horizontal cruze um item alocado, no entanto com menor número de restrições. O parâmetro M_j^1 pode ser definido como $\sum_{k \in K} (L - l_k + 1) * (w_k - 1)$, que é um limitante grande suficiente para o *left-hand-side* (lhs) da restrição. A Figura 3.6(a) ilustra casos proibidos por essas restrições. As restrições (3.15b) substituem as restrições (3.14b) do Modelo (3.14) e limitam que as alocações ocorram sobre um corte horizontal, sendo que o parâmetro M_j^2 pode ser definido como $\sum_{k \in K} (L - l_k + 1)$.

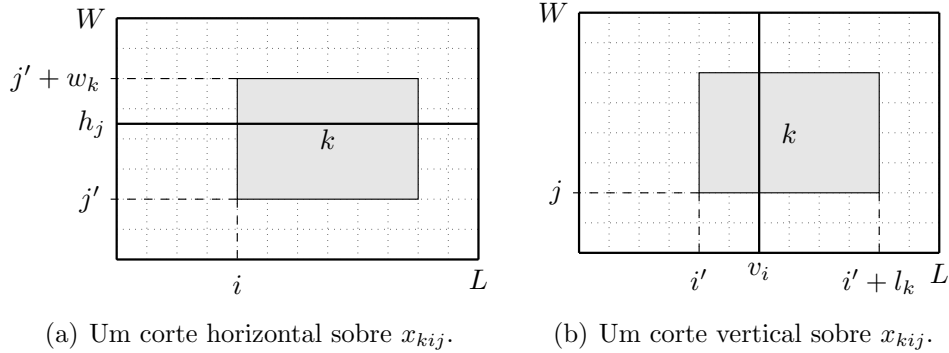


Figura 3.6: Ilustrações de cortes guilhotinados sobre um item alocado para padrões 2-estágios.

3.2.2 Formulações para padrões 1-grupo

Um padrão de corte 1-grupo é formado apenas por tiras horizontais e verticais no objeto. Assim, em adição as tiras horizontais h_j , as variáveis $v_{i'j}$ podem ser redefinidas conforme a Equação (3.16), porque todos os cortes verticais são de borda-a-borda no objeto original.

$$v_i = \begin{cases} 1, & \text{se um corte vertical é realizado } (i, 0) \text{ até } (i, W), \\ 0, & \text{caso contrário,} \end{cases} \quad 0 \leq i < L. \quad (3.16)$$

Tem-se no Modelo (3.17) uma formulação pseudopolinomial de PLI para o problema PCBGR que considera apenas padrões 1-grupo não-exatos, obtida a partir do modelo definido na Seção 3.1.2. As variáveis x_{kij} permanecem conforme definição na Seção 3.1, enquanto as variáveis $p_{i_1 i_2 j_1 j_2}$ não são mais necessárias, porque todos os cortes 1-grupo conduzem a padrões guilhotinados.

Max (3.8a),

s.a

(3.8b), (3.8c), (3.8i), (3.14b), (3.14c),

$$\sum_{j < j' < j + w_k} h_{j'} + \sum_{i < i' < i + l_k} v_{i'} \leq (1 - x_{kij}) M_{kij}, \quad k \in K, (i, j) \in R_k, \quad (3.17a)$$

$$x_{kij} \leq v_i, \quad k \in K, (i, j) \in R_k. \quad (3.17b)$$

$$v_i \in \{0, 1\}, \quad 0 \leq i < L. \quad (3.17c)$$

As restrições (3.17a) evitam que qualquer corte horizontal ou vertical cruze um item alocado, sendo que M_{kij} é um parâmetro que pode ser definido como $w_k + l_k - 2$. As restrições (3.17b) proíbem alocações sobre um corte vertical. As restrições (3.17c) tratam o domínio das variáveis. As demais restrições são como previamente definidas. Similarmente aos modelos da Seção 3.1, a borda esquerda do objeto é considerada cortada (isto é, $v_0 = 1$).

Assim como na subseção anterior, pode-se reformular o Modelo (3.17) para considerar menos variáveis e restrições, resultando no Modelo (3.18), a fim de se ter melhor desempenho médio em *solvers* de propósito geral.

Max (3.8a),

s.a

(3.8b), (3.8c), (3.8i), (3.14c), (3.15a), (3.15b), (3.17c),

$$\sum_{k \in K} \sum_{0 \leq j \leq W - w_k} \sum_{i' < i < i' + l_k} x_{ki'j} \leq (1 - v_i) N_i^1, \quad 0 < i < L, \quad (3.18a)$$

$$\sum_{k \in K} \sum_{0 \leq j \leq W - w_k} x_{kij} \leq N_i^2 v_i, \quad 0 < i < L, \quad (3.18b)$$

As restrições (3.18a) substituem as restrições (3.17a) do Modelo (3.17) e garantem que nenhum item alocado cruze um corte vertical, sendo que N_i^1 é um parâmetro que pode ser definido como $\sum_{k \in K} (W - w_k + 1) * (l_k - 1)$, que é um limitante grande suficiente para o lhs das restrições. A Figura 3.6(b) ilustra como essas restrições agem no modelo. As restrições (3.18b) substituem (3.17b) e impõem que as alocações devem ocorrer sobre um corte vertical, sendo que o parâmetro N_i^2 pode ser definido como $\sum_{k \in K} (W - w_k + 1)$.

Dado que o conjunto das variáveis $p_{i_1 i_2 j_1 j_2}$ não está mais presente em qualquer modelo dessa seção, as linhas 7 até 23 do Algoritmo 3.1.1 não são mais necessárias. Assim, as variáveis de corte h_j e v_i devem considerar apenas as posições dos conjuntos \bar{X} e \bar{Y} , respectivamente. Ressalta-se que os Modelos (3.14), (3.15), (3.17) e (3.18) podem ser adaptados facilmente para abordar o caso exato. Para tanto, é necessário adicionalmente impor que um tipo de item k pode ser alocado à posição (i, j) , se e somente se $h_{j'}$ e $v_{i'}$ são cortes realizados, para $j' = j + w_k$ e $i' = i + l_k$.

3.3 Algoritmo *branch-and-Benders-cut* para o PCBGR

Como método de solução ao PCBGR, um algoritmo *branch-and-Benders-cut* (B&BC) baseado no Modelo (3.11) é proposto. Ele consiste em utilizar a decomposição de Benders (BENDERS, 1962) para reformular o Modelo (3.11), e então utiliza-se um algoritmo *branch-and-cut* para resolver o Problema-Mestre-de-Benders resultante (PMB). A geração dos cortes de Benders são adicionados ao algoritmo *branch-and-cut* e, assim, podem ser realizados em cada nó da árvore. Para descrições detalhadas e *surveys* sobre decomposição de Benders e o algoritmo B&BC, o leitor é direcionado aos trabalhos de Costa (2005), Rahmaniani et al. (2017) e Moreno, Munari e Alem (2019).

O método proposto considera a decomposição de Benders para reformular o Modelo (3.11) como segue:

1. O PMB é definido pelo conjunto de restrições de não-sobreposição de itens e do caso restrito, conforme apresentado no Modelo (3.19). Assim, o PMB busca padrões de corte promissores (soluções de teste, do inglês *trial solutions*), representadas pelo vetor de solução \bar{x}_{kij} . Note que o PMB é exatamente a formulação de PLI proposta por Beasley (1985b) para o caso não-guilhotinado restrito — veja a Seção 3.1.1.

$$\mathbf{Max} \quad (3.8a), \quad \mathbf{s.a} \quad (3.8b), (3.8c), (3.8i). \quad (3.19)$$

2. Cada solução de teste obtida pelo PMB é verificada considerando a restrição de guilhotina por meio do Subproblema-de-Benders (SPB), que é definido pelo Modelo

(3.20). Ele consiste nas variáveis $h_{i'j}$, $v_{ijj'}$ e $p_{i_1i_2j_1j_2}$ e restrições do Modelo (3.11) que não estão incluídas no PMB.

$$\mathbf{Max} \ 0, \tag{3.20a}$$

$$\mathbf{s.a} \ (3.8j), (3.8k), (3.10), \tag{3.20b}$$

$$\sum_{(i'j'j'') \in S_{kij}^h} h_{i'i'j'} + \sum_{(i'j'j'') \in S_{kij}^v} v_{i'j'j''} \leq 0, \quad (k, i, j) \in \bar{\mathcal{X}}, \tag{3.20c}$$

$$1 \leq \sum_{i' \leq i; i+l_k \leq i''} h_{i'i''j}, \quad (k, i, j) \in \bar{\mathcal{X}}, \tag{3.20d}$$

$$1 \leq \sum_{j' \leq j; j+w_k \leq j''} v_{ij'j''}, \quad (k, i, j) \in \bar{\mathcal{X}}. \tag{3.20e}$$

As restrições (3.20c)–(3.20e) são adaptadas das restrições (3.8f)–(3.8h) do Modelo (3.11) para considerar apenas os componentes unitários da solução de teste \bar{x}_{kij} . Assim, elas são definidas considerando $\bar{\mathcal{X}} = \{(k, i, j) \mid \bar{x}_{kij} = 1\}$, que é o conjunto de triplas correspondentes aos itens alocados dessa solução. Assim como na formulação original, essas restrições proíbem cortes sobre os itens alocados, e garantem que esses itens são alocados aos cantos cortados.

O algoritmo B&BC é como segue. Em dado nó da árvore *branch-and-cut*, cada solução de teste \bar{x}_{kij} obtida pelo PMB é verificada pelo SPB. Se o SPB é factível para essa solução, então o padrão de corte correspondente é guilhotinado; assim, a solução é aceita, e armazenada como melhor solução incumbente, se ela melhora o limitante inferior da árvore. Caso contrário, o padrão de corte é não-guilhotinado, e a solução é utilizada para gerar um corte combinatório de Benders, conforme a desigualdade (3.21). Essa desigualdade é adicionada ao PMB do nó corrente, que é então re-otimizado.

$$\sum_{(k,i,j) \in \bar{\mathcal{X}}} x_{kij} \leq |\bar{\mathcal{X}}| - 1. \tag{3.21}$$

Como em implementações ordinárias do algoritmo B&BC, os cortes combinatórios de Benders são adicionados como *lazy constraints* no método proposto, usando procedimentos do tipo *callback* dos *solvers* de propósito geral. Assim, o PMB é resolvido uma única vez, dentro de única árvore *branch-and-cut*.

Nota-se que os conjuntos de variáveis $h_{i'j}$, $v_{ijj'}$ e $p_{i_1i_2j_1j_2}$ podem ser significativamente reduzidos no Modelo (3.20), dado que a solução de teste \bar{x}_{kij} é (momentaneamente) fixada durante análise no SPB. Em tal caso, o domínio dessas variáveis podem ser restringidos, em adição as posições zero e do tamanho do objeto, para: (i) posições horizontais i (resp. posições verticais j) das triplas em $\bar{\mathcal{X}}$ (isto é, os itens alocados); e (ii) segmentos horizontais $i' - i$ ou $i_2 - i_1$ (resp. segmentos verticais $j' - j$ ou $j_2 - j_1$) que são maiores ou iguais ao comprimento (resp. largura) do menor item alocado em $\bar{\mathcal{X}}$. Essa

redução é considerada nos experimentos computacionais descritos neste capítulo. Assim, as linhas 7 até 23 do Algoritmo (3.1.1) não são necessárias na definição do SPB considerado no algoritmo B&BC proposto. Para pesquisas futuras, essa decomposição poderia ser analisada adicionalmente no contexto dos modelos de padrões 2-estágios e 1-grupo da Seção 3.2.

Por fim, destaca-se que o algoritmo de Benders proposto é uma decomposição primal do Modelo (3.11). De forma sucinta, um conjunto de variáveis e restrições foi mantido no PMB (restrições de não sobreposição de pares de itens e de caso restrito), enquanto outro conjunto de variáveis e restrições foi levado ao SBP (restrições do quadro de guilhotina, para evitar cortes sobre itens alocados, e de itens alocados em cantos cortados). Essa tese também estudou um algoritmo de Benders seguindo uma decomposição dual do Modelo(3.11) para o SPB. No entanto, nessa segunda estratégia assumia-se, sem prova matemática, que as variáveis binárias $h_{ijj'}$, $v_{i'j}$ e $p_{i_1i_2j_1j_2}$ podiam ser relaxadas no intervalo 0-1, e por isso foi descontinuada. Trata-se de uma oportunidade para pesquisas futuras.

3.4 Experimentos computacionais

Experimentos computacionais foram realizados usando instâncias de *benchmark* da literatura, para avaliar o desempenho das formulações de PLI e o algoritmo B&BC propostos neste capítulo. O objetivo dessa seção é duplo. Primeiro, para o PCBGR, são comparados a qualidade de solução e tempos de processamento da formulação de PLI proposta – Modelo (3.11) – com aquelas obtidas pelas outras duas formulações da literatura (BEN MESSAOUD; CHU; ESPINOUSE, 2008; FURINI; MALAGUTI; THOMOPULOS, 2016), todas elas resolvidas por um *solver* de propósito geral, descrito a seguir. Adicionalmente, é verificado o desempenho do algoritmo B&BC proposto com respeito a essas formulações, usando o mesmo conjunto de instâncias. A partir desses resultados, foram identificados cenários mais adequados para cada uma das abordagens, sem o objetivo de ser um estudo sistemático. Segundo, são analisados os desempenhos das formulações de PLI propostas para casos especiais do PCBGR que consideram padrões 2-estágios e 1-grupo – Modelos (3.15) e (3.18). Seus desempenhos são comparados aos modelos propostos por Lodi, Martello e Monaci (2002) e Yanasse e Morabito (2006), para as mesmas variantes.

A implementação do Modelo (3.11) considera as melhorias do Algoritmo 3.1.1 e é denotado como Grelha. Os Modelos (3.15) e (3.18) foram implementados considerando as melhorias de discretização discutidas na Seção 3.2.2 e são denotadas como Grelha-2-estágios e Grelha-1-grupo, respectivamente. Todas as abordagens e de *benchmark* foram implementadas em C++ utilizando a biblioteca do Concert, que é parte do IBM CPLEX Optimization Studio v.12.8. Os experimentos foram realizados em um computador com processador Intel Core i7-3770 (3.40GHz), 16 GB de RAM, sob Sistema

Operacional Ubuntu 18.04.

Na Tabela 3.1 são mostradas as características das instâncias de *benchmark* tomadas da literatura. Os conjuntos de instâncias *cgcut* e *gcut* foram obtidos do repositório *online* OR-Library¹, enquanto as demais instâncias foram obtidas de Wang (1983) e Oliveira e Ferreira (1990). É reportado para cada instância seu nome (coluna Instância), o comprimento do objeto (coluna L), largura do objeto (coluna W), o número de tipos de itens (coluna m) e o número máximo de itens (coluna n), sendo que $n = \sum_{k \in K} u_k$. Também são reportados o valor ótimo da função objetivo de cada instância com respeito as versões não-estagiada, 2-estágios e 1-grupo do PCBGR. Os valores ótimos nas colunas não-estagiada e 2-estágios são resultados conhecidos e foram tomados de Lodi, Martello e Monaci (2002) e Furini, Malaguti e Thomopoulos (2016); enquanto os valores na coluna 1-grupo foram obtidos ao resolver a formulação de PLI de Yanasse e Morabito (2006) pelo CPLEX, sem impor limite de tempo. É importante destacar que quando tem-se *gap* = 0.00% e o tempo limite de processamento “tl” foi alcançado, uma solução ótima foi obtida para a correspondente formulação, mas o *solver* não foi capaz de provar a otimalidade da solução no intervalo de tempo considerado; note que o valor das soluções ótimas das instâncias consideradas são conhecidas da literatura.

Tabela 3.1: Conjunto de instâncias de *benchmark* para o PCBGR.

Instância	L	W	m	n	Valor da solução ótima (OPT)		
					não-estagiada	2-estágios	1-grupo
<i>cgcut1</i>	15	10	7	16	244	240	240
<i>cgcut2</i>	40	70	10	23	2.892	2.535	2.069
<i>cgcut3</i>	40	70	19	62	1.860	1.720	1.580
<i>OF1</i>	70	40	10	23	2.737	2.713	2.361
<i>OF2</i>	70	40	10	24	2.690	2.515	2.342
<i>wang20</i>	70	40	19	42	2.721	2.623	2.470
<i>gcut1</i>	250	250	10	10	48.368	43.024	43.024
<i>gcut2</i>	250	250	20	20	59.307	57.996	55.680
<i>gcut3</i>	250	250	30	30	60.241	59.895	59.895
<i>gcut4</i>	250	250	50	50	60.942	60.504	60.504
<i>gcut5</i>	500	500	10	10	195.582	193.379	192.907
<i>gcut6</i>	500	500	20	20	236.305	224.399	224.399
<i>gcut7</i>	500	500	30	30	238.974	238.974	231.494
<i>gcut8</i>	500	500	50	50	245.758	245.758	242.656
<i>gcut9</i>	1.000	1.000	10	10	919.476	919.476	806.912
<i>gcut10</i>	1.000	1.000	20	20	903.435	856.445	856.445
<i>gcut11</i>	1.000	1.000	30	30	955.389	942.219	915.219
<i>gcut12</i>	1.000	1.000	50	50	970.744	970.744	970.744

¹<http://people.brunel.ac.uk/~mastjjb/jeb/orlib/files/>

3.4.1 Resultados para o PCBGR

O desempenho de Grelha é comparado aos desempenhos das formulações de PLI de Ben Messaoud, Chu e Espinouse (2008) e Furini, Malaguti e Thomopulos (2016). Note que a formulação de Ben Messaoud, Chu e Espinouse (2008) foi originalmente proposta para o GSPP, e foi adaptada para o PCBGR, conforme discutido na Seção 2.4.3, e é denotado aqui por BMCE. Para o modelo de Furini, Malaguti e Thomopulos (2016), foi considerado o procedimento enumerativo de variáveis desses autores, assim como as duas reduções propostas por eles. No que segue, esse modelo é referido como FMT.

Na Tabela 3.2 são mostrados os resultados dessas três formulações. Para cada formulação e instância, é reportado o número de variáveis (coluna var), o número de restrições (coluna rest), o *gap* de otimalidade (coluna gap) como uma porcentagem, o valor da relaxação linear (coluna solr) e o tempo de processamento para resolver a instância (coluna t[s]) em segundos. O *gap* é calculado como $(OPT - sol)/(OPT + 10^{-10}) * 100$, sendo que *OPT* é o valor ótimo correspondente tomado da Tabela 3.1 e *sol* é o valor da melhor solução inteira encontrada pelo *solver* usando tal formulação. O tempo limite para busca de soluções inteiras do CPLEX foi definido como 1 hora, e é denotado por “tl” quando ocorrido. O símbolo “*” significa que o CPLEX teve interrupções por falta de memória durante sua execução, e o *gap* de “100.0%” significa que nenhuma solução inteira foi encontrada. Em todas as formulações, foi fornecido a solução ótima 2-estágios para cada instância como solução inicial, que foi obtida ao resolver o modelo Lodi, Martello e Monaci (2002) pelo CPLEX. O fornecimento de solução inicial contribuiu principalmente para BMCE, dado que o *solver* é capaz de encontrar boas soluções factíveis com as formulações FMT e Grelha.

Primeiro, os resultados de BMCE para o PCBGR são analisados, dado que a formulação originalmente proposta foi testada apenas para o GSPP em Ben Messaoud, Chu e Espinouse (2008). Nota-se que foi considerado $n = \bar{n}$ ao PCBGR, em vez de $n = \sum_{k \in K} u_k$ como no GSPP, sendo que \bar{n} foi calculado como o número máximo de itens que cabem no objeto obtido ao resolver uma mochila unidimensional, cujo peso de cada item é sua área ($l_k \times w_k$), a capacidade da mochila é $L \times W$, e a contribuição de cada item é unitária. Essa consideração se mostrou útil ao modelo, visto que segundo os autores, ele tem em torno de $3n^4/4$ variáveis binárias e $2n^4$ restrições. Por exemplo, para as instâncias *gcut*, os números de variáveis e restrições desse modelo são menores do que os outros dois modelos, dado que para essas instâncias $\bar{n} \leq 9$ (o limitante superior obtido pela mochila unidimensional). No entanto, sua relaxação linear fornece limitantes muito fracos, o que contribuíram para o desempenho ruim do CPLEX. O *solver* provou otimalidade apenas em 2 das 18 instâncias considerando BMCE, obtendo limite de tempo em 15 delas.

Tabela 3.2: Resultados para as três formulações de PLI para o PCBGR.

Instância	BMCE (BEN MESSAOUD; CHU; ESPINOUSE, 2008)					FMT (FURINI; MALAGUTI; THOMOPULOS, 2016)					Grelha				
	var	rest	gap	solr	t[s]	var	rest	gap	solr	t[s]	var	rest	gap	solr	t[s]
cgcut1	23.010	52.820	0,0	346,0	tl	752	147	0,0	244,5	0,02	1.417	3.979	0,0	244,5	0,15
cgcut2	83.223	198.417	100,0	*	*	49.724	2.027	0,0	2.897,4	10,49	24.860	68.500	9,9	2.906,1	tl
cgcut3	17.809	26.746	6,5	5.240,0	tl	32.727	1.879	0,0	1.919,2	8,35	5.793	15.520	0,0	1.962,5	121,24
OF1	13.090	26.707	0,9	4.735,0	tl	34.908	2.108	0,0	2.754,6	3,73	15.051	41.229	0,9	2.789,9	tl
OF2	9.395	18.082	6,2	4.747,0	tl	31.795	2.120	0,0	2.734,1	5,64	8.746	21.938	0,0	2.747,2	124,61
wang20	15.389	26.726	3,6	10.834,0	tl	32.725	1.879	0,0	2.721,0	5,53	5.793	15.520	0,0	2.784,5	32,61
gcut1	660	1.038	0,0	112.502,0	94,05	2.814	525	0,0	50.739,5	0,28	520	1.553	0,0	50.823,5	0,09
gcut2	2.611	4.176	0,0	138.133,0	tl	626.515	28.813	0,0	60.188,5	2.429,35	9.626	26.148	0,0	61.240,4	84,59
gcut3	4.736	7.244	0,6	165.699,0	tl	1.127.094	32.711	0,0	60.638,5	2.705,92	29.591	78.053	0,0	61.731,7	1.550,54
gcut4	8.604	11.766	0,0	226.788,0	tl	2.024.833	35.226	0,0	61.640,7	tl	105.456	276.254	100,0	*	*
gcut5	1.227	2.204	0,0	414.013,0	tl	170.931	24.480	0,0	218.543,0	481,30	1.408	3.987	0,0	228.381,9	4,42
gcut6	1.587	2.214	0,0	509.976,0	tl	994.207	73.834	0,0	237.651,5	tl	4.103	12.087	0,0	242.292,2	7,36
gcut7	3.101	4.186	0,0	771.605,0	tl	2.512.109	100.730	100,0	*	*	12.094	34.089	0,0	242.976,0	67,66
gcut8	8.604	11.766	0,0	859.788,0	tl	9.809.683	136.775	100,0	*	*	145.214	379.646	100,0	*	*
gcut9	1.227	2.204	0,0	1.478.097,0	tl	311.561	36.162	0,0	937.266,0	1.243,86	1.803	4.720	0,0	947.014,0	1,30
gcut10	910	1.048	0,0	2.126.738,0	1.758,86	20.735	2.557	0,0	962.059,7	6,11	3.180	9.600	0,0	962.059,7	10,95
gcut11	4.736	7.244	0,1	2.660.197,0	tl	18.863.364	400.100	100,0	*	*	42.477	118.138	100,0	*	*
gcut12	4.081	4.206	0,0	2.999.501,0	tl	31.289.676	489.478	100,0	*	*	91.051	261.285	100,0	*	*

Furini, Malaguti e Thomopulos (2016) compararam o desempenho de seu modelo com respeito ao modelo de Ben Messaoud, Chu e Espinouse (2008), no contexto do GSPP. Os resultados da Tabela 3.2 estão de acordo para aqueles obtidos esses autores, dado que FMT também tem melhor desempenho médio que BMCE para o PCBGR. Considerando o FMT, CPLEX foi capaz de provar otimalidade em 12 das instâncias, e excedeu o limite de memória apenas para aquelas instâncias cujo número de variáveis excede dois milhões. A relaxação linear desse modelo é a mais apertada entre as três formulações, e seu desempenho nas instâncias de tamanho médio (primeiras seis linhas da tabela) é proeminente, especialmente em relação ao número de restrições e tempo de solução. No entanto, é importante destacar que FMT é um modelo pseudopolinomial (formulação não-compacta) que requer uma fase de pré-processamento, apesar de em geral ser rápida.

Os resultados de Grelha na Tabela 3.2 mostram que o modelo proposto tem melhor desempenho médio que FMT nas instâncias gcut, dado que foi capaz de provar otimalidade em duas instâncias adicionais e com tempos de processamento significativamente menores. Em particular, para as instâncias com $m = 10$ tipos de itens (gcut1, gcut5 e gcut9), o desempenho de Grelha é significativamente melhor que FMT. Para a instância gcut9, o CPLEX levou 1,30 segundos para provar otimalidade usando Grelha, enquanto ele levou 1.243,86 segundos com FMT. Para algumas instâncias com $m = 20$ tipos de itens como, por exemplo, gcut2 e gcut7, ele levou menos que 90 segundos para provar otimalidade com Grelha, enquanto CPLEX alcançou o limite de memória com FMT. Note que para as instâncias gcut, à medida que o valor de m aumenta, o tempo de processamento também aumenta. Essas instâncias são caracterizadas com itens com tamanho relativo grande em relação ao objeto, o que resulta em conjuntos \bar{X} e \bar{Y} com menos elementos e, assim, menos variáveis e restrições no modelo. Os experimentos também destacaram a relevância do PEV. Por exemplo, a instância gcut2 teria 742.201 variáveis e 2.770.181 restrições na ausência das linhas 7 até 23 do Algoritmo 3.1.1. Portanto, para essa instância, o PEV conduziu a reduções de 98.70% e 99.06% nos números de variáveis e restrições, respectivamente.

Sobre a relaxação linear, Grelha forneceu limitantes significativamente melhores do que aqueles de BMCE, mas inferiores ou similares que aqueles de FMT. Por outro lado, Grelha não parece adequado para cenários com muitos tipos de itens, dado que atingiu o limite de memória em todas as instâncias com $m = 50$ (gcut4, gcut8 e gcut12) e para a instância gcut11. Particularmente, FMT tem melhor desempenho médio que Grelha nas seis primeiras instâncias da Tabela 3.2 (conjuntos cgcut, OF e wang20), que são caracterizadas por itens que relativamente ao objeto são menores que nas instâncias gcut. Nota-se que quando o tempo limite de execução é atingido e o *gap* de otimalidade é zero, a solução ótima da instância foi obtida (ou fornecida pela solução inicial), mas o certificado de otimalidade não foi atestado.

Apesar da relevância de evitar soluções simétricas em formulações de PLI,

esse autor desconhece outros procedimentos para quebra de simetrias para os modelos de Beasley (1985b) e/ou Grelha, além daquelas adotadas nesse trabalho. Assim, apesar da restrição de cortes guilhotinados até a metade dos retângulos, um retângulo com comprimento $i_2 - i_1$ e largura $j_2 - j_1$ pode ser representado diversas vezes por variáveis $p_{i_1 i_2 j_1 j_2}$ apesar de sua posição.

Na Tabela 3.3 são mostrados os resultados obtidos pelo algoritmo B&BC proposto. Para cada instância, são reportados os tamanhos dos conjuntos \bar{X} e \bar{Y} , o número de soluções de teste encontradas pelo PMB e verificadas pelo SPB (coluna testes) e o número de cortes adicionais ao PMB (coluna cortes). Os números de variáveis e restrições dessa tabela correspondem ao PMB inicial. Em comparação com Grelha, o algoritmo B&BC foi capaz de provar otimalidade mais rápido que o *solver* com o modelo em 5 instâncias, conhecidamente cgcut1, OF2, wang20, gcut1 e gcut5. Uma dessas instâncias é caracterizada por ter seu valor ótimo da função objetivo (OPT) próximo do valor ótimo da função objetivo do problema de padrões não-guilhotinados e, assim, necessitou de poucos cortes de factibilidade para convergir para uma solução ótima (veja as colunas “testes” e “cortes”). O algoritmo B&BC forneceu uma solução para a instância gcut11, uma das instâncias que o CPLEX atingiu o limite de memória, tanto considerando Grelha quanto FMT. O algoritmo B&BC provou otimalidade em 2 duas instâncias de tamanho médio que Grelha havia atingido o limite de tempo (cgcut2 and OF1), mas os tempos de processamento piores que aqueles de FMT.

Note que o algoritmo B&BC obteve soluções ótimas para todas as instâncias em que não atingiu o limite de memória, no entanto não foi capaz de provar otimalidade em duas delas (gcut3 e gcut11) dentro do limite de tempo. Como esperado, o método não foi capaz de provar otimalidade para soluções com $m = 30$ ou $m = 50$ tipos de itens no conjunto gcut. À medida que o número de tipos de itens aumenta, o número de soluções simétricas (guilhotinadas e não-guilhotinadas) também aumenta no PMB, o que necessita da geração de mais cortes para remover as soluções de teste não-guilhotinadas. É importante ressaltar que em experimentos considerando limite de tempo de poucos minutos, o método foi capaz de obter soluções de boa qualidade para todas essas instâncias, mas sem o certificado de otimalidade. No entanto, para o limite de tempo de 1 hora, o número de cortes no PMB levou o método a atingir o limite de memória para algumas instâncias. Adicionalmente, nota-se que o tempo de processamento no SPB é desprezível, e os números de testes e de cortes na Tabela 3.3 podem diferir dado que as soluções de teste não são cortadas quando são do tipo guilhotinado, e a solução inicial fornecida não é verificada pelo SPB.

3.4.2 Resultados para as versões 2-estágios e 1-grupo do PCBGR

Neste seção, são analisados os desempenhos do Modelo (3.15) para padrões 2-estágios (Grelha-2-estágios) e do Modelo (3.18) para padrões 1-grupo (Grelha-1-grupo)

Tabela 3.3: Resultados do algoritmo B&BC.

Instância	$ \bar{X} $	$ \bar{Y} $	Algoritmo B&BC					
			var	rest	gap	t[s]	testes	cortes
cgcut1	12	11	470	117	0,0	0,02	0	0
cgcut2	22	51	5.920	1.060	0,0	213,47	7	4
cgcut3	18	46	3.029	784	0,0	183,46	116	114
OF1	41	30	3.659	1.170	0,0	254,58	109	108
OF2	30	30	3.243	851	0,0	14,99	1	0
wang20	46	18	3.029	784	0,0	29,25	168	165
gcut1	23	9	361	186	0,0	0,06	0	0
gcut2	35	48	6.443	1.618	0,0	407,88	1.872	1.872
gcut3	78	41	19.636	3.110	0,0	tl	3.100	3.099
gcut4	85	81	71.393	6.770	100,0	*	*	*
gcut5	16	23	892	340	0,0	1,91	2	1
gcut6	33	38	3.053	1.204	0,0	13,43	0	0
gcut7	63	33	9.244	2.014	0,0	99,66	156	156
gcut8	95	133	97.795	12.458	100,0	*	*	*
gcut9	26	11	1.137	260	0,0	7,45	881	881
gcut10	28	49	2.335	1.316	0,0	357,56	5.472	5.472
gcut11	63	104	31.800	6.416	0,0	tl	1.414	1.413
gcut12	148	115	73.295	16.808	100,0	*	*	*

em relação aos modelos propostos por Lodi, Martello e Monaci (2002) e Yanasse e Morabito (2006), que foram especificamente desenvolvidos pra esses tipos de padrões de corte guilhotinados. Para os modelos de padrões 2-estágios, considerou-se tiras horizontais apenas.

Nas Tabelas 3.4 e 3.5 são reportados os resultados dos quatro modelos. Elas indicam que os modelos de Lodi, Martello e Monaci (2002) e Yanasse e Morabito (2006) têm melhor desempenho médio que Grelha-2-estágios e Grelha-1-grupo, principalmente em relação ao tempo de processamento para solução. Por exemplo, na Tabela 3.5, Grelha-1-grupo resultou em menor tempo de processamento apenas para a instância gcut9. Além disso, os números de variáveis e restrições do modelo Lodi, Martello e Monaci (2002) são significativamente menores que Grelha-2-estágios, enquanto para o modelo de Yanasse e Morabito (2006) e Grelha-1-grupo, esses números são mais próximos. Esse comportamento é de alguma forma esperado, dado que o modelo de Beasley (1985b) – uma sub-parte dos modelos Grelha-2-estágios e Grelha-1-grupo – já conduziria a maiores tempos que esses modelos de *benchmark* para algumas dessas instâncias, o que indica a dificuldade que as restrições de não-sobreposição trazem aos modelos propostos.

Por outro lado, Grelha-2-estágios e Grelha-1-grupo forneceram limitantes de relaxação linear melhores que aqueles dos outros modelos, especialmente em relação ao modelo de Yanasse e Morabito (2006), o que tende a ser útil para o desenvolvimento de métodos de solução. Além disso, Grelha-2-estágios e Grelha-1-grupo são mais indicados

Tabela 3.4: Comparação entre o modelo de Lodi, Martello e Monaci (2002) e Grelha-2-estágios.

Instância	Lodi, Martello e Monaci (2002)					Grelha-2-estágios				
	var	rest	gap	solr	t[s]	var	rest	gap	solr	t[s]
cgcut1	136	32	0,0	257,8	0,10	480	136	0,0	244,5	2,45
cgcut2	276	46	0,0	2.878,0	0,18	5.970	1.159	0,0	2.906,1	1.158,17
cgcut3	1.953	124	0,0	2.005,3	0,43	3.074	873	0,0	1.995,5	105,33
OF1	276	46	0,0	2.800,0	0,47	3.688	1.227	0,0	2.789,9	138,83
OF2	300	48	0,0	2.800,0	1,78	3.272	908	0,0	2.747,2	41,20
wang20	903	84	0,0	2.800,0	1,98	3.046	817	0,0	2.791,1	39,67
gcut1	55	20	0,0	62.192,6	0,17	369	201	0,0	50.960,0	1,72
gcut2	210	40	0,0	62.397,1	0,45	6.490	1.711	0,0	61.240,4	39,02
gcut3	465	60	0,0	62.500,0	1,07	19.676	3.189	0,0	61.785,3	571,51
gcut4	1.275	100	0,0	62.500,0	1,34	71.473	6.929	100,0	*	*
gcut5	55	20	0,0	247.475,5	0,60	914	383	0,0	228.381,9	1,96
gcut6	210	40	0,0	249.494,5	0,05	3.090	1.277	0,0	242.556,1	32,44
gcut7	465	60	0,0	250.000,0	0,06	9.276	2.077	0,0	243.671,0	96,12
gcut8	1.275	100	0,0	250.000,0	1,33	97.927	12.721	100,0	*	*
gcut9	55	20	0,0	990.591,4	0,11	1.147	279	0,0	947.014,0	1,35
gcut10	210	40	0,0	997.715,9	1,21	2.383	1.411	0,0	967.987,4	15,32
gcut11	465	60	0,0	999.177,6	2,91	31.903	6.621	0,0	983.752,5	2.244,23
gcut12	1.275	100	0,0	1.000.000,0	1,21	73.409	17.035	100,0	*	*

Tabela 3.5: Comparação entre o modelo de Yanasse e Morabito (2006) e Grelha-1-grupo.

Instance	Yanasse e Morabito (2006)					Grelha-1-grupo				
	var	rest	gap	solr	t[s]	var	rest	gap	solr	t[s]
cgcut1	170	174	0,0	364,0	0,03	491	157	0,0	244,5	2,15
cgcut2	837	600	0,0	4.449,0	0,57	5.991	1.200	0,0	2.906,1	tl
cgcut3	6.956	2.624	0,0	20.631,7	3,61	3.091	906	0,0	1.995,5	491,27
OF1	1.096	872	0,0	7.670,0	0,20	3.728	1.306	0,0	2.789,9	304,17
OF2	962	708	0,0	8.652,0	0,19	3.301	965	0,0	2.747,2	207,12
wang20	7.126	3.098	0,0	23.910,0	6,44	3.091	906	0,0	2.791,1	109,15
gcut1	1.185	852	0,0	163.562,0	0,44	391	244	0,0	50.960,0	1,71
gcut2	7.779	2.986	0,0	274.563,0	5,52	6.524	1.778	0,0	61.240,4	250,47
gcut3	23.945	6.766	0,0	407.651,0	144,70	19.753	3.342	0,0	61.785,3	675,46
gcut4	101.794	16.508	1,2	731.408,0	tl	71.557	7.096	100,0	*	*
gcut5	1.274	832	0,0	545.300,0	2,68	929	412	0,0	228.381,9	2,78
gcut6	8.159	3.005	0,0	1.232.057,0	15,42	3.122	1.340	0,0	242.556,1	59,77
gcut7	24.709	6.584	0,0	2.004.791,0	103,97	9.338	2.200	0,0	243.671,0	164,93
gcut8	106.416	17.256	3,2	2.805.462,0	tl	98.021	12.908	100,0	*	*
gcut9	1.318	952	0,0	2.021.830,0	8,45	1.172	328	0,0	947.014,0	8,35
gcut10	9.049	3.362	0,0	5.355.377,0	18,16	2.410	1.464	0,0	967.987,4	92,85
gcut11	28.331	7.292	0,0	6.536.520,0	136,76	31.965	6.744	100,0	*	*
gcut12	121.197	19.980	0,0	12.521.992,0	1.327,06	73.556	17.328	100,0	*	*

para casos onde a discretização do objeto original pode fornecer vantagens. Por exemplo, se o valor dos tipos de itens dependerem de sua área e posição, devido a diferentes espessuras ao longo do objeto, como discutido por Gilmore e Gomory (1965), então pode-se considerar os modelos Grelha-2-estágios ou Grelha-1-grupo após facilmente substituir o parâmetro v_k por v_{kij} . Os modelos da literatura não parecem ser facilmente adaptados para esse caso.

3.5 Comentários finais do capítulo

Nesse capítulo abordou-se o PCBGR, uma variante muito estudada no campo dos PCE. Sua aplicabilidade em diferentes ambientes produtivos motiva a proposição de diversos métodos de solução. No entanto, os primeiros modelos que formularam este problema (BEN MESSAOUD; CHU; ESPINOUSE, 2008; FURINI; MALAGUTI; THOMOPULOS, 2016) apareceram na literatura mais de 30 anos depois dos artigos seminais (CHRISTOFIDES; WHITLOCK, 1977; WANG, 1983).

Foram propostas uma formulação pseudopolinomial de PNLI que modela o problema, e sua formulação pseudopolinomial de PLI equivalente, ambas baseadas no modelo de Beasley (1985b). Elas são extensões desse modelo para o caso guilhotinado, dado que a formulação de Beasley foi projetada para padrões não-guilhotinados. Além disso, um procedimento que, sem perda de generalidade, reduz os números de variáveis e restrições da formulação de PLI foi proposto. Ao reformular tal formulação linear, novas formulações pseudopolinomiais foram desenvolvidas para casos especiais de padrões de corte guilhotinado, isto é, padrões 2-estágios e 1-grupo. Como método de solução para o PCBGR, foi proposto um algoritmo *branch-and-Benders-cut* (B&BC), baseado na aplicação da decomposição de Benders à formulação linear proposta. O Problema-Mestre-de-Benders resultante é o modelo de Beasley (1985b).

Os resultados dos experimentos computacionais usando instâncias de *benchmark* da literatura mostraram que a formulação de PLI proposta é apropriada para cenários de poucos tipos de itens, que são grandes em relação ao objeto (como nas instâncias *gcut*), pois resultam em menos variáveis. Nesse caso, o modelo proposto apresentou melhor desempenho médio que as outras duas formulações da literatura. Por outro lado, não é indicado para cenários com muitos tipos de itens, dado que o comportamento oposto tende a aparecer.

Também foi verificado que o algoritmo B&BC apresenta melhor desempenho médio que a formulação de PLI proposta, diante de um *solver* de propósito geral, especialmente quando o valor ótimo da função objetivo é próximo do valor ótimo da função objetivo de Beasley (1985b) (isto é, dos padrões não-guilhotinados). No entanto, o desempenho do método é afetado negativamente por cenários com muitos tipos de itens, dado que os números de soluções simétricas de padrões (guilhotinados e não-guilhotinados) tendem a crescer e, assim, o método necessita mais cortes combinatórios para convergir.

Com respeito aos casos especiais de padrões guilhotinados 2-estágios e 1-grupo, os modelos propostos atingiram desempenho inferior em relação a outros modelos da literatura.

A extensão das formulações propostas para o caso de rotação de itens é direta. Conforme discutido anteriormente, basta-se adicionar um tipo de item $w \times l$ para cada tipo de item $l \times w$ da instância original, considerando agregadas suas cópias quando no caso restrito. No entanto, isso gera um alto custo computacional, dado que os números de variáveis e restrições aumentam consideravelmente. Assim como as formulações de Ben Messaoud, Chu e Espinouse (2008) e Furini, Malaguti e Thomopulos (2016), as formulações desse capítulo não parecem ser facilmente adaptadas para lidar com o caso d -estágios, com $d \geq 3$. No entanto, diferentemente dessas formulações da literatura, a consideração do caso com defeitos no objeto a ser cortado é direta na formulação baseada em grelha, e é objeto de análise do próximo capítulo.

Capítulo 4

Problema de corte bidimensional guilhotinado e restrito de objeto com defeitos: propostas de formulação matemática e de métodos de solução

Neste capítulo aborda-se o Problema de Corte Bidimensional Guilhotinado e Restrito de Objeto com Defeitos (PCBGR.D), uma extensão do PCBGR que considera um objeto heterogêneo devido a regiões defeituosas. A formulação matemática e algoritmo de decomposição Benders, originalmente propostos no Capítulo 3, são estendidos ao PCBGR.D neste capítulo nas Seções 4.2 e 4.3, respectivamente. Um algoritmo baseado em Programação por Restrições (PR) é proposto na Seção 4.4. Experimentos computacionais são realizados para os casos irrestrito e restrito, a partir de instâncias de *benchmark* na Seção 4.5. Por questão de clareza, optou-se por descrever a formulação e o método propostos anteriormente de forma completa a seguir, no contexto de defeitos.

Objetivos do capítulo:

- Estender a formulação de PLI não-estagiada e o algoritmo de decomposição de Benders, ambos baseados em grelha e originalmente propostos ao PCBGR, para o PCBGR.D;
- Desenvolver um algoritmo baseado em Programação por Restrições (PR), para o PCBGR.D.

Parte dos resultados deste capítulo foram recentemente publicados:

★ Martin, M.; Hokama, P. H.; Morabito, R.; Munari, P. *The constrained two-dimensional guillotine cutting problem with defects: an ILP formulation, a Benders decomposition and a CP-based algorithm*. International Journal of Production Research (2019), p. 1-18. DOI: <https://doi.org/10.1080/00207543.2019.1630773>.

4.1 Descrição do problema

De acordo com Wäscher, Haußner e Schumann (2007), o Problema da Alocação Bidimensional de Único Objeto Retangular (do inglês, *Two-dimensional Rectangular Single Large Object Placement Problem* de sigla 2D_R_SLOPP) é um problema padrão entre os PCE, no qual um objeto retangular grande é cortado para produzir itens retangulares pequenos de diferentes dimensões ou tipos. O objetivo é maximizar a área total ou valor total dos itens cortados (problema de maximização de saídas), e os itens são alocados de forma ortogonal às bordas do objeto. A variante desse capítulo, isto é, o Problema de Corte Bidimensional Guilhotinado e Restrito de Objeto com Defeitos (PCBGR_D) considera as seguintes restrições adicionais:

- Todos os cortes são guilhotináveis, assim um corte em um retângulo gera sempre dois sub-retângulos menores (classe de guilhotina);
- Há um limitante no número máximo de cópias para cada tipo de item no padrão (caso restrito);
- Não há limitante sobre o número máximo de estágios guilhotináveis (caso não-estagiado);
- Há única orientação, assim a rotação de itens é proibida (restrição rotacional);
- O objeto é heterogêneo devido a alguns defeitos, que são representados por um conjunto de retângulos;
- Um defeito é representado pelo menor conjunto de retângulos sobrepostos capaz de contê-lo e suas posições no objeto – isto é sem perda de generalidade pois os cortes são ortogonais guilhotinados. Cada item cortado é obtido de uma área livre de defeitos do objeto.

Destaca-se a relevância de considerar o caso não-estagiado restrito no contexto de materiais defeituosos por: (i) evitar superprodução, dado que esses ambientes produtivos geralmente lidam com cenários de baixa demanda para cada tipo de item; (ii) permitir buscar taxas de utilização de material maiores com diversos estágios guilhotinados, dado que esses ambientes produtivos lidam com materiais custosos. A Figura 4.1 ilustra diferentes tipos de padrões guilhotinados bidimensionais com defeitos.

O PCBGR_D pode ser formalmente declarado de acordo com os seguintes conjuntos e parâmetros.

- Conjuntos:
 - $K = \{1, \dots, m\}$: conjuntos de tipos de itens;
 - $D = \{1, \dots, \delta\}$: conjuntos de defeitos;

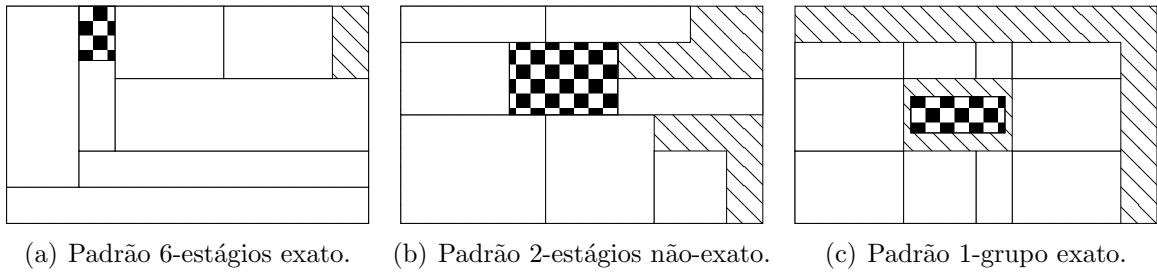


Figura 4.1: Padrões de corte bidimensionais guilhotinados com único defeito.

• Parâmetros:

- L e W : comprimento e largura do objeto defeituoso, respectivamente;
- L_k , W_k e V_k : comprimento, largura e valor de um tipo de item $k \in K$, respectivamente;
- U_k : número máximo de peças permitidas do tipo de item $k \in K$ no padrão;
- (ox_d, oy_d) : posição do canto inferior-esquerdo do defeito $d \in D$ em um sistema de coordenadas Cartesianas;
- l_d e w_d : comprimento e largura do defeito $d \in D$, respectivamente.

Para lidar com a ausência da restrição de rotação de itens, pode-se adicionar um novo tipo de item k' de comprimento $L_{k'} = W_k$ e largura $W_{k'} = L_k$, para cada $k \in K$, se necessário. A Tabela 4.1 mostra uma instância ilustrativa do PCBGR_D para um objeto de dimensões $(L \times W) = (12 \times 8)$, $m = 3$ tipos de itens e $\delta = 3$ defeitos. A Figura 4.2 esboça o objeto defeituoso desse exemplo e sua correspondente solução ótima. O canto inferior-esquerdo do objeto é a origem de um sistema de coordenadas Cartesianas, no qual cada área defeituosa pode ser representada e posicionada.

Tabela 4.1: Instância ilustrativa do PCBGR_D.

Parâmetros	Tipos de itens			Parâmetros	Defeitos		
	1	2	3		1	2	3
L_k	8	5	6	ox_d	2	6	9
W_k	2	4	6	oy_d	1	5	3
V_k	17	18	40	l_d	2	1	3
U_k	2	2	1	w_d	1	2	1

4.2 Formulação matemática

Esse autor não está ciente de qualquer outra formulação de PLI para o PCBGR_D. Essa seção propõe uma formulação de PLI, baseada na discretização do objeto ao problema de corte bidimensional guilhotinado e restrito com defeitos. Essa formulação

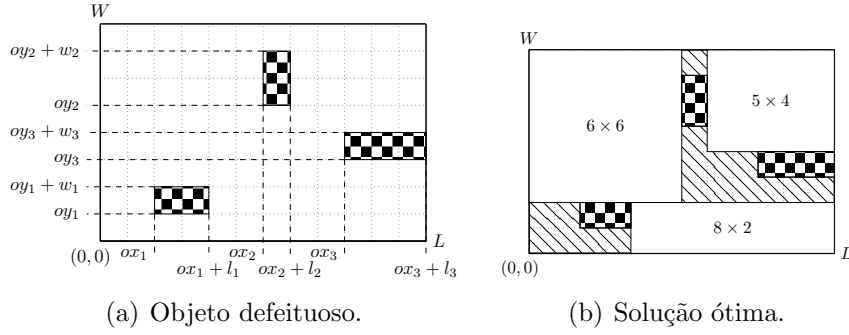


Figura 4.2: Exemplo ilustrativo com três áreas defeituosas.

pode ser vista como uma generalização da formulação proposta no Capítulo 3. Optou-se por lidar com PCBGR.D a partir do conceito de grelha, pois os defeitos estão posicionados e fixos no objeto, e tal abordagem pode facilmente lidar com isso ao proibir a alocação de itens aos pontos sobre os quais as peças poderiam sobrepor um defeito.

