



Ministério da
Ciência e Tecnologia



INPE-16549-TDI/1561

**HEURÍSTICAS BASEADAS EM GERAÇÃO
SEQUENCIAL DE PADRÕES PARA O PROBLEMA DE
CORTE DE ESTOQUE UNIDIMENSIONAL COM
NÚMERO REDUZIDO DE PADRÕES**

Gonçalo Renildo Lima Cerqueira

Tese de Doutorado do Curso de Pós-Graduação em Computação Aplicada,
orientada pelo Dr. Horácio Hideki Yanasse, aprovada em 04 de agosto de 2009.

Registro do documento original:

<<http://urlib.net/sid.inpe.br/mtc-m18@80/2009/07.08.22.22>>

INPE
São José dos Campos
2009

PUBLICADO POR:

Instituto Nacional de Pesquisas Espaciais - INPE

Gabinete do Diretor (GB)

Serviço de Informação e Documentação (SID)

Caixa Postal 515 - CEP 12.245-970

São José dos Campos - SP - Brasil

Tel.:(012) 3945-6911/6923

Fax: (012) 3945-6919

E-mail: pubtc@sid.inpe.br

CONSELHO DE EDITORAÇÃO:**Presidente:**

Dr. Gerald Jean Francis Banon - Coordenação Observação da Terra (OBT)

Membros:

Dr^a Maria do Carmo de Andrade Nono - Conselho de Pós-Graduação

Dr. Haroldo Fraga de Campos Velho - Centro de Tecnologias Especiais (CTE)

Dr^a Inez Staciarini Batista - Coordenação Ciências Espaciais e Atmosféricas (CEA)

Marciana Leite Ribeiro - Serviço de Informação e Documentação (SID)

Dr. Ralf Gielow - Centro de Previsão de Tempo e Estudos Climáticos (CPT)

Dr. Wilson Yamaguti - Coordenação Engenharia e Tecnologia Espacial (ETE)

BIBLIOTECA DIGITAL:

Dr. Gerald Jean Francis Banon - Coordenação de Observação da Terra (OBT)

Marciana Leite Ribeiro - Serviço de Informação e Documentação (SID)

Jefferson Andrade Ancelmo - Serviço de Informação e Documentação (SID)

Simone A. Del-Ducca Barbedo - Serviço de Informação e Documentação (SID)

REVISÃO E NORMALIZAÇÃO DOCUMENTÁRIA:

Marciana Leite Ribeiro - Serviço de Informação e Documentação (SID)

Marilúcia Santos Melo Cid - Serviço de Informação e Documentação (SID)

Yolanda Ribeiro da Silva Souza - Serviço de Informação e Documentação (SID)

EDITORAÇÃO ELETRÔNICA:

Viveca Sant´Ana Lemos - Serviço de Informação e Documentação (SID)



Ministério da
Ciência e Tecnologia



INPE-16549-TDI/1561

**HEURÍSTICAS BASEADAS EM GERAÇÃO
SEQUENCIAL DE PADRÕES PARA O PROBLEMA DE
CORTE DE ESTOQUE UNIDIMENSIONAL COM
NÚMERO REDUZIDO DE PADRÕES**

Gonçalo Renildo Lima Cerqueira

Tese de Doutorado do Curso de Pós-Graduação em Computação Aplicada,
orientada pelo Dr. Horácio Hideki Yanasse, aprovada em 04 de agosto de 2009.

Registro do documento original:

<<http://urlib.net/sid.inpe.br/mtc-m18@80/2009/07.08.22.22>>

INPE
São José dos Campos
2009

Dados Internacionais de Catalogação na Publicação (CIP)

Cerqueira, Gonçalo Renildo Lima.
C335h Heurísticas baseadas em geração sequencial de padrões para o problema de corte de estoque unidimensional com número reduzido de padrões / Gonçalo Renildo Lima Cerqueira. – São José dos Campos : INPE, 2009.
140p. ; (INPE-16549-TDI/1561)

Tese (Doutorado em Computação Aplicada) – Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2009.

Orientador : Dr. Horácio Hideki Yanasse.

1. Redução de padrões. 2. Padrões de corte. 3. Heurística.
4. Programação linear. 5. Problema de corte. I. Título.

CDU 519.6

Copyright © 2009 do MCT/INPE. Nenhuma parte desta publicação pode ser reproduzida, armazenada em um sistema de recuperação, ou transmitida sob qualquer forma ou por qualquer meio, eletrônico, mecânico, fotográfico, reprográfico, de microfilmagem ou outros, sem a permissão escrita da Editora, com exceção de qualquer material fornecido especificamente no propósito de ser entrado e executado num sistema computacional, para o uso exclusivo do leitor da obra.

Copyright © 2009 by MCT/INPE. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, microfilming or otherwise, without written permission from the Publisher, with the exception of any material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use of the reader of the work.

**Aprovado (a) pela Banca Examinadora
em cumprimento ao requisito exigido para
obtenção do Título de Doutor(a) em
Computação Aplicada**

Dr. Solon Venâncio de Carvalho



Presidente / INPE / SJCampos - SP

Dr. Horacio Hideki Yanasse



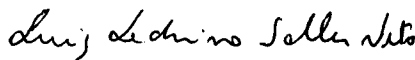
Orientador(a) / INPE / SJCampos - SP

Dr. José Demisio Simões da Silva



Membro da Banca / INPE / SJCampos - SP

Dr. Luiz Leduíno de Salles Neto



Convidado(a) / UNIFESP / São José dos Campos - SP

Dr. Reinaldo Morabito Neto



Convidado(a) / UFSCAR / São Carlos - SP

Aluno (a): Gonçalo Renildo Lima Cerqueira

São José dos Campos, 04 de agosto de 2009

*“Para onde quer que se dilate o nosso olhar, em parte alguma vemos
contradição entre Ciências Naturais e Religião; antes encontramos
plena convergência nos pontos decisivos. Ciências Naturais e
Religião não se excluem mutuamente, como hoje em dia muitos
pensam e receiam, mas completam-se e apelam uma para a outra.
Para o crente DEUS está no começo; para o físico, DEUS está no
ponto de chegada de toda a sua reflexão”.*

MAX PLANCK (1858-1947)

“Físico alemão, criador da Teoria do Quantum ”
Prêmio Nobel de Física 1918

A meus pais Jaime e Regina, e a minha avó Rosa

IN MEMORIAM
Eloísa e Daniel
Almir Olimpio Alves

AGRADECIMENTOS

A Deus, pelos dons da ciência e sabedoria e pela sua Divina presença em minha vida.

A meus pais Jaime e Regina que não pouparam esforço e dedicação na minha educação e dos meus irmãos: Fernando, Jeronice, Zé Mário, Juçara, Cris, Rene, Sueli e Nildo, principalmente pelos valores éticos e morais que nos passaram e pela família que juntos formamos.

A minha namorada Grazyelle, pelo amor, carinho, amizade, apoio e incentivo.

Ao meu orientador Horacio Hideki Yanasse, pela competência, paciência, dedicação e seriedade com as quais sempre conduziu a orientação deste trabalho, a quem manifesto os meus mais sinceros agradecimentos.

Aos meus amigos do INPE: Elias Canhadas, Heitor Moraes, Francisco Correa, Antônio Chaves, Geraldo Regis Mauri, Eduardo Luz, Guilherme Nadal, Renata Sampaio, Daniel Lamosa, Denilson Paulo e Jorge Formiga pela amizade e companherismo ao longo destes anos.

Aos professores, funcionários da biblioteca e da CAP-INPE, em especial aos professores José Demísio Simões Silva, Solon Venâncio de Carvalho e Reinaldo Roberto Rosa pela valiosa contribuição à minha formação.

Aos meus colegas de trabalho do DCE-UESB, pela confiança depositada durante meu período de afastamento para cursar o doutorado, em especial aos professores Jorge Anderson, Silvânio Bezerra, Ivanor Nunes, Carlos Takiya, Luiz Darcy e Benedito Acioly.

À Maria José Pinto Lamosa, Daniel Massaru e Luiz Leduino que gentilmente cederam alguns códigos utilizados neste trabalho.

A meus sobrinhos Jadson, Ítalo, Carol, Fernanda e Eduardo, meus afilhados: Felipe I, Felipe II, Brenda e César e meus amigos Ernesto e Solange pela agradável companhia e inestimável respeito que sempre nos acompanharam.

Aos meus amigos da UEMS: Haroldo Benatti, Hildete Magalhães Rocha e Marcos André e da UFPE: Cláudio Vidal, Hildeberto Cabral e Paulo Renato pelo valioso

período que juntos passamos .

À CAPES, FAPESP, CNPq e ao Governo do Estado da Bahia pelo apoio financeiro.

RESUMO

Neste trabalho, foca-se o problema de corte de estoque unidimensional em que são considerados dois objetivos: a minimização do desperdício e a redução do número de padrões na solução. Quatro novas heurísticas para resolver este problema são propostas. As três primeiras heurísticas possuem três fases: na primeira fase um conjunto inicial de padrões é gerado, na segunda fase um problema residual é resolvido, e, na terceira fase uma técnica de redução de padrões é aplicada à solução do problema. A metodologia aplicada nas fases 2 e 3 é a mesma para estas 3 heurísticas consideradas, que diferem apenas na fase 1 quanto à metodologia adotada na obtenção dos padrões. A quarta heurística tem apenas duas fases. Na primeira fase utiliza-se uma variante de uma heurística recentemente proposta na literatura em que várias soluções são geradas e a melhor é mantida. A segunda fase desta heurística 4 é idêntica à fase 3 das 3 heurísticas anteriores. Na heurística 1 os padrões são gerados levando-se em consideração a demanda média dos itens remanescentes, valorizando os itens maiores. Na heurística 2, os padrões são gerados a partir de uma divisão dos itens em dois grupos disjuntos de acordo com suas demandas. A heurística 3 é uma variante da heurística 2, em que modificamos a maneira de dividir os itens em grupos, que não necessariamente são disjuntos e a quantidade destes grupos é definida de acordo com a demanda residual dos itens a cada iteração. Na heurística 4 são gerados padrões que podem ser cortados com diferentes frequências pré-estabelecidas de acordo com as demandas dos itens existentes no problema. Propomos também uma variação do procedimento de redução de padrões da literatura conhecido como KOMBI, a fim de melhorar seu desempenho, possibilitando um maior número e combinação de padrões, contribuindo assim, para reduzir mais o número de padrões na solução do problema, o qual denominamos de KOMBI MODIFICADO ou KOMBI-M. São realizados testes computacionais com instâncias da literatura e observa-se que as heurísticas propostas apresentam um bom desempenho em termos de qualidade de solução quando comparadas com outras soluções geradas por métodos recentes da literatura, com média de tempo computacional aceitável.

HEURISTICS BASED ON SEQUENTIAL PATTERN GENERATION FOR THE ONE-DIMENSIONAL CUTTING STOCK PROBLEM WITH A REDUCED NUMBER OF PATTERNS

ABSTRACT

In this work, the one-dimensional cutting stock problem with the objectives of minimizing waste and reducing the number of patterns in the solution is considered. Four new heuristics to solve this problem are proposed. The first three heuristics have three phases: in the first phase an initial set of patterns is generated, in the second phase a residual problem is solved and in the third phase a pattern reduction technique is applied to the solution of the problem. The methodology applied in the second and third phases is the same for all these three heuristics. The heuristics differ only in phase 1 regarding the methodology used to generate the patterns. The fourth heuristic has only two phases. In the first phase, a variation of a heuristic procedure recently proposed in the literature is used, where several solutions are generated and the best is chosen. The second phase of heuristic 4 is identical to phase 3 of the three previous heuristics. In the first heuristic the patterns are generated taking into consideration the average demand of the remaining items, valuing the larger items. In the second heuristic, the patterns are generated dividing the items in two disjoint groups according to their demands. The third heuristic is a variant of the second heuristic, modifying the way the items are divided in groups, that are not necessarily disjoint sets and the quantity of items in these groups is determined according to the residual demand of the items at each iteration. In the fourth heuristic, patterns that can be cut with different pre-established frequencies, set according to the demands of the existing items, are generated. We also propose a variant of the procedure in of the literature known as KOMBI, that we denominated MODIFIED KOMBI or KOMBI-M. We performed computational tests with instances of the literature and we observed that the proposed heuristics present a good performance in terms of quality of solutions when compared with other solutions generated by recent methods of the literature, with an acceptable computational time average.

SUMÁRIO

Pág.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

1	INTRODUÇÃO	25
2	O PROBLEMA DE CORTE DE ESTOQUES UNIDIMENSIONAL (PCEU)	31
2.1	Geração de Padrões - Problema da Mochila	31
2.1.1	Programação Dinâmica	33
2.1.2	Método da Enumeração Implícita	34
2.2	Heurísticas Simples para Resolução do PCEU	35
2.2.1	Heurística FFD	36
2.2.2	Heurística Gulosa	37
2.3	Heurísticas Baseadas em formulação matemática do problema	38
2.3.1	Solução Inicial para o Método Simplex	40
2.3.2	Geração de Colunas	41
2.3.3	Métodos de Arredondamento de Soluções	43
2.3.3.1	Aproximação Básica de Padrões	44
2.3.3.2	Aproximação por Problemas Residuais	46
2.3.3.3	Aproximação Composta	47
2.3.4	Outras Heurísticas Residuais	48
2.3.4.1	Heurística nova 1	49
2.3.4.2	Heurística nova 2	49
2.3.4.3	Heurística nova 3	50
2.4	Outras Abordagens para Resolução do Problema de Corte Unidimensional	50
2.4.1	Modelo de Fluxo de Arcos	50
2.4.2	Modelos Onecut	52
2.4.3	Modelo do Bin Packing Problem	56
2.5	Extensão dos Modelos para Diferentes Objetos	57

3	MINIMIZAÇÃO DE PADRÕES DE CORTE	63
3.1	Revisão Bibliográfica	63
4	NOVAS HEURÍSTICAS PARA O PROBLEMA DE REDUÇÃO DE PADRÕES DE CORTE	83
4.1	Heurística 1	83
4.1.1	Fase 1: Processo Inicial de Geração de Padrões	83
4.1.2	Fase 2: Resolução do Problema Residual	85
4.1.3	Fase 3: Aplicação de um método de redução de padrões	86
4.2	Heurística 2	91
4.3	Heurística 3	96
4.4	Heurística 4	101
4.4.1	Exemplo Comum às 4 heurísticas	103
5	IMPLEMENTAÇÕES E RESULTADOS COMPUTACIONAIS	107
5.1	Resultados dos Testes Computacionais	109
6	KOMBI MODIFICADO - KOMBI-M	115
6.1	Experimentos Computacionais com o KOMBI-M	118
6.2	Resultados dos Testes Computacionais com KOMBI-M	121
7	CONCLUSÕES	133
	REFERÊNCIAS BIBLIOGRÁFICAS	135