Sem perda de generalidade, assumem-se que os dados da instância são inteiros positivos. Sejam o conjunto R a discretização do objeto e o conjunto R_k os pontos que um tipo de item $k \in K$ pode ter seu canto inferior-esquerdo alocado, sem considerar as áreas defeituosas, conforme definido nas Equações (4.1) e (4.2), respectivamente. O conjunto G_k é definido na Equação (4.3) e consiste dos pontos factíveis para alocação do canto inferior-esquerdo de um tipo de item k , isto é, pontos no qual a sobreposição com uma área defeituosa não é possível. Note que as bordas direita e inferior de um item alocado (ou defeito) não são consideradas cobertas por tal item (ou defeito).

$$R = \{(i, j) \in \mathbb{Z}^2 \mid 0 \leq i \leq L, 0 \leq j \leq W\}. \quad (4.1)$$

$$R_k = \{(i, j) \in R \mid i \leq L - L_k, j \leq W - W_k\}. \quad (4.2)$$

$$G_k = \{(i, j) \in R_k \mid \forall d \in D : (ox_d + l_d \leq i) \text{ ou } (ox_d \geq i + L_k) \\ \text{ou } (oy_d + w_d \leq j) \text{ ou } (oy_d \geq j + W_k)\}. \quad (4.3)$$

A variável de decisão que lida com a alocação de um tipo de item k à posição (i, j) é definida na Equação (4.4), e ilustrada na Figura 4.3(b). Beasley (1985b) e Neidlein, Scholz e Wäscher (2016) sugerem usar o conjunto R_k como domínio para as posições (i, j) na definição das variáveis x_{kij} , conjuntamente com restrições adicionais para proibir alocações que resultariam em sobreposição a defeitos. No entanto, na formulação aqui proposta, optou-se por não criar tais variáveis pois essas alocações são conhecidas como infactíveis. Assim, restringe-se o domínio de (i, j) para G_k . A alocação esboçada na Figura 4.3(a) está no conjunto R_k , mas não no conjunto G_k . Por exemplo, o conjunto G_k do exemplo da Tabela 4.1 indica que o tipo de item $k = 1$ pode ser alocado às posições $\{(0, 2), (0, 3), (1, 2), (1, 3), (4, 0), (4, 1)\}$, o tipo de item $k = 2$ às posições $\{(0, 2), (0, 3), (0, 4), (1, 2), (1, 3), (1, 4), (4, 0), (4, 1), (7, 4)\}$ e o tipo de item $k = 3$

às posições $\{(0, 2)\}$.

$$x_{kij} = \begin{cases} 1, & \text{se um tipo de item } k \text{ é alocado à posição } (i, j), \\ 0, & \text{caso contrário.} \end{cases} \quad k \in K, (i, j) \in G_k. \quad (4.4)$$

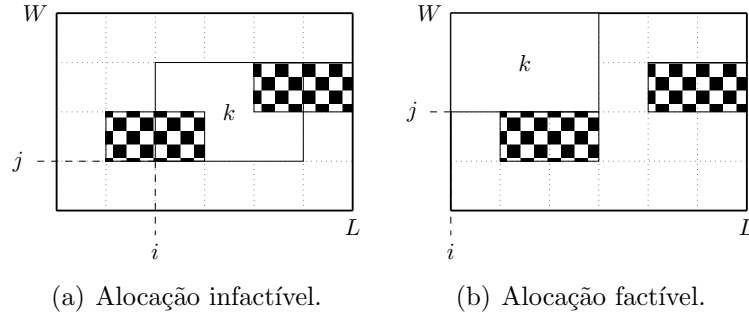


Figura 4.3: Ilustração da variável x_{kij} .

A formulação também inclui três outros tipos de decisões para endereçar cortes horizontais, verticais e existência de retângulos. As correspondentes variáveis de decisão são definidas, respectivamente, nas Equações (4.5), (4.6) and (4.7). Nas Figuras 4.4(a), 4.4(b) e 4.4(c) esboçam-se tais variáveis. Observe que a variável $p_{i_1 i_2 j_1 j_2}$ representa a área de possíveis subpadrões sobre os quais cortes são realizados para obter itens. Quando $p_{i_1 i_2 j_1 j_2} = 1$, diz-se que o retângulo $(i_1 i_2 j_1 j_2)$ está ativo.

$$h_{ii'j} = \begin{cases} 1, & \text{se um corte horizontal é realizado de } (i, j) \text{ até } (i', j), \\ 0, & \text{caso contrário,} \end{cases} \quad 0 \leq i < i' \leq L, 0 \leq j \leq W. \quad (4.5)$$

$$v_{ijj'} = \begin{cases} 1, & \text{se um corte vertical é realizado de } (i, j) \text{ até } (i, j'), \\ 0, & \text{caso contrário,} \end{cases} \quad 0 \leq i \leq L, 0 \leq j < j' \leq W. \quad (4.6)$$

$$p_{i_1 i_2 j_1 j_2} = \begin{cases} 1, & \text{se um retângulo de canto inferior-esquerdo} \\ & (i_1, j_1) \text{ e canto superior-direito } (i_2, j_2) \text{ existe,} \\ 0, & \text{caso contrário,} \end{cases} \quad 0 \leq i_1 < i_2 \leq L, 0 \leq j_1 < j_2 \leq W. \quad (4.7)$$

Note que V_k é o valor do tipo de item $k \in K$. A função objetivo consiste

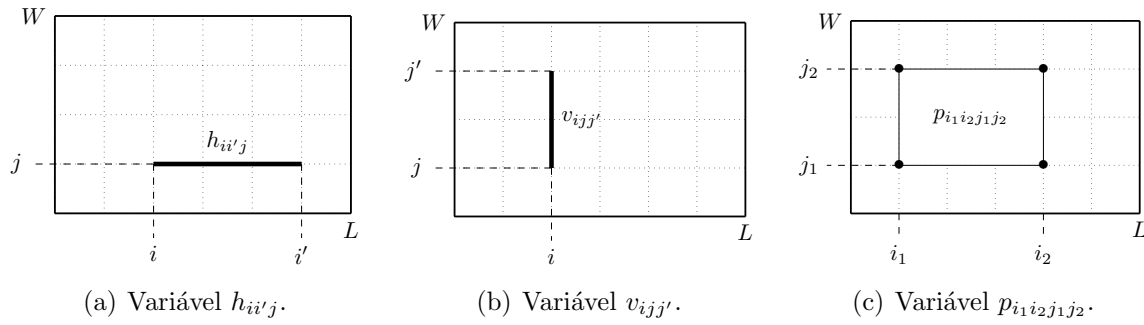


Figura 4.4: Ilustração das variáveis $h_{ii'j}$, $v_{ijj'}$ e $p_{i_1 i_2 j_1 j_2}$.

em maximizar o valor total dos itens alocados no objeto, dado por:

$$\sum_{k \in K} \sum_{(i,j) \in G_k} V_k x_{kij}. \quad (4.8)$$

Por questão de clareza, no que segue, a formulação é apresentada por blocos de restrições com suas respectivas explicações.

Restrições de não-sobreposição e caso restrito

Como formalmente proposto por Beasley (1985b), define-se o seguinte parâmetro para as restrições de não-sobreposição entre quaisquer pares de itens alocados:

$$f_{kij'j'} = \begin{cases} 1, & \text{se } 0 \leq i \leq i' \leq i + l_k - 1 < L \text{ e } 0 \leq j \leq j' \leq j + w_k - 1 < W, \\ 0, & \text{caso contrário.} \end{cases}$$

Usando esse parâmetro, as restrições (4.9) garantem que cada posição (i', j') é coberta por até um item alocado.

$$\sum_{k \in K} \sum_{(i,j) \in G_k} f_{kij'j'} x_{kij} \leq 1, \quad \forall (i', j') \in R, i' < L, j' < W. \quad (4.9)$$

As restrições (4.10) garantem o caso restrito do PCBGR.D. Observe que pode-se impor um limitante inferior ao número de vezes que cada tipo de item $k \in K$ deve aparecer no padrão de forma similar ao limitante superior, se necessário.

$$\sum_{(i,j) \in G_k} x_{kij} \leq U_k, \quad \forall k. \quad (4.10)$$

Quadro de guilhotina

Um corte de guilhotina requer a existência de um retângulo ativo, tal que ele seja feito de borda a borda do retângulo. Assim, um corte horizontal ou vertical existe apenas se há um retângulo ativo que suporte esse segmento, como ilustrado na Figura 4.5. As restrições (4.11) até (4.14) garantem que um retângulo $p_{i_1 i_2 j_1 j_2}$ está ativo apenas se suas bordas estão cortadas, o que só pode acontecer quando o *right-hand-side* (rhs) de cada uma dessas restrições é ao menos 1.

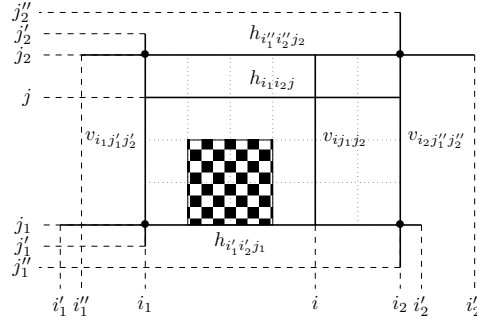


Figura 4.5: Caracterização das restrições de guilhotina.

$$p_{i_1 i_2 j_1 j_2} \leq \sum_{i_1' \leq i_1, i_2 \leq i_2'} h_{i_1' i_2' j_1}, \quad 0 \leq i_1 < i_2 \leq L, 0 \leq j_1 < j_2 \leq W, \quad (4.11)$$

$$p_{i_1 i_2 j_1 j_2} \leq \sum_{i_1' \leq i_1, i_2 \leq i_2'} h_{i_1' i_2' j_2}, \quad 0 \leq i_1 < i_2 \leq L, 0 \leq j_1 < j_2 \leq W, \quad (4.12)$$

$$p_{i_1 i_2 j_1 j_2} \leq \sum_{j_1' \leq j_1, j_2 \leq j_2'} v_{i_1 j_1' j_2'}, \quad 0 \leq i_1 < i_2 \leq L, 0 \leq j_1 < j_2 \leq W, \quad (4.13)$$

$$p_{i_1 i_2 j_1 j_2} \leq \sum_{j_1' \leq j_1, j_2 \leq j_2'} v_{i_2 j_1' j_2'}, \quad 0 \leq i_1 < i_2 \leq L, 0 \leq j_1 < j_2 \leq W. \quad (4.14)$$

As restrições (4.15) e (4.16), respectivamente, mostram como a formulação proposta modela os cortes guilhotinados horizontais e verticais sob essa perspectiva.

$$h_{i_1 i_2 j} \leq \sum_{j_1 < j < j_2} p_{i_1 i_2 j_1 j_2}, \quad 0 \leq i_1 < i_2 \leq L, 0 < j < W, \quad (4.15)$$

$$v_{i_1 j_1 j_2} \leq \sum_{i_1 < i < i_2} p_{i_1 i_2 j_1 j_2}, \quad 0 < i < L, 0 \leq j_1 < j_2 \leq W. \quad (4.16)$$

Evitar cortes sobre itens alocados

Sejam S_{kij}^h e S_{kij}^v respectivamente os conjuntos de todos segmentos horizontais (j', j'', j') e verticais (i', j', j'') que cruzam um item alocado x_{kij} (isto é, quando $x_{kij} = 1$), definidos como:

$$S_{kij}^h = \{(i', i'', j') \in \mathbb{Z}^3 \mid j < j' < j + w_k, 0 \leq i' < i + l_k, \max\{i, i'\} < i'' \leq L\},$$

$$S_{kij}^v = \{(i', j', j'') \in \mathbb{Z}^3 \mid i < i' < i + l_k, 0 \leq j' < j + w_k, \max\{j, j'\} < j'' \leq W\}.$$

Ambos os conjuntos são ilustrados na Figura 4.6, que mostra exemplos de segmentos que pertencem aos conjuntos e suas sobreposições com certo tipo de item k alocado à posição (i, j) . As restrições (4.17) garantem que nenhum corte horizontal nem vertical que cruze um item alocado x_{kij} assuma valor unitário. Assim, nenhum corte pode existir sobre um item alocado. Para cada $k \in K$ e $(i, j) \in G_k$, o parâmetro M_{kij} é definido como $|S_{kij}^h| + |S_{kij}^v|$.

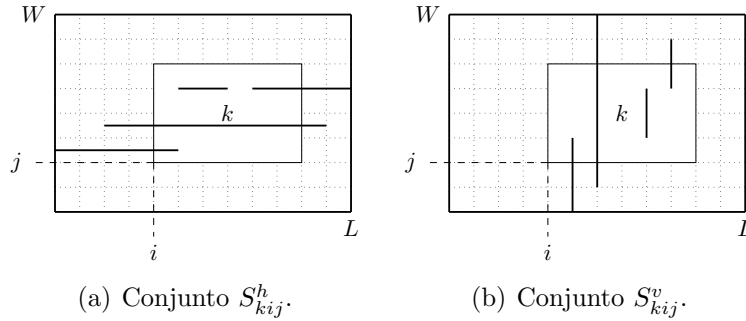


Figura 4.6: Ilustrações de alguns segmentos de S_{kij}^h e S_{kij}^v .

$$\sum_{(i', i'', j') \in S_{kij}^h} h_{i', i'', j'} + \sum_{(i', j', j'') \in S_{kij}^v} v_{i', j', j''} \leq (1 - x_{kij}) M_{kij}, \quad \forall k, \forall (i, j) \in G_k. \quad (4.17)$$

Alocações de itens a cantos cortados

As restrições (4.18) e (4.19) restringem a alocação de um tipo de item k às posições (i, j) de cantos horizontais e verticais cortados. Essas restrições agem conjuntamente com as restrições de guilhotina (4.11)–(4.16) e restrições que proíbem cortes sobre itens alocados (4.17) para promover o comportamento guilhotinado da formulação.

$$x_{kij} \leq \sum_{\substack{0 \leq i' < i, \\ i + l_k \leq i'' \leq L}} h_{i', i'', j}, \quad \forall k, \forall (i, j) \in G_k, \quad (4.18)$$

$$x_{kij} \leq \sum_{\substack{0 \leq j' < j, \\ j + w_k \leq j'' \leq W}} v_{i, j', j''}, \quad \forall k, \forall (i, j) \in G_k. \quad (4.19)$$

Bordas do objeto cortadas e domínio das variáveis

As restrições (4.20) garantem que as bordas do objeto original estão cortadas, o que é necessário para as restrições (4.15) e (4.16). As restrições (4.21) até (4.24) impõem o domínio das variáveis.

$$h_{0L0} = 1, h_{0LW} = 1, v_{00W} = 1, v_{L0W} = 1, p_{0L0W} = 1, \quad (4.20)$$

$$x_{kij} \in \{0, 1\}, \quad \forall k, \forall (i, j) \in G_k, \quad (4.21)$$

$$h_{iiv'j} \in \{0, 1\}, \quad \forall (i, j) \in R, 0 \leq i < i' \leq L, \quad (4.22)$$

$$v_{ijj'} \in \{0, 1\}, \quad \forall (i, j) \in R, 0 \leq j < j' \leq W, \quad (4.23)$$

$$p_{i_1i_2j_1j_2} \in \{0, 1\}, \quad 0 \leq i_1 < i_2 \leq L, 0 \leq j_1 < j_2 \leq W. \quad (4.24)$$

Tendo definidos todos os elementos necessários, tem-se o Modelo (4.25), que é uma formulação de PLI para o PCBGR_D baseado na discretização do objeto.

$$\mathbf{Max} \text{ (4.8)}, \quad \mathbf{s.a} \text{ (4.9) – (4.24)}. \quad (4.25)$$

Por exemplo, a solução ótima esboçada na Figura 4.2(b) é obtida pelo Modelo (4.25) com as seguintes variáveis assumindo valores não-nulos: $x_{1,4,0}$, $x_{2,7,4}$, $x_{3,0,2}$, $h_{0,12,2}$, $v_{7,2,8}$, $h_{7,12,4}$, $p_{0,12,0,2}$, $p_{0,12,2,8}$, $p_{0,7,2,8}$ e $p_{7,12,2,8}$ – as bordas cortadas do objeto são omitidas. Os números de variáveis e restrições do Modelo (4.25) são ambos em ordem de $O(mLW + L^2W^2)$. Note que pode-se desenvolver desigualdades válidas, associando as variáveis x_{kij} e $p_{i_1i_2j_1j_2}$ de forma a melhorar a relaxação linear do modelo, como pesquisas futuras.

4.2.1 Adaptação da discretização dos conjuntos normais

A discretização dos conjuntos normais é uma técnica proposta por Herz (1972) e por Christofides e Whitlock (1977) que considera, sem perda de generalidade, apenas posições dadas por combinações lineares inteiras positivas das dimensões dos tipos de itens, em vez de todas as posições inteiras do objeto. Diversos autores têm usado tal discretização melhorada buscando menos variáveis e restrições em modelos, e/ou acelerar a convergência de algoritmos (BEASLEY, 1985b; CHRISTOFIDES; WHITLOCK, 1977; SCHEITHAUER; TERNO, 1988; MORABITO; PUREZA, 2010; AFSHARIAN; NIKNEJAD; WÄSCHER, 2014; FURINI; MALAGUTI; THOMOPULOS, 2016). A ideia

principal é que, para qualquer padrão de corte, há um padrão equivalente com seus itens alocados justificados ao canto inferior-esquerdo, de tal forma que um item alocado tem seu canto inferior-esquerdo sobre uma borda do objeto e/ou um outro item alocado.

Na presença de defeitos, esse princípio deve ser estendido porque um item alocado pode estar à direita ou sobre: (i) um outro item alocado; ou, (ii) um defeito. As Equações (4.26) e (4.27) definem os conjuntos de discretização para o PCBGR.D. A discretização do objeto representada pelo conjunto R , previamente definido na Equação (4.1), pode ser redefinido usando os conjuntos \bar{X} e \bar{Y} , sem perda de generalidade, conforme apresentado em (4.28). Similarmente, os conjuntos R_k e G_k podem ser redefinidos de acordo com a nova definição do conjunto R .

$$\bar{X} = \left\{ \begin{array}{l} q_x \mid q_x = \sum_{k \in K} n_k L_k + \sum_{d \in D} (ox_d + l_d) \delta_d, \ 0 \leq q_x \leq L - \min_{k \in K} \{L_k\}, \\ n_k \in \mathbb{N}, n_k \leq U_k, \ \delta_d \in \{0, 1\} \end{array} \right\} \cup \{L\}. \quad (4.26)$$

$$\bar{Y} = \left\{ \begin{array}{l} q_y \mid q_y = \sum_{k \in K} n_k W_k + \sum_{d \in D} (oy_d + w_d) \delta_d, \ 0 \leq q_y \leq W - \min_{k \in K} \{W_k\}, \\ n_k \in \mathbb{N}, n_k \leq U_k, \ \delta_d \in \{0, 1\} \end{array} \right\} \cup \{W\}. \quad (4.27)$$

$$R = \{(i, j) \in \mathbb{Z}^2 \mid i \in \bar{X}, j \in \bar{Y}\}. \quad (4.28)$$

Como os defeitos estão fixos no objeto, a definição de \bar{X} e \bar{Y} pode levar a alguns pontos desnecessários abaixo ou à esquerda de cada defeito. Observe que o conjunto \bar{X} (respectivamente \bar{Y}) pode ser obtido por meio de algoritmos tradicionais (CINTRA et al., 2008), ao considerar os pontos defeituosos $ox_d + l_d$ (respectivamente $oy_d + w_d$), $\forall d \in D$, como elementos do conjunto inicial de pontos discretizados no início de tais algoritmos. Por exemplo na Tabela 4.1, tem-se os conjuntos $\bar{X} = \{0, 4, 5, 6, 7, 12\}$ e $\bar{Y} = \{0, 2, 4, 6, 8\}$. Na ausência de defeitos, o conjunto \bar{X} não incluiria os elementos 4 e 7.

4.3 Algoritmo de decomposição de Benders

A formulação de PLI proposta na seção anterior é indicada para resolver instâncias pequenas, a partir de *solvers* de propósito geral, visto que os números de variáveis e restrições tendem a crescer rapidamente segundo as dimensões do objeto.

Para obter soluções para instâncias maiores e evitar questões de discretização, essa seção estende o algoritmo de decomposição de Benders do Capítulo 3 para o PCBGR.D. A decomposição proposta parte do Modelo (4.29), chamado Problema-Mestre-de-Benders (PMB) aqui, e iterativamente busca por padrões de corte factíveis com respeito às restrições de não-sobreposição e caso restrito. Nota-se, mais uma vez,

que o PMB é a formulação de Beasley (1985b) para o caso não-guilhotinado restrito.

$$\mathbf{Max} \text{ (4.8), } \quad \mathbf{s.a} \text{ (4.9) – (4.10), (4.21).} \quad (4.29)$$

Após resolver o PMB, obtém-se uma solução \bar{x}_{kij} que tem de ser analisada no Subproblema-de-Benders (SPB) com respeito à restrição de guilhotina. A solução é aceita apenas se ela é guilhotinada; caso contrário, ela é rejeitada e conduz a um corte combinatório do tipo $\sum_{\forall(kij)|\bar{x}_{kij}=1} x_{kij} \leq \sum_{\forall(kij)} \bar{x}_{kij} - 1$ que é adicionado ao PMB, então uma nova iteração do algoritmo inicia. As variáveis do Modelo (4.25) que não estão incluídas no Modelo (4.29) aparecem no SPB. Assim, o SPB é apenas um verificador da restrição de guilhotina e pode resultar em cortes de factibilidade dado que essas variáveis – $h_{i'j}$, $v_{ijj'}$ e $p_{i_1i_2j_1j_2}$ – não estão na função objetivo (4.8). Observe que o SPB pode ter menos variáveis e restrições que o Modelo (4.25), dado que a solução \bar{x}_{kij} é usada para fixar o valor das correspondentes variáveis. Seja $\bar{\mathcal{X}}$ o conjunto de índices de variáveis (k, i, j) tais que $\bar{x}_{kij} > 0$; e sejam os conjuntos $\bar{\mathcal{H}}$, $\bar{\mathcal{V}}$ e $\bar{\mathcal{P}}$, respectivamente, os domínios reduzidos das variáveis $h_{i'j}$, $v_{ijj'}$ e $p_{i_1i_2j_1j_2}$ devido à fixação de x_{kij} . As definições dos conjuntos $\bar{\mathcal{H}}$, $\bar{\mathcal{V}}$ e $\bar{\mathcal{P}}$ consideram:

- a discretização \bar{X} (resp. \bar{Y}) com elementos 0 e L (resp. W), os elementos i (resp. j) de todos $\bar{\mathcal{X}}$ e os $ox_d + l_d$ (respectivamente $oy_d + w_d$) de todos os defeitos.;
- os comprimentos $i' - i$ e $i_2 - i_1$ (resp. larguras $j' - j$ e $j_2 - j_1$) maiores ou iguais ao tipo de item alocado com menor comprimento L_k (resp. largura W_k) em $\bar{\mathcal{X}}$.

Para o exemplo descrito na Tabela 4.1, essa discretização simplificada resulta nos conjuntos $\bar{X} = \{0, 4, 7, 12\}$ e $\bar{Y} = \{0, 2, 4, 8\}$.

O SPB pode ser formalmente declarado como o Modelo (4.30), que é um problema linear de satisfação de restrições. Esse modelo é factível se, e somente se, a solução \bar{x}_{kij} é guilhotinada. Por abuso de notação, assumem-se que as variáveis e restrições (4.11) até (4.16) no Modelo (4.30) são definidas apenas pelos índices em $\bar{\mathcal{H}}$, $\bar{\mathcal{V}}$ e $\bar{\mathcal{P}}$.

$$\mathbf{Max} \ 0 \quad (4.30a)$$

$$\mathbf{s.a} \text{ (4.11) – (4.16), (4.20),} \quad (4.30b)$$

$$\sum_{(i''j') \in S_{kij}^h \cup \bar{\mathcal{H}}} h_{i''j'} + \sum_{(i'j'j'') \in S_{kij}^v \cup \bar{\mathcal{V}}} v_{i'j'j''} \leq 0, \quad \forall(k, i, j) \in \bar{\mathcal{X}}, \quad (4.30c)$$

$$1 \leq \sum_{(i'j'j'') \in \bar{\mathcal{H}} | i' \leq i, i + l_k \leq i''} h_{i'j'j''}, \quad \forall(k, i, j) \in \bar{\mathcal{X}}, \quad (4.30d)$$

$$1 \leq \sum_{(ij'j'') \in \bar{\mathcal{V}} | j' \leq j, j + w_k \leq j''} v_{ij'j''}, \quad \forall(k, i, j) \in \bar{\mathcal{X}}, \quad (4.30e)$$

$$h_{ii'j} \in \{0, 1\}, \quad \forall (i, i', j) \in \bar{\mathcal{H}}, \quad (4.30f)$$

$$v_{ijj'} \in \{0, 1\}, \quad \forall (i, j, j') \in \bar{\mathcal{V}}, \quad (4.30g)$$

$$p_{i_1 i_2 j_1 j_2} \in \{0, 1\}, \quad \forall (i_1, i_2, j_1, j_2) \in \bar{\mathcal{P}}. \quad (4.30h)$$

Seguindo a nomenclatura apresentada no detalhado *survey* de decomposições de Benders de Rahmaniani et al. (2017), foi implementado um método *Branch-and-Benders-cut*, visto que o BPM é resolvido dentro de uma única árvore de busca. Os cortes combinatórios, se existirem, são adicionados como *lazy constraints* após verificar se uma solução inteira \bar{x}_{kij} é do tipo guilhotinado usando o SPB – o leitor é direcionado ao *survey* mencionado para uma discussão completa sobre o tópico. Daqui em diante, a decomposição de Benders apresentada ao PCBGR.D é chamada de B&BC.D, que é resumida pelo Algoritmo 4.3.1. Observe que quando um tempo limite é considerado como critério de parada, a abordagem é heurística.

Algoritmo 4.3.1: Algoritmo de decomposição de Benders para o PCBGR.D.

Entrada: Instância $(L \times W)$, (L_k, W_k, V_k, U_k) com $k \in K$, e (ox, oy, l_d, w_d) com $d \in D$.

- 1 Gerar a discretização dos conjuntos normais \bar{X} e \bar{Y} .
 - 2 Montar o Modelo (4.29); /* PMB */
 - 3 Fornecer uma solução inicial ao Modelo (4.29) como discutido na Seção 4.5;
 - 4 **repita**
 - 5 Encontrar uma solução inteira \bar{x}_{kij} ao resolver o Modelo (4.29);
 - 6 Computar $\bar{\mathcal{H}} = \{0, L\} \cup \{i \mid \bar{x}_{kij}\}$, $\bar{\mathcal{V}} = \{0, W\} \cup \{j \mid \bar{x}_{kij}\}$, e
 $\bar{\mathcal{P}} = \{(i_1, i_2, j_1, j_2) \mid i_1 \text{ e } i_2 \in \bar{\mathcal{H}}, j_1 \text{ e } j_2 \in \bar{\mathcal{V}}, i_2 - i_1 \geq$
 $\min_{k \in K \mid \bar{x}_{kij}=1} \{l_k\}, j_2 - j_1 \geq \min_{k \in K \mid \bar{x}_{kij}=1} \{w_k\}\}$;
 - 7 Resolver o Modelo (4.30) com as cardinalidades reduzidas dos conjuntos $\bar{\mathcal{H}}$, $\bar{\mathcal{V}}$
e $\bar{\mathcal{P}}$; /* SPB */
 - 8 **se** Modelo (4.30) *é infactível* **então**
 - 9 Insira o corte combinatório $\sum_{\forall (kij) \mid \bar{x}_{kij}=1} x_{kij} \leq \sum_{\forall (kij)} \bar{x}_{kij} - 1$ no Modelo (4.29);
 - 10 **até** *provar otimalidade*.
- Saída:** Uma solução ótima.
-

4.4 Algoritmo baseado em Programação por Restrições

Um método de solução alternativo é proposto ao PCBGR.D nessa seção. Trata-se de algoritmo *branch-and-cut* baseado em Programação por Restrições (PR) para resolver o PCBGR.D, chamado daqui em diante de B&C-CP. O algoritmo decompõe o problema em duas fases, alternando entre essas fases até atingir um critério de parada. Na fase 1, ele usa um modelo de PLI para encontrar um sub-conjunto promissor de itens. Então, a fase 2 parte de um algoritmo baseado em PR para verificar se o sub-conjunto

encontrado na fase 1 tem uma alocação factível no objeto com defeitos, tal que os itens e defeitos não se sobreponham. Se uma alocação factível não existe, um corte combinatório mínimo é adicionado ao modelo usado na fase 1 para eliminar a família de sub-conjuntos que não conduziriam a uma solução factível.

Para descrever o algoritmo B&C-CP, define-se primeiramente as variáveis de decisão y_k^n :

$$y_k^n = \begin{cases} 1, & \text{se } n \text{ itens do tipo } k \text{ estão presentes no padrão,} \\ 0, & \text{caso contrário.} \end{cases} \quad k \in K, n \in \{0, \dots, U_k\}. \quad (4.31)$$

O modelo de PLI usado na fase 1 do algoritmo é então declarado como:

$$\text{Max} \sum_{k \in K} \sum_{n=0}^{U_k} n V_k y_k^n. \quad (4.32a)$$

$$\text{s.a} \sum_{k \in K} \sum_{n=0}^{U_k} n y_k^n l_k w_k \leq L W - \sum_{d \in D} l_d w_d \quad (4.32b)$$

$$\sum_{n=0}^{U_k} y_k^n = 1 \quad k \in K, \quad (4.32c)$$

$$\sum_{k \in K} \sum_{n=I(k)}^{U_k} y_k^n \leq |K| - 1 \quad \forall I \in \mathcal{I}, \quad (4.32d)$$

$$y_k^n \in \{0, 1\}, \quad \forall k \in K, n = \{0, \dots, U_k\}. \quad (4.32e)$$

A desigualdade (4.32b) garante que a área dos itens selecionados não excede a área total disponível, dada pela área do objeto defeituoso menos a área total dos defeitos em D . As restrições (4.32c) impõem que apenas um $n \in \{0, \dots, U_k\}$ é selecionado para cada tipo de item $k \in K$. Nas restrições (4.32d), \mathcal{I} é a família de todos os possíveis sub-conjuntos I de itens tal que não há alocação factível no objeto defeituoso para todos os itens em I . Para cada $I \in \mathcal{I}$, $I(k)$ denota o número de tipos de itens k in I . Claramente, o número de restrições (4.32d) é exponencial com respeito ao número de itens, e por isso, elas são relaxadas e removidas da formulação à princípio.

Recorre-se a um *software* de otimização de propósito geral para resolver o Modelo (4.32) na fase 1 do algoritmo. O modelo contém apenas um sub-conjunto de restrições (4.32d), dado por aquelas identificadas nas iterações anteriores. A solução ótima desse modelo resulta em um novo sub-conjunto de itens I . Para verificar se tal sub-conjunto tem uma alocação factível no objeto defeituoso, usa-se um algoritmo de PR que ou encontra tal alocação, ou prova que ela não existe. No primeiro caso, o algoritmo

termina com uma solução ótima. Caso contrário, cria-se um corte do tipo (4.32d) usando o sub-conjunto I , que é inserido no Modelo (4.32) como uma *lazy constraint*, e uma nova iteração do algoritmo inicia.

O algoritmo PR funciona como segue. Dado um sub-conjunto de itens I encontrado na fase 1 do algoritmo, define-se as variáveis X_i e Y_i para cada item i em I para indicar a posição (X_i, Y_i) no qual o item i será alocado. Considera-se aqui cópias individuais de itens, isto é, se n itens do tipo k são selecionadas, então I contém n itens de comprimento l_k e largura w_k . Por abuso de notação, diz-se que l_i e w_i são o comprimento e largura do item $i \in I$, respectivamente. Assim, os domínios das variáveis X_i e Y_i são definidos pelos conjuntos \bar{X} e \bar{Y} dados pelas equações (4.26) e (4.27), respectivamente, isto é, $X_i \in \bar{X}$ e $Y_i \in \bar{Y}$. Para cada par de itens i e j em I , adiciona-se as seguintes restrições de não-sobreposição:

$$[X_i + l_i \leq X_j] \text{ ou } [X_j + l_j \leq X_i] \text{ ou } [Y_i + w_i \leq Y_j] \text{ ou } [Y_j + w_j \leq Y_i]. \quad (4.33)$$

Além disso, adiciona-se as restrições de não-sobreposição entre cada item $i \in I$ e cada defeito $d \in D$:

$$[X_i + l_i \leq ox_d] \text{ ou } [ox_d + l_d \leq X_i] \text{ ou } [Y_i + h_i \leq oy_d] \text{ ou } [oy_d + w_d \leq Y_i]. \quad (4.34)$$

O *solver* de PR propaga essas restrições para encontrar uma alocação de todos os itens em I . Quando todas as variáveis X_i e Y_i são designadas a um único valor, o algoritmo verifica se tal alocação pode ser obtida por um corte guilhotinado. Para esse propósito, verifica-se se a alocação tem um corte guilhotinado; se não, o algoritmo rejeita a alocação atual. Se tal corte guilhotinado existe, o algoritmo divide o problema em dois subproblemas, um para cada lado do corte, e recursivamente verifica se cada sub-problema tem uma sequência de cortes guilhotinados, rejeitando a alocação se qualquer sub-problema não pode ser obtido por cortes guilhotinados.

Visto que o algoritmo PR verifica todas as posições factíveis, se ele não encontra uma alocação, ele tem provado que a alocação de itens I não existe. O algoritmo então tenta remover o menor item i de I e verifica se um sub-conjunto $I' \leftarrow I \setminus \{i\}$ tem uma alocação factível. Ele repete esse processo ao remover o menor item $i \in I'$ de I' , isto é, $I' \leftarrow I' \setminus \{i\}$ até o sub-conjunto I' de itens retornar uma alocação factível. O algoritmo então coloca o último item removido de volta ao conjunto I' . Para eliminar o conjunto I' do espaço de soluções do modelo PLI usado na fase 1, usa-se a desigualdade do tipo (4.32d) obtida do conjunto I' . Observe que essa desigualdade também removerá o conjunto original I , como um corte combinatório mínimo. O B&C-CP é sumarizado no Algoritmo 4.4.1, e similarmente ao B&BC-D, quando um tempo limite é considerado

como critério de parada, a abordagem é heurística.

Algoritmo 4.4.1: Algoritmo baseado em PR para o PCBGR_D.

Entrada: Instância $(L \times W)$, (L_k, W_k, V_k, U_k) com $k \in K$, e (ox, oy, l_d, w_d) com $d \in D$.

- 1 Montar o Modelo (4.32) com as restrições (4.32d) relaxadas;
- 2 Fornecer uma solução inicial ao Modelo (4.32), conforme discutido na Seção 4.5;
- 3 **repita**
- 4 Encontrar um sub-conjunto de itens I ao resolver o Modelo (4.32);
- 5 Resolver o algoritmo de PR com as restrições (4.33)–(4.34) e o verificador da restrição de guilhotina ao sub-conjunto de itens I ;
- 6 **se nenhuma alocação é encontrada então**
- 7 Adicionar o corte combinatório de acordo com a restrição (4.32d) no Modelo(4.32);
- 8 **até Provar otimalidade.**

Saída: Um solução ótima.

4.5 Experimentos computacionais

Experimentos computacionais foram realizados para avaliar os dois métodos de solução propostos, nomeadamente o algoritmo de decomposição de Benders da Seção 4.3 e o algoritmo baseado em PR da Seção 4.4. O propósito principal é verificar seus desempenhos de acordo com qualidade de solução – e inclusive para provar otimalidade – e tempo de processamento. Não são reportados resultados computacionais para o Modelo (4.25) da Seção 4.2, porque ele tem números grandes de variáveis e restrições mesmo para instâncias pequenas, o que inviabiliza a abordagem. Essa característica se mantém mesmo estendendo o Procedimento Enumerativo de Variáveis (PEN), do Algoritmo 3.1.1, para o PCBGR_D.

Essa seção é dividida em três partes. Na Seção 4.5.1 discutem-se os descritores (parâmetros) das instâncias de *benchmark* usadas nos experimentos, que foram aleatoriamente geradas por Afsharian, Niknejad e Wäscher (2014) para o caso irrestrito do PCBGR_D — utiliza-se aqui as mesmas instâncias geradas por eles. Na Seção 4.5.2, esse conjunto de instâncias é utilizado para avaliar as abordagens propostas no caso irrestrito (isto é, sem considerar o limitante U_k) ao compará-las ao algoritmo de Programação Dinâmica (PD) de Afsharian, Niknejad e Wäscher (2014), chamado daqui em diante de DPC, isto é, o algoritmo de PD equipado com discretização completa proposto por esses autores. Na Seção 4.5.3 são reportados os resultados para o caso restrito ao usar as instâncias de *benchmark* assumindo uma cópia por tipo de item (isto é, $U_k = 1$). Nesse último experimento, os métodos de solução propostos são comparados entre si, visto que esse autor desconhece outra abordagem da literatura que aborda o PCBGR_D. Conforme previamente definido, o algoritmo de decomposição de Benders é referido como B&BC_D e o algoritmo baseado em PR como B&C-CP. Ambas abordagens foram codificadas em

C++ usando a biblioteca *Concert Studio* do *IBM CPLEX 12.8* e *IBM CPOptimizer 12.8*. Os experimentos foram realizados em um computador de processador Intel Core i7-3770 (3.40GHz), 16 GB de RAM, sob sistema operacional Ubuntu 16.04. Para ambos os métodos de solução, um padrão de corte 2-estágios foi fornecido como solução inicial, obtido pelo modelo 1 de Lodi, Martello e Monaci (2002) aliado ao CPLEX, e como esse modelo não considera o caso de materiais com defeitos, são removidos da solução qualquer item que sobreporia um defeito.

4.5.1 Conjunto de instâncias de *benchmark*

Afsharian, Niknejad e Wäscher (2014) adaptaram (uma versão prévia) do gerador de instâncias de Neidlein, Scholz e Wäscher (2016) para aleatoriamente gerar instâncias para o caso irrestrito do PCBGR_D. Os descritores e correspondentes valores usados para gerar as instâncias são como segue:

- dimensões do objeto ($L \times W$): (75×75) e (125×50) na categoria #1, (150×150) e (225×100) na categoria #2, (300×300) e (450×200) na categoria #3;
- número de tipos de itens (m): 5, 10, 15, 20 ou 25;
- valor dos tipos de itens: versão não-ponderada ($V_k = L_k W_k$);
- dimensões dos tipos de itens ($L_k \times W_k$) são tomados uniformemente dos intervalos $[L/\rho^s, 3L/4]$ e $[W/\rho^s, 3W/4]$, sendo que ρ^s is 6, 8 ou 10 — o procedimento de Neidlein, Scholz e Wäscher (2016) para evitar itens muito longos e finos foi considerado;
- número de defeitos (δ): 1, 2, 3 ou 4;
- dimensões dos defeitos são tomados uniformemente dos intervalos $[L/10, L/6]$ e $[W/10, W/6]$;
- localização dos defeitos são tomados uniformemente dos intervalos $[0, L-l_d]$ e $[0, W-w_d]$.

Note que o PCBGR_D também necessita da definição de $U_k = \lfloor L/L_k \rfloor \lfloor W/W_k \rfloor$, para cada tipo de item $k \in K$ (isto é, um número suficientemente grande) para o caso irrestrito. Essas instâncias são divididas em 90 classes ($= 6 \times 5 \times 3$) que consideram instâncias com 1 até 4 defeitos. Para cada instância, 15 instâncias foram geradas no total de 5400 ($= 90 \times 4 \times 15$) instâncias por Afsharian, Niknejad e Wäscher (2014). Essas definições de classe e categoria foram estabelecidas de acordo com os autores originais. Quando o caso restrito é analisado, define-se o parâmetro U_k como uma cópia, para cada $k \in K$, para gerar o caso mais restrito possível (isto é, $U_k = 1, k \in K$).

4.5.2 Resultados para o caso irrestrito

Os resultados do DPC foram tomados exatamente como apresentados por Afsharian, Niknejad e Wäscher (2014). Esses autores codificaram o DPC em C# e realizaram experimentos computacionais em um computador de processador AMD E-350 (1.60 GHz), 4 GB de RAM, sob Sistema Operacional Windows. De acordo com eles, o DPC fornece soluções ótimas dado que ele se baseia em discretização completa. Eles também propuseram uma abordagem heurística, chamada DPD, que foi capaz de encontrar soluções ótimas para 82,22% das instâncias da categoria #1, em média – eles não reportaram resultados para as outras duas categorias. Nessa seção, as abordagens propostas são comparadas ao DPC usando as instâncias da categoria #1 apenas. As Tabelas 4.2 e 4.3 reportam resultados para o caso irrestrito (isto é, sem considerar o limite U_k). Para cada classe, são reportados o valor médio da função objetivo (OFV) e o tempo médio de processamento (em segundos) para o DPC, B&BC_D e B&C-CP. Foi limitado a 300 segundos a execução das abordagens propostas. Cada entrada dessas tabelas é uma média sobre 60 instâncias. Para os métodos de solução propostos, também são reportados o *GAP* relativo de otimalidade médio em relação ao DPC, calculado como $(OFV_{DPC} - OFV)/(OFV_{DPC} + 10^{-10}) * 100$, sendo que OFV_{DPC} é a entrada do DPC e OFV é a entrada de B&BC_D ou B&C-CP de cada tabela.

Tabela 4.2: Caso irrestrito: resultados para objeto $(L \times W) = (75 \times 75)$ (cada entrada dessa tabela é média sobre 60 instâncias).

Classe	m	ρ^s	DPC		B&BC_D			B&C-CP		
			OFV	Tempo[s](*)	OFV	Tempo[s]	GAP[%]	OFV	Tempo[s]	GAP[%]
1	5	6	3.694,31	72,65	3.694,31	0,05	0,00	3.674,63	25,14	0,53
2	10	6	4.256,01	59,94	4.256,01	15,20	0,00	4.212,38	137,13	1,02
3	15	6	4.566,10	62,35	4.566,10	1,57	0,00	4.419,38	235,33	3,21
4	20	6	4.615,43	60,27	4.605,70	42,80	0,21	4.455,08	252,42	3,47
5	25	6	4.694,01	63,02	4.691,43	56,70	0,05	4.430,31	285,76	5,61
6	5	8	3.683,23	74,83	3.683,23	0,87	0,00	3.662,85	65,36	0,55
7	10	8	4.386,16	61,87	4.372,03	62,00	0,32	4.272,60	187,41	2,58
8	15	8	4.613,23	61,83	4.604,15	78,03	0,19	4.328,43	255,56	6,17
9	20	8	4.883,78	62,82	4.850,66	150,20	0,67	4.555,65	299,83	6,71
10	25	8	4.835,60	62,30	4.792,66	91,92	0,88	4.492,81	298,50	7,08
11	5	10	4.083,56	64,10	4.001,58	16,56	2,00	4.048,31	138,45	0,86
12	10	10	4.710,91	61,27	4.637,40	107,65	1,56	4.465,70	242,43	5,20
13	15	10	4.845,16	61,01	4.678,93	122,56	3,43	4.577,85	280,58	5,51
14	20	10	4.928,65	62,05	4.629,06	155,83	6,07	4.574,81	278,15	7,17
15	25	10	4.968,50	64,54	4.783,45	217,93	3,72	4.503,30	300,00	9,36

(*): Os resultados foram obtidos considerando ambientes computacionais diferentes.

Os resultados apresentados nas Tabelas 4.2 e 4.3 mostram que quanto maior o número de tipos de itens (m) ou do parâmetro ρ_s (isto é, menores os itens), maiores são os tempos de processamento médios e *gap* de otimalidade para B&BC_D e B&C-CP. Como esperado, o número de padrões factíveis tende a aumentar à medida que o número de itens aumenta e/ou o tamanho médio dos itens diminui. Para esses experimentos, o

Tabela 4.3: Caso irrestrito: resultados para objeto $(L \times W) = (112 \times 50)$ (cada entrada dessa tabela é média sobre 60 instâncias).

Classe	m	ρ^s	DPC		B&BC_D			B&C-CP		
			OFV	Tempo[s](*)	OFV	Tempo[s]	GAP[%]	OFV	Tempo[s]	GAP[%]
16	5	6	3.530,41	68,97	3.530,41	0,03	0,00	3.530,41	29,59	0,00
17	10	6	4.339,18	62,17	4.339,18	20,38	0,00	4.299,78	131,64	0,90
18	15	6	4.495,20	72,46	4.495,20	13,40	0,00	4.392,30	230,23	2,28
19	20	6	4.618,16	70,27	4.611,68	59,53	0,14	4.421,30	297,89	4,26
20	25	6	4.686,21	73,41	4.671,68	82,28	0,30	4.423,45	280,06	5,60
21	5	8	3.829,66	75,34	3.829,66	7,83	0,00	3.765,43	116,22	1,67
22	10	8	4.536,90	73,63	4.536,90	26,33	0,00	4.374,10	240,85	3,58
23	15	8	4.792,70	67,52	4.770,53	130,25	0,46	4.545,05	277,38	5,16
24	20	8	4.706,16	71,60	4.681,53	74,36	0,52	4.349,96	293,90	7,56
25	25	8	4.791,00	70,75	4.715,31	121,96	1,57	4.433,38	300,00	7,46
26	5	10	4.138,25	69,00	4.138,25	11,50	0,00	4.081,33	107,98	1,37
27	10	10	4.396,10	73,04	4.374,31	70,63	0,49	4.295,35	189,39	2,29
28	15	10	4.688,12	72,30	4.590,73	149,97	2,07	4.404,06	297,57	6,05
29	20	10	4.895,85	69,78	4.705,36	154,61	3,89	4.489,50	300,00	8,29
30	25	10	4.988,35	70,51	4.856,01	206,26	2,65	4.568,31	300,00	8,42

(*): Os resultados foram obtidos considerando ambientes computacionais diferentes.

tempo de processamento médio foi 74.97 segundos para o B&BC_D e 222.49 segundos para o B&C-CP. Os tempos de processamento médios do DPC variam menos possivelmente devido à discretização completa usada pelo método. Embora os resultados do DPC venham de um ambiente computacional diferente (e menos inferior), nota-se que B&BC_D e B&C-CP são competitivos ao DPC para o caso irrestrito, tanto em qualidade de solução quando em tempo computacional, especialmente quando os valores de m e ρ^s são pequenos. O *gap* médio diz respeito aos valores ótimos fornecidos pelo DPC, sobre todas as instâncias, foi 1.214% para B&BC_D e 4.462% para B&C-CP. A otimalidade foi provada em 79.89% das instâncias com B&BC_D, enquanto B&C-CP obteve soluções provadamente ótimas em 36.33% das instâncias. Assim, B&BC_D supera B&C-CP tanto em qualidade solução e tempo de processamento, para o caso irrestrito.

4.5.3 Resultados para o caso restrito

O caso restrito foi analisado ao fixar uma única cópia por tipo de item ($U_k = 1$, para cada $k \in K$) para o B&BC_D e B&C-CP, usando o conjunto de instâncias de *benchmark* gerado por Afsharian, Niknejad e Wäscher (2014). Nas Tabelas 4.4, 4.5, 4.6 e 4.7 são reportam os resultados para o caso restrito, respectivamente, de acordo com o número de tipos de itens (m), parâmetro ρ^s , número de defeitos (δ) e dimensões do objeto. Note que as instâncias em diferentes categorias tem objetos com diferentes dimensões, conforme definido na Seção 4.5.1. Cada entrada dessas tabelas apresenta o *gap* de otimalidade relativo e os tempos de processamento médios (em segundos) em parênteses, sobre todas as instâncias da mesma categoria. O *gap* relativo dos dois métodos de solução (B&BC_D e B&C-CP) é calculado de acordo com o melhor limitante superior

obtido pelo Problema-Mestre-de-Benders (OFV_{PMB}) ao fim do B&BC_D, computado como $(OFV_{PMB} - OFV)/(OFV_{PMB} + 10^{-10}) * 100$. É válido mencionar que nessa seção são apresentados médias sobre o *gap* de cada instância, enquanto na seção anterior são apresentados o *gap* de médias dos valores OFV, dado que os resultados do DPC foram obtidos de Afsharian, Niknejad e Wäscher (2014).

Como na seção anterior, as Tabelas 4.4 e 4.5 mostram que o *gap* médio e o tempo de processamento médio aumentam quando o número de tipos de itens (m) ou o parâmetro ρ^s também aumenta. No entanto, ao contrário dos experimentos anteriores, B&C-CP supera B&BC_D para o caso restrito. De fato, com menos itens no caso restrito (em relação ao caso restrito), o desempenho da análise de alocação factível da fase 2 do B&C-CP melhora significativamente, visto que o número de cortes combinatórios (possivelmente necessários) é reduzido. Por outro lado, a discretização dos conjuntos normais não desempenha tão bem quando no caso irrestrito, e assim B&BC_D requer mais tempo de processamento B&C-CP, em média.

Tabela 4.4: Caso restrito: resultados de acordo com o número de tipos de itens (cada entrada dessa tabela é média sobre 360 instâncias).

Abordagem	Categoria	Número de tipos de itens (m)				
		5	10	15	20	25
B&BC_D	1	0,00 (0,01)	0,10 (6,12)	0,45 (24,26)	0,79 (43,84)	1,93 (62,61)
	2	0,00 (0,01)	0,24 (8,84)	0,73 (30,63)	5,09 (77,77)	10,38 (102,63)
	3	0,00 (0,01)	0,29 (10,63)	3,28 (44,83)	15,73 (91,86)	25,27 (122,40)
B&C-CP	1	0,00 (0,01)	0,07 (0,70)	0,33 (7,55)	0,62 (51,84)	1,81 (114,48)
	2	0,00 (0,01)	0,18 (0,54)	0,49 (4,67)	1,78 (57,32)	3,64 (120,72)
	3	0,00 (0,01)	0,21 (0,74)	1,16 (6,20)	4,27 (61,42)	7,18 (123,38)

Tabela 4.5: Caso restrito: resultados de acordo com o parâmetro ρ^s (cada entrada dessa tabela é média sobre 600 instâncias).

Abordagem	Categoria	Parâmetro ρ^s		
		6	8	10
B&BC_D	1	0,14 (11,88)	0,60 (29,81)	1,21 (40,42)
	2	1,11 (27,04)	2,98 (46,68)	5,77 (58,21)
	3	2,16 (35,61)	9,80 (59,27)	14,78 (66,95)
B&C-CP	1	0,19 (28,40)	0,55 (35,64)	0,96 (40,71)
	2	0,53 (27,53)	1,25 (38,91)	1,86 (43,51)
	3	0,91 (29,46)	2,67 (40,14)	4,11 (45,45)

Os resultados na Tabela 4.6 indicam que, para ambos os métodos de solução propostos, o *gap* médio e os tempos de processamento reduzem à medida que o número de defeitos aumenta. Isso porque o espaço de soluções se torna menor ao evitar alocações infactíveis dos itens. No entanto, a quantidade de elementos da discretização dos conjuntos

normais tende a aumentar à medida que o número de defeitos também aumenta. Note que o algoritmo DPC tem um comportamento diferente porque sua equação de recursão básica na presença de defeitos conduz a (muito) mais espaços de estados, se comparado com a equação de recursão básica na ausência de defeitos — essa análise também se mantém para outras abordagens baseadas em PD para o problema. Na Tabela 4.7 são reportados os resultados dessa seção agregados de acordo com a categoria. A otimalidade foi provada em 77.50% das instâncias com B&BC_D, enquanto B&C-CP obteve soluções provadamente ótimas em 75.19% das instâncias. O subproblema do B&BC_D (BSP), em média, adicionou duas centenas de cortes, enquanto que o sub-problema do B&C-CP adicionou aproximadamente mil corte, aos seus respectivos problemas-mestre. Para ambos os casos, em média, o tempo gasto nos subproblemas foi inferior a 10% do tempo total de processamento.

Tabela 4.6: Caso restrito: resultados de acordo com o número de defeitos (cada entrada dessa tabela é média sobre 450 instâncias).

Abordagem	Categoria	Número de defeitos (δ)			
		1	2	3	4
B&BC_D	1	1,30 (45,24)	0,76 (31,19)	0,39 (21,81)	0,16 (11,23)
	2	4,98 (60,61)	4,10 (50,69)	2,76 (36,87)	1,30 (27,73)
	3	8,74 (69,35)	10,33 (58,03)	9,46 (49,62)	7,12 (38,79)
B&C-CP	1	0,86 (41,73)	0,61 (37,49)	0,44 (32,70)	0,36 (27,73)
	2	1,80 (42,67)	1,41 (39,51)	0,98 (34,76)	0,67 (29,66)
	3	2,68 (45,10)	2,94 (40,41)	2,62 (36,26)	2,01 (31,62)

Tabela 4.7: Caso restrito: resultados de acordo com a categoria (cada entrada dessa tabela é média sobre 1800 instâncias).

Abordagem	Categoria		
	1	2	3
B&BC_D	0,65 (27,37)	3,29 (43,98)	8,91 (53,95)
B&C-CP	0,57 (34,91)	1,21 (36,65)	2,56 (38,25)

Em síntese, os principais resultados dessa seção indicam que: (i) o B&C-CP supera o B&BC_D no caso restrito; (ii) o B&BC_D parece ser mais dependente do número de tipos de itens (m) do que B&C-CP; e (iii) B&C-CP parece ser mais dependente do número total de itens ($\sum_{k \in K} U_k$) do que B&BC_D. Nota-se que para outros cenários o aumento do número de defeitos não conduz necessariamente a cenários mais fáceis para o B&BC_D, especialmente quando as dimensões do objeto defeituoso significativamente aumentam, visto que a quantidade de elementos na discretização dos conjuntos normais e a quantidade de soluções simétricas tendem a aumentar rapidamente. Na Figura 4.7

ilustram-se soluções ótimas para duas instâncias, arbitrariamente escolhidas da categoria 3.

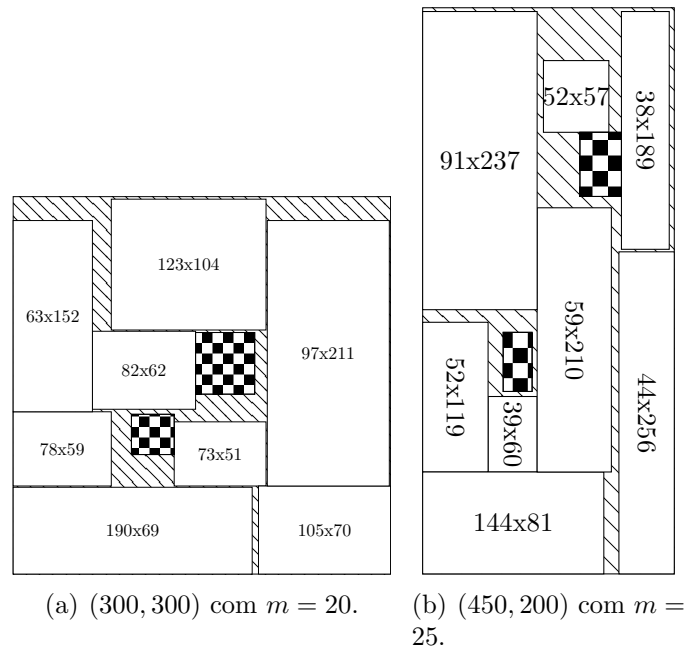


Figura 4.7: Soluções ótimas para duas instâncias da categoria 3.

4.6 Comentários finais do capítulo

O Problema de Corte Bidimensional Guilhotinado e Restrito de Objeto com Defeitos (PCBGR_D) comumente surge em ambientes industriais que lidam com objetos heterogêneos sobre os quais itens devem ser obtidos por meio de cortes de guilhotina. Apesar de seu potencial para aplicabilidade prática, a literatura contém poucos estudos que abordaram diretamente problemas de corte com materiais defeituosos.

Nesse capítulo estendeu-se a formulação compacta de PLI, originalmente proposta para o PCBGR, para lidar com o PCBGR_D. Nota-se que esse autor desconhece outra formulação matemática para esse problema. Além disso, o algoritmo de decomposição de Benders, também proposto para o PCBGR, foi estendido para lidar com o PCBGR_D. Também foi proposto um algoritmo baseado em PR como método de solução para resolver instâncias maiores.

Os experimentos computacionais usando instâncias de *benchmark* da literatura mostraram que as abordagens propostas são competitivas a outra abordagem da literatura para a versão irrestrita do problema. Para o caso de instâncias restritas, o B&C-CP supera o B&BC_D, ambos em qualidade de solução e tempo de processamento.

Conforme discutido anteriormente, basta-se adicionar um tipo de item $w \times l$ para cada tipo de item $l \times w$ da instância original, considerando agregadas suas cópias quando no caso restrito, para que as abordagens baseadas em grelha lidem com o caso

de rotação de itens. No próximo capítulo, o PCBGR volta a ser o objeto de análise, com um paradigma de modelagem distinto, isto é, a combinação de retângulos em vez da discretização do objeto, a fim de representar instâncias maiores do problema no contexto de *solvers* de propósito geral, assim como lidar com o caso de padrões d -estágios.

Capítulo 5

Problema de corte bidimensional guilhotinado e restrito: propostas de formulações matemáticas com o conceito *bottom-up*

Neste capítulo propõem-se novas formulações não-lineares e lineares ao PCBGR nas Seções 5.1 e 5.2, respectivamente, baseadas no conceito *bottom-up*. Restrições adicionais são desenvolvidas para limitar o número de estágios guilhotinados dos padrões de cortes nos modelos na Seção 5.3. A efetividade dos modelos propostos é avaliada por meio de experimentos computacionais com instâncias de *benchmark* na Seção 5.4, e comparadas ao modelo de Furini, Malaguti e Thomopoulos (2016).

Objetivos do capítulo:

- Propor uma formulação pseudopolinomial de Programação Não-Linear Inteira Mista (PNLIM), e sua formulação de PLIM equivalente ao PCBGR, baseadas no conceito *bottom-up*;
- Propor uma formulação compacta de PNLIM, e sua formulação de PLIM equivalente ao PCBGR, baseadas no conceito *bottom-up*;
- Desenvolver inequações não-lineares, e suas versões lineares equivalentes, para restringir os modelos propostos a padrões de cortes d -estágios.

Parte dos resultados deste capítulo foram recentemente publicados:

★ Martin, M.; Morabito, R.; Munari, P. *A bottom-up packing approach for modeling the constrained two-dimensional guillotine placement problem*. Computers & Operations Research, 2019. DOI: <https://doi.org/10.1016/j.cor.2019.104851>.

5.1 Formulações matemáticas não-lineares

Primeiramente, destaca-se o princípio de construções horizontais e verticais proposto por Wang (1983). Considere dois itens de dimensões $l_1 \times w_1$ e $l_2 \times w_2$. A construção horizontal desses retângulos fornece o retângulo maior (ou subpadrão) de dimensão $(l_1 + l_2) \times \max\{w_1, w_2\}$, enquanto a construção vertical fornece o retângulo maior de dimensão $\max\{l_1, l_2\} \times (w_1 + w_2)$. A Figura 5.1 ilustra as construções horizontal e vertical de dois itens. De antemão, destaca-se que os modelos propostos neste capítulo não distinguem a alocação relativa dos itens e/ou subpadrões menores que constroem um subpadrão maior. Esses modelos consideram apenas as dimensões dos itens e dos subpadrões. De fato, todos os possíveis padrões de dimensão $(l_1 + l_2) \times w_1$ que contenham os itens $l_1 \times w_1$ e $l_2 \times w_2$ são equivalentes ao subpadrão da Figura 5.1(b); por exemplo, um subpadrão equivalente teria o item $l_2 \times w_2$ à esquerda do item $l_1 \times w_1$. Note que as áreas hachuradas nessas figuras não serão ocupadas por nenhum outro item em construções futuras, isto é, elas são desperdício.

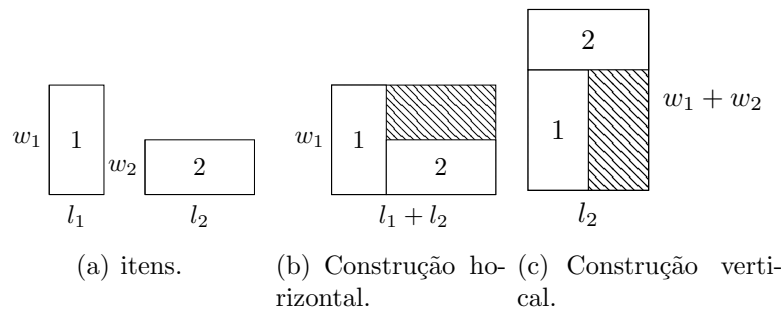


Figura 5.1: Exemplo de construções horizontal e vertical.

Para descrever os modelos, é relevante primeiro saber quantos retângulos (isto é, subpadrões) podem ser construídos em uma abordagem *bottom-up*. Seja o parâmetro \hat{n} o número de itens em certa solução ótima de uma instância ordinária do PCBGR. Para tal solução, sem perda de generalidade, pode-se considerar $\hat{n} - 1$ cortes guilhotinados em uma abordagem *top-down*, ou $\hat{n} - 1$ subpadrões gerados sucessivamente por construções horizontais e/ou verticais em uma abordagem *bottom-up*. Por exemplo, note que o padrão de corte da Figura 5.1(b) apresenta um corte/uma construção e contém dois itens, enquanto que o padrão de corte da Figura 5.6 apresenta três cortes/três construções e contém quatro itens. Dado que o parâmetro \hat{n} é difícil de ser conhecido previamente à resolução da instância, seja o parâmetro \bar{n} um limitante superior ao parâmetro \hat{n} . Por exemplo, um limitante superior simples \bar{n} é a soma de todos os itens permitidos de serem manufaturados ($\sum_{i \in I} u_i$), e um mais apertado é o valor ótimo de uma mochila unidimensional em que o peso de cada tipo de item $i \in I$ é sua área ($l_i w_i$), conforme definido pelo Modelo (5.1).

¹O conjunto de tipos de itens é denotado neste capítulo e no Capítulo 6 por $I = \{1, \dots, m\}$, diferentemente do conjunto K do Capítulos 3 e 4.

$$\bar{n} = \mathbf{Max} \sum_{i \in I} q_i, \mathbf{s.a.} \sum_{i \in I} (l_i w_i) q_i \leq LW, q_i \leq u_i, \quad q_i \in \mathbb{Z}_+, i \in I. \quad (5.1)$$

Nessa seção desenvolvem-se duas formulações de Programação Não-Linear Inteira Mista (PNLIM) que consideram até $\bar{n} - 1$ construções de itens para o PCBGR. A diferença principal entre os dois modelos propostos está no uso de uma estrutura de árvore binária explícita ou implícita. Observe que qualquer solução de uma instância do PCBGR pode ser representada por uma árvore binária cuja raiz é o objeto ou o maior retângulo capaz de ser contido por ele, dado que qualquer solução parcial é construída a partir de dois retângulos menores. O modelo da Seção 5.1.1 formula o problema com o conhecimento prévio de uma árvore binária, que pode conduzir a todos os possíveis padrões de corte. O modelo da Seção 5.1.2 formula o problema com um conjunto adicional de variáveis que, sem perda de otimalidade, implicitamente gera a estrutura da árvore binária das soluções.

5.1.1 Formulação não-linear 1: abordagem *bottom-up* hierárquica

O primeiro modelo considera que cada retângulo, se existir, corresponde a um dos seguintes: uma construção horizontal, uma construção vertical, ou uma cópia de itens. A Figura 5.2 ilustra tal compreensão e destaca as variáveis relacionadas que serão definidas no modelo. Esse modelo é chamado abordagem *bottom-up* hierárquica, porque os retângulos pequenos que podem construir um retângulo maior são previamente definidos em uma árvore binária, na qual os nós representam os retângulos.

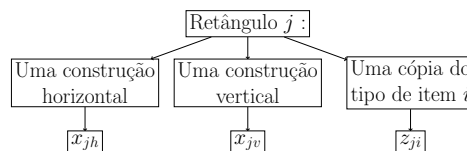


Figura 5.2: Compreensão do modelo hierárquico.

A árvore binária é gerada pelo Algoritmo 5.1.1, que segue uma abordagem de corte *top-down* dado que ele inicia a partir do limitante superior \bar{n} para o objeto. Cada corte guilhotinado em um subpadrão maior sempre gera dois subpadrões menores. Assim, em um caso extremo, um desses dois subpadrões menores consiste de um único item, enquanto o outro fornece o resto dos itens para o subpadrão maior. Por outro lado, no outro caso extremo, cada um dos dois subpadrões pequenos fornece metade dos itens para o subpadrão maior. O Algoritmo 5.1.1 cria o nó $j = 1$, define que ele pode conter até \bar{n} itens (capacidade do nó), e insere-o em uma fila. Então ele seleciona e remove o primeiro nó j da fila, cria seus dois nós-filho j^- e j^+ se o nó j tem capacidade maior que 1 item, adiciona os nós j^- e j^+ na árvore binária ao definir a tupla (j, j^-, j^+) , e os insere na fila. O algoritmo itera até que a fila esteja vazia. Ao criar os nós-filho do nó

j , sem perda de generalidade é arbitrariamente definido que: (i) o nó-filho da esquerda j^- é capaz de conter a capacidade do nó $j - 1$ itens; (ii) o nó-filho da direita j^+ é capaz de conter a $\lfloor \text{“capacidade do nó } j \text{”} / 2 \rfloor$ itens. Em outras palavras, considera-se que cada nó-filho da direita j^+ forneça ao nó j : (i) apenas um item, como mencionado no primeiro caso extremo mencionado acima; e, (ii) metade dos itens, como no outro caso extremo. O número total de nós na árvore é representado pelo parâmetro \mathcal{N} . Seja $J = \{1, \dots, \mathcal{N}\}$ o conjunto de todos os nós da árvore binária. O Algoritmo 5.1.1 também classifica cada nó $j \in J$ nos conjuntos disjuntos J_A, J_B, J_C e \bar{J} , o que é útil para descrever o modelo:

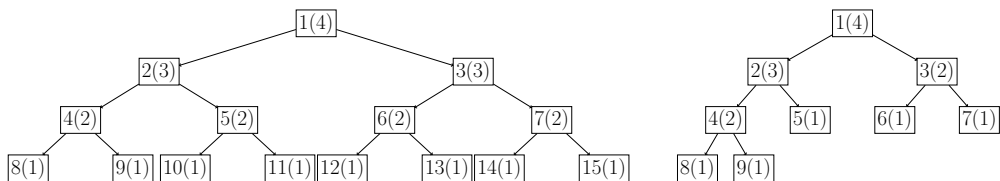
1. retângulo $j \in J_A$ se seus dois filhos são nós não-folha;
2. retângulo $j \in J_B$ se seu filho da esquerda é nó não-folha, e seu filho da direita é um nó-folha;
3. retângulo $j \in J_C$ se seus dois filhos são nós-folhas;
4. retângulo $j \in \bar{J}$ se ele é um nó-folha.

Note que o parâmetro \mathcal{N} seria igual a $2^{\bar{n}} - 1$ supondo que os nós-filho j^- e j^+ sempre tivessem a capacidade do nós j menos 1, isto é, assume-se enumeração completa. A Figura 5.3 ilustra essas duas formas de enumerar a árvore binária, isto é, a enumeração completa e a enumeração do Algoritmo 5.1.1, sendo que cada nó é denotado por seu número de identificação e sua capacidade entre parênteses. Por exemplo, em ambas árvores binárias, os nós $j^- = 2$ e $j^+ = 3$ são filhos de $j = 1$, enquanto os nós $j^- = 4$ e $j^+ = 5$ são filhos de $j = 2$; no entanto, os nós $j = 10, \dots, 15$ existem apenas na árvore binária gerada pela enumeração completa. A enumeração do Algoritmo 5.1.1 é sem perda de generalidade e drasticamente reduz o número de nós na árvore. Diferentemente da enumeração completa, ela também elimina soluções simétricas porque não permite que o nó-filho da direita j^+ forneça mais do que a metade do itens ao nó j , diante de sua máxima capacidade. Assume-se arbitrariamente que a capacidade do nó-filho da esquerda j^- é sempre maior que a capacidade do nó-filho da direita j^+ . Quanto aos conjunto, note que, na Figura 5.3(b), $J_A = \{1\}$, $J_B = \{2\}$, $J_C = \{3, 4\}$ e $\bar{J} = \{5, 6, 7, 8, 9\}$. Observe que os conjuntos J_A, J_B e J_C contém apenas nós não-folha, enquanto o conjunto \bar{J} contém apenas nós-folha. Além disso, esses quatro conjuntos são disjuntos, e sua união resulta no conjunto de todos os nós $J = J_A \cup J_B \cup J_C \cup \bar{J}$. O conjunto de todos os nós não-folha é $J \setminus \bar{J}$.

Para ilustrar como a árvore binária proposta pelo Algoritmo 5.1.1 pode representar padrões de corte sem perda de generalidade, considere os exemplos da Figura 5.4, sendo que cada nó cinza contém um único item; observe que esses exemplos também são para um limitante superior de $\bar{n} = 4$ itens. Por exemplo, supondo que o padrão ótimo, que é representado pelo nó $j = 1$, contém 4 itens exatamente (isto é, quando $\hat{n} = 4$), há duas opções de construções: (i) seu nó-filho da esquerda é um subpadrão com três itens,

Algoritmo 5.1.1: Árvore binária para abordagem *bottom-up* hierárquica.**Entrada:** Limitante superior \bar{n} .

- 1 $r \leftarrow 1$; $J_A \leftarrow \{\}$; $J_B \leftarrow \{\}$; $J_C \leftarrow \{\}$; $\bar{J} \leftarrow \{\}$;
 - 2 Criar nó raiz r com capacidade de \bar{n} itens;
 - 3 Iniciar fila $T = \{r\}$;
 - 4 **repita**
 - 5 Selecionar e remover o primeiro nó t da fila T ;
 - 6 **se** capacidade do nó $t > 1$ **então**
 - 7 Criar novo nó $r + 1$ de capacidade igual a “capacidade do nó t ”-1;
 - 8 Criar novo nó $r + 2$ de capacidade igual a \lfloor “capacidade do nó t ”/2 \rfloor ;
 - 9 Definir o nó $r + 1$ como nó-filho esquerdo (j^-) do nó j , e definir o nó $r + 2$ como nó-filho direito (j^+) do nó j , isto é, definir a tupla (j, j^-, j^+) na árvore binária;
 - 10 **se** ambos os nós $r + 1$ e $r + 2$ têm capacidade maior que 1 **então**
 - 11 $J_A \leftarrow J_A \cup \{t\}$;
 - 12 **se** a capacidade do nó $r + 1$ é maior que 1 e a capacidade do nó $r + 2$ é igual a 1 **então**
 - 13 $J_B \leftarrow J_B \cup \{t\}$;
 - 14 **se** ambos os nós $r + 1$ e $r + 2$ têm capacidade igual a 1 **então**
 - 15 $J_C \leftarrow J_C \cup \{t\}$;
 - 16 Inserir os nós $r + 1$ e $r + 2$ no fim da fila T ;
 - 17 $r \leftarrow r + 2$;
 - 18 **senão**
 - 19 $\bar{J} \leftarrow \bar{J} \cup \{t\}$;
 - 20 $\mathcal{N} \leftarrow r$;
 - 21 **até** fila T esteja vazia
- Saída:** Tuplas (j, j^-, j^+) da árvore binária, parâmetro \mathcal{N} , e conjuntos J_A , J_B , J_C e \bar{J} .

Figura 5.3: Esquemas de enumeração da árvore binária para um exemplo com $\bar{n} = 4$.

no qual cada um dos nós 5, 8 e 9 contém um único item, e seu nó-filho da direita $j^+ = 3$ também contém um único item, como ilustrado pela Figura 5.4(a); ou, (ii) seus nós-filhos $j^- = 2$ e $j^+ = 3$ são subpadrões com dois itens cada, no qual cada um dos nós 4, 5, 6 e 7 contém um único item, como ilustrado na Figura 5.4(b). Em contraste, supondo apenas dois itens no padrão ótimo (isto é, quando $\hat{n} = 2$), teria-se que o nó $j = 1$ é um subpadrão formado pelos nós $j^- = 2$ e $j^+ = 3$, que contém um único item cada, como ilustrado na Figura 5.4(c). Apesar das capacidades do nó $j = 3$ na Figura 5.4(a), do nó $j = 4$ na Figura 5.4(b) e dos nós $j = 2$ e $j = 3$ na Figura 5.4(c) serem maiores que um, cada um desses nós contém apenas um único item para representar os respectivos padrões de corte. Em outras palavras, cada nó-filho de um nó que representa/contém um único item nos exemplos da Figura 5.4 não é necessário para o respectivo padrão de corte; no entanto, esses nós são necessários na árvore binária para representar todos os outros possíveis padrões de corte. Por essa razão, foram mantidos os nós 6 e 7 na Figura 5.4(a), os nós 8 e 9 na Figura 5.4(b), e os nós 4 até 9 na Figura 5.4(c). Dessa forma, quando um no-pai j é definido, algumas decisões devem ser tomadas sobre seus filhos j^- e j^+ . Se o nó-pai for uma construção horizontal ou vertical (isto é, um subpadrão), então os nós-filhos devem ser subpadrões ou itens. No entanto, se o nó-pai representar um item, seus nós-filhos existem na árvore binária, mas seus respectivos valores nas variáveis do modelo a seguir serão zero. Observe que outros padrões de corte são também possíveis, incluindo aquele cujo nó $j = 1$ representa um item e nenhuma construção é necessária (isto é, quando $\hat{n} = 1$).

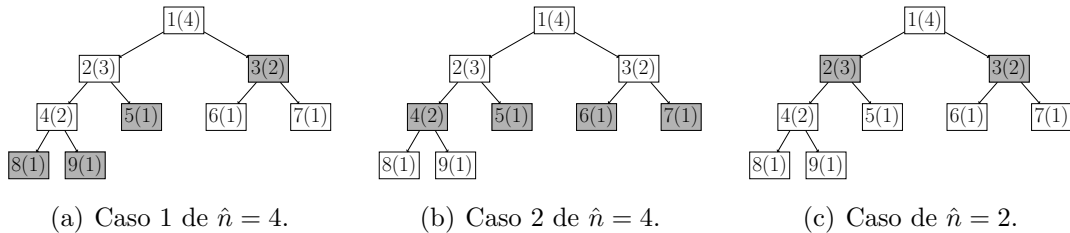


Figura 5.4: Representações de padrões de corte na árvore binária proposta para um exemplo com $\bar{n} = 4$ (cada nó cinza contém um único item).

Seja $O = \{h, v\}$ o conjunto de todas as orientações possíveis para construir um nó não-folha, sendo que h e v são construções horizontal e vertical, respectivamente. Sobre as variáveis de decisão, há quatro conjuntos de variáveis no modelo, que são definidas em (5.2) até (5.5). Note que apenas os nós não-folha $j \in J \setminus \bar{J}$ podem ser construídos de itens ou subpadrões e, assim, existem variáveis para definir seus correspondentes comprimentos e larguras. Em oposição, qualquer nó $j \in J$ pode ser uma cópia de qualquer tipo de item. A Figura 5.5, a seguir, esboça os casos possíveis para um nó $j \in J$ na abordagem *bottom-up* hierárquica: (i) cópia de um tipo de item; (ii) combinação de dois subpadrões menores; (iii) combinação de um subpadrão menor e de uma cópia de tipo de item; ou, (iv) duas cópias de tipos de itens. Esses casos estão de acordo com a categoria

do nó $j \in J$.

$$z_{ji} = \begin{cases} 1, & \text{se o nó } j \text{ contém uma cópia do tipo de item } i, \\ 0, & \text{caso contrário.} \end{cases} \quad j \in J, i \in I. \quad (5.2)$$

$$x_{jo} = \begin{cases} 1, & \text{se o nó } j \text{ é construído com orientação } o, \\ 0, & \text{caso contrário.} \end{cases} \quad j \in J \setminus \bar{J}, o \in O. \quad (5.3)$$

$$L_j: \text{ é o comprimento do nó } j, \quad j \in J \setminus \bar{J}. \quad (5.4)$$

$$W_j: \text{ é a largura do nó } j, \quad j \in J \setminus \bar{J}. \quad (5.5)$$

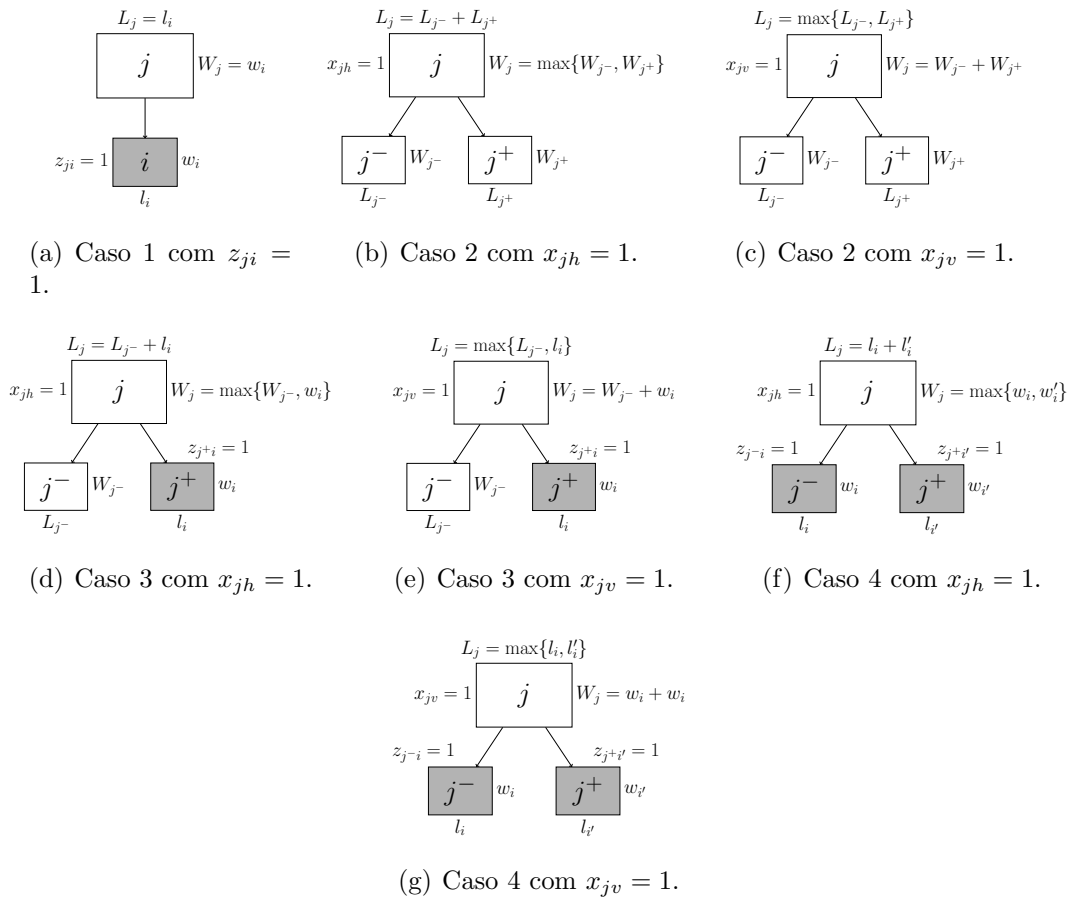


Figura 5.5: Modelo hierárquico. (a) Qualquer nó $j \in J$ pode conter uma cópia de um tipo de item. Um nó não-folha $j \in J \setminus \bar{J}$ pode ser construído de: (b) e (c) dois subpadrões menores (se $j \in J_A$); (d) e (e) um subpadrão menor e uma cópia de um tipo de item (se $j \in J_A \cup J_B$); ou, (f) e (g) duas cópias de tipos de item.

Tendo definidos todas as variáveis e parâmetros, o Modelo (5.6) declara uma formulação pseudopolinomial de PNLIM para o PCBGR. Essa formulação é não-compacta, pois suas variáveis e restrições são definidas em termos dos nós da estrutura de árvore binária gerada pelo Algoritmo 5.1.1.

$$\mathbf{Max} \sum_{j \in J} \sum_{i \in I} v_i z_{ji}, \quad (5.6a)$$

s.a

$$\sum_{j \in J} z_{ji} \leq u_i, \quad i \in I, \quad (5.6b)$$

$$\sum_{i \in I} z_{ji} + \sum_{o \in O} x_{jo} \leq 1, \quad j \in J \setminus \bar{J}, \quad (5.6c)$$

$$\sum_{i \in I} z_{ji} \leq 1, \quad j \in \bar{J}, \quad (5.6d)$$

$$\sum_{o \in O} x_{j-o} + \sum_{i \in I} z_{j-i} + \sum_{o \in O} x_{j+o} + \sum_{i \in I} z_{j+i} = 2 \sum_{o \in O} x_{jo}, \quad j \in J_A, \quad (5.6e)$$

$$\sum_{o \in O} x_{j-o} + \sum_{i \in I} z_{j-i} + \sum_{i \in I} z_{j+i} = 2 \sum_{o \in O} x_{jo}, \quad j \in J_B, \quad (5.6f)$$

$$\sum_{i \in I} z_{j-i} + \sum_{i \in I} z_{j+i} = 2 \sum_{o \in O} x_{jo}, \quad j \in J_C, \quad (5.6g)$$

$$L_j = \sum_{i \in I} l_i z_{ji} + (L_{j^-} + L_{j^+}) * x_{jh} + \max\{L_{j^-}, L_{j^+}\} * x_{jv}, \quad j \in J_A, \quad (5.6h)$$

$$L_j = \sum_{i \in I} l_i z_{ji} + (L_{j^-} + \sum_{i \in I} l_i z_{j+i}) * x_{jh} + \max\{L_{j^-}, \sum_{i \in I} l_i z_{j+i}\} * x_{jv}, \quad j \in J_B, \quad (5.6i)$$

$$L_j = \sum_{i \in I} l_i z_{ji} + (\sum_{i \in I} l_i z_{j-i} + \sum_{i \in I} l_i z_{j+i}) * x_{jh} + \max\{\sum_{i \in I} l_i z_{j-i}, \sum_{i \in I} l_i z_{j+i}\} * x_{jv}, \quad j \in J_C, \quad (5.6j)$$

$$W_j = \sum_{i \in I} w_i z_{ji} + \max\{W_{j^-}, W_{j^+}\} * x_{jh} + (W_{j^-} + W_{j^+}) * x_{jv}, \quad j \in J_A, \quad (5.6k)$$

$$W_j = \sum_{i \in I} w_i z_{ji} + \max\{W_{j^-}, \sum_{i \in I} w_i z_{j+i}\} * x_{jh} + (W_{j^-} + \sum_{i \in I} w_i z_{j+i}) * x_{jv}, \quad j \in J_B, \quad (5.6l)$$

$$W_j = \sum_{i \in I} w_i z_{ji} + \max \left\{ \sum_{i \in I} w_i z_{j-i}, \sum_{i \in I} w_i z_{j+i} \right\} * x_{jh} + \left(\sum_{i \in I} w_i z_{j-i} + \sum_{i \in I} w_i z_{j+i} \right) * x_{jv}, \quad j \in J_C, \quad (5.6m)$$

$$z_{ji} \in \{0, 1\}, \quad j \in J, i \in I, \quad (5.6n)$$

$$x_{jo} \in \{0, 1\}, \quad j \in J \setminus \bar{J}, o \in O, \quad (5.6o)$$

$$0 \leq L_j \leq L, \quad j \in J \setminus \bar{J}, \quad (5.6p)$$

$$0 \leq W_j \leq W, \quad j \in J \setminus \bar{J}. \quad (5.6q)$$

A função objetivo (5.6a) consiste em maximizar a soma total dos itens selecionados. As restrições (5.6b) garantem que cada tipo de item $i \in I$ é selecionado até seu número máximo de cópias permitido de ser manufaturado. As restrições (5.6c) limitam que um nó não-folha (isto é, em J_A , J_B ou J_C) pode conter, no máximo, uma cópia de algum tipo de item, ou representar uma construção horizontal ou vertical, conforme ilustrado na Figura 5.5. As restrições (5.6d) limitam que um nó-folha (isto é, $j \in \bar{J}$) pode ser, no máximo, uma cópia de um tipo de item.

As restrições (5.6e), (5.6f) e (5.6g) associam as decisões de nós pais aos seus correspondentes nós filhos, de acordo com as tuplas (j, j^-, j^+) . Nessas restrições, se um nó pai não é um subpadrão (isto é, quando $\sum_{o \in O} x_{jo} = 0$), então o *right-hand-side* (rhs) da correspondente restrição é zero e, assim, as variáveis de seus filhos no *left-hand-side* (lhs) não podem assumir o valor de 1. Por outro lado, se um nó pai é um subpadrão (isto é, quando $\sum_{o \in O} x_{jo} = 1$), então o rhs da correspondente restrição é igual a 2 e, assim, seus dois filhos devem ser cópias de itens, construções horizontais, ou construções verticais, de acordo com a categoria de cada nó pai. Por exemplo, quando o rhs é igual a 2, as restrições (5.6e) consideram que o filho da esquerda j^- e o filho da direita j^+ do nó $j \in J_A$ podem ser uma cópia de qualquer tipo de item ou algum tipo de construção. Para o mesmo caso, as restrições (5.6f) permitem que o filho da direita j^+ do nó j pode ser, no máximo, uma cópia de um tipo de item, no qual é também a opção para os dois filhos de um nó $j \in J_C$ nas restrições (5.6g). Note que não é possível satisfazer as restrições (5.6e), (5.6f) e (5.6g) com apenas o filho da esquerda ou o filho da direita do nó j se o rhs é igual a 2, devido as restrições (5.6c) e (5.6d).

As restrições (5.6h), (5.6i) e (5.6j) definem a variável L_j dos nós não-folha $j \in J \setminus \bar{J}$, de acordo com as tuplas (j, j^-, j^+) . Nessas restrições, a variável L_j assume um comprimento l_i do tipo de item $i \in I$ quando o nó j é definido ser um tipo de item (isto é, quando $\sum_{i \in I} z_{ji} = 1$). Caso contrário, se um nó j é definido ser uma construção horizontal (isto é, quando $x_{jh} = 1$), a variável L_j assume a soma dos comprimentos de seus dois filhos, que são dados pelas variáveis L_{j^-} e L_{j^+} para filhos não-folha nas restrições (5.6h), e por expressões $\sum_{i \in I} l_i z_{j-i}$ e $\sum_{i \in I} l_i z_{j+i}$ para os filhos folhas nas restrições (5.6j). Nas restrições (5.6i), há um caso intermediário, no qual o filho da esquerda j^- é um nó não-folha com comprimento denotado por L_{j^-} , e o filho da direita j^+ é um nó folha

com comprimento denotado por $\sum_{i \in I} l_i z_{j+i}$. Por último, se um nó j é definido ser uma construção vertical (isto é, quando $x_{jv} = 1$), a variável L_j assume o maior comprimento entre os comprimentos de seus dois filhos, usando a função $\max\{.\}$ e as mesmas variáveis dos casos de construções horizontais de filhos folha e não-folha.

As restrições (5.6k), (5.6l) e (5.6m) atuam de forma similar as restrições (5.6h), (5.6i) e (5.6j), mas na definição da variável W_j , usando as tuplas (j, j^-, j^+) . Por uma questão de clareza, apresenta-se uma descrição completa a seguir. Nessas restrições, W_j assume o comprimento w_i de um tipo de item $i \in I$, quando o nó j é definido ser um tipo de item (isto é, quando $\sum_{i \in I} z_{ji} = 1$). No entanto, se o nó j é definido ser uma construção horizontal (isto é, quando $x_{jh} = 1$), W_j assume a maior largura entre as larguras de seus dois filhos, que são as variáveis W_{j^-} e W_{j^+} para os filhos não-folha nas restrições (5.6k), e as expressões $\sum_{i \in I} w_i z_{j-i}$ e $\sum_{i \in I} w_i z_{j+i}$ para os filhos folhas nas restrições (5.6m). Nas restrições (5.6l), há um caso intermediário, no qual o filho da esquerda j^- é um nó não-folha com largura denotada por uma variável W_{j^-} , e o filho da direita j^+ é um nó folha com largura denotada por $\sum_{i \in I} w_i z_{j+i}$. Por último, se um nó j é definido ser uma construção vertical (isto é, quando $x_{jv} = 1$), a variável W_j assume a soma das larguras de seus dois filhos nas restrições (5.6k), (5.6l) e (5.6m), que são representados como no caso de construções verticais de filhos folha e não-folha.

As restrições (5.6n), (5.6o), (5.6p) e (5.6q) definem o domínio das variáveis de decisão. O Modelo (5.6) é chamado daqui em diante Modelo Não-linear 1. Ele é uma formulação não-linear devido aos produtos de variáveis e o uso da função $\max\{.\}$ nas restrições (5.6h) até (5.6m). Observe que esse modelo não requer que os dados de entrada (isto é, L , W , l_i e w_i , para cada $i \in I$) sejam inteiros.

Por questão de clareza, apresenta-se um exemplo ilustrativo para o PCBGR a seguir. A Tabela 5.1 mostra uma instância não-ponderada (isto é, $v_i = l_i w_i$) para um objeto de dimensões $(L \times W) = (35 \times 30)$ e $m = 5$ tipos de itens. A Tabela 5.2 detalha uma solução ótima de tal instância obtida ao resolver Modelo Não-linear 1 usando o *IBM CPLEX CP Optimizer v12.8*, e mostra apenas os valores das variáveis que assumiram valores não-nulos, o que é esboçado na Figura 5.6. Os nós folha de Modelo Não-linear 1 são representados pelo símbolo “—”, e o número 0 significa que todas as variáveis x_{jo} e z_{ji} para tal nó j são iguais a zero. Observe que o Modelo (5.1) fornece um limitante superior $\bar{n} = 4$ para essa instância. Particularmente, a árvore binária do Modelo Não-linear 1 é aquela da Figura 5.3(b), dado que o limitante superior é o mesmo. O Seção 5.5 discute o Algoritmo 5.5.1, que é um pseudo-código para gerar o padrão de corte a partir de uma solução do Modelo Não-linear 1.

Tabela 5.1: Exemplo ilustrativo para o PCBGR.

	Tipo de item $i \in I$				
	1	2	3	4	5
l_i	15	30	5	15	20
w_i	15	15	30	20	20
u_i	3	1	4	1	1

j	j^-, j^+	$(L \times W)$	x_{jo} ou z_{ji}
1	2, 3	(35×30)	$x_{1h} = 1$
2	4, 5	(30×30)	$x_{2v} = 1$
3	6, 7	—	$z_{33} = 1$
4	8, 9	(30×15)	$x_{4h} = 1$
5	—	—	$z_{52} = 1$
6	—	—	0
7	—	—	0
8	—	—	$z_{81} = 1$
9	—	—	$z_{91} = 1$

Tabela 5.2: Solução ótima do exemplo ilustrativo da Tabela 5.1 para o Modelo Não-linear 1.

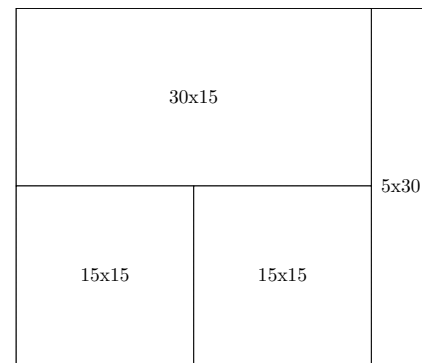


Figura 5.6: Padrão de corte da Tabela 5.2.

5.1.2 Formulação não-linear 2: abordagem *bottom-up* implícita

O segundo modelo considera que cada retângulo, se existir, é sempre um subpadrão (isto é, um retângulo construído) de itens e/ou outros subpadrões. A Figura 5.7 ilustra essa compreensão e destaca as correspondentes variáveis usadas no modelo. É importante destacar que o Modelo Não-linear 1 funciona de forma diferente dado que ele considera que cada retângulo é um subpadrão (construído horizontal ou verticalmente) ou uma cópia de algum tipo de item. Esse segundo modelo é chamado de abordagem *bottom-up* implícita, porque os retângulos menores que podem construir um retângulo maior são definidos implicitamente por variáveis de decisão do modelo, em contraste à abordagem hierárquica da seção anterior, que requer a saída do Algoritmo 5.1.1.

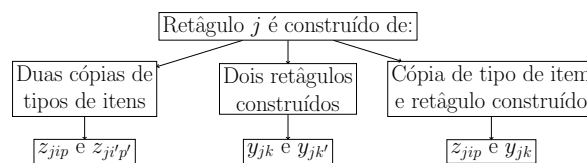


Figura 5.7: Compreensão do modelo implícito.

Para permitir que ambos os modelos sejam facilmente comparados com relação a suas variáveis e restrições, optou-se por uma notação similar. Dessa forma,

chama-se agora de “retângulo” o que foi chamado de “nó” na seção anterior. Seja $J = \{1, \dots, \bar{n} - 1\}$ o conjunto de todos os retângulos possíveis de serem construídos (isto é, subpadrões), que é diferente da definição da seção anterior. Seja $O = \{h, v\}$ o conjunto de todas as orientações possíveis para construir um retângulo, sendo que h e v são para construções horizontais e verticais, respectivamente. Seja $Q_i = \{1, \dots, \min\{2, u_i\}\}$ o conjunto possível de cópias de um tipo de item $i \in I$ a ser contido em qualquer retângulo $j \in J$.

Sobre as variáveis de decisão, existem cinco conjuntos de variáveis nesse modelo, que são definidas em (5.7) até (5.11). É importante destacar que as variáveis y_{jk} são responsáveis por gerar subpadrões ao associar subpadrões menores e, assim, elas atuam como a árvore binária do modelo anterior. Na Figura 5.8 esboçam-se os três casos para construção de um subpadrão $j \in J$ na abordagem *bottom-up* implícita ao combinar: (i) duas cópias de tipos de itens, (ii) uma cópia de um tipo de item e um subpadrão menor, ou (iii) dois subpadrões menores. Observe que ao combinar duas cópias de tipos de itens (Figuras 5.8(a) e 5.8(b)), os tipos de itens i e i' podem representar o mesmo tipo de item. Para representar esse caso, o índice $p \in Q_i$ é necessário nas variáveis z_{jip} no modelo implícito, que é diferente da definição da seção anterior. O índice p nas variáveis z_{jip} é limitado em um conjunto de até dois elementos, visto que qualquer subpadrão $j \in J$ contém (diretamente) no máximo duas cópias de um tipo de item $i \in I$. De acordo com a definição dos conjuntos Q_i , se $u_i = 1$, então um subpadrão $j \in J$ pode conter no máximo uma cópia do tipo de item $i \in I$.

$$z_{jip} = \begin{cases} 1, & \text{se o retângulo } j \text{ contém a } p\text{-ésima cópia do tipo de item } i, \\ 0, & \text{caso contrário,} \end{cases} \quad j \in J, i \in I, p \in Q_i. \quad (5.7)$$

$$x_{jo} = \begin{cases} 1, & \text{se o retângulo } j \text{ é construído com orientação } o, \\ 0, & \text{caso contrário,} \end{cases} \quad j \in J, o \in O. \quad (5.8)$$

$$y_{jk} = \begin{cases} 1, & \text{se o retângulo } j \text{ contém o retângulo } k, \\ 0, & \text{caso contrário,} \end{cases} \quad j \in J, k \in J, j < k. \quad (5.9)$$

$$L_j: \text{ é o comprimento do retângulo } j, \quad j \in J. \quad (5.10)$$

$$W_j: \text{ é a largura do retângulo } j, \quad j \in J. \quad (5.11)$$

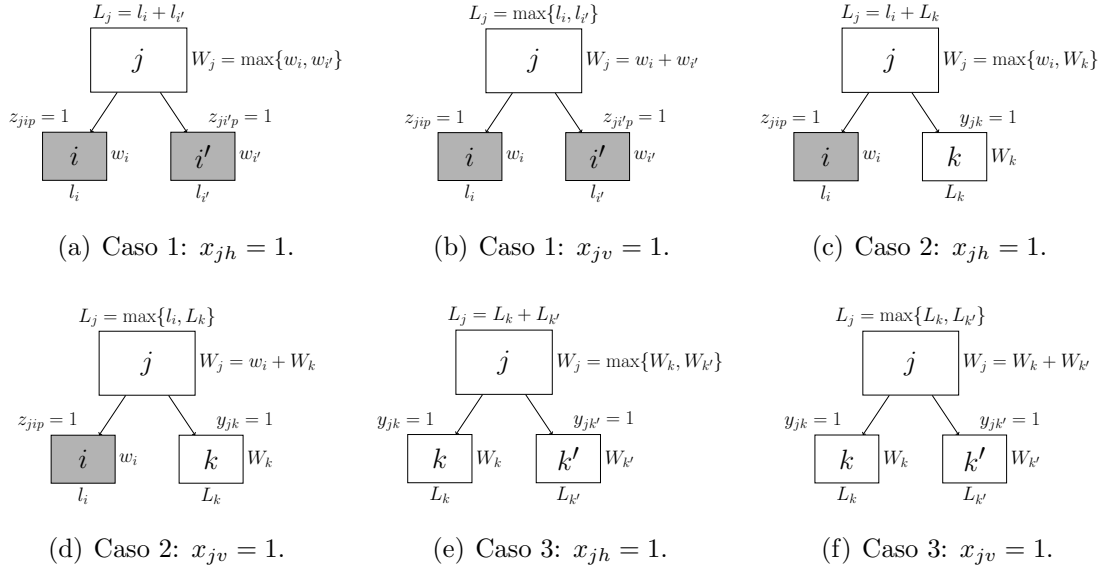


Figura 5.8: Modelo implícito. Um subpadrão é construído de: (a) e (b) duas cópias de tipos de itens; (c) e (d) uma cópia de tipo de item e um subpadrão menor; or, (e) e (f) dois subpadrões menores.

Usando as variáveis e parâmetros definidos até então, o Modelo (5.12) é uma formulação compacta de PNLIM para o PCBGR.

$$\text{Max} \sum_{j \in J} \sum_{i \in I} \sum_{p \in Q_i} v_i z_{jip}, \quad (5.12a)$$

s.a

$$\sum_{j \in J} \sum_{p \in Q_i} z_{jip} \leq u_i, \quad i \in I, \quad (5.12b)$$

$$\sum_{o \in O} x_{jo} \leq 1, \quad j \in J, \quad (5.12c)$$

$$\sum_{i \in I} \sum_{p \in Q_i} z_{1ip} + \sum_{k \in K, 1 < k} y_{1k} \leq 2 \sum_{o \in O} x_{1o}, \quad (5.12d)$$

$$\sum_{i \in I} \sum_{p \in Q_i} z_{jip} + \sum_{k \in K, j < k} y_{jk} = 2 \sum_{o \in O} x_{jo}, \quad j \in J \setminus \{1\}, \quad (5.12e)$$

$$\sum_{j \in J, j < k} y_{jk} = \sum_{o \in O} x_{ko}, \quad k \in J \setminus \{1\} \quad (5.12f)$$

$$L_j = \left(\sum_{i \in I} \sum_{p \in Q_i} l_i z_{jip} + \sum_{k \in J, j < k} L_k y_{jk} \right) * x_{jh} \\ + \max\{(i, p) \in (I \times Q_i) : l_i z_{jip}; k \in J, j < k : L_k y_{jk}\} * x_{jv}, \quad j \in J, \quad (5.12g)$$

$$W_j = \max\{(i, p) \in (I \times Q_i) : w_i z_{jip}; k \in J, j < k : W_k y_{jk}\} * x_{jh} \\ + \left(\sum_{i \in I} \sum_{p \in Q_i} w_i z_{jip} + \sum_{k \in J, j < k} W_k y_{jk} \right) * x_{jv}, \quad j \in J, \quad (5.12h)$$

$$z_{jip} \in \{0, 1\}, \quad j \in J, i \in I, p \in Q_i, \quad (5.12i)$$

$$y_{jk} \in \{0, 1\}, \quad j \in J, k \in J, j < k, \quad (5.12j)$$

$$x_{jo} \in \{0, 1\}, \quad j \in J, o \in O, \quad (5.12k)$$

$$0 \leq L_j \leq L, \quad j \in J, \quad (5.12l)$$

$$0 \leq W_j \leq W, \quad j \in J. \quad (5.12m)$$

A função objetivo (5.12a) consiste em maximizar a soma total dos valores dos itens selecionados. As restrições (5.12b) garantem que cada tipo de item $i \in I$ é selecionado em até seu número máximo de cópias permitido de ser manufaturado. As restrições (5.12c) limitam que cada retângulo, se existir, é um subpadrão construído horizontal ou verticalmente.

As restrições (5.12d) e (5.12e) garantem que se um retângulo $j \in J$ é construído (isto é, quando $\sum_{o \in O} x_{jo} = 1$), então ele deve conter exatamente duas cópias de tipos de itens, ou dois subpadrões, ou uma cópia de um tipo de item e um subpadrão — veja Figura 5.8 —, desde que o rhs da correspondente restrição seja 2. Por outro lado, se o retângulo $j \in J$ não é construído (isto é, quando $\sum_{o \in O} x_{jo} = 0$), então ele não deve conter itens nem subpadrões. Observe que as restrições (5.12d), (5.12e) e (5.12f) também garantem implicitamente que o retângulo 1 é o último retângulo construído e, assim, ele corresponde à solução final (isto é, padrão de corte final). A restrição (5.12d) é similar as restrições (5.12e) mas aplicada ao retângulo 1 apenas, visto que ele permite que a solução seja formada de apenas um único item neste caso.

As restrições (5.12f) garantem que cada retângulo $k \in J \setminus \{1\}$ é um subpadrão, se e somente se, ele está contido em um subpadrão maior $j \in J$, com $j < k$. Observe que essas restrições também limitam que um retângulo $k \in J \setminus \{1\}$ é contido no máximo em um subpadrão $j \in J$, com $j < k$, de acordo com as restrições (5.12c).