LISTA DE FIGURAS

	<u>Pág.</u>
1.1 Problema de corte de estoque: (a) objetos em estoque, (b) itens deman- dados, (c) padrões de corte	26
1.2 corte bidimensional: (a) placa retangular, (b) itens, (c) padrão de corte .	27
1.3 corte tridimensional: (a) objeto (contêiner), (b) itens (embalagens), (c) empacotamento	27
2.1 Árvore de decisões	33
2.2 Grafo e um caminho (padrão de corte).	51
2.3 Exemplo de padrões equivalentes	52
2.4 (a) Diferentes objetos em estoque; (b) itens; (c) onecut (simetria).	53
2.5 Exemplo de programação linear para modelos onecut	55
2.6 Grafo para diferentes objetos em estoque.	59
2.7 Modelo de PL para diferentes objetos em estoque.	59
3.1 Algoritmo SHP	66

LISTA DE TABELAS

	<u>Pág.</u>
1.1 Tipologia de Alguns Problemas, reproduzido de Dyckhoff (1990).	29
4.1 Exemplo 1: dados de um problema de corte	89
4.2 Solução Inteira do Problema de Corte	90
4.3 Solução da Heurística	90
4.4 Solução da Heurística Após Aplicar Método KOMBI	91
4.5 Exemplo 2: dados do problema de corte	94
4.6 Solução do Problema de Corte	94
4.7 Dados do problema residual obtido do exemplo 2	95
4.8 Solução da Heurística	96
4.9 Solução da Heurística pós KOMBI	96
4.10 Exemplo 3: dados do problema de corte	99
4.11 Exemplo 3: Solução do Problema de Corte	99
4.12 Exemplo 3: dados do problema residual	100
4.13 Solução da Heurística	101
4.14 Solução da Heurística pós KOMBI	101
4.15 Exemplo 4: dados do problema de corte	102
4.16 Exemplo 4: Solução do Problema de Corte	103
4.17 Exemplo 4: Solução da Heurística	103
4.18 Solução do exemplo 1 com a Heurística 2	104
4.19 Solução do exemplo 1 com a Heurística 3	104
4.20 Solução do exemplo 1 com a Heurística 3, após KOMBI	105
4.21 Solução do exemplo 1 com a Heurística 4	105
4.22 Solução do exemplo 1 com a Heurística 4 após KOMBI	105
5.1 Problemas Testes Utilizados	108
5.2 Desempenho da heurística 1	110
5.3 Desempenho da heurística 2	111
5.4 Desempenho da heurística 3	112
5.5 Desempenho da heurística 4	113
5.6 Testes Computacionais Realizados com as Heurísticas (KOMBI)	114
6.1 Exemplo 5: dados do problema de corte	115
6.2 Solução obtida com a Heurística 2	115
6.3 Solução obtida com KOMBI-M	116

6.4	Exemplo 2: solução da Heurística pós KOMBI-M	117
6.5	Exemplo 4: solução da Heurística pós KOMBI-M	117
6.6	Kombi Modificado	119
6.7	Redução de Padrões em Porcentagem	120
6.8	Média do Tempo Computacional (s)	121
6.9	Desempenho da heurística 1 com KOMBI-M	122
6.10	Desempenho da heurística 2 com KOMBI-M	123
6.11	Desempenho da heurística 3 com KOMBI-M	124
6.12	Desempenho da heurística 4 com KOMBI-M	125
6.13	Testes Computacionais sem H_4^*	126
6.14	Testes Computacionais (KOMBI-M)	128
6.15	Tempo Médio das 4 Heurísticas	129
6.16	Testes Computacionais Adicionais	131

LISTA DE ABREVIATURAS E SIGLAS

PCE	–	Problema de Corte de Estoque
PCEU	–	Problema de Corte de Estoque Unidimensional
PMR	–	Problema da Mochila Restrito
PMI	–	Problema da Mochila Irrestrito
PMP	–	Problema de Minimização de Padrões
FFD	–	First Fit Decreasing
MIRUP	–	Modified Integer Round-Up Property
IRUP	–	Integer Round-Up Property
BPP	–	Bin Packing Problem
SHP	–	Sequential Heuristic Procedure
NO	–	Número de Objetos
ILS	–	Iterated Local Search
SHPC	–	Sequential Heuristic Procedure C

1 INTRODUÇÃO

Na tentativa de melhor utilizar os recursos disponíveis, muitas empresas de papel, de móveis, de vidro, metalúrgica, plástica, têxtil etc, que lidam no dia a dia com a tarefa de cortar peças menores a partir de peças maiores têm demonstrado um crescente interesse em técnicas que visam a otimização do processo de corte.

Problemas de corte consistem em cortar diferentes tipos de itens a partir de peças maiores em estoque (objetos), buscando otimizar alguma função objetivo, que pode ser, por exemplo, a minimização dos custos ou da quantidade de objetos cortados. É também comum na literatura considerar-se vários objetivos para o problema, como em [Wascher \(1990\)](#), [Sweeney e Haessler \(1990\)](#), [Moretti e Neto \(2008\)](#) e [Lopes e Araujo \(2009\)](#).

Determinar a melhor maneira de alocar os itens menores dentro dos objetos maiores, dá origem aos conhecidos problemas de empacotamento. É comumente observado no setor industrial em operações logísticas como a alocação de produtos para transporte em caminhões baús de volumes variados, vagões, contêineres de navio e outros espaços menores, como caixas e recipientes de isopor, muito usados pela indústria farmacêutica.

Planejar tais operações é de fundamental importância nestes casos, a fim de melhor aproveitar os espaços disponíveis nestes recipientes. Segundo [Arenales et al. \(36.,2004\)](#) o problema de corte pode ser pensado como um problema de empacotamento e (vice-versa), pois a parte do material que será cortado para a produção de um item, pode ser identificada como o espaço ocupado por este, e por esta razão, tais problemas são referidos como Problemas de Corte e Empacotamento (PCE) e são paralelamente estudados.

A variedade no tamanho dos itens envolvidos e as suas demandas, permitem diferentes maneiras para cortar o objeto em estoque. Esta maneira de cortar o objeto para produzir os diferentes itens é o que chamamos de *padrão de corte*. Embora a escolha dos melhores padrões de corte contribuam para um melhor planejamento de produção numa empresa, observa-se ainda, principalmente nas pequenas e médias empresas, que a escolha destes padrões é feita com base na experiência de algum funcionário ou decisor, o que pode levar a produzir mais sobras de material que o inevitável, gerando um custo maior que o necessário na produção. Um exemplo de

problema de corte de estoque com alguns padrões possíveis é mostrado na [Figura 1.1](#). A área hachurada representa o desperdício gerado.

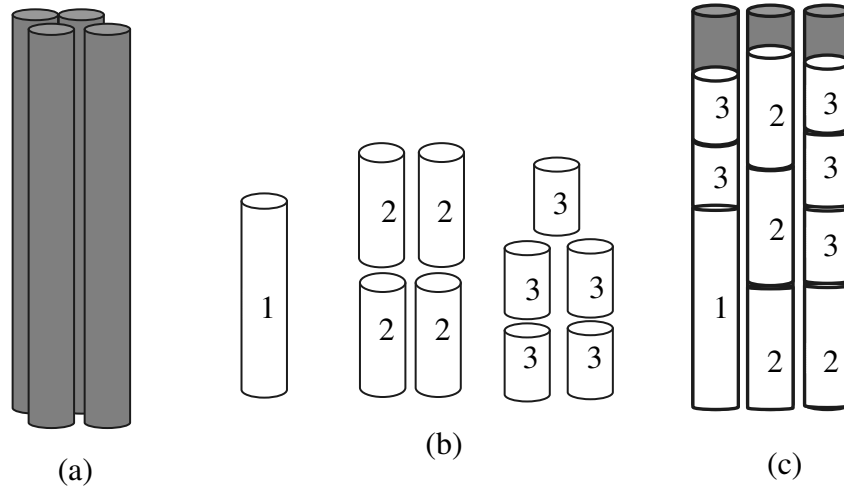


Figura 1.1 - Problema de corte de estoque: (a) objetos em estoque, (b) itens demandados, (c) padrões de corte

Uma classificação dos problemas de corte segundo suas dimensões, seleção de objetos e itens, e variedade de objetos e itens, foi sugerida por [Dyckhoff \(1990\)](#).

1-Dimensão

Quanto à dimensão os problemas de corte podem ser classificados em:

- Problemas Unidimensionais: quando apenas uma das dimensões é relevante para o processo de corte como mostra a [Figura 1.1](#). É o que ocorre, por exemplo, no corte de barras de aço, bobinas de papel, tubos de plástico, etc.
- Problemas bidimensionais: quando duas dimensões são relevantes para o processo de corte. Por exemplo, no corte de chapas de aço para produzir peças menores, duas dimensões (comprimento e largura) são relevantes para o processo de geração de padrões, como pode ser observado da [Figura 1.2](#).

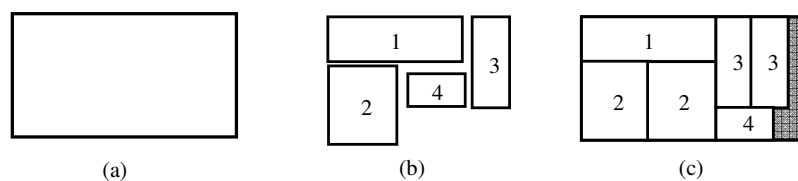


Figura 1.2 - corte bidimensional: (a) placa retangular, (b) itens, (c) padrão de corte

- Problemas tridimensionais: quando três dimensões são relevantes para o processo de corte (por exemplo, comprimento, largura e altura). Uma das aplicações mais importantes destes problemas ocorre no carregamento de contêineres de navios. Ver [Figura 1.3](#).

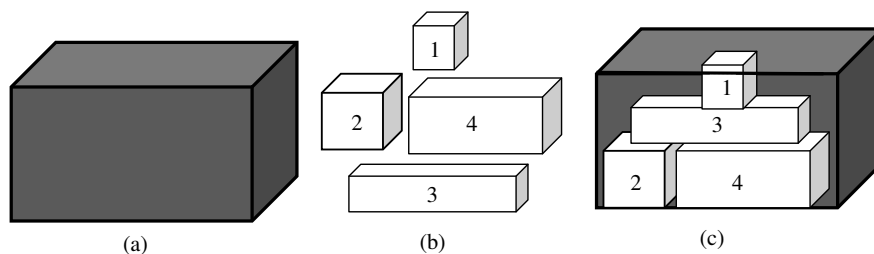


Figura 1.3 - corte tridimensional: (a) objeto (contêiner), (b) itens (embalagens), (c) empacotamento

- Problemas 1,5-dimensionais: quando duas dimensões são relevantes para a solução, sendo uma delas variável, por exemplo no corte de tecidos.
- Problemas 2,5 dimensionais: quando três dimensões são relevantes, com uma delas variável. Por exemplo, o problema de se efetuar o carregamento de unidades dentro de caixas abertas, isto é, as bases estão definidas, mas a altura é variável. Quando mais de três dimensões são relevantes para o problema, tem-se um problema N -dimensional com $N > 3$.

3-Variedade de Objetos e Itens

A seleção de objetos e itens, segundo [Dyckhoff \(1990\)](#), associa a produção dos itens menores ao objeto em estoque, segundo as seguintes possibilidades de atribuição:

- (B) utilização de todos os objetos e atribuição de alguns itens a todos eles. Exemplo: problema da mochila (clássico) e problema do carregamento de palete.

- (V) atribuição de todos os itens a alguns objetos. Exemplo: problemas clássicos de corte de estoque e do *bin-packing*.

Quanto ao tipo de objetos:

- (O) um único objeto;
- (I) objetos iguais quanto aos tamanhos e geometrias;
- (D) objetos com diferentes tamanhos e geometria;

Quanto à variedade de itens tem-se as seguintes classificações:

- (F) poucos itens (com diferentes tamanhos);
- (M) muitos itens (com diferentes tamanhos);
- (R) muitos itens (com poucos tamanhos diferentes)
- (C) itens congruentes ou de mesmo tamanho.

Combinando as diferentes características listadas, diferentes tipos de problemas de corte e empacotamento podem ser determinados, cuja tipologia representada pela quádrupla $\alpha/\beta/\gamma/\delta$ indica: dimensão/itens e objetos selecionados/variedade de objetos/variedade de itens. Para exemplificar, a quádrupla $1/V/I/R$, representa (1) um problema unidimensional, com (V) todos os itens cortados a partir de alguns (I) objetos de mesmo tamanho com (R) poucos itens diferentes, em quantidades relativamente grandes. Na [Tabela 1.1](#) são apresentados alguns problemas clássicos da literatura de corte e empacotamento e suas respectivas tipologias.

Devido à sua importância e por se tratarem geralmente de problemas pertencentes à classe NP-difíceis, muitos pesquisadores têm se ocupado na busca pelo desenvolvimento de novos métodos para resolver os problemas de corte, sendo a grande maioria métodos heurísticos, pois a resolução destes problemas por métodos exatos é muitas vezes inviável em situações práticas devido ao alto custo computacional.

Uma lista com mais de 400 trabalhos entre livros, artigos, dissertações e outros e suas bibliografias que tratam de problemas de corte e empacotamento é apresentada em [Sweeney e Paternoster \(1992\)](#).

Tabela 1.1 - Tipologia de Alguns Problemas, reproduzido de [Dyckhoff \(1990\)](#).

Problema	Tipologia
Problema da Mochila (clássico)	1/B/O
Problema do Carregamento de Paletes	2/B/O/C
Problema da Mochila Multidimensional	/B/O
Problema do Bin Packing Dual	1/B/O/M
Problema do Carregamento de Veículos (clássico)	1/V/I/F
Problema do Carregamento de Contêiner (clássico)	3/V/I
Problema do Bin Packing (clássico)	1/V/I/M
Problema de Balanceamento de uma Linha de Montagem	1/V/I/M
Problema de Alocação de Memória	1/V/I/M
Problema de Câmbio Monetário	1/B/O/R

Devido à grande diversidade de problemas de corte e correlatos existentes na prática que tem sido tratados na literatura, [Wascher et al. \(2007\)](#) sugeriram uma nova tipologia que, segundo os autores, é mais abrangente e adequada que a anterior, cobrindo vários novos problemas que não eram adequadamente classificados com a tipologia de [Dyckhoff \(1990\)](#). Consideramos, no entanto, a tipologia de [Dyckhoff \(1990\)](#) mais simples e mais intuitiva, o que é um atrativo para a manutenção do seu uso.