As restrições (5.12g) garantem que a variável L_j esteja de acordo com as definições do retângulo $j \in J$ com respeito a orientação $o \in O$ da construção, e os comprimentos dos itens e/ou subpadrões a partir dos quais ele é construído. Por exemplo, se o retângulo $j \in J$ é uma construção horizontal (isto é, quando $x_{jh} = 1$), então a variável L_j assume a soma dos comprimentos dos itens e/ou subpadrões que constroem tal retângulo j . Por outro lado, se o retângulo $j \in J$ é uma construção vertical (isto é, quando $x_{jv} = 1$), então a variável L_j assume o maior comprimento dos itens e/ou subpadrões que constroem tal retângulo j ao usar a função $\max\{\cdot\}$. Observe que se um retângulo $j \in J$ contém outro subpadrão $k \in J$ (isto é, quando $y_{jk} = 1$), o comprimento e largura de tal subpadrão k pode ser definido como $L_k y_{jk}$ e $W_k y_{jk}$, respectivamente, o que é útil para definir as dimensões do retângulo j . No entanto, se o retângulo $j \in J$ não contém o subpadrão $k \in J$ (isto é, quando $y_{jk} = 0$), ambos os produtos de variáveis $L_k y_{jk}$ e $W_k y_{jk}$ são zero.

As restrições (5.12h) são similares as restrições (5.12g), mas relacionadas às variáveis W_j . Elas garantem que W_j esteja de acordo com as definições do retângulo

$j \in J$ com respeito a orientação $o \in O$ da construção, e as larguras dos tipos de itens e/ou subpadrões a partir dos quais ele é construído. Por exemplo, se o retângulo $j \in J$ é uma construção horizontal (isto é, quando $x_{jh} = 1$), então a variável W_j assume a maior largura dos itens e/ou subpadrões que constroem tal retângulo j ao usar a função $\max\{\cdot\}$. No entanto, se o retângulo $j \in J$ é uma construção vertical (isto é, quando $x_{jv} = 1$), então W_j assume a soma das larguras dos itens e/ou subpadrões que constroem tal retângulo j .

As restrições (5.12i) até (5.12m) definem o domínio das variáveis de decisão. Observe que esse modelo tem a premissa que o retângulo 1 é sempre construído e, assim, $\sum_{o \in O} x_{1o} = 1$ pode ser fixado. O Modelo (5.12) é chamado daqui em diante de Modelo Não-linear 2. Esse modelo é não-linear devido ao produto de variáveis e o uso da função $\max\{\cdot\}$ nas restrições (5.12g) e (5.12h). De forma similar ao modelo proposto na seção anterior, esse modelo não requer que os dados de entrada (isto é, L, W, l_i e w_i , para cada $i \in I$) sejam inteiros.

Na Tabela 5.3 mostra-se uma solução ótima (valores não-nulos apenas) do exemplo ilustrativo da Tabela 5.1, que é esboçado na Figura 5.9. Essa solução foi obtida ao resolver o Modelo Não-linear 2 usando o *IBM CPLEX CPOptimizer v12.8*. Observe que as soluções apresentadas nas Figuras 5.6 e 5.9 são idênticas; optou-se por replicá-la aqui apenas por clareza na comparação com a Tabela 5.3. Na Seção 5.5 discute-se o Algoritmo 5.5.2, que é um pseudo-código para gerar o padrão de corte a partir de uma solução do Modelo Não-linear 2.

j	$(L \times W)$	x_{jo}	y_{jk} e/ou z_{jip}
1	(35×30)	$x_{1h} = 1$	$y_{12} = 1$ e $z_{131} = 1$
2	(30×30)	$x_{2v} = 1$	$y_{23} = 1$ e $z_{221} = 1$
3	(30×15)	$x_{3h} = 1$	$z_{312} = 1$ e $z_{311} = 1$

Tabela 5.3: Solução ótima do exemplo ilustrativo da Tabela 5.1 ao Modelo Não-linear 2.

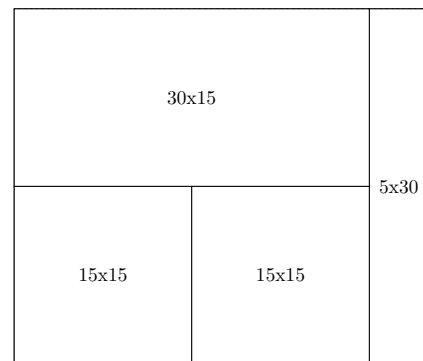


Figura 5.9: Padrão de corte da Tabela 5.3.

Particularmente, esforços para reformular os Modelos (5.6) e (5.12) em uma abordagem de corte *top-down*, em vez de *bottom-up*, são praticamente diretas. O Apêndice A apresenta os modelos da Seção 5.1 reformulados nessa perspectiva. Nota-se, no entanto, que os experimentos computacionais preliminares não motivaram novos desenvolvimentos para esses modelos.

5.2 Formulações matemáticas lineares

Nessa seção propõem-se dois modelos de PLIM ao reformular o Modelo Não-linear 1 e o Modelo Não-linear 2 com variáveis e restrições adicionais para permitir o uso de *solvers* de propósito geral. Adicionalmente, propõem-se desigualdades válidas para ambos os

modelos visando melhorar suas relaxações lineares e acelerar a convergência de métodos *branch-and-cut* usados pelos *solvers* de PLIM.

5.2.1 Formulação linear 1

O modelo Modelo Não-linear 1 pode ser reformulado, sem perda de otimalidade, como um modelo de PLIM por meio de desigualdades disjuntivas tomando lugar das restrições não-lineares (5.6h) até (5.6m). Assim, tem-se o Modelo (5.13), que é uma formulação pseudopolinomial de PLIM para o PCBGR.

Max (5.6a),

s.a

(5.6b) – (5.6g), (5.6n) – (5.6q),

$$L_j \geq \sum_{i \in I} l_i z_{ji}, \quad j \in J \setminus \bar{J}, \quad (5.13a)$$

$$L_j \geq L_{j^-} + L_{j^+} - 2L(1 - x_{jh}), \quad j \in J_A, \quad (5.13b)$$

$$L_j \geq L_{j^-} - L(1 - x_{jv}), \quad j \in J_A \cup J_B, \quad (5.13c)$$

$$L_j \geq L_{j^+} - L(1 - x_{jv}), \quad j \in J_A, \quad (5.13d)$$

$$L_j \geq L_{j^-} + \sum_{i \in I} l_i z_{j+i} - 2L(1 - x_{jh}), \quad j \in J_B, \quad (5.13e)$$

$$L_j \geq \sum_{i \in I} l_i z_{j+i} - L(1 - x_{jv}), \quad j \in J_B \cup J_C, \quad (5.13f)$$

$$L_j \geq \sum_{i \in I} l_i z_{j-i} + \sum_{i \in I} l_i z_{j+i} - 2L(1 - x_{jh}), \quad j \in J_C, \quad (5.13g)$$

$$L_j \geq \sum_{i \in I} l_i z_{j-i} - L(1 - x_{jv}), \quad j \in J_C, \quad (5.13h)$$

$$W_j \geq \sum_{i \in I} w_i z_{ji}, \quad j \in J \setminus \bar{J}, \quad (5.13i)$$

$$W_j \geq W_{j^-} - W(1 - x_{jh}), \quad j \in J_A \cup J_B, \quad (5.13j)$$

$$W_j \geq W_{j^+} - W(1 - x_{jh}), \quad j \in J_A, \quad (5.13k)$$

$$W_j \geq W_{j^-} + W_{j^+} - 2W(1 - x_{jv}), \quad j \in J_A, \quad (5.13l)$$

$$W_j \geq \sum_{i \in I} w_i z_{j+i} - W(1 - x_{jh}), \quad j \in J_B \cup J_C, \quad (5.13m)$$

$$W_j \geq W_{j^-} + \sum_{i \in I} w_i z_{j+i} - 2W(1 - x_{jv}), \quad j \in J_B, \quad (5.13n)$$

$$W_j \geq \sum_{i \in I} w_i z_{j-i} - W(1 - x_{jh}), \quad j \in J_C, \quad (5.13o)$$

$$W_j \geq \sum_{i \in I} w_i z_{j-i} + \sum_{i \in I} w_i z_{j+i} - 2W(1 - x_{jv}), \quad j \in J_C. \quad (5.13p)$$

Foram substituídas as restrições não-lineares (5.6h) do Modelo Não-linear 1 pelas quatro correspondentes restrições lineares (5.13a) até (5.13d), sem perda de otimalidade. Particularmente, quando $j \in J_A$ é uma cópia de qualquer tipo de item, as restrições (5.13b) até (5.13d) estão inativas, visto que $x_{jh} = x_{jv} = 0$. Se o retângulo $j \in J_A$ é uma construção horizontal (isto é, quando $x_{jh} = 1$), apenas a restrição (5.13b) tem um rhs positivo. No entanto, se o retângulo é uma combinação vertical (isto é, quando $x_{jv} = 1$), apenas as restrições (5.13c) e (5.13d) têm rhs positivos. As restrições adicionais (5.13e) até (5.13p) seguem esses princípios para as três categorias de retângulos não-folha J_A , J_B e J_C com respeito as variáveis L_j e W_j . Observe que as restrições (5.13b) até (5.13h) e restrições (5.13j) até (5.13p) são do tipo M-Grande, e optou-se por já indicar números suficientemente grandes nessas restrições, de acordo com os parâmetros L e W .

Foram realizados experimentos computacionais preliminares para analisar o desempenho médio do Modelo (5.6), e observou-se que sua relaxação linear não é apertada. Note que o Modelo (5.13) não possui restrições de alocação, isto é, ele considera apenas variáveis binárias associadas à estrutura da árvore binária. Particularmente, em alguns desses experimentos, o valor ótimo de sua relaxação linear foi exatamente a soma dos valores de todos os itens (isto é, $\sum_{i \in I} v_i u_i$). Para contornar isso, foram desenvolvidas algumas desigualdades válidas (5.14) que apertam sua relaxação linear e, assim, aceleram sua convergência em pacotes de *software* de propósito geral para PLIM.

$$\sum_{j \in J} \sum_{i \in I} (l_i w_i) z_{ji} \leq UB_1, \quad (5.14a)$$

$$\sum_{j \in J} \sum_{i \in I} v_i z_{ji} \leq UB_2, \quad (5.14b)$$

$$\sum_{j \in J} \sum_{i \in I} z_{ji} \leq \bar{n}, \quad (5.14c)$$

$$\sum_{j \in J \setminus \bar{J}} \sum_{o \in O} x_{jo} = \sum_{j \in J} \sum_{i \in I} z_{ji} - 1, \quad (5.14d)$$

$$L_j \leq L \left(\sum_{o \in O} x_{jo} + \sum_{j \in J} \sum_{i \in I} z_{ji} \right), \quad j \in J \setminus \bar{J}, \quad (5.14e)$$

$$W_j \leq W \left(\sum_{o \in O} x_{jo} + \sum_{j \in J} \sum_{i \in I} z_{ji} \right), \quad j \in J \setminus \bar{J}. \quad (5.14f)$$

$$\sum_{o \in O} x_{j-o} \geq \sum_{o \in O} x_{j+o}, \quad j \in J_A, \quad (5.14g)$$

$$\sum_{i \in I} i z_{j-i} \leq \sum_{i' \in I} i' z_{j+i'}, \quad j \in J \setminus \bar{J}. \quad (5.14h)$$

A restrição (5.14a) limita que a área total dos itens selecionados não ultrapasse o parâmetro UB_1 , que indica um limitante superior a área útil do objeto original. Similarmente, a restrição (5.14b) limita que o valor total dos itens selecionados não ultrapasse o parâmetro UB_2 , que indica um limitante superior para o valor da função objetivo

– esta desigualdade é imposta apenas para as instâncias ponderadas. Os parâmetros UB_1 e UB_2 podem ser estimados por meio de qualquer relaxação do PCBGR. Observe que o impacto de tais desigualdades nos modelos depende de quão apertados esses limitantes são. Por exemplo, uma forma simples de estimar os parâmetros UB_1 e UB_2 é considerar, para cada tipo de item $i \in I$, sua área ($l_i w_i$) ou valor (v_i) nos coeficientes da função objetivo do Modelo (5.1), respectivamente.

As restrições (5.14c) garantem que a solução tem no máximo \bar{n} itens, conforme assumido no procedimento de enumeração dado pelo Algoritmo 5.1.1, sem perda de generalidade. A restrição (5.14d) garante a relação entre os números de construções e itens. O número de itens na solução é representado por $\sum_{j \in J} \sum_{i \in I} z_{ji}$ e, como discutido no início da Seção 5.1, o número de construções é igual ao número de itens menos 1. As restrições (5.14e) e (5.14f) definem, respectivamente, que as variáveis L_j e W_j podem assumir valores positivos apenas se o retângulo $j \in J$ é uma construção (horizontal ou vertical) ou uma cópia de um tipo de item.

As restrições (5.14g) eliminam algumas simetrias do modelo ao impor que o nó-filho da direita de um nó $j \in J_A$ pode ser um subpadrão apenas quando seu nó-filho da esquerda também é um subpadrão. Por exemplo, supondo a Figura 5.3(b) e uma solução ótima com $\hat{n} = 3$, há dois casos possíveis: (i) nós 3, 4 e 5 são itens; or, (ii) nós 2, 6 e 7 são itens. No entanto, a última opção não é permitida diante dessas restrições adicionais, dado que o nó 3 não pode ser um subpadrão quando o nó 2 contém um item. As restrições (5.14h) eliminam outras simetrias do modelo, quando um subpadrão $j \in J \setminus \bar{J}$ é construído de duas cópias de itens i e i' , ao limitar que seu nó-filho da esquerda tenha um tipo de item com identificação i menor ou igual à identificação i' do nó-filho da direita. Destaca-se que as restrições (5.14a) e (5.14b), dentre todas as desigualdades válidas (5.14), foram as mais relevantes em melhorar a performance do modelo no contexto de um *solver* de propósito geral nos experimentos computacionais preliminares. O Modelo (5.13) com as desigualdades válidas (5.14) é chamado daqui em diante de Modelo Linear 1.

5.2.2 Formulação linear 2

O Modelo Não-linear 2 pode ser reformulado, sem perda de otimalidade, como um modelo de PLIM. Para isso, são necessários dois conjuntos de variáveis e restrições disjuntivas para substituir as restrições não-lineares (5.12g) até (5.12h), similarmente ao desenvolvimento apresentado na seção anterior. As Equações (5.15) e (5.16) definem as variáveis adicionais necessárias à reformulação.

$$y_{jk}^L: \text{ é o comprimento do retângulo } k \text{ contido pelo retângulo } j, \quad j \in J, k \in J, j < k. \quad (5.15)$$

y_{jk}^W : é a largura do retângulo k contido pelo retângulo j , $j \in J, k \in J, j < k$. (5.16)

Ao reformular o Modelo Não-linear 2 usando essas variáveis, tem-se uma formulação compacta de PLIM para o PCBGR, dada pelo Modelo (5.17).

Max (5.12a),

s.a

(5.12b) – (5.12f), (5.12i) – (5.12m),

$$0 \leq y_{jk}^L \leq L y_{jk}, \quad j \in J, k \in J, j < k, \quad (5.17a)$$

$$y_{jk}^L \leq L_k, \quad j \in J, k \in J, j < k, \quad (5.17b)$$

$$L_k - L(1 - y_{jk}) \leq y_{jk}^L, \quad j \in J, k \in J, j < k, \quad (5.17c)$$

$$0 \leq y_{jk}^W \leq W y_{jk}, \quad j \in J, k \in J, j < k, \quad (5.17d)$$

$$y_{jk}^W \leq W_k, \quad j \in J, k \in J, j < k, \quad (5.17e)$$

$$W_k - W(1 - y_{jk}) \leq y_{jk}^W, \quad j \in J, k \in J, j < k, \quad (5.17f)$$

$$L_j \geq \sum_{i \in I} \sum_{p \in Q_i} l_i z_{jip} + \sum_{k \in K, j < k} y_{jk}^L - 2L(1 - x_{jh}), \quad j \in J, \quad (5.17g)$$

$$L_j \geq l_i z_{jip} - L(1 - x_{jv}), \quad j \in J, i \in I, p \in Q_i, \quad (5.17h)$$

$$L_j \geq y_{jk}^L - L(1 - x_{jv}), \quad j \in J, k \in K, j < k, \quad (5.17i)$$

$$W_j \geq w_i z_{jip} - W(1 - x_{jh}), \quad j \in J, i \in I, p \in Q_i, \quad (5.17j)$$

$$W_j \geq y_{jk}^W - W(1 - x_{jh}), \quad j \in J, k \in K, j < k, \quad (5.17k)$$

$$W_j \geq \sum_{i \in I} \sum_{p \in Q_i} w_i z_{jip} + \sum_{k \in K, j < k} y_{jk}^W - 2W(1 - x_{jv}), \quad j \in J. \quad (5.17l)$$

As restrições (5.17a) até (5.17c) definem que a variável y_{jk}^L é igual a L_k quando $y_{jk} = 1$, e ela é igual a zero quando $y_{jk} = 0$. Similarmente, as restrições (5.17d) até (5.17f) definem que a variável y_{jk}^W é igual a W_k quando $y_{jk} = 1$, e ela é igual a zero quando $y_{jk} = 0$. Cada restrição não-linear (5.12g) do Modelo Não-linear 2 pode ser substituída, sem perda de otimalidade, pelas três correspondentes restrições lineares (5.17g) até (5.17i). Observe que quando o retângulo $j \in J$ é uma construção horizontal (isto é, quando $x_{jh} = 1$), apenas a restrição (5.17g) está ativa, mas se ele é uma construção vertical (isto é, quando $x_{jv} = 1$), apenas as restrições (5.17h) e (5.17i) estão ativas. As restrições (5.17j) até (5.17l) impõem uma comportamento similar às variáveis W_j .

Os experimentos computacionais preliminares também mostraram que a relaxação linear do Modelo (5.17) não fornece limitantes superiores apertados, assim como aqueles do Modelo (5.13). Assim, as desigualdades válidas (5.14) a serem usadas no Modelo (5.17) foram adaptadas, conforme dado por (5.18).

$$\sum_{j \in J} \sum_{i \in I} \sum_{p \in Q_i} (l_i w_i) z_{jip} \leq UB_1, \quad (5.18a)$$

$$\sum_{j \in J} \sum_{i \in I} \sum_{p \in Q_i} v_i z_{jip} \leq UB_2, \quad (5.18b)$$

$$\sum_{j \in J} \sum_{i \in I} \sum_{p \in Q_i} z_{jip} \leq \bar{n}, \quad (5.18c)$$

$$\sum_{j \in J} \sum_{o \in O} x_{jo} = \sum_{j \in J} \sum_{i \in I} \sum_{p \in Q_i} z_{jip} - 1, \quad (5.18d)$$

$$L_j \leq L \sum_{o \in O} x_{jo}, \quad j \in J, \quad (5.18e)$$

$$W_j \leq W \sum_{o \in O} x_{jo}. \quad j \in J, \quad (5.18f)$$

As restrições (5.18a) até (5.18f) são, respectivamente, equivalente as desigualdades válidas (5.14a) até (5.14f). Observe que as desigualdades (5.14g) e (5.14h) não são diretamente adaptáveis ao Modelo (5.17), e que a desigualdade (5.18d) supõe que a solução ótima não é formada de um único item. O Modelo (5.17) com as desigualdades válidas (5.18) é chamado daqui em diante de Modelo Linear 2.

5.3 Extensões

Os modelos propostos das Seções 5.1 e 5.2 podem também abordar o caso de rotação de itens ao simplesmente adicionar um novo tipo de item i' com comprimento w_i e largura l_i , para cada tipo de item $i \in I$. Em adição, é necessário adaptar as restrições que limitam a produção de até u_i cópias de forma a considerar não apenas os tipo de item i , mas também a equivalente cópia rotacionada i' .

Nessa seção discute-se como adaptar os modelos não-lineares e lineares propostos para o caso do *Guillotine Strip Packing Problem* (GSPP), e como estendê-los para estritamente abordar padrões d -estágios, com $d \in \mathbb{Z}_+$. Observe que os modelos matemáticos relacionados às versões d -estágios do PCBGR, como em Lodi e Monaci (2003) para padrões 2-estágios e Puchinger e Raidl (2007) para 3-estágios, foram publicados antes dos modelos propostos por Ben Messaoud, Chu e Espinouse (2008) e Furini, Malaguti e Thomopulos (2016) para o caso não-estagiado. Segundo o conhecimento desse autor, os modelos de Ben Messaoud, Chu e Espinouse (2008) e Furini, Malaguti e Thomopulos (2016) não são diretamente adaptáveis para estritamente considerar padrões d -estágios. Além disso, um modelo que é capaz de fornecer soluções ótimas limitadas em até d estágios guilhotinados pode ser relevante em casos no qual importa o compromisso (*trade-off*) entre a utilização de material e as taxas de produção (MORABITO; ARENALES, 2000).

5.3.1 Problema da Dimensão Aberta Guilhotinado

O Problema da Dimensão Aberta Guilhotinado (do inglês, *Guillotine Strip Packing Problem* – GSPP) considera o corte de um conjunto de itens a partir de um único retângulo grande (objeto), de comprimento fixo e largura variável. O objetivo é cortar todos os itens, minimizando a largura variável do objeto, ao usar cortes guilhotinados ortogonais. Os modelos não-lineares e lineares propostos nas Seções 5.1 e 5.2 podem ser facilmente adaptadas para o GSPP. Observe que o limitante superior \bar{n} torna-se desnecessário, visto que uma solução do GSPP deve cortar todos os itens e, assim, $\hat{n} = \sum_{i \in I} u_i$. Particularmente, a adaptação é indicada para o Modelo Não-linear 1, que é definida pelo Modelo (5.19).

$$\mathbf{Min} \quad W_1 \tag{5.19a}$$

s.a

$$(5.6c) - (5.6p),$$

$$\sum_{j \in J} z_{ji} = u_i, \quad i \in I, \tag{5.19b}$$

$$0 \leq W_j \leq \bar{W}, \quad j \in J \setminus \bar{J}. \tag{5.19c}$$

A função objetivo (5.19a) consiste de minimizar a largura do objeto (isto é, nó 1). As restrições (5.19b) garantem que cada tipo de item $i \in I$ é manufaturado de acordo com sua demanda u_i . As restrições (5.19c) tratam do domínio de variáveis W_j . O parâmetro \bar{W} é um limitante superior à largura do objeto, que pode ser definida como o valor suficientemente grande $\sum_{i \in I} w_i u_i$.

5.3.2 Restrições para padrões d -estágios

A premissa é contar o número de estágios guilhotinados necessários para cada retângulo j que representa subpadrões. Assim, o conjunto de variáveis adicionais δ_j é criado.

Por exemplo, a Figura 5.10(a) ilustra quatro cortes guilhotinados, sendo que o primeiro corte é um corte vertical (primeiro estágio) e os outros cortes são horizontais (segundo estágio). Note que nessa figura: cada item A, B, C, D e E é gerado em zero estágio; cada um dos retângulos AE, BC e BCD é gerado em um estágio; e, o retângulo AE-BCE é gerado em dois estágios. Outro exemplo é ilustrado na Figura 5.10(b) que contém cinco cortes guilhotinados, sendo que o primeiro corte é um corte horizontal (primeiro estágio), o segundo e terceiro cortes são verticais (segundo estágio), e os há outros dois cortes horizontais adicionais (terceiro estágio). Note que nessa figura: cada item A, B, C, D, E e F é gerado em zero estágio; cada um dos retângulos BC e DE é gerado em um estágio; cada um dos retângulos ABC e DEF é gerado em dois estágios; e

o retângulo ABC-DEF é gerado em três estágios. Observe que a operação de *trimming*, i.e., o corte adicional para separar um item e o desperdício não é considerado como um corte guilhotinado adicional.

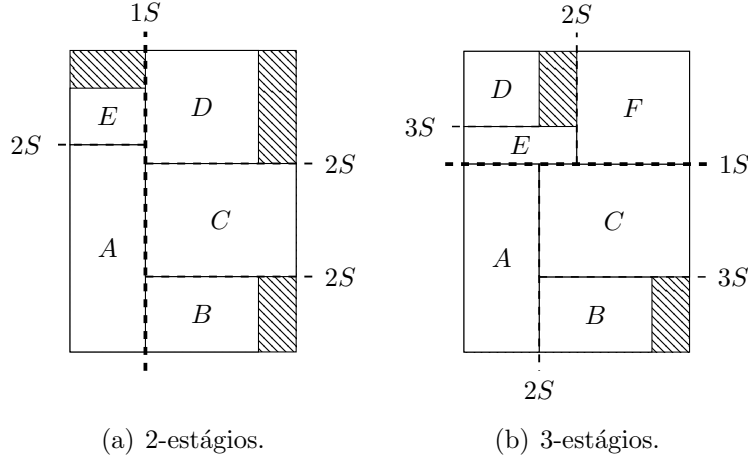


Figura 5.10: Exemplos de padrões de corte 2-estágios e 3-estágios.

Modelo Não-linear 1

As restrições (5.20) para o Modelo Não-linear 1 são propostas para estritamente forçar a geração de padrões de corte de até d estágios.

$$\delta_j \geq \delta_{j^-} + x_{jh}x_{j-v} + x_{jv}x_{j-h} + \sum_{i \in I} z_{j-i}, \quad j \in J_A \cup J_B, \quad (5.20a)$$

$$\delta_j \geq \delta_{j^+} + x_{jh}x_{j+v} + x_{jv}x_{j+h} + \sum_{i \in I} z_{j+i}, \quad j \in J_A, \quad (5.20b)$$

$$\delta_j = \sum_{o \in O} x_{jo}, \quad j \in J_C, \quad (5.20c)$$

$$\delta_j \leq d \sum_{o \in O} x_{jo}, \quad j \in J \setminus \bar{J}, \quad (5.20d)$$

$$\delta_j \in \mathbb{Z}_+, \quad j \in J \setminus \bar{J}. \quad (5.20e)$$

As restrições (5.20a) estabelecem que a quantidade de estágios guilhotinados de um nó j (δ_j) dos conjuntos J_A ou J_B é maior ou igual a: (i) a quantidade de estágios guilhotinados do filho da esquerda j^- mais uma possível mudança na orientação de construção (isto é, quando $x_{jh}x_{j-v} = 1$ ou $x_{jv}x_{j-h} = 1$); ou (ii) 1 quando o nó j^- é uma cópia de qualquer tipo de item (isto é, quando $\sum_{i \in I} z_{j-i} = 1$). As restrições (5.20b) são similares as previamente definidas e são para o filho da direita j^+ de cada nó $j \in J_A$. As restrições (5.20c) limitam δ_j ao valor unitário quando o nó $j \in J_C$ é um subpadrão (isto é, quando $\sum_{o \in O} x_{jo} = 1$). As restrições (5.20d) garantem que cada nó tem até d estágios

guilhotinados quando ele é um subpadrão. As restrições (5.20e) tratam do domínio de variáveis.

Modelo Não-linear 2

Para o Modelo Não-linear 2 estritamente considerar padrões de corte de até d estágios, tem-se a adição de restrições adicionais (5.21).

$$\delta_j \leq d \sum_{o \in O} x_{jo}, \quad j \in J, \quad (5.21a)$$

$$\delta_j \geq \sum_{o \in O} x_{jo}, \quad j \in J, \quad (5.21b)$$

$$\delta_j \geq \delta_k + x_{jh}x_{kv} + x_{jv}x_{kh} - d(1 - y_{jk}), \quad j \in J, k \in K, j < k, \quad (5.21c)$$

$$\delta_j \in \mathbb{Z}_+, \quad j \in J. \quad (5.21d)$$

As restrições (5.21a) e (5.21b) garantem que para cada retângulo $j \in J$ sua variável δ_j está entre 1 e d , quando tal retângulo é um subpadrão. As desigualdades disjuntivas (5.21c) estabelecem que quando um retângulo j contém um retângulo k (isto é, quando $y_{jk} = 1$), a quantidade de estágios guilhotinados do retângulo j é maior ou igual do que a quantidade de estágios guilhotinados do retângulo k mais uma possível alteração na orientação de construção (isto é, quando $x_{jh}x_{kv} = 1$ or $x_{jv}x_{kh} = 1$). As restrições (5.21d) definem o domínio das variáveis de decisão. Ressalta-se que os conjuntos J no Modelo Não-linear 1 e no Modelo Não-linear 2 têm definições diferentes.

Modelos lineares

As restrições (5.20a), (5.20b) e (5.21c) são não-lineares, visto que elas têm produtos de variáveis binárias para representar a possível alteração na orientação de construção. No entanto, como amplamente conhecimento na literatura de otimização linear, o produto de duas variáveis binárias pode ser facilmente linearizado usando uma variável auxiliar e restrições adicionais. Assim, parte-se disso para obter modelos lineares para esses casos.

O Modelo Linear 1 com as restrições (5.22) estritamente consideram padrões d -estágios, que é chamado daqui em diante de Modelo Linear d -estágios 1.

(5.20c) – (5.20e),

$$\delta_j \geq \delta_{j^-} + w_{j,h,j^-,v} + w_{j,v,j^-,h} + \sum_{i \in I} z_{j^-i}, \quad j \in J_A \cup J_B, \quad (5.22a)$$

$$\delta_j \geq \delta_{j^+} + w_{j,h,j^+,v} + w_{j,v,j^+,h} + \sum_{i \in I} z_{j^+i}, \quad j \in J_A, \quad (5.22b)$$

$$w_{j_o, k\bar{o}} \leq x_{j_o}, \quad j \in J_A \cup J_B, o \in O, k \in \{j^-, j^+\}, \quad (5.22c)$$

$$w_{j_o k\bar{o}} \leq x_{k\bar{o}}, \quad j \in J_A \cup J_B, o \in O, k \in \{j^-, j^+\}, \quad (5.22d)$$

$$w_{j_o k\bar{o}} \geq x_{j_o} + x_{k\bar{o}} - 1, \quad j \in J_A \cup J_B, o \in O, k \in \{j^-, j^+\}, \quad (5.22e)$$

$$w_{j_o k\bar{o}} \in \{0, 1\}, \quad j \in J_A \cup J_B, o \in O, k \in \{j^-, j^+\}. \quad (5.22f)$$

As restrições (5.22a) e (5.22b) são lineares e equivalentes às restrições (5.20a) e (5.20b), respectivamente. As restrições (5.22c) até (5.22e) garantem que a variável auxiliar $w_{j_o k\bar{o}}$ é igual a um, apenas se, $x_{j_o} = 1$ e $x_{k\bar{o}} = 1$, e 0 caso contrário. Observe que \bar{o} é o complemento da orientação $o \in O$, isto é, se $o = h$ então $\bar{o} = v$, e se $o = v$ então $\bar{o} = h$. Abusa-se da notação na definição de $k \in \{j^-, j^+\}$, dado que os nós no nó J_B têm apenas filhos da esquerda.

O Modelo Linear 2 com as restrições (5.23) estritamente consideram padrões d -estágios, que é chamado daqui em diante de Modelo Linear d -estágios 2.

$$\delta_j \geq \delta_k + w_{jhkv} + w_{vkh} - d(1 - y_{jk}), \quad j \in J, k \in K, j < k, \quad (5.23a)$$

$$w_{j_o k\bar{o}} \leq x_{j_o}, \quad j \in J, o \in O, k \in K, j < k, \quad (5.23b)$$

$$w_{j_o k\bar{o}} \leq x_{k\bar{o}}, \quad j \in J, o \in O, k \in K, j < k, \quad (5.23c)$$

$$w_{j_o k\bar{o}} \geq x_{j_o} + x_{k\bar{o}} - 1, \quad j \in J, o \in O, k \in K, j < k, \quad (5.23d)$$

$$w_{j_o k\bar{o}} \in \{0, 1\}, \quad j \in J, o \in O, k \in K, j < k, \quad (5.23e)$$

As restrições (5.23a) são lineares e equivalente às restrições(5.21c). As restrições (5.23b) até (5.23d) garantem que a variável auxiliar $w_{j_o k\bar{o}}$ é igual a um, apenas se $x_{j_o} = 1$ e $x_{k\bar{o}} = 1$, e 0 caso contrário.

5.4 Experimentos computacionais

Foram realizados experimentos computacionais para verificar a qualidade de soluções e desempenho computacional do Modelo Linear 1 e do Modelo Linear 2. O modelo pseudopolinomial de Furini, Malaguti e Thomopoulos (2016), formulação estado-da-arte da literatura para o PCBGR, é considerado para fins de comparação. Esse modelo, chamado daqui em diante de Modelo FMT, foi implementado com as reduções *cut-position* e *redundant-cut* propostas por esses autores. Os três modelos (Modelo Linear 1, Modelo Linear 2 e Modelo FMT) foram codificados em C++ usando a biblioteca *Concert Studio* do *IBM CPLEX Optimization Studio v12.8*. Os experimentos foram realizados em um computador de processador Intel Xeon E5-2680v2 (2.8 GHz), limitado a 20 *threads*, 16 GB of RAM, sob sistema operacional CentOS Linux 7.2.1511. A execução do *solver* foi limitado a 900 segundos para os modelos, e o símbolo “tl” é usado nas tabelas para quando esse limite foi alcançado; esse intervalo de tempo foi considerado como aceitável

Tabela 5.4: Três conjuntos de instâncias para o PCBGR.

Conjunto	Instâncias
conjunto A	cgcut1-3 (ponderada), OF1-2, wang20, okp1-5 (ponderada), gcut1-12
conjunto B	APT30-39, APT40-49 (ponderada), gcut13
conjunto C	300 instâncias geradas aleatoriamente

no contexto de sistemas produtivos.

Essa seção é dividida em três partes. Cada uma delas considera um conjunto de instâncias de *benchmark*, descritas na Tabela 5.4. As instâncias dos conjuntos A e B foram obtidas da literatura. Elas estão disponíveis em repositórios *online* como, por exemplo, OR-Library² e ESICUP³. Todas as instâncias do conjunto A foram consideradas nos experimentos computacionais de Furini, Malaguti e Thomopulos (2016). Eles argumentaram que as instâncias do conjunto B são intratáveis ao Modelo FMT, visto que elas têm itens muito pequenos em comparação ao objeto, resultando em muitas variáveis. O conjunto C é composto de 300 instâncias aleatoriamente geradas, que foram criadas usando o gerador de instâncias de Silva, Oliveira e Wäscher (2014).

Note que os modelos propostos requerem os parâmetros \bar{n} , UB_1 e UB_2 , no qual o último é usado para instâncias ponderadas apenas. Cada um desses parâmetros foi obtido ao resolver um problema da mochila unidimensional, exceto seja dito o contrário. Por exemplo, o Modelo (5.1) foi resolvido usando o CPLEX para determinar o parâmetro $\bar{n} = \sum_{i \in I} q_i^*$, sendo que q_i^* é a solução ótima obtida por este modelo. Para obter os parâmetros UB_1 e UB_2 , o Modelo (5.1) foi resolvido usando o CPLEX, considerando as áreas e os valores dos itens como coeficientes na função objetivo, respectivamente (isto é, $UB_1 = \sum_{i \in I} (l_i w_i) q_i^*$ e $UB_2 = \sum_{i \in I} v_i q_i^*$). Para cada um desses cálculos, o tempo de execução do *solver* foi limitado a 60 segundos. Quando tal limite de tempo foi alcançado, o respectivo parâmetro foi definido como o melhor limitante superior encontrado pelo *solver* durante a busca arredondado para baixo.

5.4.1 Resultados ao conjunto de instâncias A

O conjunto de instâncias A é composto de 23 instâncias clássicas da literatura de tamanho médio relacionadas ao PCBGR. Na Tabela 5.5 para cada instância, são reportados seu nome (Instância), o comprimento do objeto (L), a largura do objeto (W), o número de tipos de itens (m), número de total de itens ($n = \sum_{i \in I} u_i$), o limitante superior \bar{n} e o valor da função objetivo da solução ótima (OPT). Os valores OPT foram obtidos de Furini, Malaguti e Thomopulos (2016). Como esses autores, uma cópia por tipo de item $i \in I$ foi considerada para as instâncias gcut (isto é, $u_i = 1$). As instâncias cgcut1-3, OF1-2, wang20 and okp1-5 são caracterizadas por terem itens com área menor

²<http://people.brunel.ac.uk/~mastjjb/jeb/orlib>

³<https://www.euro-online.org/websites/esicup/data-sets/>

do que 25% da área do objeto, enquanto as instâncias *gcut*1-12 tem itens com dimensões uniformemente amostradas nos intervalos $[\bar{L}/4, 3\bar{L}/4]$ e $[\bar{W}/4, 3\bar{W}/4]$; para mais detalhes, o leitor é direcionado a Christofides e Whitlock (1977), Wang (1983), Beasley (1985a), Oliveira e Ferreira (1990) e Fekete e Schepers (1997).

Na Tabela 5.5 são reportados os resultados para todas as instâncias do conjunto A. Para cada um desses três modelos, são reportados o número de variáveis (*var*), número de restrições (*rest*), *gap* de otimalidade relativo como uma porcentagem (*gap*[%]), tempo de processamento de solução em segundos (*tempo*[s]), e o valor da função objetivo da relaxação linear (RL) para cada instância. O *gap* relativo de otimalidade é calculado como $(OPT - OFV)/(OPT + 10^{-10}) * 100$, sendo que *OFV* é o valor da função objetivo do correspondente modelo ao fim da execução, que pode não ser ótimo se o tempo limite foi atingido. O símbolo “#” foi usado para indicar que o tempo limite foi alcançado quando se resolvia a relaxação linear do respectivo modelo. Os valores da relaxação linear do Modelo Linear 2 não são reportados, porque eles são iguais àqueles do Modelo Linear 1, conforme discutido a seguir. Observe que os valores da relaxação linear do Modelo Linear 1 são inteiros, porque os parâmetros UB_1 e UB_2 também são inteiros para todos as instâncias A.

Particularmente para os experimentos reportados nessa seção, os parâmetros \bar{n} , UB_1 e UB_2 foram obtidos ao resolver uma versão relaxada e adaptada do “Contiguous relaxations based model of the Two-dimensional Orthogonal Packing Problem” do livro-texto de Scheithauer (2018), dado pelo Modelo (5.16), mas sem as restrições (5.16g). Esse modelo é pseudopolinomial e fornece limitantes superiores melhores que o problema da mochila unidimensional.

É importante destacar que a formulação Grelha do Capítulo 3 e as formulações Modelo Linear 1 e Modelo Linear 2 propostas neste capítulo se destacam em cenários cujos itens são relativamente grandes em comparação às dimensões do objeto a ser cortado como, por exemplo, as instâncias *gcut*. No contexto da formulação Grelha, esse tipo de instância conduz à discretização de conjuntos normais de poucos elementos e, assim, menores são os números de variáveis e restrições. Por outro lado, no contexto das formulações *bottom-up*, esse tipo de instância tende a ter solução ótima formada de poucas construções horizontais e/ou verticais; em outras palavras, o parâmetro \bar{n} tende a ser pequeno e, por consequência, menores são os números de variáveis e restrições dos modelos. No entanto, a comparação dos resultados das Tabelas 3.2 (do Capítulo 3) e 5.5 mostram que o desempenho médio das formulações *bottom-up* é superior à formulação Grelha nas instâncias *gcut*. Além disso, as formulações *bottom-up* dificilmente atingem o limite de memória computacional, o que acontece rapidamente na formulação Grelha, à medida que se aumenta a quantidade de tipos de itens. Dessa forma, optou-se, a seguir, por comparar diretamente os resultados das formulações Modelo Linear 1 e Modelo Linear 2 à formulação de Furini, Malaguti e Thomopulos (2016), que representa o estado-da-arte da literatura.

Os resultados da Tabela 5.5 indicam que o Modelo FMT tem melhor desempenho que o Modelo Linear 1 e o Modelo Linear 2 para as primeiras 11 instâncias da tabela (exceto para as instâncias okp2 e okp3), em média, enquanto que os modelos propostos alcançaram melhor performance que o Modelo FMT nas 12 instâncias gcut. Por exemplo, não foi possível provar otimalidade em nenhuma instâncias okp com o Modelo Linear 1, e apenas em uma instância com Modelo Linear 2, enquanto isso a otimalidade foi provada com Modelo FMT em 3 das 5 instâncias. Em contraste, para as instâncias gcut, a otimalidade foi provada em todas as instâncias usando Modelo Linear 1, em todas exceto uma instância usando Modelo Linear 2, e em apenas 4 instâncias usando Modelo FMT. Observe que o CPLEX atingiu o limite de memória na instância gcut4 com Modelo FMT, o que é representada pelo símbolo “*”, e para 7 dentre 12 instâncias o *solver* não foi capaz de obter uma solução de valor não-nulo dentro do tempo limite, o que é indicado por gap de 100%. Além disso, o *solver* atingiu tempo limite quando resolvia a relaxação linear de Modelo FMT para as instâncias gcut6, gcut7, gcut8, gcut11 e gcut12. A relaxação linear do Modelo Linear 1 e do Modelo Linear 2 forneceram os mesmos limitantes superiores por causa das desigualdades válidas relacionadas aos parâmetros UB_1 e UB_2 (sem elas, os modelos propostos forneceriam limitantes superiores mais fracos). Essas duas desigualdades válidas poderiam ser facilmente adaptadas e incluídas em Modelo FMT, mas elas não seriam relevantes visto que essa formulação já é capaz de fornecer bons limitantes superiores para a relaxação linear.

Para a maioria das instâncias, os modelos propostos apresentam significativamente menos variáveis e restrições que o Modelo FMT. Os desempenhos do Modelo Linear 1 e do Modelo Linear 2 são diretamente relacionados ao parâmetro \bar{n} , isto é, ao número possível de padrões a ser construído. Por exemplo, o parâmetro \bar{n} não excedeu 10 itens para as instâncias gcut, visto que elas são caracterizadas por itens relativamente grandes em relação às dimensões do objeto. No entanto, o parâmetro \bar{n} é de até 32 para as instâncias okp, o que explica o aumento nos números de variáveis e restrições em comparação às instâncias gcut, e também a otimalidade não provada. Note que a definição do número de nós/retângulos no conjunto J do Modelo Linear 1 e do Modelo Linear 2 é diferente, o que explica a diferença dos números de variáveis desses modelos. Por outro lado, o Modelo FMT parece ser mais dependente da dimensão do objeto ao atingir dezenas de milhões de variáveis para algumas instâncias gcut, o que tende a inviabilizar a abordagem para esses cenários.

Experimentos computacionais também foram executados com tempo limite de 3600 segundos, considerando apenas as instâncias da Tabela 5.5 em que a otimalidade não foi provada. Modelo Linear 1 e Modelo Linear 2 tiveram melhoria marginal em apenas duas instâncias cada. Com tal limite de tempo, a otimalidade foi provada com Modelo FMT em quase 3000 segundos às instâncias gcut2 e gcut3.

Tabela 5.5: Resultados dos modelos para o conjunto de instâncias A.

Instância	\bar{L}	\bar{W}	m	n	\bar{n}	OPT	Modelo Linear 1				Modelo Linear 2				Modelo FMT					
							var	rest	gap[%]	tempo[s]	RL	var	rest	gap[%]	tempo[s]	var	rest	gap[%]	tempo[s]	RL
cgcut1	15	10	7	16	13	244	1,015	818	0.00	1.94	244	390	910	0.00	5.70	752	147	0.00	0.05	244.5
cgcut2	40	70	10	23	18	2,892	3,394	2,044	5.77	tl	2,896	765	1,798	1.24	tl	49,724	2,027	0.00	8.52	2,897.4
cgcut3	40	70	19	62	10	1,860	1,237	441	0.00	11.23	1,860	477	1,039	0.00	24.56	32,727	1,879	0.00	12.67	1,919.2
OF1	70	40	10	24	10	2,737	874	532	0.00	tl	2,747	345	782	0.88	tl	34,908	2,108	0.00	6.05	2,754.6
OF2	70	40	10	23	11	2,690	706	431	0.00	111.49	2,690	297	669	0.00	15.44	31,795	2,120	0.00	6.89	2,734.1
wang20	70	40	19	42	10	2,721	1,237	440	0.00	216.15	2,726	423	930	0.00	tl	32,725	1,879	0.00	6.86	2,721.0
okp1	100	100	15	50	32	27,589	31,057	13,166	0.00	tl	28,402	2,356	5,629	6.66	tl	254,392	7,962	0.00	42.58	28,055.2
okp2	100	100	30	30	16	22,502	6,430	1,474	2.60	tl	23,713	825	1,878	2.60	tl	282,311	9,536	100.00	tl	24,496.1
okp3	100	100	30	30	14	24,019	4,446	1,028	0.00	tl	24,191	676	1,528	1.15	tl	257,694	8,386	4.26	tl	26,283.0
okp4	100	100	33	61	29	32,893	45,253	9,340	1.61	tl	32,893	2,674	6,112	0.00	113.26	286,622	9,082	0.00	231.60	32,898.9
okp5	100	100	29	97	31	27,923	50,373	11,726	7.70	tl	27,923	2,955	6,782	8.22	tl	366,077	9,535	0.00	30.63	27,974.4
gcut1	250	250	10	10	4	48,368	106	71	0.00	0.11	48,368	51	117	0.00	0.10	2,814	525	0.00	0.41	50,739.5
gcut2	250	250	20	20	6	59,307	416	153	0.00	2.94	59,798	150	337	0.00	3.80	626,515	28,813	100.00	tl	60,188.5
gcut3	250	250	30	30	7	60,241	798	206	0.00	9.95	61,275	249	554	0.00	82.16	1,127,094	32,711	100.00	tl	60,638.5
gcut4	250	250	50	50	9	60,942	2,338	370	0.00	63.51	61,380	516	1,132	0.00	tl	2,024,833	35,226	*	*	*
gcut5	500	500	10	10	5	195,582	154	100	0.00	0.27	195,582	74	168	0.00	0.57	170,931	24,480	0.00	285.19	218,543.0
gcut6	500	500	20	20	5	236,305	284	110	0.00	2.11	236,305	114	258	0.00	0.26	994,207	73,834	100.00	tl	1,232,057.0#
gcut7	500	500	30	30	6	238,974	606	163	0.00	5.66	240,143	200	447	0.00	8.45	2,512,109	100,575	100.00	tl	2,004,791.0#
gcut8	500	500	50	50	8	245,758	1,818	298	0.00	3.88	245,758	441	969	0.00	57.62	9,809,683	136,574	100.00	tl	2,805,462.0#
gcut9	1,000	1,000	10	10	5	919,476	154	100	0.00	0.84	939,600	74	168	0.00	8.75	311,561	36,162	0.00	731.34	937,266.0
gcut10	1,000	1,000	20	20	5	903,435	284	110	0.00	1.36	937,349	114	258	0.00	6.55	20,735	2,557	0.00	8.35	962,059.7
gcut11	1,000	1,000	30	30	7	955,389	798	206	0.00	15.95	969,709	249	554	0.00	90.36	18,863,364	400,100	100.00	tl	6,536,520.0#
gcut12	1,000	1,000	50	50	7	970,744	1,298	226	0.00	12.08	979,521	369	814	0.00	249.11	31,289,676	489,478	100.00	tl	12,521,992.0#
Médias						219,243.09	6,742.43	1,893.61	0.77	293.89	222,829.26	642.78	1,471.00	0.90	342.03	3,016,663.00	61,552.00	36.56	430.51	1,251,965.25

5.4.2 Resultados ao conjunto de instâncias B

Os modelos propostos também foram analisados usando instâncias maiores da literatura, isto é, as instâncias APT e gcut13. Particularmente, as instâncias APT são caracterizadas por ter itens com dimensões uniformemente amostradas nos intervalos $[0, 05\bar{L}, 0, 4\bar{L}]$ e $[0, 05\bar{W}, 0, 4\bar{W}]$, enquanto as dimensões do objeto são uniformemente amostradas no intervalo $[100, 1000]$; para mais detalhes, o leitor é direcionado a Álvarez-Valdés, Parajón e Tamarit (2002a). De acordo com Furini, Malaguti e Thomopulos (2016), seu modelo pseudopolinomial torna-se intratável diante das instâncias APT, porque elas têm itens muito pequenos, o que conduz a um número de variáveis que é muito grande na prática. De fato, nos experimentos computacionais preliminares, o *solver* teve estouro de memória em quase todas essas instâncias que apresentaram dezenas de milhões de variáveis. Nas poucas instâncias tratáveis, o *solver* sempre obteve solução de valor nulo. Assim, os modelos propostos são comparados entre si nessa seção.

A Tabela 5.6 mostra os resultados para as instâncias do conjunto B. Para cada instância, também é reportado o valor da solução ótima ou melhor limitante superior conhecido (coluna OPT/UP), conforme obtido de Velasco e Uchoa (2019) e Dolatabadi, Lodi e Monaci (2012). Na coluna OPT/UP, todos os valores são ótimos, exceto aqueles com símbolo “•”, que são limitantes superiores. Para cada um dos três modelos, são reportados o desvio de *gap* relativo como uma porcentagem (coluna des[%]), que é calculado como $(OPT/UP - OFV)/(OPT/UP + 10^{-10}) * 100$, sendo que *OPT/UP* é a entrada correspondente na coluna OPT/UP para a instância. Observe que quando o valor OPT/UP da instância corresponde a seu valor de solução ótima, então o *des* é igual ao *gap* relativo de otimalidade. São reportados resultados de experimentos sem fornecer uma solução inicial (sem sol. inicial) e fornecendo um padrão de corte 2-estágios como solução inicial (com sol. inicial), obtida ao resolver o Modelo 1 de (LODI; MARTELLO; MONACI, 2002) usando o CPLEX em até 60 segundos. De fato, o modelo de (LODI; MARTELLO; MONACI, 2002) foi executado duas vezes para obter o melhor padrão de corte considerando tiras horizontais e verticais.

Quando nenhuma solução inicial é fornecida aos modelos, os resultados da Tabela 5.6 mostram que o *solver* foi capaz de encontrar soluções com desvio médio de apenas 3.40% com o Modelo Linear 2, enquanto de 47.06% com o Modelo Linear 1. O desempenho de ambos os modelos são diretamente dependentes do parâmetro \bar{n} , que é significativamente maior para as instâncias do conjunto B em comparação àquelas do conjunto A. Particularmente, os números de variáveis e restrições de Modelo Linear 1 são, em média, 1.092.815,48 and 169.690,62, respectivamente. Em contraste, o Modelo Linear 2 apresentou médias de variáveis de 9.592,24 e de restrições de 22.483,76. Assim, apesar da performance do *solver* com o Modelo Linear 1 ter sido ligeiramente superior que com o Modelo Linear 2 para as instâncias do conjunto A, o Modelo Linear 2 superou o Modelo Linear 1 no conjunto B composto de instâncias maiores, isto é, que requerem mais

subpadrões diante de valores maiores de \bar{n} .

Os padrões de corte 2-estágios forneceram soluções iniciais satisfatórias, com média de desvio ($\text{des}[\%]$) de 1.48%. Dessa forma, o *solver* com o Modelo Linear 1 não foi capaz de melhorar a solução inicial fornecida para nenhuma das instâncias dentro do intervalo de tempo limite. No entanto, o *solver* com o Modelo Linear 2 melhorou as soluções das instâncias APT35, ATP36, ATP42, ATP44, ATP45, ATP46, ATP47 e ATP49. Além disso, ele encontrou a solução ótima da instância ATP45 com o Modelo Linear 2, apesar de não ter sido capaz de provar a otimalidade dentro do intervalo de tempo limite. Apesar do *solver* não ter provado otimalidade de nenhuma das instâncias do conjunto B, ele foi capaz de obter soluções factíveis satisfatórias usando os modelos propostos para essas instâncias maiores. Foram realizados também experimentos computacionais considerando 3600 segundos como tempo limite, mas o Modelo Linear 1 e o Modelo Linear 2 apenas mostraram melhorias marginais.

Tabela 5.6: Resultados dos modelos para o conjunto de instâncias B.

Instância	\bar{L}	\bar{W}	m	n	\bar{n}	OPT/UP	Modelo Linear 1						Modelo Linear 2						
							var	rest	sem sol. inicial		com sol. inicial		RL	var	rest	sem sol. inicial		com sol. inicial	
									des[%]	tempo[s]	des[%]	tempo[s]				des[%]	tempo[s]	des[%]	tempo[s]
APT30	927	152	38	192	65	140,904	1,166,598	210,008	32.32	tl	0.50	tl	140,904	11,040	26,088	4.70	tl	0.50	tl
APT31	856	964	51	258	73	823,976	2,562,707	348,159	85.24	tl	0.50	tl	825,184	14,940	34,973	2.73	tl	0.50	tl
APT32	307	124	56	249	70	38,068	2,335,484	289,953	89.27	tl	0.25	tl	38,068	14,559	33,799	3.17	tl	0.25	tl
APT33	241	983	44	224	52	236,611	529,596	82,920	11.18	tl	0.44	tl	236,903	8,313	19,171	2.05	tl	0.44	tl
APT34	795	456	27	130	75	361,398	1,580,527	392,393	99.44	tl	1.01	tl	362,520	12,099	29,555	2.92	tl	1.01	tl
APT35	960	649	29	153	55	621,021	447,669	103,987	28.74	tl	1.06	tl	623,040	7,479	17,797	3.52	tl	0.49	tl
APT36	537	244	28	153	52	130,744	345,388	82,904	54.72	tl	1.13	tl	131,028	6,783	16,095	2.48	tl	0.99	tl
APT37	440	881	43	222	56	387,276	701,773	112,310	50.85	tl	0.38	tl	387,640	9,240	21,440	4.31	tl	0.38	tl
APT38	731	358	40	202	65	261,395	1,224,928	210,010	36.22	tl	0.89	tl	261,698	11,296	26,602	3.57	tl	0.89	tl
APT39	538	501	33	163	42	268,750	173,703	35,754	1.50	tl	0.88	tl	269,538	5,166	11,966	1.93	tl	0.88	tl
APT40	683	138	56	290	67	67,154	1,927,048	239,257	87.15	tl	2.34	tl	68,518	13,893	32,069	3.63	tl	2.34	tl
APT41	837	367	36	177	47	206,542	290,964	55,151	21.62	tl	2.05	tl	215,699	6,463	14,989	2.05	tl	2.05	tl
APT42	167	291	59	325	80	33,655●	4,446,959	524,878	100.00	tl	1.91	tl	34,098	18,486	43,117	8.50	tl	1.87	tl
APT43	362	917	49	259	72	216,870●	2,323,711	328,073	87.34	tl	2.22	tl	222,570	14,413	33,777	3.52	tl	1.83	tl
APT44	223	496	39	196	60	73,868	852,675	149,762	51.73	tl	3.96	tl	77,768	9,499	22,403	5.34	tl	3.20	tl
APT45	188	578	33	156	54	74,691	469,943	96,691	47.27	tl	0.65	tl	77,891	7,685	18,109	0.00	tl	0.00	tl
APT46	416	514	42	197	55	149,911	635,402	104,001	13.90	tl	1.93	tl	154,488	8,883	20,619	1.09	tl	0.17	tl
APT47	393	554	43	204	44	150,234	267,163	42,776	4.81	tl	3.94	tl	157,053	6,278	14,365	2.21	tl	2.74	tl
APT48	931	254	34	167	49	167,660	325,474	65,113	20.43	tl	1.33	tl	173,358	6,696	15,637	5.70	tl	1.33	tl
APT49	759	449	25	119	52	219,354	310,849	82,902	61.87	tl	3.45	tl	226,346	6,426	15,379	5.92	tl	3.12	tl
gcut13	3000	3000	32	32	26	8,618,394●	30,564	6,501	2.71	tl	0.30	tl	9,000,000	1,800	4,209	1.99	tl	0.30	tl
Médias						630,879.81	1,092,815.48	169,690.62	47.06	900.00	1.48	900.00	651,633.90	9,592.24	22,483.76	3.40	900.00	1.20	900.00

5.4.3 Resultados ao conjunto de instâncias C gerado aleatoriamente

O gerador de instâncias 2DCPackGen de Silva, Oliveira e Wäscher (2014) foi usado para gerar aleatoriamente 300 instâncias não-ponderadas. O 2DCPackGen foi desenvolvido para amostrar a distribuição beta de acordo com as dimensões mínimas e máximas do objeto e dos itens para diversos problemas de corte bidimensionais. Os autores propuseram valores aos parâmetros α e β da distribuição beta para considerar diferentes formas para o objeto e itens, que são representados por um número de identificação (id). Para uma descrição completa, o leitor é direcionado a Silva, Oliveira e Wäscher (2014).

Os descritores e valores correspondentes usados para gerar as instâncias do conjunto C são apresentados a seguir:

- dimensões mínima e máxima do objeto: 100 e 200;
- dimensões mínima e máxima dos itens: 25 e 100;
- formato id do objeto (*os*): 2 e 6;
- formato id dos itens (*is*): 1, 2, 3, 4, 6, 7, 8, 11, 12 e 16;
- número de tipos de itens (*m*): 10, 20 e 40;
- número máximo de cópias a ser produzido: distribuição uniforme no intervalo $[1, 5]$;

Esses valores aos descritores foram considerados para gerar um conjunto de instâncias que o *solver* pudesse obter boas soluções para os três modelos (isto é, Modelo Linear 1, Modelo Linear 2 e Modelo FMT). Observe que as instâncias do conjunto B são praticamente intratáveis para Modelo FMT, e o *solver* com Modelo Linear 1 obteve soluções satisfatórias apenas quando uma solução inicial foi fornecida. Sobre o formato do objeto e dos itens, $os = 6$ significa que o objeto foi amostrado como “grande e quadrado” em seu intervalo, enquanto $is = 7$ significa itens amostrados com comprimentos médios e largos em seus intervalos; veja Silva, Oliveira e Wäscher (2014) para uma descrição completa. Particularmente, o 2D_R_CG_SLOPP corresponde ao problema de id 2 no 2DCPackGen. Foi utilizado 2018 como valor para a semente. Uma classe é caracterizada pelo formato id do objeto, o formato id dos itens e o número de tipos de itens em um total de 60 ($= 2 \times 10 \times 3$) classes. Cada classe consiste de 5 instâncias, e assim o conjunto C tem 300 ($= 60 \times 5$) instâncias. Para cada tipo de item $i \in I$, o número final máximo de cópias a serem produzidas foi considerado como o mínimo entre o valor gerado pelo 2DCPackGen e o limitante geométrico (isto é, $\lfloor L/l_i \rfloor \lfloor W/w_i \rfloor$).

A Tabela 5.7 mostra os resultados médios das instâncias do conjunto C. Eles estão agregados em relação aos números de tipos de itens m , e assim apenas os descritores os e is são exibidos. Cada entrada nessa tabela é uma média sobre 15

instâncias. São reportados as médias para o valor da função objetivo (OFV), o *gap* de otimalidade relativo, o tempo de processamento computacional em segundos (tempo[s]), e a porcentagem de instâncias com otimalidade provada (opt%) para cada classe agregada. O Modelo FMT atingiu o limite de memória em 9 das 300 instâncias, logo os resultados médios correspondentes não são diretamente comparáveis. Foi utilizado o símbolo \oplus na coluna OFV para indicar as classes agregadas com instâncias em que Modelo FMT atingiu o limite de memória. O *gap* de otimalidade relativo é calculado como $(OFV_b - OFV)/(OFV_b + 10^{-10}) * 100$, sendo que OFV_b é o melhor limitante superior para o valor da função objetivo obtido pelo CPLEX durante a busca do correspondente modelo.

Tabela 5.7: Resultados dos modelos para o conjunto de instâncias C.

os	is	\bar{n}_{avg}	Modelo Linear 1				Modelo Linear 2				Modelo FMT			
			OFV	gap[%]	tempo[s]	opt%	OFV	gap[%]	tempo[s]	opt%	OFV	gap[%]	tempo[s]	opt%
2	1	16.27	21,370.73	2.41	788.87	13.33	21,285.47	3.45	tl	0.00	14,982.71 \oplus	22.29	633.88	60.00
2	2	9.33	20,971.73	0.96	275.80	73.33	20,962.00	2.03	653.62	40.00	19,057.40	6.67	347.86	93.33
2	3	13.13	21,282.47	2.49	728.42	20.00	21,323.00	2.43	785.01	13.33	15,257.86 \oplus	21.79	495.14	66.67
2	4	16.47	21,343.60	2.34	781.70	13.33	21,310.00	3.09	841.07	6.67	10,567.86 \oplus	43.10	634.51	46.67
2	6	5.07	19,968.27	0.00	12.76	100.00	19,968.27	0.00	13.46	100.00	19,968.27	0.00	59.78	100.00
2	7	7.67	20,458.33	0.08	98.83	93.33	20,468.07	1.03	288.53	73.33	18,533.27	6.67	162.80	93.33
2	8	10.33	20,628.47	0.96	363.72	66.67	20,628.47	3.54	666.10	26.67	18,692.67	6.67	176.29	93.33
2	11	11.07	21,558.67	1.93	583.26	40.00	21,657.20	1.95	780.53	13.33	14,289.07	26.67	473.69	73.33
2	12	13.53	21,370.60	2.49	779.67	20.00	21,296.13	3.00	781.71	13.33	15,314.00	21.28	503.29	66.67
2	16	15.80	21,348.33	3.01	780.77	13.33	21,252.80	3.45	781.55	13.33	13,722.73	27.89	560.63	66.67
6	1	20.53	28,072.60	3.51	tl	0.00	27,980.67	3.72	tl	0.00	4,551.80	81.10	867.86	6.67
6	2	12.27	27,772.00	1.79	587.43	40.00	27,845.73	2.60	784.26	13.33	13,668.07 \oplus	42.86	525.67	53.33
6	3	16.60	28,120.93	2.89	786.80	13.33	28,050.40	3.66	tl	0.00	7,899.08 \oplus	67.80	731.37	20.00
6	4	20.47	28,143.33	3.23	844.07	6.67	27,989.93	3.90	tl	0.00	6,671.86 \oplus	71.43	784.65	26.67
6	6	6.80	27,767.07	0.00	21.74	100.00	27,767.07	0.30	179.35	86.67	18,983.87	26.67	315.55	73.33
6	7	9.93	28,296.33	1.48	526.65	53.33	28,373.80	2.22	699.60	26.67	18,012.13	33.33	547.71	66.67
6	8	13.20	28,049.33	2.81	689.52	26.67	28,060.73	3.22	787.19	13.33	13,792.67	49.02	650.09	40.00
6	11	13.67	28,330.60	2.83	780.55	13.33	28,314.80	2.88	781.73	13.33	8,071.93	66.67	734.07	33.33
6	12	16.60	28,342.40	2.84	788.29	13.33	28,259.87	3.52	844.61	6.67	8,614.71 \oplus	64.29	727.35	33.33
6	16	19.13	28,203.87	2.76	783.41	13.33	28,149.27	2.85	833.26	13.33	9,167.93	61.36	786.11	20.00
Médias			24,569.98	2.04	595.11	36.67	24,547.18	2.64	705.08	23.67	13,490.99	37.38	535.92	56.67

O CPLEX provou otimalidade em 56.67% das instâncias para o Modelo FMT, 36.67% para o Modelo Linear 1 e 22.67% para o Modelo Linear 2. Para esses três modelos, o *solver* provou otimalidade em todas as instâncias das classes $os = 2$ e $is = 6$. Essas classes correspondem a instâncias com objeto longo e estreito, e itens grandes e quadrados. Nesse contexto, o Modelo Linear 2 obteve os melhores tempos de processamento, seguido de Modelo Linear 1 e Modelo FMT. O CPLEX com Modelo Linear 1 também conduziu a soluções ótimas para a classe com $os = 6$ e $is = 6$ (objeto grande e quadrado, itens grandes e quadrados). Em 35.33% das instâncias, Modelo FMT resultou em OFV igual a zero, o que explica os valores da correspondente coluna gap[%]. De fato, semelhante ao observado para algumas instâncias *gcut* na Seção 5.4.1, o *solver* com Modelo FMT não foi capaz de obter soluções de valor não-nulo dentro do tempo limite; para essas instâncias, ele gastou todo o tempo de processamento na fase de pré-processamento do *branch-and-cut* e/ou resolvendo o nó raiz. A média do número de variáveis foi maior que 1,2 milhão para essas instâncias.

Em resumo, os modelos propostos foram capazes de fornecer soluções sa-

tisfatórias no contexto de um *solver* de propósito geral. A árvore binária explícita do Modelo Linear 1 contribui para que o CPLEX provasse otimalidade mais rapidamente para a maioria das instâncias moderadas. Por outro lado, a formulação compacta Modelo Linear 2 superou a formulação pseudopolinomial Modelo Linear 1 para instâncias maiores. O Modelo FMT parece obter limitantes de relaxação linear mais apertados, o que contribui para uma prova mais rápida de otimalidade; ele também mostrou boa performance nas instâncias moderadas que não requerem muitas variáveis/restrições.

O Modelo Linear d-estágios 1 e o Modelo Linear d-estágios 2 foram analisados também com o conjunto de instâncias C ao considerar $d = 2$ estágios guilhotinados. A Tabela 5.8 mostra os resultados relacionados aos padrões 2-estágios. O modelo 1 de Lodi, Martello e Monaci (2002) foi tomado como modelo de *benchmark* para essas abordagens, descritas na tabela pela coluna Modelo LM 1. Como esperado, o modelo *benchmark* de 2-estágios superou os resultados dos modelos propostos, dado que ele foi especificamente projetado para este tipo de padrões de corte. Ainda assim, para as classes com otimalidade provada em todas as instâncias na Tabela 5.7, o CPLEX provou otimalidade mais rápido com o Modelo Linear d-estágios 1 e o Modelo Linear d-estágios 2, do que com o Modelo FMT.

Tabela 5.8: Resultados dos modelos para o conjunto de instâncias C para padrões 2-estágios.

os	is	\bar{n}_{avg}	Modelo Linear 1				Modelo Linear 2				Modelo LM 1			
			OFV	gap[%]	tempo[s]	opt%	OFV	gap[%]	tempo[s]	opt%	OFV	gap[%]	tempo[s]	opt%
2	1	16.27	21,111.73	3.12	782.69	13.33	21,017.87	4.01	879.54	6.67	21,190.27	0.00	9.59	100.00
2	2	9.33	20,891.20	0.11	111.84	93.33	20,891.20	2.32	586.58	40.00	20,905.13	0.00	5.08	100.00
2	3	13.13	21,156.87	1.40	526.82	53.33	21,126.53	3.25	783.96	13.33	21,172.27	0.00	5.88	100.00
2	4	16.47	21,227.40	2.54	741.68	20.00	21,048.13	3.55	807.35	13.33	21,287.13	0.00	9.67	100.00
2	6	5.07	19,916.47	0.00	6.59	100.00	19,916.47	0.00	9.69	100.00	19,916.47	0.00	0.61	100.00
2	7	7.67	20,299.33	0.00	16.23	100.00	20,299.33	1.39	264.94	73.33	20,299.33	0.00	2.75	100.00
2	8	10.33	20,324.60	1.60	263.42	73.33	20,337.20	4.01	660.95	33.33	20,337.20	0.00	2.87	100.00
2	11	11.07	21,383.60	1.07	349.09	73.33	21,302.27	3.34	781.02	13.33	21,383.60	0.00	6.39	100.00
2	12	13.53	21,056.73	2.44	667.83	33.33	21,015.60	3.90	781.32	13.33	21,081.73	0.00	3.97	100.00
2	16	15.80	21,102.20	3.24	742.80	26.67	21,096.40	3.71	780.70	13.33	21,190.27	0.00	5.73	100.00
6	1	20.53	28,035.40	2.78	834.86	13.33	27,531.93	5.21	900.00	0.00	28,170.47	0.00	8.59	100.00
6	2	12.27	27,686.20	1.34	420.36	66.67	27,643.07	3.20	785.13	13.33	27,689.80	0.00	7.78	100.00
6	3	16.60	28,053.20	3.00	782.55	13.33	27,887.80	3.84	858.21	6.67	28,163.47	0.00	7.07	100.00
6	4	20.47	27,759.67	3.91	786.66	13.33	27,829.80	4.48	900.00	0.00	28,144.93	0.00	7.00	100.00
6	6	6.80	27,657.20	0.00	6.83	100.00	27,657.20	0.17	155.94	93.33	27,657.20	0.00	0.99	100.00
6	7	9.93	28,162.73	0.06	168.50	93.33	28,162.73	2.86	676.86	26.67	28,162.73	0.00	5.32	100.00
6	8	13.20	27,980.93	2.21	623.16	33.33	27,894.93	3.62	783.78	13.33	27,990.00	0.00	6.65	100.00
6	11	13.67	28,251.73	2.16	641.19	33.33	28,126.27	3.45	781.67	13.33	28,308.40	0.00	9.06	100.00
6	12	16.60	28,100.60	3.34	772.03	20.00	27,971.87	4.44	842.99	6.67	28,326.33	0.00	5.64	100.00
6	16	19.13	27,883.60	3.31	780.85	13.33	27,589.00	4.26	804.84	13.33	28,075.20	0.00	31.83	100.00
Averages			24,402.07	1.88	501.30	49.33	24,317.28	3.25	691.27	25.33	24,472.60	0.00	7.12	100.00

Foram realizados experimentos computacionais considerando 3600 segundos como tempo limite para algumas classes agregadas das Tabelas 5.7 e 5.8, nomeadamente $os = 6$ e $is = 1$, $os = 6$ e $is = 4$, e $os = 6$ e $is = 16$. Observe que a otimalidade foi provada em menos do que um terço das instâncias na Tabela 5.7. Novamente, o Modelo Linear 1 e o Modelo Linear 2 tiveram melhorias marginais. A otimalidade foi provada com o Modelo FMT em quase metade das instâncias. Seu gap de otimalidade médio foi de 25%, mostrando que mesmo diante de um maior tempo limite, o *solver* gastou a maior parte

de seu tempo de processamento na solução do nó raiz.

5.5 Pseudocódigos para gerar a alocação dos modelos *bottom-up*

O Algoritmo 5.5.1 é um pseudocódigo para gerar um padrão de corte a partir de uma solução do Modelo Não-linear 1, do Modelo Linear 1 ou do Modelo Linear d-estágios 1. Observe que o índice j varia de 1 até \mathcal{N} na linha 2, visto que o padrão de corte final (isto é, o nó 1) é posicionado em $(0,0)$. Particularmente, foi escolhido arbitrariamente que os nós-filho da esquerda j^- estejam abaixo ou à esquerda dos nós-filho da direita j^+ . A linha 16 exhibe as posições onde os itens selecionados devem ser posicionados no padrão de corte final

Algoritmo 5.5.1: Gerar um padrão de corte a partir da solução do Modelo Hierárquico.

Data: Instância, árvore binária, valores das variáveis z_{ji} , x_{jo} , L_j e W_j .

```

1  $x_1 \leftarrow 0, y_1 \leftarrow 0;$ 
2 for  $j \in J \setminus \bar{J}, o \in O \mid x_{jo} = 1$  do
3    $x_{j^-} \leftarrow x_j, y_{j^-} \leftarrow y_j;$ 
4   if  $j \in J_A \cup J_B$  then
5     if  $o = h$  then
6        $x_{j^+} \leftarrow x_j + L_{j^-}, y_{j^+} \leftarrow y_j;$ 
7     else
8        $x_{j^+} \leftarrow x_j, y_{j^+} \leftarrow y_j + W_{j^-};$ 
9   else
10     $i' \leftarrow \underset{i \in I}{\arg\{z_{j-i} = 1\}};$ 
11    if  $o = h$  then
12       $x_{j^+} \leftarrow x_j + l_{i'}, y_{j^+} \leftarrow y_j;$ 
13    else
14       $x_{j^+} \leftarrow x_j, y_{j^+} \leftarrow y_j + w_{i'};$ 
15 for  $j \in J, i \in I \mid z_{ji} = 1$  do
16    $\lfloor$  nó  $j$  é uma cópia do tipo de item  $i$  posicionado em  $(x_j, y_j)$ ;
   Result: Posições  $(x_j, y_j)$  dos itens selecionados.

```

O Algoritmo 5.5.2 é um pseudocódigo para gerar um padrão de corte a partir de uma solução do Modelo Não-linear 2, do Modelo Linear 2 ou do Modelo Linear d-estágios 2. Observe que j varia de 1 até $\bar{n} - 1$ na linha 2, dado que o padrão de corte final (isto é, o retângulo 1) é posicionado em $(0,0)$. Particularmente, foi arbitrariamente definido que: (i) o subpadrão k_1 , caso exista, está abaixo ou à esquerda do subpadrão k_2 ou item i_2 ; e, (ii) o item i_2 , caso exista, está acima ou à direita do subpadrão k_1 ou item i_1 . A linha 30 exhibe as posições onde os itens selecionados deverão ser posicionados no padrão de corte final.

Algoritmo 5.5.2: Gerar um padrão de corte a partir da solução do Modelo Implícito.

Data: Instâncias, valores das variáveis z_{jip} , x_{jo} , y_{jk} , L_j e W_j .

```

1  $\alpha_1 \leftarrow 0, \beta_1 \leftarrow 0;$ 
2 for  $j \in J, o \in O \mid x_{jo} = 1$  do
3    $k_1 \leftarrow -1, i_1 \leftarrow -1, p_1 \leftarrow -1, l_1 \leftarrow -1, w_1 \leftarrow -1;$ 
4    $k_2 \leftarrow -1, i_2 \leftarrow -1, p_2 \leftarrow -1;$ 
   // Linhas 5 até 14 identificam os subpadrões menores e/ou cópias de itens que
   // compõem o subpadrao  $j$ 
5   for  $k \in J, j < k \mid y_{jk} = 1$  do
6     if  $l_1 = -1$  then
7        $k_1 \leftarrow k, l_1 \leftarrow L_k, w_1 \leftarrow W_k;$ 
8     else
9        $k_2 \leftarrow k;$ 
10  for  $i \in I, p \in Q_i \mid z_{jip} = 1$  do
11    if  $l_1 = -1$  then
12       $i_1 \leftarrow i, p_1 \leftarrow p, l_1 \leftarrow l_i, w_1 \leftarrow w_i;$ 
13    else
14       $i_2 \leftarrow i, p_2 \leftarrow p;$ 
   // Linhas 15 até 21 definem as posições dos subpadrões menores  $k_1$  e  $k_2$ , caso
   // existam
15  if  $k_1 > -1$  then
16     $\alpha_{k_1} \leftarrow \alpha_j, \beta_{k_1} \leftarrow \beta_j;$ 
17  if  $k_2 > -1$  then
18    if  $o = h$  then
19       $\alpha_{k_2} \leftarrow \alpha_j + l_1, \beta_{k_2} \leftarrow \beta_j;$ 
20    else
21       $\alpha_{k_2} \leftarrow \alpha_j, \beta_{k_2} \leftarrow \beta_j + w_1;$ 
   // Linhas 22 até 28 definem as posições dos itens  $i_1$  e  $i_2$ , caso existam
22  if  $i_1 > -1$  then
23     $x_{ji_1p_1} \leftarrow \alpha_j, y_{ji_1p_1} \leftarrow \beta_j;$ 
24  if  $i_2 > -1$  then
25    if  $o = h$  then
26       $x_{ji_2p_2} \leftarrow \alpha_j + l_1, y_{ji_2p_2} \leftarrow \beta_j;$ 
27    else
28       $x_{ji_2p_2} \leftarrow \alpha_j, y_{ji_2p_2} \leftarrow \beta_j + w_1;$ 
29 for  $j \in J, i \in I, p \in Q_i \mid z_{jip} = 1$  do
30    $\square$  retângulo  $j$  é a  $p$ -ésima cópia do tipo de item  $i$  posicionado em  $(x_{jip}, y_{jip});$ 

```

Result: Posições (x_{jip}, y_{jip}) dos itens selecionados.

5.6 Comentários finais do capítulo

Neste capítulo revisitou-se a ideia de Wang (1983), que propôs sucessivamente combinar itens para gerar soluções para o PCBGR. Baseado nesse conceito, foram propostas uma formulação pseudopolinomial de PNLIM, e uma formulação pseudopolinomial de PLIM equivalente, que completamente modelam o PCBGR. Essas formulações são pseudopolinomiais nos números de variáveis e restrições, dado que elas requerem uma estrutura de árvore binária explícita para suas definições. Tal árvore binária foi gerada por um procedimento proposto de fácil implementação. Foram propostas também uma formulação compacta de PNLIM e uma formulação compacta de PLIM equivalente, que modelam o PCBGR, e consideram a representação de árvore binária de forma implícita, isto é, que não requerem sua geração *a priori*.

Experimentos computacionais usando instâncias da literatura indicaram que os modelos propostos são apropriados para cenários nos quais o número máximo de itens que cabem em um objeto é pequeno, no contexto de certificados de otimalidade diante de *solvers* de propósito geral. Para além disso, neste contexto, uma vantagem dos modelos é que eles são capazes de resolver instâncias grandes, que são quase intratáveis para o modelo estado-da-arte atual da literatura. Adicionalmente, foram desenvolvidas restrições para limitar o número de estágios guilhotinados nos modelos propostos, o que pode ser útil para cenários nos quais importa o compromisso entre a utilização de material e as taxas de produção.

Como discutido anteriormente, as formulações desse capítulo facilmente abordam os casos de rotação de itens e de padrões d -estágios. Por outro lado, a consideração de placa com defeitos não é diretamente aplicável ao Modelo Linear 2. No Capítulo 7, como pesquisa futura, discute-se um possível caminho para o Modelo Linear 1 considerar o caso de objeto com defeitos.

Particularmente, esforços para reformular esses modelos em uma abordagem de corte *top-down*, em vez de *bottom-up*, são praticamente diretas. Tais novos modelos poderiam ser comparadas aos modelos *bottom-up* aqui propostos. No Apêndice A são apresentados os modelos da Seção 5.1 reformulados nessa perspectiva. Nota-se, no entanto, que os experimentos computacionais preliminares não motivaram novos desenvolvimentos para esses modelos.

No próximo capítulo, o objeto de análise passa a ser o PCTGR, que aqui é interpretado como a generalização tridimensional do PCBGR. O paradigma de modelagem deste capítulo, isto é, a combinação sucessiva dos itens é mantido.

Capítulo 6

Problema de corte tridimensional guilhotinado e restrito: propostas de formulações matemáticas e de método de solução

Neste capítulo aborda-se o Problema de Corte Tridimensional Guilhotinado e Restrito (PCTGR), que pode ser visto como uma generalização do PCBGR ao caso tridimensional. A formulação compacta baseada no conceito *bottom-up*, originalmente proposta no Capítulo 5, é estendida ao PCTGR na Seção 6.1, que também conta com uma formulação matemática ao PCTGR de 3-estágios. Como método de solução, propõe-se estender o algoritmo *bottom-up* de Wang (1983) ao caso tridimensional na Seção 6.2, no contexto de padrões não-estagiados e 3-estágios. Os experimentos computacionais consideram os casos irrestrito e restrito, a partir de instâncias de *benchmark* na Seção 6.3, e comparadas às abordagens em De Queiroz et al. (2012) e em Egeblad e Pisinger (2009).

Objetivos do capítulo:

- Estender a formulação compacta baseada no conceito *bottom-up*, originalmente proposta ao problema bidimensional, para o PCTGR;
- Propor uma formulação matemática para o PCTGR de 3-estágios;
- Estender e propor modificações no algoritmo *bottom-up* de Wang (1983) para abordar o PCTGR e sua versão de 3-estágios.

6.1 Formulações matemáticas

A formulação baseada em grelha e as formulações baseadas no conceito *bottom-up* propostas nesta tese, e mesmo as formulações de Ben Messaoud, Chu e Espinouse (2008) e Furini, Malaguti e Thomopoulos (2016), são estendíveis ao caso tridimensional a fim de abordar o PCTGR. No entanto, os números grandes de variáveis e de restrições de algumas dessas formulações inviabilizariam as respectivas abordagens no contexto de *solvers* de propósito geral para resolver instâncias realistas do problema. Assim, optou-se por representar o PCTGR por meio da formulação compacta *bottom-up*, que é caracterizada por números menores de variáveis e de restrições, e que apresentou melhor desempenho nas instâncias de grande porte em comparação à formulação pseudopolinomial *bottom-up*. Ainda motivado por uma aplicação industrial, que corta blocos de até 40 toneladas (com alguns metros em cada dimensão) em poucos estágios guilhotinados, uma formulação de Programação Linear Inteira (PLI) foi proposta à versão do PCTGR que é limitada a 3-estágios. Note que um padrão de corte d -estágios busca tempos menores de produção em detrimento da utilização de material ao rotacionar a serra de corte em até d vezes. Como provar otimalidade no caso tridimensional tende a ser um desafio significativo, o algoritmo de combinação sucessiva de itens, originalmente proposto por Wang (1983), também é aqui estendido ao caso tridimensional no contexto de padrões não-estagiados e 3-estágios, como método de solução.

Até onde esse autor tem conhecimento, não existem modelos matemáticos ao PCTGR propostos na literatura. Destaca-se que o Problema de Carregamento de Contêineres é um outro PCE tridimensional, mas que geralmente não considera a restrição de guilhotina, que é considerada pelo o PCTGR. Mais uma vez, destaca-se que o desenvolvimento de modelos contribui para a caracterização dos problemas, e para a proposição de futuros métodos de solução baseados em decomposição desses modelos, que podem permitir a resolução de instâncias maiores do problema. Assim, na Seção 6.1.1, a formulação compacta do Capítulo 5 é estendida ao PCTGR. Na Seção 6.2 é proposto um método de solução ao PCTGR. Neste capítulo, o objeto a ser cortado é caracterizado por $\bar{L} \times \bar{W} \times \bar{H}$.

6.1.1 Formulação não-linear inteira para padrões não-estagiados

O PCTGR permite três tipos de construções na abordagem de empacotamento *bottom-up*, como ilustrado na Figura 6.1. Considere duas cópias de tipos de itens de tamanhos $l_1 \times w_1 \times h_1$ e $l_2 \times w_2 \times h_2$. A construção horizontal desses itens gera o bloco de tamanho $(l_1 + l_2) \times \max\{w_1; w_2\} \times \max\{h_1; h_2\}$; a construção em profundidade gera o bloco de tamanho $\max\{l_1; l_2\} \times (w_1 + w_2) \times \max\{h_1; h_2\}$; e, a construção vertical gera o bloco de tamanho $\max\{l_1; l_2\} \times \max\{w_1; w_2\} \times (h_1 + h_2)$. Uma construção na abordagem *bottom-up* envelopa dois blocos para gerar um bloco maior. Ela geralmente não distin-

gue a posição relativa dos dois blocos menores no bloco maior, porque essa abordagem se baseia apenas nos tamanhos dos blocos. Assim, o volume não ocupado por cópias de tipos de itens em uma construção é considerado como desperdício, porque esse volume não será ocupado por nenhuma outra cópia de tipo de item em construções futuras. Note que trata-se de corte guilhotinado.

Assim como no caso bidimensional, é importante primeiro conhecer quantos blocos (i.e., subpadrões) podem ser construídos para definir o modelo. Seja o parâmetro \hat{n} o número de cópias de tipos de itens em certa solução ótima de uma instâncias ordinária do PCTGR. Para essa solução, sem perda de otimalidade, pode-se considerar $\hat{n}-1$ subpadrões gerados por construções sucessivas horizontais, em profundidade e verticais de cópias de tipos de itens. Seja o parâmetro \bar{n} um limitante superior ao parâmetro \hat{n} como, por exemplo, $\hat{n} = \sum_{i \in I} u_i$, dado que o parâmetro \hat{n} é de difícil conhecimento antes da resolução da instância.

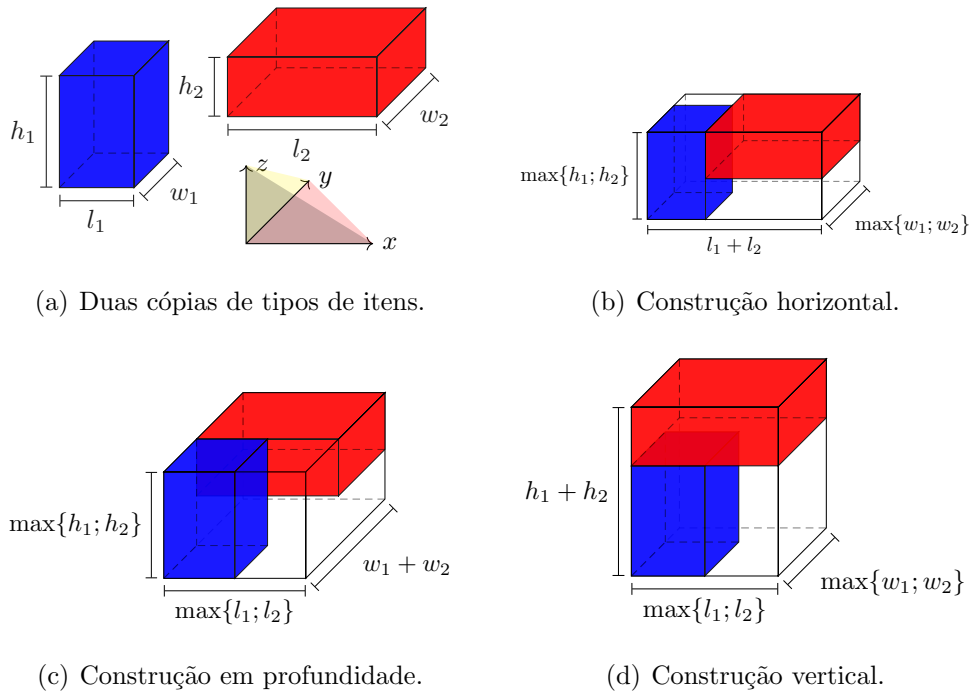


Figura 6.1: Exemplos de construções na abordagem *bottom-up*.

No modelo considera-se que cada bloco, caso exista, é sempre um subpadrão gerado ao combinar cópias de tipos de itens ou blocos menores. Seja $J = \{1, \dots, \bar{n} - 1\}$ o conjunto de todos os subpadrões (i.e., os blocos). Seja $O = \{h, d, v\}$ o conjunto de todas as orientações possíveis para construir um subpadrão, sendo que h , d e v são para construção horizontal, em profundidade e vertical, respectivamente. Seja $Q_i = \{1, \dots, \min\{2, u_i\}\}$ o conjunto de cópias possíveis de um tipo de item $i \in I$ a ser contido diretamente por qualquer subpadrão $j \in J$. Há seis conjuntos de variáveis nesse modelo, que são definidos em (6.1) até (6.6).

$$z_{jip} = \begin{cases} 1, & \text{se o bloco } j \text{ contém a } p\text{-ésima cópia do tipo de item } i, \\ 0, & \text{caso contrário,} \end{cases} \quad j \in J, i \in I, p \in Q_i. \quad (6.1)$$

$$x_{jo} = \begin{cases} 1, & \text{se o bloco } j \text{ é construído com orientação } o, \\ 0, & \text{caso contrário,} \end{cases} \quad j \in J, o \in O. \quad (6.2)$$

$$y_{jk} = \begin{cases} 1, & \text{se o bloco } j \text{ contém o bloco } k, \\ 0, & \text{caso contrário,} \end{cases} \quad j \in J, k \in J, j < k. \quad (6.3)$$

$$L_j: \text{ comprimento do bloco } j, \quad j \in J. \quad (6.4)$$

$$W_j: \text{ largura do bloco } j, \quad j \in J. \quad (6.5)$$

$$H_j: \text{ altura do bloco } j, \quad j \in J. \quad (6.6)$$

Tendo definidos todas as variáveis e parâmetros necessários, no Modelo (6.7) declara-se uma formulação compacta de PNLIM ao PCTGR. Note que esse modelo é estendível ao caso n -dimensional ao simplesmente adicionar uma nova orientação no conjunto O , e um novo bloco de restrições relacionado a essa nova dimensão do objeto, para cada dimensão adicional. De fato, o Modelo (6.7) difere do Modelo (5.12), porque ele tem uma terceira dimensão para as construções, assim como uma dimensão adicional para os subpadrões.

$$\mathbf{Max} \sum_{j \in J} \sum_{i \in I} \sum_{p \in Q_i} v_i z_{jip}, \quad (6.7a)$$

s.a

$$\sum_{j \in J} \sum_{p \in Q_i} z_{jip} \leq u_i, \quad i \in I, \quad (6.7b)$$

$$\sum_{o \in O} x_{jo} \leq 1, \quad j \in J, \quad (6.7c)$$

$$\sum_{i \in I} \sum_{p \in Q_i} z_{1ip} + \sum_{k \in K, 1 < k} y_{1k} \leq 2 \sum_{o \in O} x_{1o}, \quad (6.7d)$$

$$\sum_{i \in I} \sum_{p \in Q_i} z_{jip} + \sum_{k \in K, j < k} y_{jk} = 2 \sum_{o \in O} x_{jo}, \quad j \in J \setminus \{1\}, \quad (6.7e)$$

$$\sum_{j \in J, j < k} y_{jk} = \sum_{o \in O} x_{ko}, \quad k \in J \setminus \{1\} \quad (6.7f)$$

$$L_j = \left(\sum_{i \in I} \sum_{p \in Q_i} l_i z_{jip} + \sum_{k \in J, j < k} L_k y_{jk} \right) * x_{jh} + \max\{(i, p) \in (I \times Q_i) : l_i z_{jip}; k \in J, j < k : L_k y_{jk}\} * (x_{jd} + x_{jv}), \quad j \in J, \quad (6.7g)$$

$$W_j = \left(\sum_{i \in I} \sum_{p \in Q_i} w_i z_{jip} + \sum_{k \in J, j < k} W_k y_{jk} \right) * x_{jd} + \max\{(i, p) \in (I \times Q_i) : w_i z_{jip}; k \in J, j < k : W_k y_{jk}\} * (x_{jh} + x_{jv}), \quad j \in J, \quad (6.7h)$$

$$H_j = \left(\sum_{i \in I} \sum_{p \in Q_i} h_i z_{jip} + \sum_{k \in J, j < k} H_k y_{jk} \right) * x_{jv} + \max\{(i, p) \in (I \times Q_i) : h_i z_{jip}; k \in J, j < k : H_k y_{jk}\} * (x_{jh} + x_{jd}), \quad j \in J, \quad (6.7i)$$

$$z_{jip} \in \{0, 1\}, \quad j \in J, i \in I, p \in Q_i, \quad (6.7j)$$

$$y_{jk} \in \{0, 1\}, \quad j \in J, k \in J, j < k, \quad (6.7k)$$

$$x_{jo} \in \{0, 1\}, \quad j \in J, o \in O, \quad (6.7l)$$

$$0 \leq L_j \leq \bar{L}, \quad j \in J, \quad (6.7m)$$

$$0 \leq W_j \leq \bar{W}, \quad j \in J, \quad (6.7n)$$

$$0 \leq H_j \leq \bar{H}, \quad j \in J. \quad (6.7o)$$

A função objetivo (6.7a) consiste em maximizar a soma total do valor das cópias de itens selecionadas. As restrições (6.7b) garantem que cada tipo de item $i \in I$ é selecionado em até seu máximo de cópias a ser cortado. As restrições (6.7c) garantem que cada bloco, caso exista, é um subpadrão construído horizontalmente, em profundidade ou verticalmente.

As restrições (6.7d) e (6.7e) garantem que se um subpadrão $j \in J$ é construído (i.e., quando $\sum_{o \in O} x_{jo} = 1$), então ele deve conter exatamente ou duas cópias de tipos de itens, ou dois subpadrões, ou uma cópia de tipo de item e um subpadrão; note que o *right-hand-side* (RHS) da respectiva restrição se torna 2. Em contraste, se o bloco $j \in J$ não é um subpadrão (i.e., quando $\sum_{o \in O} x_{jo} = 0$), então ele não deve conter cópias de tipos de itens nem subpadrões. Particularmente, as restrições (6.7d), (6.7e) e (6.7f) também asseguram que o subpadrão $j = 1$ é o último a ser construído, i.e., ele é o padrão de corte final. A restrição (6.7d) é o caso especial das restrições (6.7e) para o subpadrão $j = 1$, dado que a solução pode ser formada de apenas uma única cópia de qualquer tipo de item.

As restrições (6.7f) garantem que o bloco $k \in J \setminus \{1\}$ é um subpadrão, se e somente se, ele esteja contido em um subpadrão maior $j \in J, j < k$. Note que essas restrições também limitam que um bloco $k \in J \setminus \{1\}$ está contido no máximo por um subpadrão $j \in J, j < k$, segundo às restrições (6.7c).

As restrições (6.7g) garantem que a variável L_j assume o tamanho do bloco que envelopa as cópias de tipos de itens e/ou subpadrões que o construíram, segundo a orientação $o \in O$. Por exemplo, se um subpadrão $j \in J$ é construído horizontalmente (i.e., quando $x_{jh} = 1$ e $x_{jd} + x_{jv} = 0$), então a variável L_j assume a soma dos comprimentos das cópias de tipo de itens e/ou subpadrões que construíram esse subpadrão j . Mas se o bloco $j \in J$ é construído em profundidade ou verticalmente (i.e., quando $x_{jh} = 0$ e $x_{jd} + x_{jv} = 1$), então a variável L_j assume o maior comprimento das cópias de tipos de itens e/ou subpadrões que construíram esse subpadrão j ao usar a função $\max\{\cdot\}$. Note que se um bloco $j \in J$ contém outro bloco $k \in J$ (i.e., quando $y_{jk} = 1$), o comprimento, a largura e a altura desse subpadrão k podem ser definidos como $L_k y_{jk}$, $W_k y_{jk}$ e $H_k y_{jk}$, respectivamente, o que é útil para descrever os tamanho do subpadrão j . No entanto, se o bloco $j \in J$ não contém o bloco $k \in J$ (i.e., quando $y_{jk} = 0$), esses produtos de variáveis $L_k y_{jk}$, $W_k y_{jk}$ e $H_k y_{jk}$ resultam em zero. As restrições (6.7h) e (6.7i) são similares às restrições (6.7g), mas relacionadas às variáveis W_j e H_j , respectivamente.

As restrições (6.7j) e (6.7o) definem o domínio das variáveis. Note que esse modelo tem a premissa que o subpadrão $j = 1$ é sempre construído e, assim $\sum_{o \in O} x_{1o} = 1$ pode ser fixado. O Modelo (6.7) é não-linear devido ao produto de variáveis e o uso da função $\max\{\cdot\}$ nas restrições (6.7g) até (6.7i). Além disso, ele não requer que os dados de entrada (i.e., \bar{L} , \bar{W} , \bar{H} , l_i , w_i e h_i , para cada $i \in I$) sejam inteiros.

Na Seção B.1 do Apêndice B, similarmente ao realizado no Capítulo 5, uma formulação de PLIM é obtida ao reformular o Modelo (6.7) por meio de estratégias de linearização, que permitem o uso de *solvers* de propósito geral. Na Seção B.2 do Apêndice B discute-se o Algoritmo B.2.1 que é um pseudo-código para gerar o padrão de corte a partir de uma solução do Modelo (6.7) ou de sua versão linear.

6.1.2 Formulação linear inteira para padrões 3-estágios

Alguns sistemas produtivos limitam suas operações de corte em até $d \in \mathbb{Z}_+$ estágios guilhotinados para buscar tempos menores de produção, em detrimento da utilização de material. Como o corte guilhotinado de blocos retangulares pode ser demorado (mais de uma hora por corte guilhotinado), é proposta nessa seção uma formulação de PLI que estritamente considera padrões 3-estágios, principalmente motivado pelos modelos de Lodi e Monaci (2003), Yanasse e Morabito (2006), Yanasse e Morabito (2008), Silva, Alvelos e Valério de Carvalho (2010) e Macedo, Alves e Valério de Carvalho (2010) para tipos de padrões 2-estágios para problemas bidimensionais de corte guilhotinado.

Primeiramente, é proposto um modelo que considera cortes verticais no primeiro estágio (i.e., paralelos ao eixo xy), cortes em profundidade no segundo estágio (i.e., paralelos ao eixo xz), e cortes horizontais no terceiro estágio (i.e., paralelos ao eixo yz). Em seguida, é discutido como adaptar o modelo para outras sequências de corte guilhotinado. A Figura 6.2 ilustra os cortes de sequência vertical-profundidade-horizontal;

os blocos coloridos na Figura 6.2(d) são cópias de tipos de itens, enquanto que o volume em branco é desperdício. Assim, a partir do objeto $\bar{L} \times \bar{W} \times \bar{H}$, os cortes de primeiro estágio geram os subobjetos de tamanho $\bar{L} \times \bar{W} \times \bar{h}_j$. Os cortes de segundo estágio geram as tiras de tamanho $\bar{L} \times \bar{w}_k \times \bar{h}_j$. No último estágio, as cópias de tipos de itens são cortadas dessas tiras. Seja $J = \{\bar{h}_1, \bar{h}_2, \dots, \bar{h}_{|J|}\}$ o conjunto de todas as diferentes alturas (e suas cópias) dos tipos de itens $i \in I$, e seja $K = \{\bar{w}_1, \bar{w}_2, \dots, \bar{w}_{|K|}\}$ o conjunto de todas as diferentes larguras (e suas cópias) dos tipos de itens $i \in I$. Seja $I(j, k) = \{i \in I \mid h_i \leq \bar{h}_j, w_i \leq \bar{w}_k\}$ o subconjunto de tipos de itens capaz de serem contidos na tira $\bar{L} \times \bar{w}_k \times \bar{h}_j$. A diferente altura \bar{h}_j é duplicada no conjunto J conforme expressado por $\min\left\{\sum_{i \in I \mid h_i = \bar{h}_j} u_i; \lfloor \bar{H} / \bar{h}_j \rfloor\right\}$, enquanto que a diferente largura \bar{w}_k é duplicada no conjunto K conforme expressado por $\min\left\{\sum_{i \in I \mid w_i = \bar{w}_k} u_i; \lfloor \bar{W} / \bar{w}_k \rfloor\right\}$.

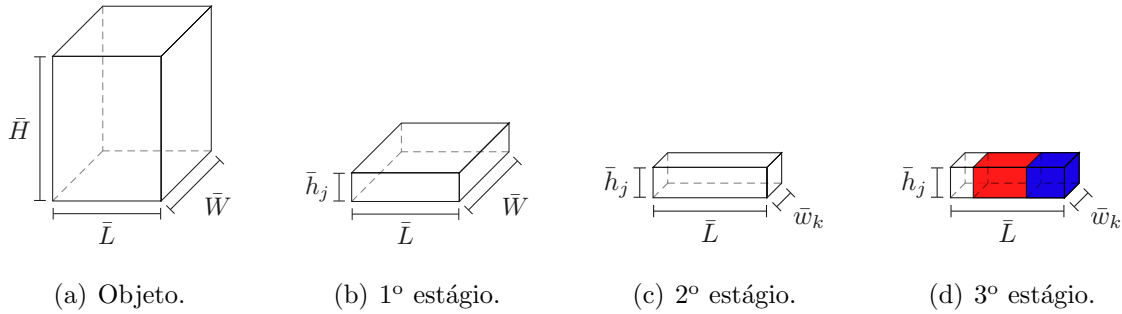


Figura 6.2: Padrão 3-estágios para o PCTGR com cortes na sequência vertical-profundidade-horizontal.

O modelo tem três conjuntos de variáveis de decisão, que são definidos em (6.8) até (6.10). As variáveis x_j são associadas aos subobjetos obtidos no primeiro estágio, e as variáveis y_{jk} às tiras obtidas no segundo estágio. As variáveis z_{jki} indicam o número de cópias do tipo de item $i \in I(i, j)$ cortadas da tira $\bar{L} \times \bar{w}_k \times \bar{h}_j$ no terceiro estágio.

$$x_j = \begin{cases} 1, & \text{se o subobjeto } \bar{L} \times \bar{W} \times \bar{h}_j \text{ é cortado,} \\ 0, & \text{caso contrário,} \end{cases} \quad j \in J. \quad (6.8)$$

$$y_{jk} = \begin{cases} 1, & \text{se a tira } \bar{L} \times \bar{w}_k \times \bar{h}_j \text{ é cortada,} \\ 0, & \text{caso contrário,} \end{cases} \quad j \in J, k \in K. \quad (6.9)$$

$$z_{jki} : \text{número de cópias do tipo de item } i \text{ cortado da tira } \bar{L} \times \bar{w}_k \times \bar{h}_j, \quad j \in J, k \in K, i \in I(j, k). \quad (6.10)$$

Tendo definidos todas as variáveis e parâmetros, o Modelo (6.11) é uma formulação de PLI que estritamente considera padrões 3-estágios para o PCTGR.

$$\text{Max} \sum_{j \in J} \sum_{k \in K} \sum_{i \in I(j,k)} v_i z_{jki}, \quad (6.11a)$$

s.a

$$\sum_{j \in J} \bar{h}_j x_j \leq \bar{H}, \quad (6.11b)$$

$$\sum_{k \in K} \bar{w}_k y_{jk} \leq \bar{W} x_j, \quad j \in J, \quad (6.11c)$$

$$\sum_{i \in I(j,k)} l_i z_{jki} \leq \bar{L} y_{jk}, \quad j \in J, k \in K, \quad (6.11d)$$

$$x_j \in \{0, 1\}, \quad j \in J, \quad (6.11e)$$

$$y_{jk} \in \{0, 1\}, \quad j \in J, k \in K, \quad (6.11f)$$

$$z_{jki} \in \mathbb{Z}_+, \quad j \in J, k \in K, i \in I(j, k). \quad (6.11g)$$

A função objetivo (6.11a) consiste em maximizar o valor total dos itens cortados. A restrição (6.11b) garante que a soma das alturas \bar{h}_j dos subobjetos não ultrapasse a altura do objeto. As restrições (6.11c) garantem que a soma das largura \bar{w}_k das tiras não ultrapasse a largura do subobjeto $j \in J$. As restrições (6.11d) garantem que a soma dos comprimentos l_i das cópias dos tipos de itens $i \in I$ não ultrapasse o comprimento da tira. Esses três blocos de restrições são restrições de mochila. As restrições (6.11e) até (6.11g) definem o domínio das variáveis.