Neste trabalho, abordamos o problema de corte de estoque unidimensional (1/V/I) tendo como objetivo reduzir o número de diferentes padrões na solução, com o compromisso de não aumentar muito o desperdício, ou seja, encontrar "boas" soluções para o problema com número reduzido de padrões. Desenvolvemos novas heurísticas que são competitivas ou melhores que outros métodos recentemente propostos na literatura. A motivação para a realização desta pesquisa justifica-se pelo interesse de empresas em reduzir os custos de produção. Em alguns ambientes de corte, cada novo padrão de corte utilizado requer a preparação do equipamento de corte, o que consome recursos como tempo e mão de obra. Por isso o interesse de algumas empresas em reduzir a quantidade de padrões distintos.

A apresentação deste trabalho está organizada conforme o seguinte: no [Capítulo 2](#) revisamos o problema de corte de estoque unidimensional apresentando diferentes abordagens de resolução. No [Capítulo 3](#) apresentamos uma revisão de trabalhos da literatura que tratam da minimização de padrões no problema de corte unidimensional. No [Capítulo 4](#) apresentamos as novas heurísticas desenvolvidas para resolução do

problema de corte unidimensional visando também a redução do número de padrões na solução. No [Capítulo 5](#) apresentamos os resultados dos testes computacionais realizados com as heurísticas propostas e detalhes das implementações. No [Capítulo 6](#) sugerimos uma modificação em um procedimento de redução de padrões da literatura e testamos esta modificação com testes computacionais adicionais. Finalizando, apresentamos no [Capítulo 7](#) as conclusões e perspectivas para trabalhos futuros.

2 O PROBLEMA DE CORTE DE ESTOQUES UNIDIMENSIONAL (PCEU)

O artigo de [Kantorovich \(1960\)](#), segundo [Respicio \(2003\)](#), parece ser o primeiro trabalho encontrado na literatura em que o problema de corte de estoque (PCE) é modelado. Ele relata que este artigo, originalmente escrito em russo em 1939, foi traduzido para o inglês apenas em 1960 e completa que o interesse de Kantorovich por este problema surgiu quando em 1938, uma indústria madeireira, fornecedora de pequenas peças para produção de fuzis, o consultou acerca da forma de como efetuar os cortes dos objetos minimizando os desperdícios. Outros detalhes sobre o trabalho de Kantorovich para o PCE podem ser encontrado em [Carvalho \(2002\)](#).

O PCE tem sido bastante estudado, por diversos pesquisadores da área de Pesquisa Operacional. Embora muitos dos algoritmos desenvolvidos até hoje para este problema conseguem encontrar uma solução de boa qualidade e em alguns casos, ótima, o tempo computacional envolvido pode ser longo ou mesmo inviável para aplicações reais. Muito se tem feito no intuito de desenvolver algoritmos mais rápidos, e que permitem obter geralmente, soluções de boa qualidade em relação às soluções ótimas.

Especificamente no caso unidimensional, o problema de corte de estoque consiste em cortar peças maiores de comprimento L (objetos), para produzir peças menores de comprimento l_i (itens) de modo a atender os pedidos d_i (demanda) para $i = 1, \dots, m$ visando otimizar uma certa função objetivo, que pode ser por exemplo, a minimização dos custos de produção. Uma solução para o problema de corte de estoque consiste em determinar um conjunto de padrões de corte e suas frequências, ou seja, a quantidade de vezes que cada um deles deve ser cortado para que as demandas dos itens seja atendida. A seguir, é apresentada a maneira de como esses padrões podem ser gerados.

2.1 Geração de Padrões - Problema da Mochila

Segundo [Marques e Arenales \(2002\)](#), o termo mochila, decorre da situação prática em que um sacoleiro necessita transportar alguns utensílios (itens) em sua sacola (mochila). Associado a cada item estão o seu peso l_i e o lucro na sua revenda v_i . O objetivo do sacoleiro é, maximizar o lucro total, respeitando a capacidade física da mochila. Problemas da mochila aparecem como aplicação no mercado de capital, na busca da maximização do lucro oriundos de investimentos no mercado

financeiro, na alocação e distribuição de produtos em objetos de pequeno ou grande porte (realizadas por diversos segmentos industriais), carregamento de contêineres, na geração de padrões de corte e outros problemas de otimização combinatória.

Para uma formulação matemática do problema, são considerados os seguintes parâmetros:

- m : número de tipos de itens;
- l_i comprimento do item tipo i , $i = 1, \dots, m$;
- v_i valor de utilidade do item i , $i = 1, \dots, m$;
- d_i é a quantidade máxima de itens tipo i que pode ser cortada, $i = 1, \dots, m$;
- y_i variável de decisão que representa a quantidade cortada do item i , $i = 1, \dots, m$.

O problema pode então ser formulado como segue:

$$\text{maximize } g(\mathbf{y}) = \sum_{i=1}^m v_i y_i \quad (2.1)$$

$$\text{sujeito a : } \sum_{i=1}^m l_i y_i \leq L \quad (2.2)$$

$$0 \leq y_i \leq d_i, \text{ e inteiros, } i = 1, \dots, m. \quad (2.3)$$

A função objetivo maximiza a utilidade dos itens na mochila. A restrição (2.2) limita a quantidade de itens na mochila ao seu tamanho. As restrições (2.3) limitam a quantidade de itens cortados. Este problema é conhecido na literatura como Problema da Mochila Restrito (PMR). Substituindo o conjunto de restrições em (2.3) por $y_i \geq 0$ e inteiro $i = 1, \dots, m$, tem-se o problema conhecido na literatura como o Problema da Mochila Irrestrito (PMI). O vetor $\mathbf{y} = (y_1, \dots, y_m)^T$ solução do problema da mochila, corresponde a um padrão para o Problema de Corte de Estoque (PCE).

Pode-se resolver o problema da mochila por meio de uma sequência de decisões, tendo como base a inclusão de um item por vez, dentre os m diferentes disponíveis. No início do processo, o espaço ocioso é L , pois nenhum item foi ainda adicionado à mochila. Tomada a decisão de incluir o item de comprimento l_i , $i = 1, \dots, m$, no primeiro estágio, restará um espaço de $L - l_i$ a ser preenchido. Para o segundo estágio, um item do tipo j é escolhido, restando o espaço $L - l_i - l_j$. Este processo, pode então ser repetido, enquanto houver espaço suficiente para a inclusão de algum item l_i , $i = 1, \dots, m$. Quando todas as possibilidades de inclusão de itens em cada

estágio tiverem sido testadas, o algoritmo termina.

Estas decisões possíveis podem ser melhor observadas por meio de uma árvore, em que cada arco indica a decisão a ser tomada, conforme mostra a [Figura 2.1](#).

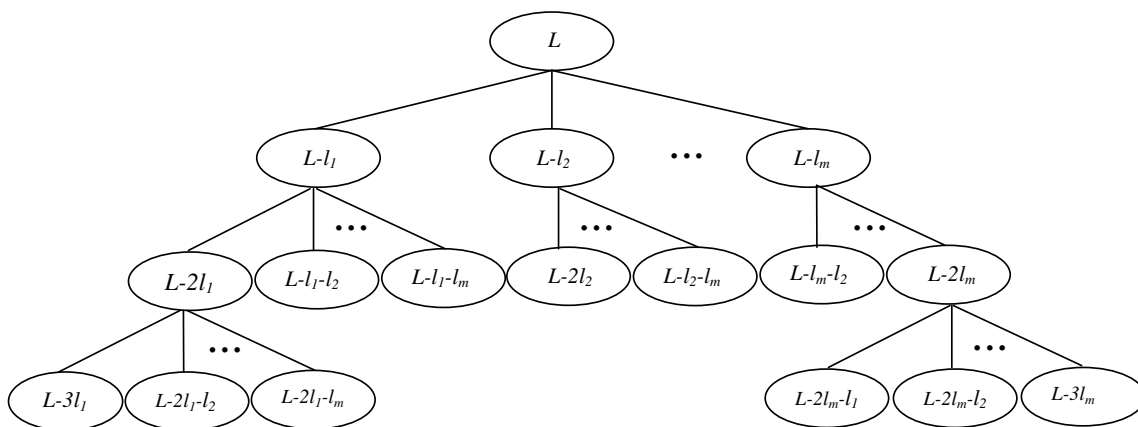


Figura 2.1 - Árvore de decisões

Embora permita determinar uma solução ótima para o problema, o método descrito anteriormente apresenta uma grande desvantagem: a necessidade de enumerar todas as possíveis soluções, o que torna inviável sua aplicação em muitas situações práticas, devido ao alto custo computacional. A seguir, dois métodos diferentes encontrados na literatura para resolução do *Problema da Mochila Restrito*, Programação Dinâmica e Enumeração Implícita, são apresentados.

2.1.1 Programação Dinâmica

Seja $X \leq L$ o espaço livre na mochila, denotamos por $G(X)$ o valor ótimo da mochila de tamanho X , ou seja, a solução ótima do seguinte problema de programação linear:

$$\begin{aligned}
 G(X) &= \max v_1 y_1 + v_2 y_2 + \dots + v_m y_m \\
 \text{sujeito a : } &l_1 y_1 + l_2 y_2 + \dots + l_m y_m \leq X \\
 &0 \leq y_i \leq d_i \text{ e inteiros, } i = 1, \dots, m.
 \end{aligned} \tag{2.4}$$

Um método de Programação Dinâmica para resolução do problema da mochila é apresentado em ([Pisinger \(1993\)](#) e [Martello e Toth \(1990\)](#)). Neste método a solução de um nó da árvore de decisões representando um espaço livre X é deter-

minada assim que os nós sucessores $X - l_1, X - l_2, \dots, X - l_m$ estiverem resolvidos. Assim, a determinação de $G(X)$ fica condicionada ao prévio conhecimento de $G(X - l_1), G(X - l_2), \dots, G(X - l_m)$.

Seja $l_0 = \min \{l_i, i = 1, \dots, m\}$ o comprimento do menor item do problema. Veja que se o espaço restante na mochila de comprimento L é menor ou igual a $l_0 - 1$, nenhum item do tipo i , $i = 1, \dots, m$ pode ser cortado, ou seja, $G(X) = 0$ para $X = 0, 1, \dots, l_0 - 1$. Portanto considerando os estados $X = l_0, X = l_0 + 1, \dots, L$, é possível determinar $G(X)$ recursivamente com tomada de decisão num dado estado X , através da relação: $G(X) = v_k + G(X - l_k) = \max\{v_i + G(X - l_i), i = 1, \dots, m\}$. (Mais detalhes veja [Pinto \(2004\)](#) ou [Gramani \(1997\)](#)).

2.1.2 Método da Enumeração Implícita

O método de enumeração implícita para resolução do problema da mochila, apresentado a seguir, sugerido por [Gilmore e Gomory \(1963\)](#), baseia-se em um esquema de busca em profundidade primeiro, permitindo através do uso de limitantes, que as piores soluções sejam descartadas, sem perder a solução ótima. O algoritmo completo é o seguinte:

INÍCIO

Passo 1: Ordene os itens mais valiosos por unidade de comprimento

Defina: $\pi_i = v_i/l_i$, $i = 1, \dots, m$, reordene as variáveis em ordem decrescente e sem perda de generalidade suponha que: $\pi_1 \geq \pi_2 \geq \dots \geq \pi_m$

Passo 2: Determine a solução inicial, utilizando busca em profundidade primeiro. Determine a solução inicial $\mathbf{y} = (y_1, \dots, y_m)$ tal que:

$$y_1 = \min \left\{ \left\lfloor \frac{L}{l_1} \right\rfloor, d_1 \right\}$$

$$y_2 = \min \left\{ \left\lfloor \frac{L - y_1 l_1}{l_2} \right\rfloor, d_2 \right\}$$

...

$$y_k = \min \left\{ \left\lfloor \frac{L - \sum_{i=1}^{k-1} l_i y_i}{l_k} \right\rfloor, d_k \right\}, k = 2, \dots, m.$$

Passo 3: Avalie a solução corrente e armazene a mais valiosa.

Determine : $g(\mathbf{y}) = \sum_{i=1}^m v_i y_i$.

Se $\underline{G} < g(\mathbf{y})$ então faça: $\underline{G} = g(\mathbf{y})$ (inicialmente $\underline{G} = 0$)
e guarde a solução correspondente $y^* = \mathbf{y}$.

Passo 4: Teste a otimalidade e calcule o limitante superior.

Determine k , o maior índice tal que $y_k \neq 0$.

Se $\mathbf{y} = 0$ então PARE, a melhor solução guardada em y^* é uma solução ótima.

Senão, calcule: (se $k = m$, considere $v_{m+1} = 0$ e $l_{m+1} = 1$.)

$$\bar{G} = v_1 y_1 + v_2 y_2 + \dots + v_k (y_k - 1) + \frac{v_{k+1}}{l_{k+1}} (L - l_1 y_1 - l_2 y_2 - \dots - l_k (y_k - 1))$$

Passo 5: Backtracking

5.1: Retorno longo

Se $\bar{G}(\mathbf{y}) \leq \underline{G}$, então faça: $y_k = 0$ e volte ao *Passo 4*.

5.2: Retorno ao nó precedente e nova busca em profundidade.

Se $\bar{G}(\mathbf{y}) > \underline{G}$ então faça $y_k \leftarrow y_k - 1$ e defina a nova solução \mathbf{y} :

$$y_j = \min \left\{ \left\lfloor \frac{L - \sum_{i=1}^{j-1} l_i y_i}{l_j} \right\rfloor, d_j \right\}, j = k + 1, \dots, m \text{ e volte ao } \textit{Passo 3}:$$

FIM

Outras abordagens diferentes para resolução do problema de mochila podem ser encontradas em [Yanasse e Soma \(1992\)](#), [Martello e Toth \(1990\)](#) e [Yanasse et al. \(2000\)](#).

2.2 Heurísticas Simples para Resolução do PCEU

Soluções para o problema de corte, podem ser obtidas por meio de simples heurísticas contrutivas conforme [Hinxman \(1980\)](#) ou [Poldi e Arenales \(2006\)](#). Alguns métodos simples sugeridos na literatura consistem em, a cada iteração, construir um bom padrão de corte, e utilizá-lo exaustivamente, o número máximo possível de vezes, sem que a demanda dos itens seja excedida. Após atualizar a demanda, o processo

é repetido até que toda ela tenha sido atendida. Ao final do processo uma solução inteira para o problema de corte é gerada. Duas heurísticas muito conhecidas que seguem esta idéia são a FFD (*First Fit Decreasing*) e Gulosa.