Observe que um padrão de corte 3-estágios para o PCTGR pode considerar cortes horizontais, em profundidade ou verticais no primeiro estágio. O tipo de corte no estágio seguinte (i.e., no segundo estágio) pode ser um dentre esses três tipos de corte, com exceção daquele realizado no estágio anterior (i.e., no primeiro estágio). Em outras palavras, eles podem ser: cortes em profundidade ou verticais, quando os cortes de primeiro estágio foram horizontais; horizontais ou verticais, quando os cortes de primeiro estágio foram em profundidade; ou mesmo, horizontais ou em profundidade, quando os cortes de primeiro estágio foram verticais. Novamente, o tipo de corte no estágio seguinte (i.e., no terceiro estágio) pode ser um dentre esses três tipos de corte, com exceção daquele realizado no estágio anterior (i.e., no segundo estágio). Dessa forma, há três tipos de cortes para o primeiro estágio, dois para o segundo estágio, e dois para o terceiro estágio, em um total de 12 ($=3 \times 2 \times 2$) permutações. Portanto, o Modelo (6.11) aliado a um *solver* de propósito geral deve ser executado 12 vezes com essas diferentes permutações para encontrar o melhor padrão de corte 3-estágios ao PCTGR. A Figura 6.3 ilustra como um objeto pode ser cortado segundo essas 12 seqüências guilhotinadas.

Para adaptar o Modelo (6.11) para as outras seqüências guilhotinadas da

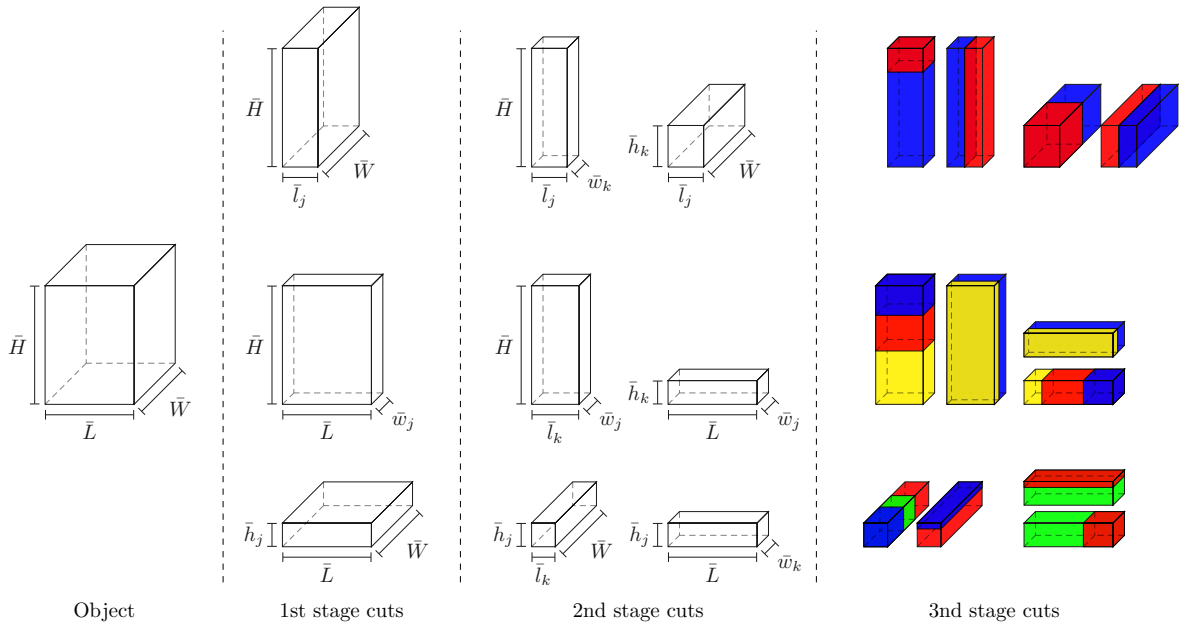


Figura 6.3: Sequências guilhotinadas para padrões 3-estágios do PCTGR.

Figura 6.3, pode-se entender o conjunto J como aquele de todas as dimensões diferentes (e suas cópias) referentes ao primeiro estágio, e o conjunto K como aquele das diferentes dimensões (e suas cópias) de segundo estágio. Por exemplo, a sequência horizontal-vertical-profundidade considera o conjunto J como aquele de todas as diferentes larguras (e suas cópias), e o conjunto K como aquele de todas as diferentes alturas (e suas cópias). Dessa forma, as constantes no RHS das restrições (6.11b) até (6.11d) devem ser \bar{L} , \bar{H} e \bar{W} , respectivamente, e w_i deve substituir l_i como coeficiente das variáveis z_{jki} nas restrições (6.11d). Adicionalmente, nota-se que essa proposta de 12 execuções do Modelo (6.11) não pode obter todos os tipos de padrões 3-estágios ao PCTGR. Particularmente, as seis sequências guilhotinadas com mesmo tipo de corte no primeiro e terceiro estágios podem requerer no conjunto J elementos que representem combinações lineares inteiras das respectivas dimensões sob análise, i.e., a discretização dos conjuntos normais (HERZ, 1972; CHRISTOFIDES; WHITLOCK, 1977). Um desses casos é ilustrado na Figura 6.4 em que um padrão de corte de sequência vertical-horizontal-vertical, cuja altura \bar{h}_j do subobjeto $\bar{L} \times \bar{W} \times \bar{h}_j$ é a soma da altura das duas cópias de tipos de itens, representados pelas caixas em vermelho e em azul. Esse tópico é mais discutido na Seção 6.3 dos experimentos computacionais.

6.2 Algoritmo *bottom-up*

Nesta seção, é proposto um método de solução para o PCTGR ao estender o algoritmo *bottom-up* de Wang (1983), que lidou originalmente com o caso bidimensional. Como dito anteriormente, esse algoritmo gera novas soluções parciais (i.e., subpadrões) ao sucessivamente combinar cópias de tipos de itens e/ou soluções parciais geradas em

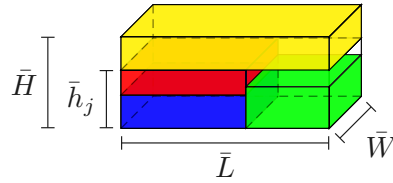


Figura 6.4: Padrão de corte de seqüência vertical-horizontal-vertical, cuja altura \bar{h}_j é soma das alturas dos tipos de itens representados em vermelho e em azul.

iterações anteriores. Para evitar o crescimento exponencial do número de soluções parciais, Wang propôs que o algoritmo rejeitasse soluções parciais com desperdício maior que certo limiar, i.e., um critério de aspiração para aceitação de novas soluções parciais. Wang propôs dois algoritmos que diferem quanto a esses critérios de aspiração. O Algoritmo 1 de Wang rejeita soluções parciais com desperdício interno maior que certa porcentagem da área do objeto, enquanto o Algoritmo 2 de Wang rejeita soluções parciais com desperdício maior que uma porcentagem da área da própria solução parcial. A autora também propôs condições de otimalidade para ambos os algoritmos a fim de declarar certificados de otimalidade, ou estimativas em relação à solução ótima. Note que esses algoritmos são projetados para instâncias não-ponderadas, i.e., instâncias cujo valor v_i de cada tipo de item $i \in I$ é sua área $l_i w_i$, no artigo original, ou seu volume $l_i w_i h_i$, neste capítulo.

Considere as seguintes definições:

- (i) O parâmetro β , $0 \leq \beta \leq 1$, é usado para definir o desperdício máximo aceitável de uma solução parcial;
- (ii) O conjunto $F^{\{k\}}$ contém todas as soluções parciais geradas na iteração k ;
- (iii) O conjunto $L^{\{k\}}$ contém todas as soluções parciais geradas até a iteração k ;
- (iv) A solução parcial T é caracterizada pelo seu tamanho $L_T \times W_T \times H_T$, desperdício T_t e número de cópias de cada tipo de item $i \in I$;
- (v) O desperdício total de uma solução parcial T é $T_t = T_i + T_o$, sendo que seu desperdício interno é $T_i = L_T W_T H_T - V$ (V é o volume de todas suas cópias de tipos de itens), e seu desperdício externo é $T_o = \bar{L} \bar{W} \bar{H} - L_T W_T H_T$, i.e., quando se aloca essa solução parcial no objeto, como ilustrado pelos volumes em branco da Figura 6.5.

O Algoritmo 6.2.1, a seguir, é o método de solução proposto ao PCTGR. Esse algoritmo é uma adaptação direta do Algoritmo 1 de Wang ao caso tridimensional, visto que o critério de aceitação de soluções parciais é baseado em uma porcentagem do volume do objeto.

O valor do parâmetro β é usado como critério para o *trade-off* entre soluções de qualidade e tempos de processamento razoáveis. Quando seu valor cresce, o espaço

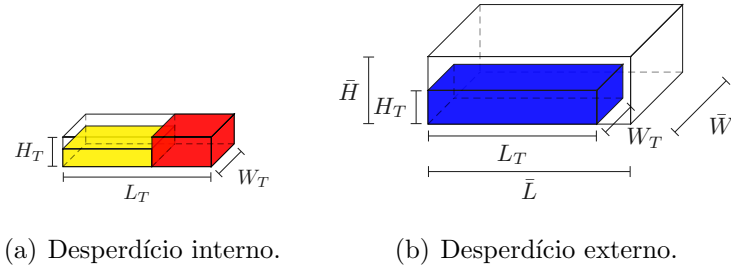


Figura 6.5: Desperdícios interno e externo de uma solução parcial T .

Algoritmo 6.2.1: Algoritmo *bottom-up* para o PCTGR.

Entrada: Instância.

- 1 Escolher um valor ao parâmetro β , $0 \leq \beta \leq 1$;
- 2 Definir o conjunto $F^{(0)} \leftarrow \{i_1, i_2, \dots, i_m\}$, o conjunto $L^{(0)} \leftarrow F^{(0)}$, e $k \leftarrow 0$;
- 3 **repita**
- 4 $k \leftarrow k + 1$;
- 5 Gerar o conjunto $F^{(k)}$ de todas as soluções parciais T ao combinar todos os elementos de $F^{(k-1)}$ com todos os elementos de $L^{(k-1)}$ satisfazendo:
 - (i) T é formado por construções horizontais, em profundidade ou verticais desses elementos;
 - (ii) o desperdício T_t não excede $\beta \bar{L} \bar{W} \bar{H}$;
 - (iii) o número de cópias de cada tipo de item $i \in I$ não excede u_i ;
- $L^{(k)} \leftarrow L^{(k-1)} \cup F^{(k)}$;
- 6 **até** $F^{(k)}$ esteja vazio
- 7 $M \leftarrow k - 1$;
- 8 Escolher a solução parcial de $L^{(M)}$ de menor desperdício T_t ;

Saída: Uma solução para o PCTGR.

de soluções também cresce, o que contribui para obter soluções com maiores taxas de utilização, em detrimento dos tempos de processamento. Em contraste, quando seu valor diminui, as taxas de utilização e tempos de processamento também tendem a diminuir. O Teorema 1 adapta ao PCTGR aquele originalmente proposto por Wang ao caso bi-dimensional. Ele fornece uma condição suficiente para a otimalidade da melhor solução T obtida pelo algoritmo, e associa o valor do parâmetro β usado para gerar as soluções parciais e seu desperdício total T_t .

Teorema 1 *Se o desperdício T_t da melhor solução T obtida ao fim do algoritmo, gerada com valor fixo β , satisfaz $T_t \leq \beta \bar{L} \bar{W} \bar{H}$, então T é uma solução ótima.*

Prova: Seja \mathcal{O} o conjunto de soluções ótimas, i.e., soluções de desperdício total mínimo T_t^* que satisfaçam as restrições do PCTGR. Seja β^* o menor valor de β que permita a criação de ao menos uma solução em \mathcal{O} . Assume-se que $T_t \leq \beta \bar{L} \bar{W} \bar{H}$. Se $T \notin \mathcal{O}$, então $\beta < \beta^*$, e assim $T_t^* \geq \beta^* \bar{L} \bar{W} \bar{H} > \beta \bar{L} \bar{W} \bar{H} \geq T_t$, o que contradiz a otimalidade de T_t^* . Portanto $T \in \mathcal{O}$. \square

Um tempo de processamento limite também foi considerado como critério de parada para o Algoritmo 6.2.1 (laço das linhas 3-5) nos experimentos computacionais da Seção 6.3. Quando esse tempo limite é alcançado, o certificado de otimalidade do Teorema 1 não é válido. A seguir, três variantes/melhorias do Algoritmo 6.2.1 são desenvolvidas. Duas delas são baseadas nas abordagens de Vasko (1989) e Oliveira e Ferreira (1990), que buscaram acelerar a convergência desse tipo de algoritmo. Na Seção 6.2.3, mostra-se como é possível estritamente considerar padrões 3-estágios no Algoritmo 6.2.1.

6.2.1 Completude de soluções parciais

Uma solução parcial que não possa ser usada em uma construção horizontal – passo 5(i) do Algoritmo 6.2.1 –, porque ao combiná-la com qualquer outra solução parcial (por exemplo, uma cópia do tipo de item de menor comprimento) se excederia o comprimento do objeto, é dita ser “completa horizontalmente”. Similarmente, uma solução parcial que não possa ser usada em uma construção em profundidade/verticalmente, porque ao combiná-la com qualquer outra solução parcial se excederia a largura/altura do objeto, é dita ser “completa em profundidade”/“completa verticalmente”. As melhorias dessa variantes são sumarizadas a seguir:

- uma solução parcial completa horizontalmente, em profundidade ou verticalmente não é considerada para futuras construções horizontais, em profundidade ou verticais, respectivamente;
- após um solução parcial ser gerada, ela é verificada quanto a completude horizontal, em profundidade e/ou vertical. Quando uma solução parcial T é completa horizontalmente, seu comprimento L_T é definido como o comprimento do objeto \bar{L} , e seu

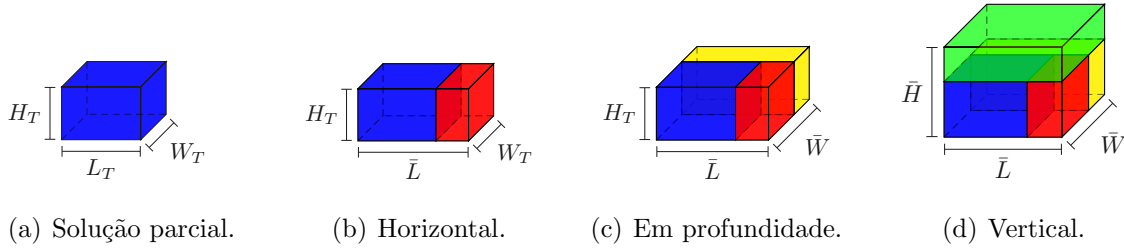


Figura 6.6: Estimativa de desperdício externo T_o na sequência H-D-V.

desperdício interno T_i é ajustado ao adicionar $(\bar{L} - L_T)W_T H_T$. Um comportamento similar é válido para soluções parciais completas em profundidade/verticalmente nas suas respectivas orientações;

- para uma solução parcial T completa horizontalmente, em profundidade e verticalmente, se $T_t < \beta \bar{L} \bar{W} \bar{H}$, então o valor de β é atualizado para $\beta = T_t / (\bar{L} \bar{W} \bar{H})$. Todas as soluções parciais em $F^{(k)}$ com desperdício maior que esse novo valor de β multiplicado pelo volume do objeto pode ser removidos (i.e., deletados), sem perda de otimalidade.

Cada tipo de item $i \in I$ é verificado quanto a completude horizontal, em profundidade e/ou vertical quando se define o conjunto $F^{(0)}$, com ajustes nos seus desperdícios internos, se necessário. Para um tipo de item completo horizontalmente, em profundidade e verticalmente, se $T_t > \beta \bar{L} \bar{W} \bar{H}$, ele não é incluído em $F^{(0)}$; caso contrário, o valor de β é atualizado. Note que se uma solução parcial T é dita completa horizontalmente, em profundidade e verticalmente, então seu desperdício externo T_o é zero.

6.2.2 Estimativa de desperdício externo T_o

O critério de aspiração proposto por Wang apenas considera o desperdício interno das soluções parciais. Oliveira e Ferreira (1990) propuseram adaptar o critério de aspiração do Algoritmo 1 de Wang para considerar também uma estimativa do desperdício externo T_o da solução parcial T , que seria rejeitada se $T_t = T_i + T_o > \beta \bar{L} \bar{W} \bar{H}$. Em outras palavras, esse outro critério de aspiração tenta identificar mais cedo quais as soluções parciais que não irão atender o critério de otimalidade. Assim, neste capítulo, T_o corresponde ao melhor (mínimo) desperdício externo de uma solução parcial maior que contenha a solução parcial T em seu canto inferior-esquerdo-frontal. Os autores estimaram T_o ao usar a formulação de PD de Gilmore e Gomory (1965), que lida com a versão irrestrita do PCBGR. A seguir, propõe-se heurísticamente estimar T_o ao usar a formulação de PD de De Queiroz et al. (2012), que lida com a versão irrestrita do PCTGR.

Considere uma solução parcial de tamanho $L_T \times W_T \times H_T$, como ilustrado na Figura 6.6(a). Ela é considerada como subpadrão de uma solução parcial

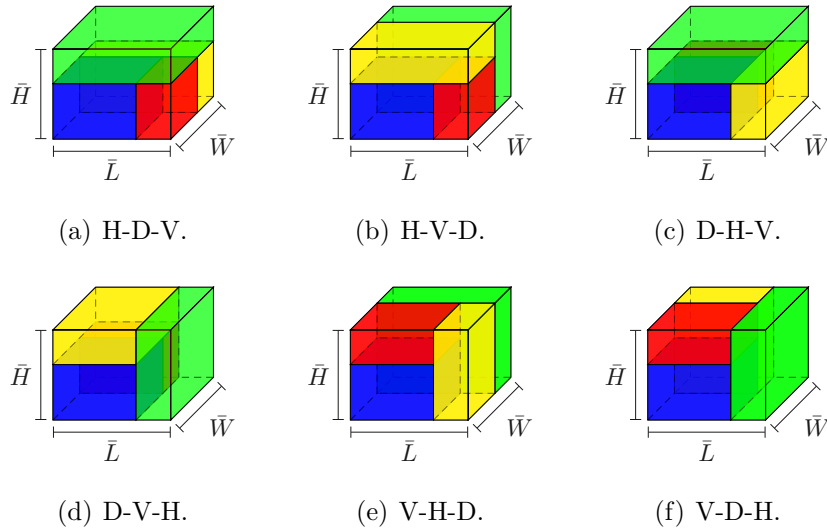


Figura 6.7: Seis permutações para estimar o desperdício externo T_o .

formada com três construções adicionais, i.e., uma construção adicional para cada dimensão. Particularmente, a Figura 6.6(d) ilustra uma solução parcial na qual a solução parcial T é tomada em uma sequência horizontal-profundidade-vertical (H-D-V). As outras cinco possíveis sequências para preencher o volume do objeto são: horizontal-vertical-profundidade, profundidade-horizontal-vertical, profundidade-vertical-horizontal, vertical-horizontal-profundidade e vertical-profundidade-horizontal. O valor do desperdício externo T_o pode ser estimado com respeito à sequência H-D-V como a soma dos volumes das soluções parciais em vermelho, amarelo e verde da Figura 6.6(d) menos seus respectivos valores ótimos na formulação de PD de De Queiroz et al. (2012), que lida com a versão irrestrita de PCTGR. Foi proposto considerar o valor mínimo entre essas seis sequências de permutações, como ilustrado na Figura 6.7, como a estimativa de desperdício externo T_o para uma solução parcial T . As soluções parciais em vermelho, amarelo e verde preenchem a primeira, segunda e terceira dimensões, respectivamente. Note que essa variante é com perda de otimalidade, porque a solução parcial T poderia ser considerada em padrões de corte com mais do que três construções adicionais, o que poderia reduzir ainda mais o valor de T_o . No entanto, a variante proposta drasticamente reduz o espaço de soluções para alguns cenários, o que contribui à convergência do algoritmo, como mostrado nos experimentos computacionais da Seção 6.3.

6.2.3 Algoritmo *bottom-up* para padrões 3-estágios

A literatura relacionada a padrões de corte d -estágios usualmente se baseia na abordagem *top-down*, no qual o objeto é cortado até os itens, após formação de pilhas ou tiras. A Figura 6.3 destaca essa abordagem no Modelo (6.11). Até onde esse autor tem conhecimento, padrões d -estágios não são abordados por abordagens *bottom-up*. No capítulo anterior, foi proposto um conjunto de restrições adicionais para cada formulação

bottom-up. Aquelas restrições são baseadas na contagem do número mínimo de estágios guilhotinados para cada subpadrão. Essa estratégia parece não ser diretamente aplicável ao PCTGR, porque ele tem três orientações possíveis (i.e., h , d e v) que gera diversos tipos de padrões de corte e dificulta a contagem, diante de degenerescência (i.e., vários tipos de padrões 2-estágios e 3-estágios).

Assim, optou-se por adaptar o Algoritmo 6.2.1 para estritamente considerar padrões 3-estágios ao enumerar de antemão por inspeção todos os possíveis tipos de padrões d -estágios, com $d \leq 3$, gerados por construções horizontais, em profundidade e verticais. Existem: (i) um tipo de padrão 0-estágio formado de única cópia de tipo de item (i.e., O); (ii) três tipos de padrões 1-estágio formados após uma construção de duas cópias de tipos de itens (i.e., H, D e V); (iii) seis tipos de padrões 2-estágios (i.e., H-D, H-V, D-H, D-V, V-D e V-H); e, (iv) doze tipos de padrões 3-estágios (i.e., H-D-H, H-D-V, H-V-D, H-V-H, D-H-D, D-H-V, D-V-D, D-V-H, V-D-H, V-D-V, V-H-D e V-H-V). Esses 22 ($=1+3+6+12$) tipos de subpadrões foram analisados entre si para cada possível orientação de construção (i.e., h , d e v). Por exemplo, a construção de orientação h dos tipos de subpadrões H-D e H-D-H gera um tipo de subpadrão H-D-H; no entanto, as construções de orientação d e v desses dois tipos de subpadrões são proibidas, porque cada um dos tipos de subpadrões resultantes iria requerer mais do que três estágios guilhotinados. O processo de inspeção gerou tabelas de acesso (do inglês, *look-up tables*) que são acessadas para verificar se uma construção é permitida ou não, como um critério adicional na linha 5 do Algoritmo 6.2.1, e informar qual é o tipo de subpadrão gerado, quando a construção é permitida. Note que cada solução parcial é agora adicionalmente caracterizada pela informação sobre o tipo de subpadrão. Essas tabelas de acesso são descritas na Seção B.3 do Apêndice B.

6.3 Experimentos computacionais

Os experimentos computacionais dessa seção foram realizados para avaliar a qualidade de soluções e tempo de processamento das abordagens propostas. A versão linear do Modelo (6.7) da Seção 6.1.1 para padrões de corte não-estagiados, reformulado como um programa linear no Apêndice B.1, com as respectivas desigualdades válidas, é chamado daqui em diante de Modelo Não-estagiado. O Modelo (6.11) da Seção 6.1.2 para padrões de corte 3-estágios é chamado daqui em diante de Modelo 3-estágios, enquanto que o algoritmo *bottom-up* com a variante/melhoria de completude de soluções parciais da Seção 6.2.1 é chamada daqui em diante de Alg-Bot. Note que o Modelo 3-estágios busca propor a melhor solução dentre 12 sequências guilhotinadas, como discutido anteriormente. É denotado por Alg-Bot+ED quando a estimativa de desperdício do volume externo é adicionalmente considerado. Quando esses algoritmos estritamente consideram padrões 3-estágios, como discutido na Seção 6.2.3, eles são denotados por Alg-Bot+3est e Alg-Bot+3est+ED, respectivamente.

Esta seção está dividida em duas partes. Cada uma delas considera um conjunto de instâncias de *benchmark* da literatura. O conjunto A considera as 12 instâncias de De Queiroz et al. (2012), enquanto o conjunto B considera as 60 instâncias de Egeblad e Pisinger (2009). Essas instâncias estão disponíveis em <http://www.loco.ic.unicamp.br/instances/3duk.html> e <http://hjemmesider.diku.dk/~pisinger/codes.html>. Elas são mais detalhadas no início de cada seção.

Para propósito de *benchmark*, as abordagens propostas foram comparadas às formulações de PD de De Queiroz et al. (2012) para padrões não-estagiados e 3-estágios na versão irrestrita do PCTGR (i.e., quando $u_i \geq \lfloor \bar{L}/l_i \rfloor \lfloor \bar{W}/w_i \rfloor \lfloor \bar{H}/h_i \rfloor$ para cada $i \in I$) na Seção 6.3.1. Até onde esse autor tem conhecimento, Morabito e Arenales (1994) é a única abordagem da literatura que lida com o PCTGR no caso restrito. Eles consideraram a restrição de guilhotina ao propor um método de solução para o Problema de Carregamento de Contêineres (PCC). No entanto, eles reportaram resultados computacionais para apenas uma instância no caso restrito. Assim, as abordagens propostas são comparadas entre si no caso restrito (i.e., quando $u_i \leq \lfloor \bar{L}/l_i \rfloor \lfloor \bar{W}/w_i \rfloor \lfloor \bar{H}/h_i \rfloor$ para cada $i \in I$, com desigualdade estrita para pelo menos um tipo de item) na Seção 6.3.2. Todas as abordagens, nomeadamente, Modelo Não-estagiado, Modelo 3-estágios, Alg-Bot, Alg-Bot+ED, Alg-Bot+3est, Alg-Bot+3est+ED e as duas formulações de PD de De Queiroz et al. (2012), foram implementadas em C++, usando a biblioteca do *Concert Studio* com o *IBM CPLEX Optimization Studio v12.8*, quando necessário. Os experimentos foram executados em ambiente computacional de Intel Xeon E5-2680v2 (2.8 GHz), limitado a 20 *threads*, com 32 GB de RAM, sob sistema operacional CentOS Linux 7.2.1511. Um tempo limite de 3600 segundos foi considerado ao *solver* para Modelo Não-estagiado e Modelo 3-estágios. O tempo limite das quatro versões do algoritmo *bottom-up* (i.e., Alg-Bot, Alg-Bot+ED, Alg-Bot+3est e Alg-Bot+3est+ED) foi de 600 segundos para evitar estouro de memória, quando a otimalidade não foi rapidamente provada.

Note que o Modelo Não-estagiado requer a definição dos limitantes superiores \bar{n} , UB_1 e UB_2 . Cada um desses parâmetros foi obtido ao resolver problemas de mochila unidimensionais. Similarmente ao realizado no capítulo anterior, o Modelo (6.12) foi resolvido pelo CPLEX para determinar o parâmetro $\bar{n} = \sum_{i \in I} q_i^*$, sendo que q_i^* é uma solução ótima desse modelo. Para os parâmetros UB_1 e UB_2 , modelos similares foram resolvidos, que consideram os volumes ou valores dos tipos de itens como coeficientes da função objetivo, respectivamente (i.e., $UB_1 = \sum_{i \in I} l_i w_i h_i q_i^*$ e $UB_2 = \sum_{i \in I} v_i q_i^*$). De fato, para cada um desses três parâmetros, diversas mochilas unidimensionais foram resolvidas no contexto de uma função de conservação de escala, como considerado por Egeblad e Pisinger (2009), que busca limitantes superiores mais apertados. Esses problemas de mochila unidimensionais diferem com respeito às dimensões dos tipos de itens, que são definidos de acordo com uma função pré-definida de conservação de escala. A premissa é que o valor ótimo mínimo entre esses problemas de mochila unidimensionais é um limitante superior válido para a instância original Fekete e Schepers (2004).

$$\bar{n} = \mathbf{Max} \sum_{i \in I} q_i, \mathbf{s.a} \sum_{i \in I} (l_i w_i h_i) q_i \leq \bar{L} \bar{W} \bar{H}, q_i \leq u_i, \quad q_i \in \mathbb{Z}_+, i \in I. \quad (6.12)$$

O valor inicial do parâmetro β fornecido às quatro versões do algoritmo *bottom-up* (linha 1 do Algoritmo 6.2.1) foi gerado ao resolver o Modelo 3-estágios com CPLEX. Foram consideradas duas permutações (dentre as 12 possíveis) com cortes verticais no primeiro estágio, como ilustrado na Figura 6.2(b). Para cada execução do *solver*, o tempo limite foi de 60 segundos.

6.3.1 Resultados para as instâncias *gcut_3d*

De Queiroz et al. (2012) propuseram as instâncias *gcut_3d* para versão ir-restrita do PCTGR ao adaptar as 12 instâncias *gcut* de Beasley (1985a), que foram inicialmente propostas para o caso bidimensional. Primeiramente, Beasley amostrou uniformemente o comprimento e a largura dessas instâncias nos intervalos $[0.25\bar{L}, 0.75\bar{L}]$ e $[0.25\bar{W}, 0.75\bar{W}]$, respectivamente. Então esses autores adicionaram a terceira dimensão para cada tipo de item ao aleatoriamente selecionar das dimensões já usadas nos outros tipos de itens. Essas instâncias são não-ponderadas, i.e., o valor de cada tipo de item é seu volume ($v_i = l_i w_i h_i$ para $i \in I$). Para cada uma delas, as dimensões do objeto ($\bar{L} \times \bar{W} \times \bar{H}$), o número de tipos de itens (m) e o número total de itens ($n = \sum_{i \in I} u_i$) são reportados na Tabela 6.1. As instâncias *gcut_3d_cons* foram geradas ao adaptar as instâncias *gcut_3d* para o caso mais restrito possível (i.e., quando $u_i = 1$ para cada $i \in I$); essas instâncias também são consideradas a seguir. É reportado a razão (em porcentagem) entre a soma do volume de todas as cópias de tipos de itens e o volume do objeto, i.e., $\sum_{i \in I} (l_i w_i h_i u_i) / (\bar{L} \times \bar{W} \times \bar{H}) * 100$ na coluna V_{cons} para o caso restrito. Essa razão indica que a decisão associada a seleção do PCTGR permanece mesmo diante do caso mais restrito possível, para essas instâncias.

A Tabela 6.2 mostra os resultados das abordagens propostas e da abordagem de *benchmark* para padrões não-estagiados, no caso irrestrito. Para cada abordagem e instância, são reportados a utilização de material como uma porcentagem (coluna $sol[\%]$), e o tempo de processamento em segundos (coluna $tempo[s]$). Para o Modelo Não-estagiado, também são reportados o número de variáveis (coluna var), o número de restrições (coluna $rest$) e o *gap* como uma porcentagem (coluna $gap[\%]$). O *gap* é calculado como $(UB - sol) / (UB + 10^{-10}) * 100$, sendo que UB é o melhor valor de limitante superior e sol é a melhor solução inteira encontrados pelo *solver* usando a formulação. O limitante superior \bar{n} também é reportado para o Modelo Não-estagiado. Para o Alg-Bot e o Alg-Bot+ED, também são reportados o número de soluções parciais geradas (coluna $nGer$), inseridas nos conjuntos $L^{\{k\}}$ (coluna $nIns$) e no conjunto $L^{\{M\}}$ (coluna $nEnd$). É denotado por “tl” quando o tempo limite foi atingido. Para cada abordagem, as soluções ótimas

Tabela 6.1: Características das instâncias *gcut_3d*.

Instâncias	\bar{L}	\bar{W}	\bar{H}	m	n	$V_{cons}[\%]$
gcut1_3d	250	250	250	10	57	117.64
gcut2_3d	250	250	250	20	113	198.66
gcut3_3d	250	250	250	30	207	254.21
gcut4_3d	250	250	250	50	274	572.18
gcut5_3d	500	500	500	10	60	104.04
gcut6_3d	500	500	500	20	82	270.97
gcut7_3d	500	500	500	30	132	400.62
gcut8_3d	500	500	500	50	270	524.15
gcut9_3d	1000	1000	1000	10	82	87.75
gcut10_3d	1000	1000	1000	20	75	307.60
gcut11_3d	1000	1000	1000	30	133	316.77
gcut12_3d	1000	1000	1000	50	251	587.08

encontradas são destacadas em negrito, mesmo quando a otimalidade não foi provada em tal abordagem.

Primeiramente, os resultados da formulação de Programação Dinâmica de De Queiroz et al. (2012) para padrões não-estagiados são analisados. A otimalidade foi provada pelo algoritmo para todas as instâncias do conjunto A em poucos décimos de segundo. Como proposto pelos autores, os conjuntos de discretização dos *reduced raster points* foram considerados, que segundo eles é sem perda de generalidade para o caso irrestrito. Essas instâncias são caracterizadas por tipos de itens com dimensões grandes em relação às dimensões do objeto, o que conduz a conjuntos de discretização com poucos elementos, e que por sua vez, à rápida convergência do algoritmo. Esses resultados indicam a efetividade das formulações de PD para problemas de corte guilhotinado irrestritos (CINTRA et al., 2008; DE QUEIROZ et al., 2012).

Os resultados do Modelo Não-estagiado na Tabela 6.2 mostram que os números de variáveis e de restrições são relativamente pequenos. No entanto, o tempo limite foi atingido para todas as instâncias com *gap* médio de 15.73%. Particularmente, a solução ótima da instância *gcut10_3d* foi encontrada, mas não provada dentro do tempo limite (*gap* de 14.84%). Esses *gaps* são relacionados principalmente à fraca relaxação linear do modelo, que é precisamente os valores fornecidos pelo parâmetro UB_1 . Por exemplo, se os valores das soluções ótimas obtidas pela formulação de PD forem considerados em vez da relaxação linear para calcular os *gaps*, o *gap* de otimalidade médio seria de 5.11%.

O Alg-Bot foi capaz de encontrar soluções e provar otimalidade em todas as instâncias do conjunto A com tempo de processamento médio de 86.05 segundos. O número de soluções parciais geradas (coluna nGer) parece crescer rapidamente, quando o número de tipo de itens também cresce. No entanto, o número de soluções parciais inseridas nos conjuntos $L^{\{k\}}$ (coluna nIns) é muito menor devido ao critério de aspiração, que rejeita a maioria delas. Particularmente, o número de soluções no conjunto $L^{\{M\}}$

(coluna nEnd) é ainda menor, porque nenhuma solução duplicada é permitida, e o valor do parâmetro β pode ser melhorado durante a busca, diante da completude de soluções parciais. A análise dos resultados da abordagem heurística Alg-Bot+ED mostra que essa versão do algoritmo foi capaz de encontrar soluções com *gap* de otimalidade médio de 0.23%, com melhoria de 67.20% sobre o tempo de Alg-Bot. Observe que soluções ótimas foram encontradas em 8 das 12 instâncias. Além disso, o número de soluções parciais manuseadas foi reduzido drasticamente em relação ao Alg-Bot, ao usar a estimativa de volume externo de soluções parciais. Para essas instâncias irrestritas, em média, o CPLEX com o Modelo 3-estágios forneceu ao Alg-Bot e ao Alg-Bot+ED um valor inicial ao parâmetro β de 0,1344 (i.e., uma taxa de utilização de material média de 86,56%).

A Tabela 6.3 mostra os resultados para padrões não-estagiados com as instâncias *gcut_3d* adaptadas ao caso mais restrito possível (i.e., quando $u_i = 1$ for $i \in I$). Para esses experimentos, resultados da abordagem de *benchmark* não são reportados, dado que ela apenas aborda o caso irrestrito. O Alg-Bot foi capaz de encontrar soluções e provar otimalidade em todas as instâncias adaptadas do conjunto A, com tempo de processamento médio de 69.07 segundos. As variações nos números de soluções parciais nas colunas nGer, nIns e nEnd com respeito àquelas da Tabela 6.2 são principalmente relacionados ao: (i) número de cópias de tipos de itens disponíveis (i.e., $n = \sum_{i \in I} u_i$); e, (ii) valor do parâmetro β , que é maior no caso restrito. Para essas instâncias restritas, em média, o CPLEX com o Modelo 3-estágios forneceu ao Alg-Bot e ao Alg-Bot+ED um valor inicial ao parâmetro β de 0,2656 (i.e., uma taxa de utilização de material média de 73,44%). A análise dos resultados da abordagem heurística Alg-Bot+ED mostra que essa versão do algoritmo foi capaz de encontrar soluções ótimas para todas as instâncias adaptadas, com melhoria de 84.72% em tempo sobre o Alg-Bot. Com relação ao modelo, CPLEX com o Modelo Não-estagiado foi capaz de provar otimalidade para as instâncias *cons_gcut1_3d* e *cons_gcut5_3d*. Para três instâncias adicionais (*cons_gcut6_3d*, *cons_gcut9_3d* e *cons_gcut10_3d*), soluções ótimas foram encontradas, mas não provadas dentro do intervalo de tempo. O *gap* de otimalidade médio foi de 3.18%, e o tempo de processamento médio foi de 3.085,49 segundos. Como dito antes, os *gaps* da Tabela 6.3 estão principalmente relacionados à fraca relaxação linear do modelo. Por exemplo, para as instâncias *cons_gcut6_3d* e *cons_gcut10_3d*, o valor das solução da instância é de fato ótimo, mas a estimativa de *gap* é superior a 20.00%.

As Tabelas 6.4 e 6.5 mostram os resultados das abordagens propostas e de *benchmark* para padrões 3-estágios nos casos irrestrito e restrito, respectivamente. Similarmente aos resultados da Tabela 6.2, a formulação de PD de De Queiroz et al. (2012) para padrões 3-estágios foi capaz de provar otimalidade em todas as instâncias do conjunto A no caso irrestrito, em poucos décimos de segundos. Para o Modelo 3-estágios, os valores nas colunas *var*, *rest*, *sol*[%] e *gap*[%] estão relacionados à melhor solução encontrada entre as 12 sequências guilhotinadas, enquanto que o respectivo tempo de processamento total é mostrado na coluna *tempo*[s]. O tempo limite total foi de 43200 (=12*3600) segundos para

o Modelo 3-estágios. Nas Tabelas 6.4 e 6.5, os resultados do Modelo 3-estágios mostram que os números de variáveis e de restrições são relativamente pequenos, mas aumentam rapidamente quando o número de tipos de itens também aumenta. Além disso, para cada instância, a otimalidade foi provada nas 12 sequências guilhotinadas, com tempo de processamento médio de 476.84 segundos na Tabela 6.4 e de 86.90 segundos na Tabela 6.5.

O Alg-Bot+3est foi capaz de encontrar soluções e provar otimalidade em 11 das 12 instâncias com tempo de processamento médio de 129.42 segundos e 169.64 segundos nas Tabelas 6.4 e 6.5, respectivamente. A otimalidade das instâncias gcut03_3d e cons_gcut04_3d não foi provada por essa versão do algoritmo; no entanto, diante de um tempo limite maior de 1800 segundos, essas soluções foram provadas serem ótimas. Como esperado, os números de soluções parciais geradas para padrões 3-estágios são menores do que aqueles obtidos nos respectivos casos para padrões não-estagiados. Em contraste, os tempos de processamento médios são ligeiramente superiores, porque os valores do parâmetro β são maiores, o que contribui para espaços de soluções maiores. Em contraste, os tempos de processamento médios são ligeiramente maiores, porque a taxa de utilização de material médio (i.e., sol[%]) dos padrões 3-estágios é menor do que a dos padrões não-estagiados, o que resulta em valores maiores do parâmetro β durante a busca, e por sua vez, a espaço de soluções maiores. Em relação ao Alg-Bot+3est+ED, essa versão do algoritmo *bottom-up* foi capaz de encontrar soluções ótimas em 11 das 12 instâncias no caso irrestrito, e em todas as instâncias no caso restrito, com melhoria sobre o tempo do Alg-Bot+3est de 78.41% e de 94.03%, respectivamente. Observe que o valor da solução ótima de padrões 3-estágios e não-estagiados é igual em três instâncias no caso irrestrito (gcut01_3d, gcut10_3d e gcut11_3d), e em quatro instâncias no caso restrito (cons_gcut01_3d, cons_gcut05_3d, cons_gcut10_3d e cons_gcut12_3d).

Em resumo, os principais resultados dessa seção são:

- (i) as formulações de Programação Dinâmica de De Queiroz et al. (2012), que são projetadas ao caso irrestrito, superam as abordagens propostas nesse caso;
- (ii) O Alg-Bot superou o Modelo Não-estagiado, tanto em qualidade de solução e em tempo de processamento nos casos irrestrito e restrito. Particularmente, a relaxação linear fraca do Modelo Não-estagiado prejudica a prova de otimalidade nessa abordagem;
- (iii) O Modelo 3-estágios teve melhor desempenho que o Alg-Bot+3est com respeito aos tempos de processamento no caso restrito, mas o contrário acontece no caso irrestrito;
- (iv) as estimativas de desperdício em Alg-Bot+ED e Alg-Bot+3est+ED significativamente aceleraram a convergência dos algoritmos.

Destaca-se a qualidade das soluções iniciais geradas pelo CPLEX com o Modelo 3-estágios para as quatro versões do algoritmo *bottom-up*. O *gap* de otimalidade médio dessas soluções iniciais foi menor que 3.50% para padrões não-estagiados, enquanto que de 1.75% para padrões 3-estágios. Para ambos casos, o tempo de processamento médio foi menor que 25 segundos. De forma geral, o Modelo Não-estagiado e as versões do Algoritmo *Bottom-up*, assim como o Modelo 3-estágios, também encontram soluções de boa qualidade rapidamente, e tendem a levar significativa parte de seus tempos de processamento para provar a otimalidade dessas soluções.

Tabela 6.2: Resultados das instâncias do conjunto A para padrões não-estagiados no caso irrestrito.

Instance	Modelo Não-estagiado						Alg-Bot					Alg-Bot+ED					De Queiroz et al. (2012)	
	\bar{n}	var	rest	sol[%]	gap[%]	tempo[s]	sol[%]	tempo[s]	nGer	nIns	nEnd	sol[%]	tempo[s]	nGer	nIns	nEnd	sol[%]	tempo[s]
gcut01_3d	18	952	2,664	77.57	22.43	tl	80.57	0.55	68,265	3,737	929	80.57	0.52	2,714	193	20	80.57	0.10
gcut02_3d	27	2,444	7,042	80.93	19.07	tl	84.88	113.29	26,303,103	127,381	36,213	84.88	9.80	32,146	561	35	84.88	0.10
gcut03_3d	27	3,016	8,768	87.49	12.51	tl	92.48	519.92	166,244,800	165,048	15,437	92.48	42.63	47,178	550	48	92.48	0.13
gcut04_3d	33	5,184	15,220	89.00	11.00	tl	95.43	98.87	22,760,214	25,908	7,670	95.04	69.58	5,231	99	62	95.43	0.37
gcut05_3d	21	1,280	3,612	81.77	18.23	tl	84.34	0.66	146,386	3,137	1,446	84.34	0.49	683	53	24	84.34	0.10
gcut06_3d	22	1,743	4,999	78.96	21.04	tl	84.84	4.05	282,406	9,543	2,284	84.84	1.92	9,994	269	27	84.84	0.10
gcut07_3d	22	2,100	6,080	79.98	20.02	tl	88.11	15.95	4,352,043	19,101	6,057	88.11	12.75	9,605	226	41	88.11	0.10
gcut08_3d	22	3,003	8,809	87.90	12.10	tl	93.20	112.96	26,912,145	38,971	15,691	92.89	69.67	5,974	112	69	93.20	0.31
gcut09_3d	18	969	2,715	82.45	17.55	tl	93.16	0.89	137,163	2,249	746	93.16	1.80	1,033	71	41	93.16	0.10
gcut10_3d	12	715	2,035	85.16	14.84	tl	85.16	1.17	22,615	1,196	579	85.16	3.80	439	34	30	85.16	0.10
gcut11_3d	18	1,598	4,622	90.78	9.22	tl	91.44	6.16	1,375,346	12,269	5,380	90.78	5.71	1,453	51	48	91.44	0.10
gcut12_3d	27	3,952	11,596	89.24	10.76	tl	92.65	158.08	24,553,935	33,038	12,017	91.50	120.01	4,007	99	90	92.65	0.28
Médias	22.25	2,246.33	6,513.50	84.27	15.73	tl	88.86	86.05	22,763,201.75	36,798.17	8,704.08	88.65	28.22	10,038.08	193.17	44.58	88.86	0.16

Tabela 6.3: Resultados das instâncias do conjunto A para padrões não-estagiados no caso restrito.

Instance	Modelo Não-estagiado						Alg-Bot					Alg-Bot+ED				
	\bar{n}	var	rest	sol[%]	gap[%]	tempo[s]	sol[%]	tempo[s]	nGer	nIns	nEnd	sol[%]	tempo[s]	nGer	nIns	nEnd
cons_gcut01_3d	8	196	516	62.73	0.00	545.57	62.73	0.33	2,006	625	328	62.73	0.61	1,001	205	139
cons_gcut02_3d	13	576	1,606	72.90	27.10	tl	73.83	12.74	523,935	36,729	16,197	73.83	17.24	117,126	3,999	2,226
cons_gcut03_3d	17	1,056	3,008	77.32	22.68	tl	87.09	212.29	26,488,139	117,588	24,315	87.09	21.64	572,098	3,168	318
cons_gcut04_3d	18	1,496	4,336	84.00	16.00	tl	90.86	263.36	65,271,546	90,607	18,641	90.86	13.44	1,113,256	3,523	293
cons_gcut05_3d	8	196	516	56.86	0.00	480.26	56.86	1.34	3,337	1,609	711	56.86	0.47	2,974	818	483
cons_gcut06_3d	10	378	1,048	76.28	23.72	tl	76.28	13.87	13,778	2,764	1,045	76.28	9.22	8,158	655	78
cons_gcut07_3d	13	696	1,976	80.59	19.41	tl	80.88	19.28	612,818	11,305	7,527	80.88	10.90	23,364	485	394
cons_gcut08_3d	17	1,376	3,988	80.28	19.72	tl	89.01	169.89	40,316,874	92,327	21,816	89.01	23.24	1,665,605	5,458	391
cons_gcut09_3d	9	240	636	64.14	8.72	tl	64.14	0.57	20,993	6,321	1,606	64.14	0.47	16,943	4,044	1,087
cons_gcut10_3d	9	320	886	71.63	28.37	tl	71.63	1.24	4,902	1,521	672	71.63	1.11	2,911	347	100
cons_gcut11_3d	14	780	2,216	74.90	25.10	tl	78.21	69.71	2,534,631	128,253	38,890	78.21	8.42	495,506	11,535	2,589
cons_gcut12_3d	16	1,260	3,652	84.78	15.22	tl	87.68	64.15	9,452,812	31,233	23,326	87.68	19.90	21,198	250	243
Médias	12.67	714.17	2,032.00	73.87	17.17	3,085.49	76.60	69.07	12,103,814.25	43,406.83	12,922.83	76.60	10.55	336,678.33	2,873.92	695.08

Tabela 6.4: Resultados das instâncias do conjunto A para padrões 3-estágios no caso irrestrito.

Instance	Modelo 3-estágios					Alg-Bot+3est					Alg-Bot+3est+ED					De Queiroz et al. (2012)	
	var	rest	sol[%]	gap[%]	tempo[s]	sol[%]	tempo[s]	nGer	nIns	nEnd	sol[%]	tempo[s]	nGer	nIns	nEnd	sol[%]	tempo[s]
gcut01_3d	436	145	80.57	0.00	129.89	80.57	2.04	66,751	4,755	1,611	80.57	0.47	3,374	223	23	80.57	0.10
gcut02_3d	3,661	902	84.35	0.00	330.88	84.35	270.37	11,645,372	123,257	53,369	84.35	10.96	34,078	727	160	84.35	0.11
gcut03_3d	12,102	2,103	89.11	0.00	613.41	89.11	tl	66,855,129	166,898	117,600	89.11	43.19	32,036	474	399	89.11	0.99
gcut04_3d	35,310	3,845	94.31	0.00	2,089.49	94.31	162.40	14,581,874	32,498	25,699	94.31	69.89	5,412	101	98	94.31	3.31
gcut05_3d	799	279	83.83	0.00	129.34	83.83	1.22	117,544	3,626	2,739	83.83	0.44	827	62	59	83.83	0.10
gcut06_3d	2,834	642	81.84	0.00	221.94	81.84	5.56	236,702	11,130	5,374	81.84	1.81	11,847	290	55	81.84	0.10
gcut07_3d	13,460	1,971	87.24	0.00	317.35	87.24	26.05	2,814,318	20,613	10,352	87.24	8.14	11,081	249	75	87.24	0.14
gcut08_3d	59,042	6,530	92.56	0.00	1,058.07	92.56	182.72	14,805,081	46,189	30,008	92.33	69.35	6,405	116	111	92.56	2.60
gcut09_3d	692	233	93.16	0.00	133.86	93.16	6.10	197,381	3,251	1,922	93.16	0.67	1,443	89	77	93.16	0.10
gcut10_3d	2,366	616	85.16	0.00	237.57	85.16	6.31	29,697	1,490	1,022	85.16	4.20	498	36	36	85.16	0.10
gcut11_3d	11,629	1,871	90.78	0.00	27.02	90.78	10.61	1,135,592	14,832	10,550	90.78	6.19	1,458	50	48	90.78	0.28
gcut12_3d	40,676	4,256	91.23	0.00	433.31	91.23	217.42	12,837,596	38,524	31,925	91.23	120.01	3,835	96	95	91.23	2.12
Médias	15,250.58	1,949.42	87.85	0.00	476.84	87.85	129.42	10,443,586.42	38,921.92	24,347.58	87.83	27.94	9,357.83	209.42	103.00	87.85	0.84

Tabela 6.5: Resultados das instâncias do conjunto A para padrões 3-estágios no caso restrito.