2.2.1 Heurística FFD

A Heurística FFD, consiste em colocar o maior item no padrão, o máximo de vezes possível, sem que haja excesso na demanda. Se o item selecionado não couber mais no padrão, o segundo maior item é selecionado e assim por diante.

Algoritmo

INÍCIO

P.1. Ordene os itens em ordem não crescente de tamanho. Suponha sem perda de generalidade que:

$$l_1 \geq l_2 \geq \dots \geq l_m$$

P.2. Seja dr_i a demanda residual do item $i \in I$.

$I = \{1, \dots, m\}$. {conjunto de índices dos itens}

Inicialmente: $dr_i = d_i, \forall i \in I$.

Faça $k = 1$ {Primeiro padrão de corte }

PARE=Falso {variável lógica que indica demanda não-nula}

Enquanto PARE=Falso)

P.3. Faça: $dem_i = dr_i, Sobra = L$ e $a_{ik} = 0, \forall i \in I$

Seja $i = 1$ {comece colocando o primeiro item no padrão }

Enquanto ($i \leq m$ e $Sobra \geq l_i$) faça:

$$a_{ik} = \min \left\{ \left\lfloor \frac{Sobra}{l_i} \right\rfloor, dr_i \right\}$$

(a_{ik} é a quantidade de itens tipo i no padrão k)

Faça: $Sobra = Sobra - (a_{ik}l_i)$

$$dr_i = dr_i - a_{ik}$$

$$i = i + 1$$

Fim do Enquanto

P.4. Determine a frequência do padrão k :

$$x_k = \left\{ \begin{array}{l} \min \left\lfloor \frac{dem_i}{a_{ik}} \right\rfloor, \\ \forall i \in I \text{ tal que } a_{ik} > 0 \end{array} \right\}$$

P.5. (Critério de Parada)

Se $dr_i = 0 \forall i \in I$ então

PARE=Verdade.

Senão Faça $k = k + 1$ e volte para **P.3**

Fim do Enquanto.

FIM

2.2.2 Heurística Gulosa

A Heurística Gulosa difere da Heurística FFD somente na maneira de como o padrão de corte é gerado. Ao invés de construí-lo com prioridade para os itens maiores, a geração do padrão consiste na resolução de um problema da mochila conforme (2.5). Tal padrão deve ser utilizado o maior número possível de vezes.

$$\begin{array}{l} \text{maximize } l_1\alpha_1 + l_2\alpha_2 + \cdots + l_m\alpha_m \\ \text{sujeito a : } \left\{ \begin{array}{l} l_1\alpha_1 + l_2\alpha_2 + \cdots + l_m\alpha_m \leq L \\ 0 \leq \alpha_i \leq b_i, i = 1, \dots, m \text{ e inteiros} \end{array} \right. \end{array} \quad (2.5)$$

Algoritmo

INÍCIO

P.2. Seja dr_i a demanda residual do item $i \in I$.

$I = \{1, \dots, m\}$. {conjunto de índices dos itens}

Inicialmente: $dr_i = d_i, \forall i \in I$.

Faça $k = 1$ {Primeiro padrão de corte }

PARE=Falso {variável lógica que indica demanda não-nula}

Enquanto (PARE=Falso)

P.3. Gere o padrão de corte mais valioso resolvendo o problema (2.5) com $b_i = dr_i, \forall i \in I$

P.4. Determine a frequência do padrão k :

$$x_k = \left\{ \begin{array}{l} \min \left[\frac{dr_i}{\alpha_i} \right], \\ \forall i \in I \text{ tal que } \alpha_i > 0 \end{array} \right\}$$

P.4. Atualize a demanda: $dr_i = dr_i - x_k \alpha_i, i = 1, \dots, m$

P.5. (Critério de Parada)

Se $dr_i = 0 \forall i \in I$ **então**

PARE=Verdade.

Senão Faça $k = k + 1$ e volte para **P.3**

Fim do Enquanto.

FIM

2.3 Heurísticas Baseadas em formulação matemática do problema

Dentre as referências mais relevantes na busca de uma solução para o problema de corte de estoque pode-se citar os trabalhos de [Gilmore e Gomory \(1961\)](#) e [Gilmore e Gomory \(1963\)](#). Eles sugeriram resolver o problema através de um problema linear e geração de colunas. Seja x_j uma variável de decisão que repre-

senda a quantidade de vezes que o padrão de corte j deve ser cortado (frequência), $\mathbf{a}_j = (a_{1j}, \dots, a_{ij}, \dots, a_{mj})^T$ um dos padrões de corte que compõe as colunas da matriz A :

$$\mathbf{a}_1 = \begin{pmatrix} a_{11} \\ a_{21} \\ a_{31} \\ \cdot \\ \cdot \\ \cdot \\ a_{m1} \end{pmatrix}; \mathbf{a}_2 = \begin{pmatrix} a_{12} \\ a_{22} \\ a_{32} \\ \cdot \\ \cdot \\ \cdot \\ a_{m2} \end{pmatrix}; \mathbf{a}_3 = \begin{pmatrix} a_{13} \\ a_{23} \\ a_{33} \\ \cdot \\ \cdot \\ \cdot \\ a_{m3} \end{pmatrix}; \dots; \mathbf{a}_n = \begin{pmatrix} a_{1n} \\ a_{2n} \\ a_{3n} \\ \cdot \\ \cdot \\ \cdot \\ a_{mn} \end{pmatrix}.$$

O elemento a_{ij} indica a quantidade de itens de comprimento l_i obtidos do padrão de corte \mathbf{a}_j para $i = 1, \dots, m$ e $j = 1, \dots, n$, em que n representa o número de possíveis padrões de corte e m o número de diferentes itens. O vetor $\mathbf{a}_j = (a_{1j}, \dots, a_{ij}, \dots, a_{mj})^T$ corresponde a uma padrão de corte viável, o que significa que todos os itens de comprimento l_i a compor o padrão, devem ser obtidos a partir do corte do objeto L em estoque, portanto devem satisfazer as seguintes restrições:

$$\begin{aligned} l_1 a_{1j} + l_2 a_{2j} + \dots + l_m a_{mj} &\leq L \\ a_{ij} &\geq 0 \text{ e inteiros, } i = 1, \dots, m \quad j = 1, \dots, n. \end{aligned} \tag{2.6}$$

Considerando como objetivo minimizar o número de objetos a serem cortados, o problema de corte de estoque pode ser formulado conforme o seguinte modelo matemático :

$$\text{minimize : } \sum_{j=1}^n x_j \tag{2.7}$$

$$\text{sujeito a : } \sum_{j=1}^n a_{ij} x_j \geq d_i \quad i = 1, \dots, m \tag{2.8}$$

$$x_j \geq 0; a_{ij} \geq 0 \text{ e inteiros, } j = 1, \dots, n \quad i = 1, \dots, m \tag{2.9}$$

e restrição (2.6).

A função objetivo (2.7) minimiza a quantidade de objetos cortados, as restrições (2.8) garantem que a soma total cortada de cada item seja suficiente para atender

pelo menos a sua demanda e (2.9) são restrições de não negatividade e integralidade do número de vezes que cada padrão de corte é cortado e da quantidade de itens de cada tipo fornecida pelo padrão.