Instance	Modelo 3-estágios					Alg-Bot+3est					Alg-Bot+3est+ED				
	var	rest	sol[%]	gap[%]	tempo[s]	sol[%]	tempo[s]	nGer	nIns	nEnd	sol[%]	tempo[s]	nGer	nIns	nEnd
cons_gcut01_3d	292	94	62.73	0.00	37.68	62.73	0.40	2,303	694	536	62.73	0.30	1,137	209	174
cons_gcut02_3d	1,899	365	73.19	0.00	45.56	73.19	28.86	495,973	33,255	24,350	73.19	10.45	109,784	3,679	3,120
cons_gcut03_3d	5,619	765	83.58	0.00	76.09	83.58	589.61	17,063,870	121,783	95,390	83.58	15.59	492,418	3,313	2,742
cons_gcut04_3d	23,431	1,887	87.30	0.00	106.07	87.30	tl	41,231,549	125,989	98,647	87.30	15.54	1,043,946	4,568	3,448
cons_gcut05_3d	333	100	56.86	0.00	35.83	56.86	46.81	4,250	1,870	1,384	56.86	0.50	3,781	903	752
cons_gcut06_3d	2,472	442	71.25	0.00	82.38	71.25	11.32	17,579	3,435	2,390	71.25	1.09	10,212	721	325
cons_gcut07_3d	5,163	702	80.59	0.00	38.87	80.59	18.71	631,441	11,887	10,266	80.59	6.33	25,803	486	464
cons_gcut08_3d	32,447	2,455	86.78	0.00	164.88	86.78	387.90	32,472,683	95,619	57,999	86.78	26.38	1,689,756	5,556	1,795
cons_gcut09_3d	423	113	61.45	0.00	43.21	61.45	5.01	24,816	7,511	4,007	61.45	1.46	19,527	4,543	2,706
cons_gcut10_3d	1,770	361	71.63	0.00	32.33	71.63	5.69	5,450	1,623	877	71.63	8.43	3,117	348	100
cons_gcut11_3d	7,979	962	74.92	0.00	231.51	74.92	210.78	2,126,313	136,772	85,807	74.92	15.79	449,941	12,308	9,002
cons_gcut12_3d	27,089	2,079	87.68	0.00	148.34	87.68	88.44	6,590,510	31,101	27,025	87.68	18.79	22,116	250	246
Médias	9,076.42	860.42	74.83	0.00	86.90	74.83	169.64	8,388,894.75	47,628.25	34,056.50	74.83	10.05	322,628.17	3,073.67	2,072.83

6.3.2 Resultados para instâncias $ep3$

Egeblad e Pisinger (2009) geraram aleatoriamente as 60 instâncias $ep3$ para o PCC (chamado de Problema da Mochila Tridimensional Não-guilhotinada por esses autores). As dimensões dos tipos de itens foram uniformemente amostrados nos intervalos descritos na Tabela 6.6. Sobre o número máximo de cópias a ser cortado (i.e., u_i para $i \in I$), eles foram agrupados em cinco tipos de itens (i.e., o número de itens dividido por 5) ou em de forma aleatória (i.e., uma cópia por tipo de item). O volume do objeto é igual a 50% ou 90% do volume total de cópias de tipos de itens. A altura \bar{H} do objeto é o dobro do seu comprimento \bar{L} , que é igual a sua largura \bar{W} . Os autores propuseram a notação $ep3-n-c-t-p$, sendo que $n \in \{20, 40, 60\}$ é o número de itens, $c \in \{F, L, C, U, D\}$ descreve a classe, $t \in \{C, R\}$ descreve se ela é agrupada ou aleatória, e $p \in \{50, 90\}$ descreve a dimensão do objeto em porcentagem da soma total do volume dos itens. Essas instâncias são ponderadas, dado que o valor v_i de cada tipo de item $i \in I$ é seu volume multiplicado por um número aleatório do intervalo $\{1, 2, 3\}$.

Tabela 6.6: Características das instâncias $ep3$.

Classe	Descrição	Comprimento	Largura	Altura
F	itens são planos	[50,100]	[25,60]	[50,100]
L	itens são longos	$[1, \frac{2}{3} \cdot 100]$	[50,100]	$[1, \frac{2}{3} \cdot 100]$
C	itens são cubos	[1,100]	igual ao comprimento	igual ao comprimento
U	Maior dimensão não é maior que 200% da menor	[50,100]	[50,100]	[50,100]
D	Maior dimensão pode ser 50 vezes maior que a menor	[1,50]	[1,50]	[1,50]

A Tabela 6.7 mostra os resultados do Modelo Não-estagiado e do Modelo 3-estágios para as instâncias ponderadas do conjunto B. Lembre que o algoritmo *bottom-up* pode lidar apenas com instâncias ponderadas. Nessa tabela, são reportados os valores da função objetivo na coluna sol, em vez da utilização de material. A otimalidade foi provada pelo solver com o Modelo Não-estagiado em apenas 5 das 60 instâncias, enquanto que em 51 das 60 instâncias a otimalidade foi provada com o Modelo 3-estágios. Particularmente, as instâncias com otimalidade provada com o Modelo Não-estagiado são caracterizadas por ter o menor número de itens ($n = 20$) e o menor tipo de objeto ($p = 50$). Para ambos os modelos, os números de variáveis e de restrições aumentam quando o número de itens n também aumenta. Note que os tempos de processamento médios do Modelo 3-estágios aumentam rapidamente, quando o número de itens também aumenta. Embora o Modelo Não-estagiado lide com padrões mais gerais que o Modelo 3-estágios, o segundo superou o desempenho do primeiro em relação ao valor das soluções. O tempo de processamento médio do Modelo 3-estágios para as instâncias agrupadas foi de 62.48 segundos, e de 5728.95 segundos para as instâncias aleatórias. Destaca-se que o Modelo 3-estágios envolve 12 execuções do *solver*, mas soluções de boa qualidade são encontradas nos primeiros 30 segundos da busca, como discutido no fim da Seção 6.3.1. Além disso, o desvio médio das

Tabela 6.7: Resultados das instâncias ponderadas do conjunto B para padrões não-estagiados e 3-estágios.

Instância	\bar{L}	\bar{W}	\bar{H}	m	n	\bar{n}	Modelo Não-estagiado					Modelo 3-estágios				
							var	rest	sol	gap[%]	tempo[s]	var	rest	sol	gap[%]	tempo[s]
ep3-20-C-C-50	120	241	120	5	20	11	340	908	65,308	0.00	2.99	449	188	65,308	0.00	7.64
ep3-20-C-C-90	146	293	146	5	20	15	588	1,604	80,124	17.76	tl	617	253	80,124	0.00	4.80
ep3-20-C-R-50	116	232	116	20	20	12	506	1,409	62,364	0.00	5.87	2,856	399	62,364	0.00	20.54
ep3-20-C-R-90	141	283	141	20	20	15	728	2,039	66,844	17.43	tl	2,858	403	66,844	0.00	19.16
ep3-20-D-C-50	59	119	59	5	20	11	340	908	13,192	0.00	16.58	381	177	13,192	0.00	6.12
ep3-20-D-C-90	72	144	72	5	20	15	588	1,604	27,848	9.37	tl	485	225	27,848	0.00	15.98
ep3-20-D-R-50	52	104	52	20	20	15	728	2,039	14,619	18.08	tl	2,080	384	13,754	0.00	7.53
ep3-20-D-R-90	63	126	63	20	20	18	986	2,777	19,609	18.93	tl	2,868	443	19,265	0.00	15.69
ep3-20-F-C-50	106	213	106	5	20	10	288	764	71,816	0.00	1,749.21	309	153	71,816	0.00	45.42
ep3-20-F-C-90	129	259	129	5	20	16	660	1,808	161,011	10.57	tl	376	181	156,237	0.00	93.07
ep3-20-F-R-50	104	208	104	20	20	9	320	887	80,962	0.00	305.96	2,483	385	80,962	0.00	2.96
ep3-20-F-R-90	127	254	127	20	20	16	810	2,273	151,147	10.45	tl	2,239	405	148,204	0.00	34.88
ep3-20-L-C-50	80	160	80	5	20	12	396	1,064	20,148	26.84	tl	148	91	20,148	0.00	44.77
ep3-20-L-C-90	97	194	97	5	20	16	660	1,808	44,820	13.79	tl	265	155	44,426	0.00	45.43
ep3-20-L-R-50	77	154	77	20	20	15	728	2,039	24,729	15.92	tl	2,154	379	24,729	0.00	6.31
ep3-20-L-R-90	94	188	94	20	20	19	1,080	3,047	30,582	18.02	tl	2,306	401	30,096	0.00	25.44
ep3-20-U-C-50	125	250	125	5	20	10	288	764	91,036	36.47	tl	77	47	91,036	0.00	13.39
ep3-20-U-C-90	152	305	152	5	20	16	660	1,808	118,252	22.63	tl	440	177	132,291	0.00	2.62
ep3-20-U-R-50	129	258	129	20	20	11	440	1,223	97,358	21.28	tl	1,992	366	97,358	0.00	6.93
ep3-20-U-R-90	157	314	157	20	20	18	986	2,777	171,709	16.62	tl	2,220	385	171,709	0.00	18.41
ep3-40-C-C-50	151	303	151	5	40	27	1,716	4,844	141,418	16.97	tl	1,075	469	141,418	0.00	3.29
ep3-40-C-C-90	184	369	184	5	40	34	2,640	7,532	234,469	7.11	tl	897	400	243,447	0.00	149.61
ep3-40-C-R-50	141	282	141	40	40	28	2,646	7,657	126,069	13.10	tl	19,944	1,490	125,544	0.00	574.35
ep3-40-C-R-90	172	344	172	40	40	36	3,990	11,593	209,076	7.85	tl	21,565	1,570	213,452	0.00	3,584.24
ep3-40-D-C-50	75	150	75	5	40	27	1,716	4,844	20,464	35.86	tl	359	172	20,464	0.00	4.85
ep3-40-D-C-90	91	182	91	5	40	34	2,640	7,532	54,480	14.34	tl	1,174	529	55,092	0.00	3.65
ep3-40-D-R-50	60	121	60	40	40	31	3,120	9,043	23,860	31.83	tl	14,050	1,342	25,543	0.00	504.53
ep3-40-D-R-90	74	148	74	40	40	38	4,366	12,697	45,457	13.86	tl	17,887	1,532	46,189	0.00	1,129.75
ep3-40-F-C-50	134	268	134	5	40	24	1,380	3,872	151,848	12.45	tl	81	50	156,066	0.00	1.72
ep3-40-F-C-90	163	326	163	5	40	34	2,640	7,532	397,479	14.05	tl	531	265	405,370	0.00	12.09
ep3-40-F-R-50	131	263	131	40	40	25	2,208	6,379	215,992	20.40	tl	13,846	1,377	220,150	0.00	281.36
ep3-40-F-R-90	160	320	160	40	40	37	4,176	12,139	316,456	19.37	tl	16,709	1,417	347,868	0.00	3,956.37
ep3-40-L-C-50	100	201	100	5	40	24	1,380	3,872	54,054	11.71	tl	300	183	55,846	0.00	15.63
ep3-40-L-C-90	122	245	122	5	40	34	2,640	7,532	93,240	18.04	tl	445	223	94,884	0.00	2.60
ep3-40-L-R-50	94	188	94	40	40	29	2,800	8,107	44,004	28.19	tl	14,555	1,459	43,820	0.00	479.65
ep3-40-L-R-90	114	229	114	40	40	38	4,366	12,697	70,995	16.98	tl	19,535	1,654	70,560	0.00	808.02
ep3-40-U-C-50	158	316	158	5	40	19	900	2,492	150,434	16.74	tl	240	142	158,616	0.00	5.33
ep3-40-U-C-90	192	384	192	5	40	34	2,640	7,532	365,880	17.24	tl	219	119	388,456	0.00	27.85
ep3-40-U-R-50	164	329	164	40	40	24	2,070	5,977	216,407	19.33	tl	15,298	1,411	228,803	0.00	530.57
ep3-40-U-R-90	200	400	200	40	40	37	4,176	12,139	321,741	19.86	tl	17,424	1,531	347,272	1.10	15,711.01
ep3-60-C-C-50	173	347	173	5	60	42	3,936	11,324	219,871	38.90	tl	876	400	246,129	0.00	35.40
ep3-60-C-C-90	211	422	211	5	60	52	5,916	17,144	455,340	14.76	tl	2,440	1,027	499,572	0.00	99.50
ep3-60-C-R-50	161	322	161	60	60	46	6,930	20,313	197,795	13.84	tl	63,840	3,257	216,117	2.09	12,414.96
ep3-60-C-R-90	196	392	196	60	60	55	9,288	27,279	354,804	12.94	tl	73,137	3,551	362,563	0.00	16,030.72
ep3-60-D-C-50	85	171	85	5	60	40	3,588	10,304	65,592	11.78	tl	1,360	625	65,592	0.00	3.37
ep3-60-D-C-90	104	209	104	5	60	46	4,680	13,508	104,128	7.45	tl	1,765	801	105,600	0.00	142.75
ep3-60-D-R-50	67	135	67	60	60	50	7,938	23,289	38,738	31.10	tl	42,655	2,881	44,843	0.00	5,060.08
ep3-60-D-R-90	82	164	82	60	60	58	10,146	29,817	58,778	19.60	tl	52,116	3,273	64,682	1.01	10,632.90
ep3-60-F-C-50	153	307	153	5	60	39	3,420	9,812	443,790	12.71	tl	137	81	467,730	0.00	165.95
ep3-60-F-C-90	186	373	186	5	60	53	6,136	17,792	451,524	23.20	tl	839	362	506,304	0.00	227.00
ep3-60-F-R-50	150	301	150	60	60	34	4,290	12,537	313,020	27.80	tl	48,223	2,972	360,432	0.00	10,963.49
ep3-60-F-R-90	183	367	183	60	60	56	9,570	28,113	552,028	17.43	tl	44,723	3,064	609,403	0.50	23,926.52
ep3-60-L-C-50	115	230	115	5	60	36	2,940	8,408	110,502	12.36	tl	413	211	112,272	0.00	155.10
ep3-60-L-C-90	140	280	140	5	60	55	6,588	19,124	97,347	21.90	tl	633	362	110,796	0.00	165.43
ep3-60-L-R-50	104	209	104	60	60	45	6,688	19,599	52,586	29.10	tl	51,189	3,153	54,347	0.00	8,519.43
ep3-60-L-R-90	127	255	127	60	60	58	10,146	29,817	97,577	16.86	tl	57,316	3,442	102,089	0.98	20,539.40
ep3-60-U-C-50	180	361	180	5	60	33	2,496	7,112	330,610	22.77	tl	219	119	357,004	0.00	187.34
ep3-60-U-C-90	220	440	220	5	60	55	6,588	19,124	245,476	24.42	tl	588	310	272,664	0.00	186.81
ep3-60-U-R-50	184	369	184	60	60	37	4,896	14,319	368,386	18.61	tl	49,683	3,204	373,634	2.64	10,351.85
ep3-60-U-R-90	224	449	224	60	60	56	9,570	28,113	498,608	17.57	tl	57,202	3,504	526,282	6.36	25,711.46
Médias						30.67	3,201.33	9,273.50	162,063.35	17.08	3,334.68	12,551.52	992.18	171,077.10	0.24	2,895.72

Tabela 6.8: Resultados das instâncias adaptadas não-ponderadas do conjunto B para padrões não-estagiados e 3-estágios.

Instância	\bar{L}	\bar{W}	\bar{H}	m	n	Padrões não-estagiados					Padrões 3-estágios				
						Modelo Não-estagiado			Alg-Bot		Modelo 3-estágios			Alg-Bot+3est	
						sol[%]	gap[%]	tempo[s]	sol[%]	tempo[s]	sol[%]	gap[%]	tempo[s]	sol[%]	tempo[s]
nw-ep3-20-C-C-50	120	241	120	5	20	49.96	0.00	2.01	49.96	15.82	49.96	0.00	95.97	49.96	tl
nw-ep3-20-C-C-90	146	293	146	5	20	47.02	21.50	tl	47.02	221.84	47.02	0.00	76.26	47.02	tl
nw-ep3-20-C-R-50	116	232	116	20	20	54.38	5.20	tl	54.38	tl	54.38	0.00	220.36	54.38	tl
nw-ep3-20-C-R-90	141	283	141	20	20	44.93	23.47	tl	*	*	44.93	0.00	8.61	44.82	tl
nw-ep3-20-D-C-50	59	119	59	5	20	70.50	0.00	21.91	70.50	1.38	70.50	0.00	1.91	70.50	22.61
nw-ep3-20-D-C-90	72	144	72	5	20	56.78	12.23	tl	56.78	26.40	56.78	0.00	97.04	56.78	tl
nw-ep3-20-D-R-50	52	104	52	20	20	77.49	22.13	tl	79.94	tl	73.65	0.00	8.55	73.65	tl
nw-ep3-20-D-R-90	63	126	63	20	20	71.67	26.13	tl	67.62	tl	70.41	0.00	237.18	66.31	tl
nw-ep3-20-F-C-50	106	213	106	5	20	79.35	0.20	tl	79.35	5.17	79.35	0.00	33.44	79.35	1.34
nw-ep3-20-F-C-90	129	259	129	5	20	67.20	9.51	tl	69.29	1.31	65.76	0.00	43.13	65.76	6.59
nw-ep3-20-F-R-50	104	208	104	20	20	69.47	0.00	483.37	69.47	8.23	67.96	0.00	46.17	69.47	17.25
nw-ep3-20-F-R-90	127	254	127	20	20	68.88	17.94	tl	69.35	tl	68.17	0.00	6.28	68.17	tl
nw-ep3-20-L-C-50	80	160	80	5	20	83.01	16.99	tl	83.01	2.80	79.62	0.00	6.11	79.62	0.78
nw-ep3-20-L-C-90	97	194	97	5	20	66.53	20.49	tl	67.46	4.56	62.60	0.00	2.30	62.60	12.47
nw-ep3-20-L-R-50	77	154	77	20	20	80.77	19.23	tl	82.12	14.85	80.77	0.00	5.91	80.77	35.32
nw-ep3-20-L-R-90	94	188	94	20	20	77.59	22.00	tl	77.59	tl	75.22	0.00	10.63	68.75	tl
nw-ep3-20-U-C-50	125	250	125	5	20	61.99	36.79	tl	61.99	1.77	61.99	0.00	117.41	61.99	7.38
nw-ep3-20-U-C-90	152	305	152	5	20	73.88	17.70	tl	73.96	12.92	73.88	0.00	47.49	73.88	8.87
nw-ep3-20-U-R-50	129	258	129	20	20	71.78	26.87	tl	71.78	13.88	69.76	0.00	50.28	69.76	1.77
nw-ep3-20-U-R-90	157	314	157	20	20	71.50	25.62	tl	75.68	tl	75.56	0.00	14.55	70.78	tl
nw-ep3-40-C-C-50	151	303	151	5	40	50.09	31.74	tl	*	*	50.09	0.00	6.87	48.94	tl
nw-ep3-40-C-C-90	184	369	184	5	40	74.87	6.30	tl	*	*	75.09	0.00	4.48	75.09	tl
nw-ep3-40-C-R-50	141	282	141	40	40	53.28	29.54	tl	58.24	tl	61.61	0.00	610.24	58.24	tl
nw-ep3-40-C-R-90	172	344	172	40	40	77.46	8.08	tl	*	*	80.53	0.00	2,331.06	*	*
nw-ep3-40-D-C-50	75	150	75	5	40	72.32	24.91	tl	76.18	tl	76.04	0.00	173.61	75.97	tl
nw-ep3-40-D-C-90	91	182	91	5	40	72.01	14.47	tl	75.98	tl	74.96	0.00	14.65	73.84	tl
nw-ep3-40-D-R-50	60	121	60	40	40	81.33	18.67	tl	80.84	tl	83.95	0.00	228.90	80.84	tl
nw-ep3-40-D-R-90	74	148	74	40	40	71.55	28.45	tl	*	*	76.94	0.00	634.22	76.83	tl
nw-ep3-40-F-C-50	134	268	134	5	40	64.39	30.85	tl	73.49	2.07	73.49	0.00	48.71	73.49	9.02
nw-ep3-40-F-C-90	163	326	163	5	40	66.54	20.04	tl	74.23	tl	72.48	0.00	49.42	69.46	tl
nw-ep3-40-F-R-50	131	263	131	40	40	70.57	29.43	tl	*	*	74.95	0.00	72.73	*	*
nw-ep3-40-F-R-90	160	320	160	40	40	82.66	17.33	tl	*	*	83.61	0.00	3,592.23	82.55	tl
nw-ep3-40-L-C-50	100	201	100	5	40	83.29	16.70	tl	83.84	4.62	73.45	0.00	81.23	73.45	69.57
nw-ep3-40-L-C-90	122	245	122	5	40	74.03	18.63	tl	82.94	tl	74.83	0.00	82.13	72.27	tl
nw-ep3-40-L-R-50	94	188	94	40	40	78.58	21.42	tl	73.81	tl	79.84	0.00	346.64	73.81	tl
nw-ep3-40-L-R-90	114	229	114	40	40	73.79	26.21	tl	*	*	78.95	0.00	883.09	77.10	tl
nw-ep3-40-U-C-50	158	316	158	5	40	70.32	29.68	tl	73.44	6.75	72.38	0.00	76.49	72.38	6.10
nw-ep3-40-U-C-90	192	384	192	5	40	74.37	19.99	tl	*	*	75.33	0.00	58.73	75.33	tl
nw-ep3-40-U-R-50	164	329	164	40	40	83.75	16.25	tl	87.11	tl	88.44	0.00	876.39	87.11	tl
nw-ep3-40-U-R-90	200	400	200	40	40	72.06	27.94	tl	*	*	80.64	0.00	12,373.89	77.08	tl
nw-ep3-60-C-C-50	173	347	173	5	60	84.72	15.28	tl	*	*	86.23	0.00	76.86	86.23	tl
nw-ep3-60-C-C-90	211	422	211	5	60	66.45	22.43	tl	*	*	72.75	0.00	107.43	71.02	tl
nw-ep3-60-C-R-50	161	322	161	60	60	67.34	24.70	tl	70.23	tl	78.22	0.00	6,459.50	70.23	tl
nw-ep3-60-C-R-90	196	392	196	60	60	78.82	16.47	tl	*	*	80.16	0.00	10,401.62	*	*
nw-ep3-60-D-C-50	85	171	85	5	60	75.09	22.97	tl	80.48	tl	77.95	0.00	3.04	77.95	tl
nw-ep3-60-D-C-90	104	209	104	5	60	70.20	7.14	tl	*	*	70.25	0.00	6.14	70.25	tl
nw-ep3-60-D-R-50	67	135	67	60	60	83.50	16.50	tl	79.67	tl	83.71	0.00	578.19	79.67	tl
nw-ep3-60-D-R-90	82	164	82	60	60	75.13	24.87	tl	*	*	82.48	0.00	4,752.76	76.63	tl
nw-ep3-60-F-C-50	153	307	153	5	60	66.91	33.09	tl	89.17	194.32	88.47	0.00	4.30	86.15	tl
nw-ep3-60-F-C-90	186	373	186	5	60	68.92	23.51	tl	*	*	77.90	0.00	7.43	73.97	tl
nw-ep3-60-F-R-50	150	301	150	60	60	82.17	17.83	tl	*	*	87.57	0.90	8,416.04	*	*
nw-ep3-60-F-R-90	183	367	183	60	60	80.49	19.51	tl	*	*	84.99	1.20	14,503.98	*	*
nw-ep3-60-L-C-50	115	230	115	5	60	73.84	26.16	tl	90.82	203.61	85.27	0.00	134.55	85.27	tl
nw-ep3-60-L-C-90	140	280	140	5	60	80.07	19.93	tl	*	*	85.93	0.00	81.07	78.23	tl
nw-ep3-60-L-R-50	104	209	104	60	60	76.26	23.74	tl	81.30	tl	84.67	0.00	6,497.80	81.30	tl
nw-ep3-60-L-R-90	127	255	127	60	60	75.78	24.22	tl	*	*	82.92	0.00	18,256.03	79.28	tl
nw-ep3-60-U-C-50	180	361	180	5	60	75.46	24.54	tl	86.24	108.46	85.56	0.00	118.68	85.56	tl
nw-ep3-60-U-C-90	220	440	220	5	60	68.72	31.28	tl	*	*	83.02	0.00	94.20	83.02	tl
nw-ep3-60-U-R-50	184	369	184	60	60	85.90	14.10	tl	*	*	87.24	0.00	8,921.68	84.89	tl
nw-ep3-60-U-R-90	224	449	224	60	60	74.97	25.03	tl	*	*	85.01	3.85	23,110.23	72.49	tl
Médias						71,71	19,90	3,428,45	73,66**	330,42**	74,77	0,10	2,103,45	72,37**	487,77**

** : Esta média foi calculada com instâncias sem estouro de memória, e logo não pode ser diretamente comparada.

soluções do Modelo 3-estágios com respeito às melhores soluções obtidas por Egeblad e Pisinger (2009) (problema não-guilhotinado) foi de 3.82%. Isso indica que o Modelo 3-estágios pode ser considerado para obter boas soluções para o Problema de Carregamento de Contêineres (PCC).

Foram geradas as instâncias $nw-ep\beta$ ao adaptar as instâncias $ep\beta$ do conjunto B para o caso não-ponderado, i.e., ao considerar o valor de cada tipo de item como seu volume. A Tabela 6.8 mostra os resultados das abordagens propostas para essas instâncias adaptadas. A solução ótima da maioria dessas instâncias é perto de 100.0% de utilização de material, quando se supõe número ilimitado de cópias por tipo de item (i.e., no caso irrestrito). Por isso não são reportados resultados para o Alg-Bot+ED e o Alg-Bot+3est+ED, dado que o impacto da estimativa de desperdício do volume externo seria marginal nessas instâncias.

A análise dos resultados da Tabela 6.8 mostra que o desempenho do Modelo Não-estagiado e do Modelo 3-estágios são similares ao observado nas instâncias ponderadas da Tabela 6.7. O Alg-Bot teve estouro de memória em 23 das 60 instâncias, e foi capaz de encontrar soluções e provar otimalidade em 18 das 60 instâncias. O Alg-Bot+3est teve estouro de memória em 5 das 60 instâncias, e foi capaz de encontrar soluções e provar otimalidade em 13 das 60 instâncias. Note que as instâncias $ep\beta$ são caracterizadas por níveis moderados de utilização de material na solução ótima. Assim, o valor do parâmetro β é significativamente grande, e o critério de aspiração elimina poucas soluções parciais, logo a memória requerida para os conjuntos de soluções parciais aumenta muito. Destacase que a otimalidade foi provada na instância $nw-ep3-20-F-R-50$ pelo *solver* com o Modelo 3-estágios e pelo Alg-Bot+3est, mas seus valores ótimos diferem. Essa instância foi o único caso nos experimentos computacionais dessa seção, sendo que a premissa de considerar posições de corte no primeiro estágio iguais às dimensões dos tipos de itens conduziu a uma solução sub-ótima com respeito a padrões 3-estágios mais gerais, como aquele indicado na Figura 6.4. Por isso, optou-se por não destacar em negrito a otimalidade de soluções do Alg-Bot+3est segundo as soluções provadas ótimas pelo CPLEX com o Modelo 3-estágios.

Em resumo, os principais resultados dessa seção são: (i) o Modelo 3-estágios parece superar o desempenho do Modelo Não-estagiado, tanto em qualidade de solução e em tempo de processamento para a maioria das instâncias, apesar de lidar com um tipo de padrão de corte mais limitado; (ii) o valor do parâmetro β afeta fortemente o desempenho do Alg-Bot, que foi capaz de provar a otimalidade de algumas instâncias mesmo quando a utilização de material ótima era baixa.

6.4 Comentários finais do capítulo

Os resultados de pesquisa desse capítulo foram desenvolvidos durante estágio de pesquisa (i.e., doutorado sanduíche) na Faculdade de Engenharia da Universidade do Porto, sob orientação do Prof. Dr. José Fernando Oliveira e da Profa. Dra. Elsa Silva,

em conjunto ao Prof. Dr. Reinaldo Morabito e ao Prof. Dr. Pedro Munari. Motivado por uma aplicação industrial de corte guilhotinado de blocos retangulares de aço em empresa situada na cidade de Ovar (Portugal), o projeto consistiu inicialmente em estender algumas das abordagens previamente propostas ao caso bidimensional (i.e., o PCBGR) para o caso tridimensional (i.e., o PCTGR)

Este capítulo abordou o Problema de Corte Tridimensional Guilhotinado e Restrito (PCTGR), a variante tridimensional do PCBGR. O PCTGR surge em sistemas de manufatura, como no corte de blocos de aço, de espumas para colchões e de pedras de mármore. Uma formulação compacta de Programação Não-linear Inteira Mista e sua versão linear foram propostas para modelar o problema, ao estender a formulação proposta no capítulo anterior. Dado que alguns sistemas produtivos limitam suas operações de corte a poucos estágios guilhotinados, buscando tempos de produção menores, também foi proposto um modelo que considera padrões 3-estágios. Como método de solução ao PCTGR, foi estendido ao caso tridimensional o algoritmo de Wang (1983), que propõe sucessivamente combinar os itens até atingir as dimensões do objeto.

Os resultados computacionais usando instâncias de *benchmark* indicaram que as abordagens propostas se destacam em cenários de número moderado de cópias por tipos de itens. Notavelmente, o algoritmo *bottom-up* foi capaz de provar otimalidade em poucos segundos em algumas instâncias, especialmente naquelas cujos tamanhos dos itens são grandes em relação ao objeto. O modelo proposto para padrões 3-estágios mostrou desempenho efetivo em instâncias não-ponderadas e ponderadas, mesmo considerando um padrão de corte mais limitado que o modelo para padrões não-estagiados.

Capítulo 7

Conclusões

Nesta tese de doutorado o objeto de pesquisa foram os Problemas de Corte Guilhotinado e Restritos (PCGR). Nesses problemas buscam-se a seleção mais valiosa do conjunto de itens demandados para produção a partir de um único objeto estocado. Os PCGR têm duas restrições principais: (i) a restrição tecnológica de corte guilhotinado, que gera sempre dois sub-objetos retangulares ao cortar um objeto retangular maior; e, (ii) a restrição de produção restrita, que limita a produção de cada tipo de item a uma certa quantidade de cópias, conforme as estratégias comuns em sistemas produtivos reais. Na verdade, os PCGR conceitualmente representam as decisões associadas a um processo de produção muito comum em indústrias de manufatura, sendo que após o conhecimento da demanda dos produtos, objetos estocados são cortados para produzir os itens demandados. Esses problemas são característicos em processos de corte das indústrias moveleira, metalúrgica, de vidros, de granito e mármore, de aço, de colchões, dentre outras.

De forma geral, a proposição de novas formas efetivas para resolver os Problemas de Corte e Empacotamento (PCE), ao que se incluem os PCGR, é relevante sob dois aspectos. Primeiro sob impacto econômico dado que a resolução efetiva desses problemas contribui diretamente com a redução das matérias-primas utilizadas no sistema produtivo, que podem alcançar valores muito significativos dos custos de produção, e também indiretamente pela redução dos gastos com energia elétrica e com mão de obra. Segundo sob responsabilidade ambiental, dado que a resolução efetiva desses problemas também contribui por reduzir a extração de recursos naturais, e por reduzir a geração de resíduos e descartes industriais e também de lixo, isto é, os PCE estão indiretamente relacionados aos dois extremos das cadeias de suprimentos desses sistemas produtivos. De forma específica aos PCGR, novas formas efetivas de resolução contribuem aos sistemas produtivos de demandas reduzidas para alguns tipos de itens, e de custos de matéria-prima significativos no preço de venda dos produtos finais, visto que se respeita a restrição de produção restrita e não se limita o número de estágios guilhotinados, isto é, se busca maiores taxas de utilização dos objetos.

Inicialmente foi realizada uma cuidadosa revisão da literatura a fim de com-

preender os distintos tipos de PCE, com destaque ao três problemas abordados nesta tese, que são listados a seguir:

- Problema de Corte Bidimensional Guilhotinado e Restrito (PCBGR);
- Problema de Corte Bidimensional Guilhotinado e Restrito de Objeto com Defeitos (PCBGR_D); e,
- Problema de Corte Tridimensional Guilhotinado e Restrito (PCTGR).

Dentre esses três problemas, o PCBGR é aquele que tem sido mais abordado pela literatura, e envolve o corte de retângulos a partir de um retângulo. Como recentemente discutido no *survey* de Russo et al. (2019), que avaliou mais de 90 trabalhos ao problema, seus principais métodos de solução envolvem formulações de Programação Dinâmica, algoritmos de busca em árvores e algoritmos heurísticos/meta-heurísticos. Por outro lado, o PCBGR_D e o PCTGR foram muito pouco abordados pela literatura, isto é, nem objetos heterogêneos devido a regiões defeituosas nem o corte guilhotinado tridimensional tem recebido destaque, apesar da potencial aplicabilidade deles em alguns sistemas produtivos, como na produção de móveis customizados de madeira com nós de crescimento nos objetos, e no corte de blocos de aço e de espuma.

7.1 Contribuições desta tese de doutorado

A principal contribuição dessa tese aos PCGR está na proposição de novos modelos de Otimização Matemática, com destaque para modelos de Programação Linear Inteira Mista (PLIM). O desenvolvimento de modelos matemáticos contribui, para além da caracterização dos problemas e possível uso de *solvers* como estratégia de solução, com a pesquisa de futuros métodos de solução para o problema, a partir da análise de estruturas particulares que permitam reformular o problema por meio de técnicas de decomposição ou relaxação. Particularmente, a revisão bibliográfica realizada surpreendeu ao indicar que a literatura levou mais de três décadas para apresentar formulações matemáticas ao problema – até então, as formulações de Ben Messaoud, Chu e Espinouse (2008) e de Furini, Malaguti e Thomopoulos (2016) a partir dos trabalhos seminais de Christofides e Whitlock (1977) e Wang (1983) –, embora o problema tenha sido amplamente abordado por outras abordagens de solução exatas e heurísticas.

O primeiro esforço de modelagem do PCBGR envolveu estender para o caso guilhotinado a formulação que lida com a versão não-guilhotinada do problema, proposta por Beasley (1985b). Essa abordagem foi aqui chamada de “grelha”, pois o objeto é discretizado; trata-se de uma malha reticulada. A formulação pseudopolinomial proposta envolve decisões de alocações de tipos de itens aos pontos do objeto discretizado, assim como cortes horizontais e verticais no objeto. Inicialmente, foi proposta uma formulação não-linear, e uma formulação linear equivalente foi obtida a partir dela. Adicionalmente,

o modelo linear foi adaptado para lidar com os casos guilhotinados particulares de padrões 2-estágios e 1-grupo. Como método de solução, um algoritmo *branch-and-Benders-cut* foi desenvolvido a partir da formulação linear, em que o Problema-Mestre lida com as restrições de não-sobreposição entre pares de itens e o caso restrito, enquanto o Subproblema gera cortes de factibilidade quanto à restrição de guilhotina. Os experimentos computacionais mostraram que essas abordagens se destacam em cenários de poucos tipos de itens demandados relativamente grandes ao tamanho do objeto a ser cortado.

O segundo esforço de modelagem do PCBGR envolveu representar a abordagem de combinações sucessivas de itens demandados e subpadrões, conhecida como abordagem *bottom-up*, por meio de inequações lineares. Foram propostas duas formulações, uma pseudopolinomial e uma compacta, que partem desse princípio. A principal diferença entre elas é o conhecimento explícito ou implícito de uma estrutura de árvore binária para representar os padrões de corte. Novamente, os modelos obtidos foram não-lineares, e formulações lineares equivalentes foram obtidas. Adicionalmente, foi modelado um conjunto de restrições para essas formulações que permite restringir as soluções obtidas ao caso d -estágios, sendo que d é um escalar inteiro positivo. Os experimentos computacionais mostraram que essas formulações se destacam em cenários que exigem poucas combinações de itens, isto é, de poucos itens na solução final. Além disso, seus números de variáveis e de restrições são relativamente baixos, principalmente quando comparados à formulação estado-da-arte da literatura (FURINI; MALAGUTI; THOMOPULOS, 2016), o que permite abordar instâncias maiores sem enfrentar questões de limitação de memória computacional.

Nesta tese também foram abordadas variantes de problemas de corte guilhotinados menos tratados pela literatura, apesar de serem relevantes em vários sistemas produtivos. A versão do PCBGR de objeto com defeitos foi abordada, isto é, o PCBGR.D. Inicialmente, a formulação linear e o algoritmo *branch-and-Benders-cut*, que partem do conceito de grelha, foram adaptados para essa variante. Tal adaptação é praticamente direta, visto que a discretização do objeto permite facilmente representar os defeitos que estão fixados em determinadas posições. Um método de solução alternativo baseado em Programação por Restrições (PR) foi desenvolvido para essa variante. Os experimentos computacionais mostraram que o algoritmo *branch-and-Benders-cut* tem melhor desempenho no caso irrestrito, enquanto o algoritmo baseado em PR se destaca no caso restrito. Até onde esse autor tem conhecimento, as outras duas formulações matemáticas da literatura (BEN MESSAOUD; CHU; ESPINOUSE, 2008; FURINI; MALAGUTI; THOMOPULOS, 2016) ao PCBGR não são diretamente estendidas ao caso com defeitos.

Em seguida, a versão tridimensional do PCBGR foi abordada, isto é, o PCTGR. Primeiramente, a formulação compacta do Capítulo 5, baseada no conceito *bottom-up*, foi estendida ao caso tridimensional, visto que a adaptação é praticamente direta, e a formulação baseada em grelha conduziria a números excessivos de variáveis e de restrições. Motivado por uma aplicação industrial de corte de aço em poucos estágios

guilhotinados, foi proposta também uma formulação de Programação Linear Inteira (PLI) para o PCTGR que estritamente considera padrões 3-estágios. Como método de solução, o algoritmo combinatório de Wang (1983) foi estendido ao caso tridimensional não-estagiado e 3-estágios. Os experimentos computacionais mostraram que as abordagens propostas são capazes de fornecer soluções ótimas ou quase-ótimas, com destaque para a formulação de PLI para padrões 3-estágios e para o algoritmo combinatório proposto, diante diferentes tipos de instâncias.

Os objetivos de pesquisa desta tese de doutorado foram alcançados, visto que novos modelos matemáticos e novos métodos de solução foram propostos para os problemas estudados. Os resultados mostraram que os métodos são efetivos e têm potencial para resolver problemas práticos, conduzindo à redução dos desperdícios gerados quando comparados, eventualmente às práticas organizacionais, ou mesmo às abordagens estado-da-arte da literatura. Até o momento, três artigos foram publicados em periódicos, referentes aos Capítulos 3, 4 e 5, e outro artigo foi submetido para publicação, referente ao Capítulo 6.

7.2 Pesquisas futuras

A seguir, discute-se algumas possibilidades de pesquisas futuras para cada um dos problemas estudados.

Problema de Corte Bidimensional Guilhotinado e Restrito

As formulações baseadas em grelha ao PCBGR, do Capítulo 3, poderiam ser melhoradas com técnicas adicionais para eliminação de simetrias para o objeto discretizado. A formulação pseudopolinomial de PNLI poderia ser mais explorada no campo de técnicas de Programação por Restrições, no desenvolvimento de abordagens especificamente elaboradas, para além daquelas de *solvers* de propósito geral. Particularmente, pode-se estender o algoritmo B&BC para os casos especiais do PCBGR que estritamente consideram padrões 2-estágios ou 1-grupo. Outro caminho envolve a adaptação dessas formulações de forma a considerar outras aplicações. Por exemplo, para casos em que as distâncias entre cortes guilhotinados são restritas como, por exemplo, na indústria de vidros; esses modelos parecem ser adequados para tal caso.

As formulações baseadas no conceito *bottom-up*, do Capítulo 5, poderiam ser melhoradas com o desenvolvimento de desigualdades válidas mais fortes para apertar suas relaxações lineares e evitar possíveis soluções simétricas. Além disso, pode-se desenvolver técnicas de PLIM de grande porte como, por exemplo, decomposição de Dantzig-Wolfe, decomposição de Benders e relaxação Lagrangeana para desenvolver métodos de solução para instâncias ainda maiores. Outra possibilidade é adaptar os modelos propostos para incorporar incertezas nos dados da instância, por exemplo, no número máximo de cópias

de cada tipo de item a ser manufaturado.

As formulações baseadas no conceito *top-down*, do Apêndice A, poderiam ser investigadas mais profundamente. Essas formulações também poderiam ser exploradas no contexto de métodos de decomposição, como aqueles discutidos no parágrafo anterior. Esse estudo também poderia analisar, por meio de experimentos computacionais sistemáticos, os cenários em que esse tipo de formulação se destaca em comparação às formulações *bottom-up*, ou mesmo em relação às outras formulações da literatura.

Nota-se que também foram exploradas e propostas outras formulações matemáticas ao PCBGR nesta tese de doutorado. Uma delas era baseada no conceito de posições relativas, e buscava propor que cada item alocado estivesse relacionado a um corte horizontal ou vertical. No entanto, uma implementação prévia desse modelo mostrou que ele não representava adequadamente o problema, e admitia inadequadamente o caso não-guilhotinado em alguns cenários. Outra proposta envolveu formular o problema em uma interpretação de grafos de sub-intervalos disjuntos. A premissa era que esses grafos poderiam ser decomponíveis se houvesse um corte horizontal ou vertical que não cortasse os itens. Essa proposta foi interrompida ainda em seu estágio inicial, sem haver um modelo esboçado. Essas duas propostas poderiam ser investigadas mais profundamente a fim de gerar novas formulações ao PCBGR.

Em relação ao PCBGR em si, pode-se considerar futuramente o caso duplamente restrito, isto é, quando existem limitantes inferiores e superiores para produção de cada tipo de item, em vez de apenas limitantes superiores. Além disso, as abordagens propostas poderiam ser adaptadas à versão do PCBGR com múltiplos objetos (i.e., Problema de Corte de Estoque), por exemplo, no contexto de um algoritmo de geração de colunas.

Problema de Corte Bidimensional Guilhotinado e Restrito de Objeto com Defeitos

As formulações baseadas em grelha para padrões 2-estágios e 1-grupo, do Capítulo 3, poderiam ser estendidas para o caso com defeitos, de forma similar ao realizado na formulação de padrões não-estagiados no Capítulo 4. Além disso, essas novas formulações poderiam envolver uma generalização do PCBGR_D que considera tipos de defeitos, isto é, certo tipo de item poderia ser cortado a partir de região com o tipo de defeito A, mas não a partir do tipo de defeito B.

O algoritmo *branch-and-Benders-cut* proposto foi mais efetivo no caso com defeitos (i.e., no Capítulo 4) do que no caso sem defeitos (i.e., no Capítulo 3), provavelmente pela presença de defeitos que indiretamente eliminam soluções simétricas. Esse algoritmo poderia ser mais explorado, por exemplo, ao considerar diferentes tipos de partições de variáveis para a decomposição de Benders para o caso com defeitos. Em outras palavras, algumas variáveis (e restrições) relacionadas à restrição de guilhotina po-

deriam ser mantidas no Problema-Mestre-de-Benders a fim de reduzir o número de cortes combinatórios. Para gerar efeito semelhante no algoritmo *Branch-and-Cut* baseado em Programação por Restrições (i.e., B&C-CP), o uso de um problema de barras relaxadas contíguas (SCHEITHAUER, 2018) dentro do B&C-CP (em vez de um problema da mochila unidimensional) também parece ser bem útil.

As formulações baseadas no conceito *bottom-up*, do Capítulo 5, não parecem ser diretamente adaptáveis ao caso com defeitos. No entanto, pode-se cogitar variáveis adicionais que representassem a alocação dos itens no objeto, segundo os pseudocódigos da Seção 5.5, buscando evitar a sobreposição dos itens e dos defeitos por meio de restrições de um modelo de posições relativas. Essa proposta talvez seja viável para a formulação pseudopolinomial *bottom-up*.

Em relação ao PCBGR_D em si, pode-se considerar futuramente o caso duplamente restrito, isto é, quando existem limitantes inferiores e superiores para produção de cada tipo de item, em vez de apenas limitantes superiores. Além disso, as abordagens propostas poderiam ser adaptadas à versão do PCBGR_D com múltiplos objetos (i.e., Problema de Corte de Estoque), por exemplo, no contexto de um algoritmo de geração de colunas.

Problema de Corte Tridimensional Guilhotinado e Restrito

Como aos problemas anteriores, pode-se considerar o desenvolvimento de desigualdades válidas mais fortes para apertar a relaxação linear dos modelos propostos no Capítulo 6, buscar possíveis eliminações de soluções simétricas, ou mesmo o desenvolvimento de métodos de solução baseados em decomposição, isto é, decomposição de Dantzig-Wolfe, decomposição de Benders e relaxação Lagrangeana.

Em relação ao algoritmo *bottom-up* proposto, novas estratégias para rejeitar soluções factíveis não-promissoras contribuiriam para evitar estouro de memória em instâncias maiores. Outra pesquisa poderia envolver o problema que considera sobras reaproveitáveis durante a análise de corte; ela poderia gerar bons resultados para alguns sistemas produtivos.

Note ainda que as extensões da formulação compacta proposta e do algoritmo *bottom-up* proposto, ambos para padrões não-estagiados, parecem ser relativamente diretas ao caso n -dimensional (isto é, $n > 3$).

Em relação ao PCTGR em si, pode-se considerar futuramente o caso duplamente restrito, isto é, quando existem limitantes inferiores e superiores para produção de cada tipo de item, em vez de apenas limitantes superiores. Além disso, as abordagens propostas poderiam ser adaptadas à versão do PCTGR com múltiplos objetos (i.e., Problema de Corte de Estoque), por exemplo, no contexto de um algoritmo de geração de colunas.

Referências Bibliográficas

AFSHARIAN, M.; NIKNEJAD, A.; WÄSCHER, G. A heuristic, dynamic programming-based approach for a two-dimensional cutting problem with defects. *OR Spectrum*, v. 36, n. 4, p. 971–999, Oct 2014. ISSN 1436-6304. Disponível em: <https://doi.org/10.1007/s00291-014-0363-x>.

ÁLVAREZ-VALDÉS, R.; PARAJÓN, A.; TAMARIT, J. M. A computational study of LP-based heuristic algorithms for two-dimensional guillotine cutting stock problems. *OR Spectrum*, v. 24, n. 2, p. 179–192, 2002. ISSN 01716468.

ÁLVAREZ-VALDÉS, R.; PARAJÓN, A.; TAMARIT, J. M. A tabu search algorithm for large-scale guillotine (un)constrained two-dimensional cutting problems. *Computers & Operations Research*, v. 29, n. 7, p. 925–947, 2002. ISSN 03050548.

AMOSSEN, R. R.; PISINGER, D. Multi-dimensional bin packing problems with guillotine constraints. *Computers & Operations Research*, Elsevier, v. 37, n. 11, p. 1999–2006, 2010. ISSN 03050548.

ARENALES, M.; ARMENTANO, V.; MORABITO, R.; YANASSE, H. *Pesquisa Operacional: para cursos de engenharia*. [S.l.]: Campus, 2007.

BEASLEY, J. E. Algorithms for Unconstrained Two-Dimensional Guillotine Cutting. *The Journal of the Operational Research Society*, v. 36, n. 4, p. 297, 1985. ISSN 01605682. Disponível em: <http://www.jstor.org/stable/2582416?origin=crossref>.

BEASLEY, J. E. An Exact Two-Dimensional Non-Guillotine Cutting Tree Search Procedure. *Operations Research*, v. 33, n. 1, p. 49–64, 1985. ISSN 0030-364X.

BEN MESSAOUD, S.; CHU, C.; ESPINOUSE, M. L. Characterization and modelling of guillotine constraints. *European Journal of Operational Research*, v. 191, n. 1, p. 110–124, 2008. ISSN 03772217.

BENDERS, J. F. Partitioning procedures for solving mixed-variables programming problems. *Numer. Math.*, Springer-Verlag New York, Inc., Secaucus, NJ, USA, v. 4, n. 1, p. 238–252, dez. 1962. ISSN 0029-599X. Disponível em: <http://dx.doi.org/10.1007/BF01386316>.

BERTRAND, J. W.; FRANSOO, F. C. Operations management research methodologies using quantitative modeling. *International Journal of Operation & Production Management*, v. 22, p. 241–264, 2002.

BORTFELDT, A.; WÄSCHER, G. Constraints in container loading-A state-of-the-art review. *European Journal of Operational Research*, Elsevier B.V., v. 229, n. 1, p. 1–20, 2013. ISSN 03772217.

- CARNIERI, C.; MENDOZA, G. A.; LUPPOLD, W. G. Optimal cutting of dimension parts from lumber with a defect. *Forest Products Journal*, v. 43, p. 66–72, 1993.
- CHEN, C.; LEE, S.; SHEN, Q. An analytical model for the container loading problem. *European Journal of Operational Research*, v. 80, n. 1, p. 68–76, jan 1995. ISSN 03772217.
- CHEN, Y. A recursive algorithm for constrained two-dimensional cutting problems. *Computational Optimization and Applications*, v. 41, n. 3, p. 337–348, 2008. ISSN 09266003.
- CHRISTOFIDES, N.; HADJICONSTANTINO, E. An exact algorithm for orthogonal 2-D cutting problems using guillotine cuts. *European Journal of Operational Research*, v. 83, n. 1, p. 21–38, 1995. ISSN 03772217.
- CHRISTOFIDES, N.; WHITLOCK, C. An Algorithm for Two-Dimensional Cutting Problems. *Operations Research*, v. 25, n. 1, p. 30–44, 1977. ISSN 0030-364X.
- CINTRA, G. F.; MIYAZAWA, F. K.; WAKABAYASHI, Y.; XAVIER, E. C. Algorithms for two dimensional cutting stock and strip packing problems using dynamic programming and column generation. *European Journal of Operational Research*, v. 191, n. 1, p. 59–83, 2008. ISSN 03772217.
- COSTA, A. M. A survey on Benders decomposition applied to fixed-charge network design problems. *Computers & Operations Research*, Elsevier, v. 32, n. 6, p. 1429–1450, 2005.
- CUI, Y.; CHEN, Q. Simple heuristic for the constrained two-dimensional cutting problem. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, v. 226, p. 565–572, 2012.
- CUNG, V.-d.; HIFI, M.; Cun, Le, B. Constrained two-dimensional cutting stock problems a best- first branch-and-bound algorithm. *International Transactions in Operational Research*, v. 7, p. 185–210, 2000. ISSN 0969-6016.
- DE QUEIROZ, T. A.; MIYAZAWA, F. K.; WAKABAYASHI, Y.; XAVIER, E. C. Algorithms for 3D guillotine cutting problems: Unbounded knapsack, cutting stock and strip packing. *Computers & Operations Research*, v. 39, n. 2, p. 200–212, 2012. ISSN 03050548.
- DOLATABADI, M.; LODI, A.; MONACI, M. Exact algorithms for the two-dimensional guillotine knapsack. *Computers & Operations Research*, Elsevier, v. 39, n. 1, p. 48–53, 2012. ISSN 03050548. Disponível em: <http://dx.doi.org/10.1016/j.cor.2010.12.018>.
- DURAK, B.; AKSU, D. T. Dynamic programming and mixed integer programming based algorithms for the online glass cutting problem with defects and production targets. *International Journal of Production Research*, Taylor & Francis, v. 55, n. 24, p. 7398–7411, 2017. Disponível em: <https://doi.org/10.1080/00207543.2017.1349951>.
- DYCKHOFF, H. A new linear programming approach to the cutting stock problem. *Operations Research*, INFORMS, v. 29, n. 6, p. 1092–1104, dec 1981. ISSN 0030364X, 15265463. Disponível em: <http://www.jstor.org/stable/170363>.

- DYCKHOFF, H. A typology of cutting and packing problems. *European Journal of Operational Research*, v. 44, n. 2, p. 145 – 159, 1990. ISSN 0377-2217. Cutting and Packing. Disponível em: <http://www.sciencedirect.com/science/article/pii/037722179090350K>.
- DYCKHOFF, H.; KRUSE, H. J.; ABEL, D.; GAL, T. Trim loss and related problems. *Omega*, v. 13, p. 59–72, 1985.
- EGEBLAD, J.; PISINGER, D. Heuristic approaches for the two- and three-dimensional knapsack packing problem. *Computers & Operations Research*, v. 36, n. 4, p. 1026–1049, 2009. ISSN 03050548.
- FAYARD, D.; HIGI, M.; ZISSIMOPOULOS, V. An efficient approach for large-scale two-dimensional guillotine cutting stock problems. *Journal of the Operational Research Society*, v. 49, p. 1270–1277, 1998.
- FEKETE, S. P.; SCHEPERS, J. A new exact algorithm for general orthogonal d-dimensional knapsack problems. In: BURKARD, R.; WOEGINGER, G. (Ed.). *Algorithms — ESA '97*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1997. p. 144–156. ISBN 978-3-540-69536-3.
- FEKETE, S. P.; SCHEPERS, J. A general framework for bounds for higher-dimensional orthogonal packing problems. *Mathematical Methods of Operations Research*, v. 60, n. 2, p. 311–329, Oct 2004. ISSN 1432-5217.
- FURINI, F.; MALAGUTI, E.; Medina Durán, R.; PERSIANI, A.; TOTH, P. A column generation heuristic for the two-dimensional two-staged guillotine cutting stock problem with multiple stock size. *European Journal of Operational Research*, Elsevier B.V., v. 218, n. 1, p. 251–260, 2012. ISSN 03772217. Disponível em: <http://dx.doi.org/10.1016/j.ejor.2011.10.018>.
- FURINI, F.; MALAGUTI, E.; THOMOPULOS, D. Modeling Two-Dimensional Guillotine Cutting Problems via Integer Programming. *INFORMS Journal on Computing*, v. 28, n. 4, p. 736–751, nov 2016. ISSN 1091-9856. Disponível em: <http://pubsonline.informs.org/doi/10.1287/ijoc.2016.0710>.
- GAREY, M. R.; JOHNSON, D. S. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. New York, NY, USA: W. H. Freeman & Co., 1990. ISBN 0716710455.
- GILMORE, P. C.; GOMORY, R. E. A Linear Programming Approach to the Cutting-Stock Problem. *Operations Research*, v. 9, n. 6, p. 849–859, dec 1961. ISSN 0030-364X. Disponível em: <http://pubsonline.informs.org/doi/abs/10.1287/opre.9.6.849>.
- GILMORE, P. C.; GOMORY, R. E. Multistage Cutting Stock Problems of Two and More Dimensions. *Operations Research*, v. 13, n. 1, p. 94–120, 1965. ISSN 0030-364X. Disponível em: <http://pubsonline.informs.org/doi/abs/10.1287/opre.13.1.94>.
- GILMORE, P. C.; GOMORY, R. E. The Theory and Computation of Knapsack Functions. *Operations Research*, v. 14, n. 6, p. 1045–1074, 1966. ISSN 0030-364X. Disponível em: <http://pubsonline.informs.org/doi/abs/10.1287/opre.14.6.1045>.
- HAESSLER, R.; SWEENEY, P. Cutting stock problems and solution procedures. *European Journal of Operational Research*, v. 54, p. 141–150, 1991.

- HAHN, S. G. On the optimal cutting of defective sheets. *Operations Research*, v. 16, n. 6, p. 1100–1114, 1968. Disponível em: <https://doi.org/10.1287/opre.16.6.1100>.
- HERZ, J. C. Recursive Computational Procedure for Two-dimensional Stock Cutting. *IBM Journal of Research and Development*, v. 16, n. 5, p. 462–469, 1972. ISSN 0018-8646.
- HIFI, M. An improvement of Viswanathan and Bagchi's exact algorithm for constrained two-dimensional cutting stock. *Computers & Operations Research*, v. 24, n. 8, p. 727–736, 1997. ISSN 03050548. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0305054896000950>.
- HIFI, M. Approximate algorithms for the container loading problem. *International Transactions in Operational Research*, v. 9, n. 6, p. 747–774, 2002. ISSN 14753995.
- HIFI, M. Dynamic programming and hill-climbing techniques for constrained two-dimensional cutting stock problems. *Journal of Combinatorial Optimization*, v. 8, n. 1, p. 65–84, 2004. ISSN 13826905.
- HIFI, M. Exact algorithms for unconstrained three-dimensional cutting problems: A comparative study. *Computers & Operations Research*, v. 31, n. 5, p. 657–674, 2004. ISSN 03050548.
- HIFI, M.; M'HALLAH, R.; SAADI, T. Approximate and exact algorithms for the double-constrained two-dimensional guillotine cutting stock problem. *Computational Optimization and Applications*, v. 42, n. 2, p. 303–326, 2009. ISSN 09266003.
- JUNQUEIRA, L.; MORABITO, R.; SATO YAMASHITA, D. MIP-based approaches for the container loading problem with multi-drop constraints. *Annals of Operations Research*, v. 199, n. 1, p. 51–75, 2012. ISSN 15729338.
- JUNQUEIRA, L.; MORABITO, R.; SATO YAMASHITA, D. Three-dimensional container loading models with cargo stability and load bearing constraints. *Computers & Operations Research*, Elsevier, v. 39, n. 1, p. 74–85, 2012. ISSN 03050548.
- KANTOROVICH, L. Mathematical methods of organizing and planning production. *Management Science*, v. 6, p. 366–422, 1960.
- LIU, S.; TAN, W.; XU, Z.; LIU, X. A tree search algorithm for the container loading problem. *Computers and Industrial Engineering*, Elsevier Ltd, v. 75, n. 1, p. 20–30, 2014. ISSN 03608352.
- LODI, A.; MARTELLO, S.; MONACI, M. Two-dimensional packing problems: A survey. *European Journal of Operational Research*, v. 141, n. 2, p. 241–252, 2002. ISSN 03772217.
- LODI, A.; MONACI, M. Integer linear programming models for 2-staged two-dimensional Knapsack problems. *Mathematical Programming*, v. 94, n. 2-3, p. 257–278, 2003. ISSN 0025-5610. Disponível em: <http://link.springer.com/10.1007/s10107-002-0319-9>.
- MACEDO, R.; ALVES, C.; Valério de Carvalho, J. Arc-flow model for the two-dimensional guillotine cutting stock problem. *Computers & Operations Research*, v. 37, n. 6, p. 991 – 1001, 2010. ISSN 0305-0548. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0305054809002007>.

- MARTELLO, S.; PISINGER, D.; VIGO, D.; BOEF, E. D.; KORST, J. Algorithm 864. *ACM Transactions on Mathematical Software*, v. 33, n. 1, p. 7–es, mar 2007. ISSN 00983500.
- MISENER, R.; FLOUDAS, C. A. Glomiqo: Global mixed-integer quadratic optimizer. *Journal of Global Optimization*, v. 57, n. 1, p. 3–50, Sep 2013. ISSN 1573-2916. Disponível em: <https://doi.org/10.1007/s10898-012-9874-7>.
- MISENER, R.; FLOUDAS, C. A. Antigone: Algorithms for continuous / integer global optimization of nonlinear equations. *Journal of Global Optimization*, v. 59, n. 2, p. 503–526, Jul 2014. ISSN 1573-2916. Disponível em: <https://doi.org/10.1007/s10898-014-0166-2>.
- MORABITO, R.; ARENALES, M. An and/or-graph approach to the container loading problem. *International Transactions in Operational Research*, Blackwell Publishers Ltd, v. 1, n. 1, p. 59–73, 1994. ISSN 1475-3995. Disponível em: <http://dx.doi.org/10.1111/1475-3995.d01-8>.
- MORABITO, R.; ARENALES, M. Staged and constrained two-dimensional guillotine cutting problems: An AND/OR-graph approach. *European Journal of Operational Research*, v. 94, n. 3, p. 548–560, 1996. ISSN 03772217.
- MORABITO, R.; ARENALES, M. Optimizing the cutting of stock plates in a furniture company. *International Journal of Production Research*, Taylor & Francis, v. 38, n. 12, p. 2725–2742, 2000. Disponível em: <https://doi.org/10.1080/002075400411457>.
- MORABITO, R.; ARENALES, M.; ARCARO, V. An and or graph approach for two-dimensional cutting problems. *European Journal of Operational Research*, v. 58, n. 2, p. 263 – 271, 1992. ISSN 0377-2217. Practical Combinatorial Optimization. Disponível em: <http://www.sciencedirect.com/science/article/pii/037722179290212R>.
- MORABITO, R.; PUREZA, V. A heuristic approach based on dynamic programming and and/or-graph search for the constrained two-dimensional guillotine cutting problem. *Annals of Operations Research*, v. 179, n. 1, p. 297–315, 2010. ISSN 02545330.
- MORENO, A.; MUNARI, P.; ALEM, D. A branch-and-benders-cut algorithm for the crew scheduling and routing problem in road restoration. *European Journal of Operational Research*, v. 275, n. 1, p. 16 – 34, 2019. ISSN 0377-2217. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0377221718309299>.
- NEIDLEIN, V.; SCHOLZ, A.; WÄSCHER, G. SLOPPGEN: a problem generator for the two-dimensional rectangular single large object placement problem with defects. *International Transactions in Operational Research*, v. 23, n. 1-2, p. 173–186, 2016. ISSN 1475-3995. Disponível em: <http://dx.doi.org/10.1111/itor.12119>.
- NEIDLEIN, V.; VIANNA, A. C. G.; ARENALES, M. N.; WÄSCHER, G. *The Two-Dimensional, Rectangular, Guillotineable-Layout Cutting Problem with a Single Defect*. [S.l.], 2008. Disponível em: <https://ideas.repec.org/p/mag/wpaper/08035.html>.
- OLIVEIRA, J.; FERREIRA, J. An improved version of Wang's algorithm for two-dimensional cutting problems. *European Journal of Operational Research*, v. 44, n. 2, p. 256–266, 1990. ISSN 03772217.

- OLIVEIRA, J. F.; WÄSCHER, G. Special issue on cutting and packing. *European Journal of Operational Research*, v. 183, 2007.
- PARADA, V.; GÓMES, A.; DIEGO, J. de. Exact solutions for constrained two-dimensional cutting problems. *European Journal of Operational Research*, v. 84, n. 3, p. 633–644, 1995. ISSN 03772217.
- PARADA, V.; MUÑOZ, R.; ALVARENGA, A. G. de. A hybrid genetic algorithm for the two - dimensional guillotine cutting problem. In: _____. *Evolutionary Algorithms in Management Applications*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1995. p. 183–196. ISBN 978-3-642-61217-6. Disponível em: http://dx.doi.org/10.1007/978-3-642-61217-6_9.
- PARADA, V.; PALMA, R.; SALES, D.; GÓMES, A. A comparative numerical analysis for the guillotine two-dimensional cutting problem. *Annals of Operations Research*, v. 96, n. 1, p. 245–254, 2000. ISSN 1572-9338. Disponível em: <http://dx.doi.org/10.1023/A:1018915922011>.
- PARADA, V.; SEPÚLVEDA, M.; SOLAR, M.; GÓMES, A. Solution for the constrained guillotine cutting problem by simulated annealing. *Computers & Operations Research*, v. 25, n. 1, p. 37–47, 1998. ISSN 03050548. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0305054898800063>.
- PISINGER, D.; SIGURD, M. Using decomposition techniques and constraint programming for solving the two-dimensional bin-packing problem. *INFORMS Journal on Computing*, v. 19, n. 1, p. 36–51, 2007. ISSN 10919856.
- PUCHINGER, J.; RAIDL, G. R. Models and algorithms for three-stage two-dimensional bin packing. *European Journal of Operational Research*, v. 183, n. 3, p. 1304–1327, 2007. ISSN 03772217.
- RAHMANIANI, R.; CRAINIC, T. G.; GENDREAU, M.; REI, W. The Benders decomposition algorithm: A literature review. *European Journal of Operational Research*, Elsevier B.V., v. 259, n. 3, p. 801–817, 2017. ISSN 03772217.
- ROSA, G. *Grande Sertão: Veredas*. 1. ed. [S.l.]: Editora Jose Olympio, 1956.
- RUSSO, M.; BOCCIA, M.; SFORZA, A.; STERLE, C. Constrained two-dimensional guillotine cutting problem: upper-bound review and categorization. *International Transactions in Operational Research*, v. 0, n. 0, 2019. Disponível em: <https://onlinelibrary.wiley.com/doi/abs/10.1111/itor.12687>.
- SCHEITHAUER, G. Algorithms for the container loading problem. In: GAUL, W.; BACHEM, A.; HABENICHT, W.; RUNGE, W.; STAHL, W. W. (Ed.). *Operations Research Proceedings 1991*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1992. p. 445–452. ISBN 978-3-642-46773-8.
- SCHEITHAUER, G. *Introduction to cutting and packing optimization: problems, modeling approaches, solution methods*. [S.l.]: Springer, 2018. (International Series in Operations Research and Management Science).
- SCHEITHAUER, G.; TERNO, J. Guillotine cutting of defective boards. *Optimization*, Taylor & Francis, v. 19, n. 1, p. 111–121, 1988. Disponível em: <https://doi.org/10.1080/02331938808843323>.

- SILVA, E.; ALVELOS, F.; Valério de Carvalho, J. M. An integer programming model for two- and three-stage two-dimensional cutting stock problems. *European Journal of Operational Research*, Elsevier B.V., v. 205, n. 3, p. 699–708, 2010. ISSN 03772217. Disponível em: <http://dx.doi.org/10.1016/j.ejor.2010.01.039>.
- SILVA, E.; OLIVEIRA, J. F.; WÄSCHER, G. 2DCPackGen: A problem generator for two-dimensional rectangular cutting and packing problems. *European Journal of Operational Research*, v. 237, n. 3, p. 846 – 856, 2014. ISSN 0377-2217. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0377221714002112>.
- SILVA, E. F.; TOFFOLO, T. A. M.; WAUTERS, T. Exact methods for three-dimensional cutting and packing: A comparative study concerning single container problems. *Computers & Operations Research*, Elsevier Ltd, v. 109, p. 12–27, 2019. ISSN 03050548.
- SLACK, N.; CHAMBERS, S.; JOHNSTON, R. *Administração da Produção*. [S.l.]: Editora Atlas, 2009.
- TAWARMALANI, M.; SAHINIDIS, N. V. A polyhedral branch-and-cut approach to global optimization. *Mathematical Programming*, v. 103, n. 2, p. 225–249, Jun 2005. ISSN 1436-4646. Disponível em: <https://doi.org/10.1007/s10107-005-0581-8>.
- TOFFOLO, T. A.; ESPRIT, E.; WAUTERS, T.; Vanden Berghe, G. A two-dimensional heuristic decomposition approach to a three-dimensional multiple container loading problem. *European Journal of Operational Research*, v. 257, n. 2, p. 526–538, 2017. ISSN 03772217.
- TRICK, M. A. A Dynamic Programming Approach for Consistency and Propagation for Knapsack Constraints. *Annals of Operations Research*, v. 118, n. 1-4, p. 73–84, 2003. ISSN 02545330.
- VASKO, F. J. A computational improvement to wang’s two-dimensional cutting stock algorithm. *Computers & Industrial Engineering*, v. 16, n. 1, p. 109 – 115, 1989. ISSN 0360-8352. Disponível em: <http://www.sciencedirect.com/science/article/pii/0360835289900132>.
- VASKO, F. J.; BARTKOWSKI, C. L. Using wang’s two-dimensional cutting stock algorithm to optimally solve difficult problems. *International Transactions in Operational Research*, v. 16, n. 6, p. 829–838, 2009. Disponível em: <https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1475-3995.2009.00708.x>.
- VELASCO, A. S.; UCHOA, E. Improved state space relaxation for constrained two-dimensional guillotine cutting problems. *European Journal of Operational Research*, v. 272, n. 1, p. 106 – 120, 2019. ISSN 0377-2217. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0377221718305393>.
- VIANNA, A. C. G.; ARENALES, M. N. O problema de corte de placas defeituosas. *Pesquisa Operacional*, v. 26, p. 185 – 202, 08 2006. ISSN 0101-7438.
- VIGERSKE, S.; HEINZ, S.; GLEIXNER, A.; BERTHOLD, T. Analyzing the computational impact of MIQCP solver components. *Numerical Algebra, Control and Optimization*, American Institute of Mathematical Sciences (AIMS), v. 2, n. 4, p. 739–748, nov 2012. Disponível em: <https://doi.org/10.3934%2Fnao.2012.2.739>.

VISWANATHAN, K. V.; BAGCHI, A. Best-First Search Methods for Constrained Two-Dimensional Cutting Stock Problems. *Operations Research*, v. 41, n. 4, p. 768–776, 1993. ISSN 0030-364X. Disponível em: <http://dx.doi.org/10.1287/opre.41.4.768>.

WANG, P. Y. Two Algorithms for Constrained Two-Dimensional Cutting Stock Problems. *Operations Research*, v. 31, n. 3, p. 573–586, 1983. Disponível em: <http://dx.doi.org/10.1287/opre.31.3.573>.

WÄSCHER, G.; HAUSSNER, H.; SCHUMANN, H. An improved typology of cutting and packing problems. *European Journal of Operational Research*, v. 183, n. 3, p. 1109–1130, 2007. ISSN 03772217.

WEI, L.; LIM, A. A bidirectional building approach for the 2D constrained guillotine knapsack packing problem. *European Journal of Operational Research*, Elsevier Ltd., v. 242, n. 1, p. 63–71, 2015. ISSN 03772217. Disponível em: <http://dx.doi.org/10.1016/j.ejor.2014.10.004>.

YANASSE, H. H.; KATSURAYAMA, D. M. Checkerboard pattern: proposals for its generation. *International Transactions in Operational Research*, v. 12, n. 1, p. 21–45, 2005. Disponível em: <https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1475-3995.2005.00488.x>.

YANASSE, H. H.; KATSURAYAMA, D. M. An enumeration scheme to generate constrained exact checkerboard patterns. *Computers & Operations Research*, v. 35, n. 6, p. 2114 – 2128, 2008. ISSN 0305-0548. Part Special Issue: OR Applications in the Military and in Counter-Terrorism. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0305054806002759>.

YANASSE, H. H.; MORABITO, R. Linear models for 1-group two-dimensional guillotine cutting problems. *International Journal of Production Research*, v. 44, n. 17, p. 3471–3491, 2006. ISSN 0020-7543. Disponível em: <http://www.tandfonline.com/doi/abs/10.1080/00207540500478603>.

YANASSE, H. H.; MORABITO, R. A note on linear models for two-group and three-group two-dimensional guillotine cutting problems. *International Journal of Production Research*, v. 46, n. 21, p. 6189–6206, 2008. ISSN 0020-7543. Disponível em: <http://www.tandfonline.com/doi/abs/10.1080/00207540601011543>.

YOON, K.; AHN, S.; KANG, M. An improved best-first branch-and-bound algorithm for constrained two-dimensional guillotine cutting problems. *International Journal of Production Research*, v. 51, n. 6, p. 1680–1693, 2013. ISSN 0020-7543. Disponível em: <http://www.tandfonline.com/doi/abs/10.1080/00207543.2012.693965>.

ZHAO, X.; BENNELL, J. A.; BEKTAŞ, T.; DOWSLAND, K. A comparative review of 3D container loading algorithms. *International Transactions in Operational Research*, v. 23, n. 1-2, p. 287–320, 2016. ISSN 14753995.

Apêndices

Apêndice A

Problema de corte bidimensional guilhotinado e restrito: propostas de formulações matemáticas com o conceito *top-down*

A.1 Introdução

No Capítulo 5 foram propostas formulações matemáticas para o PCBGR baseadas no conceito de empacotamento *bottom-up*, que propõe construir horizontal e verticalmente novos retângulos, a partir os itens demandados. Essas formulações podem ser reescritas sob perspectiva do conceito de corte *top-down*, originalmente proposto por Christofides e Whitlock (1977), em que o objeto (chapa) é cortado seguidamente até obtenção dos itens demandados.

Os experimentos computacionais preliminares realizados, considerando o *solver* de propósito geral *IBM CPLEX Optimization Studio v12.8*, indicaram que os modelos *top-down* têm desempenho médio inferior aos modelos *bottom-up*, o que não motivou estudos mais detalhados. Nas Seções A.2 e A.3 apresentam-se os modelos hierárquico e implícito reformulados sob conceito *top-down*, respectivamente.

A.2 Formulação matemática *top-down* hierárquica

No modelo *top-down* hierárquico considera-se que cada retângulo, caso exista, corresponde a um dos seguintes: uma cópia de tipo de item, um corte vertical, ou um corte horizontal. Na Figura A.1 ilustra-se tal compreensão e as variáveis relacionadas que serão definidas no modelo são destacadas. Nota-se que: (i) um retângulo somente representa um tipo de item, se cada uma de suas dimensões é maior que a dimensão correspondente do tipo de item; (ii) diante corte vertical, a largura do retângulo é mantida nos dois sub-retângulos menores resultantes; (iii) diante corte horizontal, o comprimento

do retângulo é mantido nos dois sub-retângulos menores resultantes.

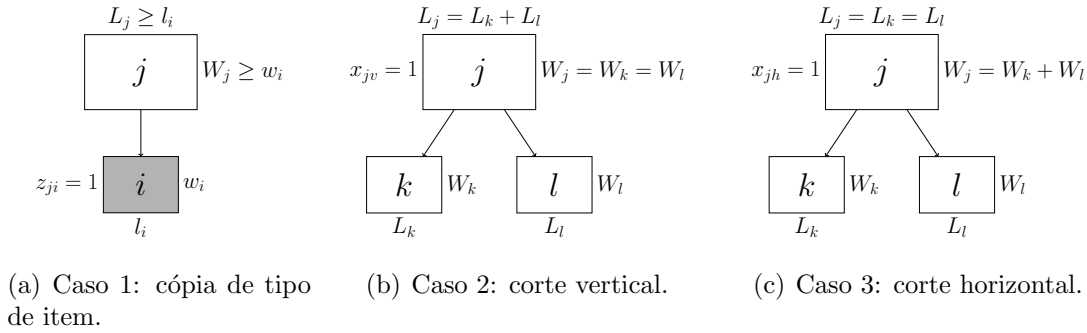


Figura A.1: Ilustrações do modelo *top-down* hierárquico.

Seja $I = \{1, \dots, m\}$ o conjunto dos tipos de itens demandados. Seja $J = \{1, \dots, \mathcal{N}\}$ o conjunto de todos os nós da árvore binária, gerada pelo Algoritmo 5.1.1 do Capítulo 5, assim como os conjuntos disjuntos J_A , J_B , J_C e \bar{J} , aqueles gerados pelo algoritmo, que categorizam cada retângulo $j \in J$. Seja $O = \{h, v\}$ o conjunto de todas as orientações possíveis para cortar um retângulo, onde h e v são cortes horizontal e vertical, respectivamente. Sobre as variáveis de decisão, há quatro conjuntos de variáveis no modelo, que são definidas em (A.1) até (A.4). Nota-se que o domínio das variáveis L_j e W_j são para todo $j \in J$ (em vez de $j \in J \setminus \bar{J}$ como no Capítulo 5), pois o modelo aqui proposto requer que cada retângulo seja obtido por um corte, para em seguida analisar o tipo de item que ele pode conter.

$$z_{ji} = \begin{cases} 1, & \text{se o nó } j \text{ representa uma cópia do tipo de item } i, \\ 0, & \text{caso contrário.} \end{cases} \quad j \in J, i \in I. \quad (\text{A.1})$$

$$x_{jo} = \begin{cases} 1, & \text{se o nó } j \text{ é cortado com orientação } o, \\ 0, & \text{caso contrário.} \end{cases} \quad j \in J \setminus \bar{J}, o \in O. \quad (\text{A.2})$$

$$L_j: \text{ é o comprimento do nó } j, \quad j \in J. \quad (\text{A.3})$$

$$W_j: \text{ é a largura do nó } j, \quad j \in J. \quad (\text{A.4})$$

Tendo definidos todas as variáveis e parâmetros, no Modelo (A.5) declara-se uma formulação pseudopolinomial de PLIM para o PCBGR, baseada no conceito *top-down*. Essa formulação é não-compacta, pois suas variáveis e restrições são definidas em termos dos nós da estrutura de árvore binária gerada pelo Algoritmo 5.1.1.

$$\text{Max} \sum_{j \in J} \sum_{i \in I} v_i z_{ji}, \quad (\text{A.5a})$$

s.a

$$\sum_{j \in J} z_{ji} \leq u_i, \quad i \in I, \quad (\text{A.5b})$$

$$\sum_{i \in I} z_{ji} + \sum_{o \in O} x_{jo} \leq 1, \quad j \in J \setminus \bar{J}, \quad (\text{A.5c})$$

$$\sum_{i \in I} z_{ji} \leq 1, \quad j \in \bar{J}, \quad (\text{A.5d})$$

$$\sum_{o \in O} x_{j-o} + \sum_{i \in I} z_{j-i} + \sum_{o \in O} x_{j+o} + \sum_{i \in I} z_{j+i} = 2 \sum_{o \in O} x_{jo}, \quad j \in J_A, \quad (\text{A.5e})$$

$$\sum_{o \in O} x_{j-o} + \sum_{i \in I} z_{j-i} + \sum_{i \in I} z_{j+i} = 2 \sum_{o \in O} x_{jo}, \quad j \in J_B, \quad (\text{A.5f})$$

$$\sum_{i \in I} z_{j-i} + \sum_{i \in I} z_{j+i} = 2 \sum_{o \in O} x_{jo}, \quad j \in J_C, \quad (\text{A.5g})$$

$$L_j \geq L_{j^-} + L_{j^+} - L(1 - x_{jv}), \quad j \in J \setminus \bar{J}, \quad (\text{A.5h})$$

$$L_j \leq L_{j^-} + L_{j^+} + L(1 - x_{jv}), \quad j \in J \setminus \bar{J}, \quad (\text{A.5i})$$

$$L_j - L_{j^*} \leq L(1 - x_{jh}), \quad j \in J \setminus \bar{J}, j^* \in \{j^-, j^+\}, \quad (\text{A.5j})$$

$$L_j - L_{j^*} \geq -L(1 - x_{jh}), \quad j \in J \setminus \bar{J}, j^* \in \{j^-, j^+\}, \quad (\text{A.5k})$$

$$W_j \geq W_{j^-} + W_{j^+} - W(1 - x_{jh}), \quad j \in J \setminus \bar{J}, \quad (\text{A.5l})$$

$$W_j \leq W_{j^-} + W_{j^+} + W(1 - x_{jh}), \quad j \in J \setminus \bar{J}, \quad (\text{A.5m})$$

$$W_j - W_{j^*} \leq W(1 - x_{jv}), \quad j \in J \setminus \bar{J}, j^* \in \{j^-, j^+\}, \quad (\text{A.5n})$$

$$W_j - W_{j^*} \geq -W(1 - x_{jv}), \quad j \in J \setminus \bar{J}, j^* \in \{j^-, j^+\}, \quad (\text{A.5o})$$

$$\sum_{i \in I} l_i z_{ji} \leq L_j, \quad j \in J, \quad (\text{A.5p})$$

$$\sum_{i \in I} w_i z_{ji} \leq W_j, \quad j \in J, \quad (\text{A.5q})$$

$$z_{ji} \in \{0, 1\}, \quad j \in J, i \in I, \quad (\text{A.5r})$$

$$x_{jo} \in \{0, 1\}, \quad j \in J \setminus \bar{J}, o \in O, \quad (\text{A.5s})$$

$$0 \leq L_j \leq L, 0 \leq W_j \leq W, \quad j \in J, \quad (\text{A.5t})$$

$$L_1 = L, W_1 = W. \quad (\text{A.5u})$$

A função objetivo (A.5a) e as restrições (A.5b) até (A.5g) estão conforme as definições feitas no Capítulo 5. As restrições (A.5u), na verdade, são uma fixação de variáveis em relação ao tamanho do objeto original (i.e., retângulo $j = 1$).

As restrições disjuntivas (A.5h), (A.5i), (A.5j) e (A.5k) são do tipo M-grande e definem as variáveis L_j dos retângulos $j \in J$. Para um $j \in J$, as restrições (A.5h) e (A.5i) definem que o valor da variável L_j corresponde a soma dos valores das variáveis L_{j^-}

e L_{j+} quando ocorre um corte vertical (isto é, quando $x_{jv} = 1$). Por outro lado, quando ocorre um corte horizontal (isto é, quando $x_{jh} = 1$), as restrições (A.5j) e (A.5k) definem que os valores das variáveis L_{j-} e L_{j+} são idênticos ao valor da variável L_j . As restrições disjuntivas (A.5l) até (A.5o) atuam de forma similar as restrições (A.5h) até (A.5k), mas para as variáveis W_j .

As restrições (A.5p) e (A.5q) garantem que um retângulo $j \in J$ contém (diretamente) a cópia do tipo de item $i \in I$, somente se $L_j \geq l_i$ e $W_j \geq w_i$.

As restrições (A.5r), (A.5s), (A.5t) e (A.5u) definem o domínio das variáveis de decisão. Observe que esse modelo não requer que os dados de entrada (isto é, L , W , l_i e w_i , para cada $i \in I$) sejam inteiros.

A.3 Formulação matemática *top-down* implícita

Assim como o modelo da seção anterior, no modelo *top-down* implícito considera-se que cada retângulo, caso exista, corresponde a um dos seguintes: uma cópia de tipo de item, um corte vertical, ou um corte horizontal. Nota-se que esse conceito é diferente daquele do correspondente modelo *bottom-up*, em que um retângulo, caso exista, é sempre uma construção horizontal ou vertical de subpadrões menores e/ou cópias de tipos de itens – veja a Figura 5.7. Dessa forma, o conjunto de todos os retângulos possíveis de serem construídos (isto é, subpadrões) é definido como $J = \{1, \dots, 2\bar{n} - 1\}$ – em vez de $J = \{1, \dots, \bar{n} - 1\}$ como no correspondente modelo *bottom-up*. Isso é necessário, pois primeiro os retângulos são gerados via corte, e depois são analisados sobre quais cópias de tipos de itens representam. Seja $O = \{h, v\}$ o conjunto de todas as orientações possíveis para construir um retângulo, onde h e v são para cortes horizontais e verticais, respectivamente. Sobre as variáveis de decisão, existem cinco conjuntos de variáveis nesse modelo, que são definidas nas Equações (A.6) até (A.10).

$$z_{ji} = \begin{cases} 1, & \text{se o retângulo } j \text{ representa uma cópia do tipo de item } i, \\ 0, & \text{caso contrário.} \end{cases} \quad j \in J, i \in I. \quad (\text{A.6})$$

$$x_{jo} = \begin{cases} 1, & \text{se o retângulo } j \text{ é cortado com orientação } o, \\ 0, & \text{caso contrário.} \end{cases} \quad j \in J, o \in O. \quad (\text{A.7})$$

$$y_{jk} = \begin{cases} 1, & \text{se o retângulo } j \text{ contém o retângulo } k, \\ 0, & \text{caso contrário,} \end{cases} \quad j \in J, k \in J, j < k. \quad (\text{A.8})$$

L_j : é o comprimento do retângulo j , $j \in J$. (A.9)

W_j : é a largura do retângulo j , $j \in J$. (A.10)

Usando as variáveis e parâmetros definidos até então, no Modelo (A.11) tem-se uma formulação compacta de PNLIM para o PCBGR, baseada no conceito *top-down*.

$$\text{Max} \sum_{j \in J} \sum_{i \in I} v_i z_{ji}, \quad (\text{A.11a})$$

s.a

$$\sum_{j \in J} z_{ji} \leq u_i, \quad i \in I, \quad (\text{A.11b})$$

$$\sum_{i \in I} z_{ji} + \sum_{o \in O} x_{jo} \leq 1, \quad j \in J, \quad (\text{A.11c})$$

$$2 \sum_{o \in O} x_{jo} = \sum_{k \in J, j < k} y_{jk}, \quad j \in J \quad (\text{A.11d})$$

$$\sum_{j \in J, j < k} y_{jk} = \sum_{o \in O} x_{ko} + \sum_{i \in I} z_{ki}, \quad k \in \{2, \dots, \bar{n}\} \quad (\text{A.11e})$$

$$L_j \geq \sum_{k \in J, j < k} L_k y_{jk} - 2L(1 - x_{jv}), \quad j \in J, \quad (\text{A.11f})$$

$$L_j \leq \sum_{k \in J, j < k} L_k y_{jk} + 2L(1 - x_{jv}), \quad j \in J, \quad (\text{A.11g})$$

$$L_j - \sum_{k \in J, j < k} L_k y_{jk} \leq L(2 - x_{jh} - y_{jk}), \quad j \in J, k \in K, j < k, \quad (\text{A.11h})$$

$$L_j - \sum_{k \in J, j < k} L_k y_{jk} \geq -L(2 - x_{jh} - y_{jk}), \quad j \in J, k \in K, j < k, \quad (\text{A.11i})$$

$$W_j \geq \sum_{k \in J, j < k} W_k y_{jk} - 2W(1 - x_{jh}), \quad j \in J, \quad (\text{A.11j})$$

$$W_j \leq \sum_{k \in J, j < k} W_k y_{jk} + 2W(1 - x_{jh}), \quad j \in J, \quad (\text{A.11k})$$

$$W_j - \sum_{k \in J, j < k} W_k y_{jk} \leq W(2 - x_{jv} - y_{jk}), \quad j \in J, k \in K, j < k, \quad (\text{A.11l})$$

$$W_j - \sum_{k \in J, j < k} W_k y_{jk} \geq -W(2 - x_{jv} - y_{jk}), \quad j \in J, k \in K, j < k, \quad (\text{A.11m})$$

$$\sum_{i \in I} l_i z_{ji} \leq L_j, \quad j \in J, \quad (\text{A.11n})$$

$$\sum_{i \in I} w_i z_{ji} \leq W_j, \quad j \in J, \quad (\text{A.11o})$$

$$z_{ji} \in \{0, 1\}, \quad j \in J, i \in I, \quad (\text{A.11p})$$

$$y_{jk} \in \{0, 1\}, \quad j \in J, k \in J, j < k, \quad (\text{A.11q})$$

$$x_{jo} \in \{0, 1\}, \quad j \in J, o \in O, \quad (\text{A.11r})$$

$$0 \leq L_j \leq L, 0 \leq W_j \leq W, \quad j \in J, \quad (\text{A.11s})$$

$$L_1 = L, W_1 = W. \quad (\text{A.11t})$$

A função objetivo (A.11a) consiste em maximizar a soma total dos valores dos itens selecionados. As restrições (A.11b) garantem que cada tipo de item $i \in I$ é selecionado em até seu número máximo de cópias permitido de ser manufaturado. As restrições (A.11c) limitam que cada retângulo, caso exista, consiste de uma cópia de tipo de item, ou de corte horizontal ou vertical.

As restrições (A.11d) garantem que cada subpadrão cortado deve conter dois subpadrões menores. As restrições (A.11e) asseguram que cada retângulo k contido em um subpadrão maior j deve ser um subpadrão (isto é, ser cortado) ou representar a cópia de um tipo de item.

As restrições disjuntivas (A.11f), (A.11g), (A.11h) e (A.11i) são do tipo M-grande e definem as variáveis L_j dos retângulos $j \in J$. Para um $j \in J$, as restrições (A.11f) e (A.11g) definem que o valor da variável L_j corresponde a soma dos valores das variáveis $L_k y_{jk}$, quando ocorre um corte vertical (isto é, quando $x_{jv} = 1$) e $y_{jk} = 1$. Por outro lado, quando ocorre um corte horizontal (isto é, quando $x_{jh} = 1$) e $y_{jk} = 1$, as restrições (A.11h) e (A.11i) definem que os valores das variáveis $L_k y_{jk}$ são idênticos ao valor da variável L_j . As restrições disjuntivas (A.11j) até (A.11m) atuam de forma similar as restrições (A.5l) até (A.5o), mas para as variáveis W_j .

As restrições (A.11n) e (A.11o) atuam de forma similar as restrições (A.5p) e (A.5q) da seção anterior; no entanto, note que seus conjuntos J têm definições diferentes. As restrições (A.11p) até (A.11s) definem o domínio das variáveis de decisão. As restrições (A.11t), na verdade, são uma fixação de variáveis em relação ao tamanho do objeto original (i.e., retângulo $j = 1$). Esse modelo é não-linear devido ao produto de variáveis nas restrições (A.11f) e (A.11m). De forma similar ao modelo proposto na seção anterior, esse modelo não requer que os dados de entrada (isto é, L , W , l_i e w_i , para cada $i \in I$) sejam inteiros.

Apêndice B

Seções adicionais do Capítulo 6 do Problema de corte tridimensional guilhotinado e restrito

B.1 Formulação linear para padrões não-estagiados

Por meio de variáveis adicionais e restrições disjuntivas, o Modelo (6.7) pode ser reformulado em uma formulação de Programação Linear Inteira Mista (PLIM). Essas variáveis adicionais são definidas em (B.1), (B.2) e (B.3).

$$y_{jk}^L: \text{comprimento do bloco } k \text{ contido pelo bloco } j, \quad j \in J, k \in J, j < k. \quad (\text{B.1})$$

$$y_{jk}^W: \text{largura do bloco } k \text{ contido pelo bloco } j, \quad j \in J, k \in J, j < k. \quad (\text{B.2})$$

$$y_{jk}^H: \text{altura do bloco } k \text{ contido pelo bloco } j, \quad j \in J, k \in J, j < k. \quad (\text{B.3})$$

Assim, o Modelo (B.4) é uma formulação compacta de PLIM para o PCTGR.

Max (6.7a),

s.t.

$$(6.7b) - (6.7f), (6.7j) - (6.7n),$$

$$0 \leq y_{jk}^L \leq \bar{L}y_{jk}, \quad j \in J, k \in J, j < k, \quad (\text{B.4a})$$

$$L_k - \bar{L}(1 - y_{jk}) \leq y_{jk}^L \leq L_k, \quad j \in J, k \in J, j < k, \quad (\text{B.4b})$$

$$0 \leq y_{jk}^W \leq \bar{W}y_{jk}, \quad j \in J, k \in J, j < k, \quad (\text{B.4c})$$

$$W_k - \bar{W}(1 - y_{jk}) \leq y_{jk}^W \leq W_k, \quad j \in J, k \in J, j < k, \quad (\text{B.4d})$$

$$0 \leq y_{jk}^H \leq \bar{H}y_{jk}, \quad j \in J, k \in J, j < k, \quad (\text{B.4e})$$

$$H_k - \bar{H}(1 - y_{jk}) \leq y_{jk}^H \leq H_k, \quad j \in J, k \in J, j < k, \quad (\text{B.4f})$$

$$L_j \geq \sum_{i \in I} \sum_{p \in Q_i} l_i z_{jip} + \sum_{k \in K, j < k} y_{jk}^L - 2\bar{L}(1 - x_{jh}), \quad j \in J, \quad (\text{B.4g})$$

$$L_j \geq l_i z_{jip} - \bar{L}(1 - x_{jv} - x_{jp}), \quad j \in J, i \in I, p \in Q_i, \quad (\text{B.4h})$$

$$L_j \geq y_{jk}^L - \bar{L}(1 - x_{jv} - x_{jp}), \quad j \in J, k \in K, j < k, \quad (\text{B.4i})$$

$$W_j \geq w_i z_{jip} - \bar{W}(1 - x_{jh} - x_{jv}), \quad j \in J, i \in I, p \in Q_i, \quad (\text{B.4j})$$

$$W_j \geq y_{jk}^W - \bar{W}(1 - x_{jh} - x_{jv}), \quad j \in J, k \in K, j < k, \quad (\text{B.4k})$$

$$W_j \geq \sum_{i \in I} \sum_{p \in Q_i} w_i z_{jip} + \sum_{k \in K, j < k} y_{jk}^W - 2\bar{W}(1 - x_{jd}), \quad j \in J, \quad (\text{B.4l})$$

$$H_j \geq h_i z_{jip} - \bar{H}(1 - x_{jh} - x_{jd}), \quad j \in J, i \in I, p \in Q_i, \quad (\text{B.4m})$$

$$H_j \geq y_{jk}^H - \bar{H}(1 - x_{jh} - x_{jd}), \quad j \in J, k \in K, j < k, \quad (\text{B.4n})$$

$$H_j \geq \sum_{i \in I} \sum_{p \in Q_i} h_i z_{jip} + \sum_{k \in K, j < k} y_{jk}^H - 2\bar{H}(1 - x_{jv}), \quad j \in J. \quad (\text{B.4o})$$

As restrições (B.4a) e (B.4b) definem que o valor da variável y_{jk}^L é L_k quando $y_{jk} = 1$, e zero caso contrário (i.e., $y_{jk} = 0$). Similarmente, as restrições (B.4c)–(B.4d) e (B.4e)–(B.4f) definem esse comportamento para as variáveis y_{jk}^W e y_{jk}^H . Cada restrição não-linear (6.7h) do Modelo (6.7) pode ser substituída, sem perda de generalidade, pelas três respectivas restrições lineares (B.4g)–(B.4i). Note que quando o subpadrão $j \in J$ é construído com orientação h (i.e., quando $x_{jh} = 1$), apenas a restrição (B.4g) está ativa, mas se ele é construído com as orientações d ou v (i.e., quando $x_{jv} + x_{jp} = 1$), apenas as restrições (B.4h) e (B.4i) estão ativas. As restrições (B.4j)–(B.4l) e (B.4m)–(B.4o) são similares as restrições (B.4g)–(B.4i), mas impõem esse comportamento para as variáveis W_j e H_j , respectivamente.

A seguir, as Desigualdades Válidas (5.18) são estendidas para o caso tridimensional nas restrições (B.5). A restrição (B.5a) limita o volume total dos itens selecionados ao parâmetro UB_1 . A restrição (B.5b) limita o valor total dos itens selecionados (i.e., a função objetivo) ao parâmetro UB_2 . Ambos parâmetros UB_1 e UB_2 podem ser calculados por qualquer relaxação do PCTGR. A restrição (B.5c) limita o número de itens selecionados ao parâmetro \bar{n} . As restrições (B.5d), (B.5e) e (B.5f) garantem que as variáveis L_j , W_j e H_j assumem valores positivos apenas quando o respectivo subpadrão j existe (i.e., quando $\sum_{o \in O} x_{jo} = 1$).

$$\sum_{j \in J} \sum_{i \in I} \sum_{p \in Q_i} (l_i w_i h_i) z_{jip} \leq UB_1, \quad (\text{B.5a})$$

$$\sum_{j \in J} \sum_{i \in I} \sum_{p \in Q_i} p_i z_{jip} \leq UB_2, \quad (\text{B.5b})$$

$$\sum_{j \in J} \sum_{i \in I} \sum_{p \in Q_i} z_{jip} \leq \bar{n}, \quad (\text{B.5c})$$

$$L_j \leq \bar{L} \sum_{o \in O} x_{jo}, \quad j \in J, \quad (\text{B.5d})$$

$$W_j \leq \bar{W} \sum_{o \in O} x_{jo}, \quad j \in J, \quad (\text{B.5e})$$

$$H_j \leq \bar{H} \sum_{o \in O} x_{jo}. \quad j \in J. \quad (\text{B.5f})$$

B.2 Pseudocódigo para gerar a alocação do modelo não-estagiado

O Algoritmo B.2.1 é um pseudocódigo para gerar o padrão de corte a partir de uma solução do Modelo (6.7) ou do Modelo (B.4). O índice j varia de 1 até $\bar{n} - 1$ na linha 2, dado que o padrão de corte final (i.e., subpadrão $j = 1$) está alocado na posição $(0, 0, 0)$. Esse algoritmo é uma adaptação direta do Algoritmo 5.5.2, que lida com o caso bidimensional.

B.3 Enumeração dos estágios guilhotinados ao Algoritmo *Bottom-up* de padrões 3-estágios

Na Seção 6.2.3, o Algoritmo *Bottom-up* foi adaptado para considerar padrões 3-estágios ao enumerar todos os possíveis tipos de padrões d -estágios, com $d \leq 3$, gerados por construções horizontais, em profundidade e verticais. As Tabelas B.1, B.2 e B.3 são as tabelas de acesso (do inglês, *look-up tables*) consideradas na linha 5 do Algoritmo 6.2.1 ao combinar horizontalmente, em profundidade e verticalmente, respectivamente. Por exemplo, a Tabela B.1 mostra que ao combinar horizontalmente uma solução parcial do tipo xy e uma solução parcial do tipo yx , isso resulta em uma nova solução parcial do tipo xyx . As entradas vazias nessas tabelas indicam que tal construção geraria uma solução parcial com mais de 3 estágios, o que é proibido. Além disso, note que ao combinar horizontalmente uma solução parcial do tipo y e uma solução parcial do tipo z , isso resulta em uma nova solução parcial do tipo yzx ou do tipo zyx . Para esses casos particulares, de forma a garantir otimalidade do método, duas soluções parciais são geradas, i.e., uma solução parcial para cada tipo possível de padrão d -estágios, e ambos são inseridos na lista $F^{\{k\}}$.

Algoritmo B.2.1: Alocação de uma solução do modelo para padrões não-estagiados.

Data: Instância, valores das variáveis z_{jip} , x_{jo} , y_{jk} , L_j , W_j , H_j .

```

1  $\alpha_1 \leftarrow 0, \beta_1 \leftarrow 0, \gamma_1 \leftarrow 0;$ 
2 for  $j \in J, o \in O \mid x_{jo} = 1$  do
3    $k_1 \leftarrow -1, i_1 \leftarrow -1, p_1 \leftarrow -1, l_1 \leftarrow -1, w_1 \leftarrow -1, h_1 \leftarrow -1;$ 
4    $k_2 \leftarrow -1, i_2 \leftarrow -1, p_2 \leftarrow -1;$ 
   // Linhas 5 até 14 identificam os componentes (subpadrões menores ou cópias
   // de tipos de itens) que constroem o subpadrao  $j$ 
5   for  $k \in J, j < k \mid y_{jk} = 1$  do
6     if  $l_1 = -1$  then
7        $k_1 \leftarrow k, l_1 \leftarrow L_k, w_1 \leftarrow W_k, h_1 \leftarrow H_k;$ 
8     else
9        $k_2 \leftarrow k;$ 
10  for  $i \in I, p \in Q_i \mid z_{jip} = 1$  do
11    if  $l_1 = -1$  then
12       $i_1 \leftarrow i, p_1 \leftarrow p, l_1 \leftarrow l_i, w_1 \leftarrow w_i, h_1 \leftarrow h_i;$ 
13    else
14       $i_2 \leftarrow i, p_2 \leftarrow p;$ 
   // Linhas 15 até 20 definem as posições dos subpadrões  $k_1$  e  $k_2$ , caso existam
15  if  $k_1 > -1$  then
16     $\alpha_{k_1} \leftarrow \alpha_j, \beta_{k_1} \leftarrow \beta_j, \gamma_{k_1} \leftarrow \gamma_j;$ 
17  if  $k_2 > -1$  then
18    if  $o = h$  then  $\alpha_{k_2} \leftarrow \alpha_j + l_1, \beta_{k_2} \leftarrow \beta_j, \gamma_{k_2} \leftarrow \gamma_j;$ 
19    if  $o = d$  then  $\alpha_{k_2} \leftarrow \alpha_j, \beta_{k_2} \leftarrow \beta_j + w_1, \gamma_{k_2} \leftarrow \gamma_j;$ 
20    if  $o = v$  then  $\alpha_{k_2} \leftarrow \alpha_j, \beta_{k_2} \leftarrow \beta_j, \gamma_{k_2} \leftarrow \gamma_j + h_1;$ 
   // Linhas 21 até 26 definem as posições dos tipos de itens  $i_1$  e  $i_2$ , caso
   // existam
21  if  $i_1 > -1$  then
22     $x_{ji_1p_1} \leftarrow \alpha_j, y_{ji_1p_1} \leftarrow \beta_j, z_{ji_1p_1} \leftarrow \gamma_j;$ 
23  if  $i_2 > -1$  then
24    if  $o = h$  then  $x_{ji_2p_2} \leftarrow \alpha_j + l_1, y_{ji_2p_2} \leftarrow \beta_j, z_{ji_2p_2} \leftarrow \gamma_j;$ 
25    if  $o = d$  then  $x_{ji_2p_2} \leftarrow \alpha_j, y_{ji_2p_2} \leftarrow \beta_j + w_1, z_{ji_2p_2} \leftarrow \gamma_j;$ 
26    if  $o = v$  then  $x_{ji_2p_2} \leftarrow \alpha_j, y_{ji_2p_2} \leftarrow \beta_j, z_{ji_2p_2} \leftarrow \gamma_j + h_1;$ 
27 for  $j \in J, i \in I, p \in Q_i \mid z_{jip} = 1$  do
28   O bloco  $j$  contém a  $p$ -ésima cópia de tipo de item  $i$  alocada em  $(x_{jip}, y_{jip}, z_{jip});$ 

```

Result: Posições $(x_{jip}, y_{jip}, z_{jip})$ dos itens selecionados.