Neste modelo, temos um problema não linear devido à restrição (2.8). Para evitar a não linearidade, pode-se enumerar todos os padrões de corte que satisfazem a restrição (2.6) e o problema passa a ser linear composto por (2.7), (2.8) e $x_j \geq 0$ e inteiro para $j = 1, \dots, n$. As dificuldades em se trabalhar com esta formulação estão no fato do número de padrões a explicitar (colunas \mathbf{a}_j , $j = 1, \dots, n$) aumentar muito à medida que aumenta o número de diferentes itens e suas demandas, e na dificuldade em resolver o problema devido às condições de integralidade impostas às variáveis de decisão.

Para solucionar o problema, Gilmore e Gomory (1961) propuseram relaxar as condições de integralidade das variáveis de decisão e a aplicação do método Simplex com Geração de Colunas. A cada iteração do simplex, gera-se uma coluna do modelo. Uma solução para o problema inteiro pode ser obtida arredondando-se a solução encontrada do problema relaxado.

A solução obtida com este procedimento tende a ser de boa qualidade em termos de erros percentuais em relação ao valor ótimo quando as demandas são elevadas. Note que ao se arredondar as soluções fracionárias, o impacto no valor da função objetivo, neste caso, é pequeno. Quando a demanda de cada item é pequena (denominado usualmente por *Bin Packing Problem*), os arredondamentos podem ter um impacto significativo no valor da função objetivo do problema.

2.3.1 Solução Inicial para o Método Simplex

Definição 2.1. *Um padrão de corte homogêneo é um padrão que produz um único tipo de item.*

Uma solução básica inicial para o problema de corte de estoque unidimensional, pode ser formada apenas por padrões homogêneos do tipo $\mathbf{a}_j = (0, \dots, \lfloor \frac{L}{l_i} \rfloor, \dots, 0)^T$, $j = 1, \dots, m$. A matriz básica, neste caso, é diagonal, em que o único elemento não

nulo de cada coluna indica a quantidade cortada do item i , $i = 1, \dots, m$:

$$\mathbf{B} = \begin{bmatrix} \lfloor \frac{L}{l_1} \rfloor & 0 & \cdots & 0 \\ 0 & \lfloor \frac{L}{l_2} \rfloor & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \lfloor \frac{L}{l_m} \rfloor \end{bmatrix}$$

2.3.2 Geração de Colunas

Seguindo o procedimento sugerido em [Gilmore e Gomory \(1961\)](#) a coluna \mathbf{a}_k a entrar na base, é aquela correspondente à variável x_k com o menor custo relativo, o que equivale a resolver o seguinte subproblema:

$$(c_k - \pi^T \mathbf{a}_k) = \min \{ (c_j - \pi^T \mathbf{a}_j), j = 1, \dots, n \}. \quad (2.10)$$

Se $(c_k - \pi^T \mathbf{a}_k) \geq 0$ então a solução básica corrente $\mathbf{x}_B = \mathbf{B}^{-1} \mathbf{d}$ é ótima, em que π é o vetor multiplicador simplex da iteração corrente, \mathbf{B} é a matriz básica $m \times m$, \mathbf{x}_B é o vetor das variáveis básicas, \mathbf{d} é o vetor de demandas.

Note que o modelo (2.7)-(2.9) que estamos considerando tem por objetivo minimizar a quantidade de objetos cortados, cujo vetor custo \mathbf{c} tem coordenadas $c_j = 1$. Precisamos determinar uma coluna não-básica $\mathbf{a}_j = (\alpha_1, \alpha_2, \dots, \alpha_m)^T$, correspondente a um padrão de corte viável, que seja candidata a entrar na base. Assim, como $\min \{ c_j - \pi^T \mathbf{a}_j \} = 1 - \max \{ \pi^T \mathbf{a}_j \}$ precisamos resolver o problema:

$$\begin{aligned} & \text{maximize} \sum_{i=1}^m \pi_i \alpha_i \\ & \text{sujeito a : } \sum_{i=1}^m l_i \alpha_i \leq L \\ & 0 \leq \alpha_i \leq d_i, \text{ e inteiros } i = 1, \dots, m. \end{aligned} \quad (2.11)$$

Algoritmo de [Gilmore e Gomory \(1961\),\(1963\)](#)

INÍCIO DO ALGORITMO

FASE 01:

1. Determine a matriz básica inicial \mathbf{B} cujas colunas são padrões homogêneos:

$$\mathbf{B} = \begin{bmatrix} \left[\frac{L}{l_1} \right] & 0 & \cdots & 0 \\ 0 & \left[\frac{L}{l_2} \right] & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \left[\frac{L}{l_m} \right] \end{bmatrix}$$

Faça: $PARE = Falso$ e $IT = 0$.

{ A variável lógica $PARE$ será $Falso$ até que a condição de otimalidade seja verificada e IT indica o número da iteração corrente }.

FASE 02:

Enquanto $PARE = Falso$, faça:

2. Determine a solução básica viável corrente: $\mathbf{x}_B = \mathbf{B}^{-1}\mathbf{d}$.

3. Determine o vetor multiplicador simplex: $\mathbf{B}^T \boldsymbol{\pi} = \mathbf{c}_B$.

4. Resolva o Problema da Mochila:

$$g(\mathbf{a}) = \text{Max } \pi_1 \alpha_1 + \pi_2 \alpha_2 + \cdots + \pi_m \alpha_m$$
$$\text{st.a. } \begin{cases} l_1 \alpha_1 + l_2 \alpha_2 + \cdots + l_m \alpha_m \leq L \\ 0 \leq \alpha_i \leq d_i, i = 1, \dots, m \text{ e inteiros} \end{cases}$$

Seja $\mathbf{a} = (\alpha_1, \alpha_2, \dots, \alpha_m)$ a solução ótima do Problema da Mochila

5. Teste de Otimalidade.

Se $1 - g(\mathbf{a}) \geq 0$

então $PARE = Verdade$ { fornece a atual solução que é ótima e que foi obtida em IT iterações }.

senão:

6. Determine a direção simplex: $\mathbf{y} = \mathbf{B}^{-1}\mathbf{a}$.

Se $\mathbf{y} < 0$ **então** o problema é ilimitado. $PARE = Verdade$. { A solução é

ilimitada}.

senão:

7. Determine o tamanho do passo:

Encontre l tal que:

$$\frac{x_l}{y_l} = \min \left\{ \frac{x_i}{y_i} \mid y_i > 0, i = 1, \dots, m \right\}$$

8. Atualização da base.

Substitua a l -ésima coluna da matriz \mathbf{B} por \mathbf{a} .

faça: $IT = IT + 1$.

Fim do Enquanto

FIM DO ALGORITMO.

Outros problemas que utilizam Geração de Colunas em sua resolução podem ser vistos em [Previero \(2001\)](#). Outras aplicações deste procedimento ao problema de corte podem ser vistos em [Carvalho \(2005\)](#), [Vanderbeck \(1999\)](#), [Carvalho \(1999\)](#), [Barnhart et al. \(1994\)](#) e [Vance \(1998\)](#).

2.3.3 Métodos de Arredondamento de Soluções

Na maioria das vezes, a solução relaxada do problema de corte, obtida pelo método simplex com geração de colunas, não é inteira. Para determinar uma solução inteira podem ser aplicadas técnicas de arredondamento. Quando as demandas dos itens são grandes, a solução obtida por arredondamento tende a ser de boa qualidade, pois o impacto no valor da função objetivo é relativamente pequeno. Quando a demanda de cada item é pequena, (é o que acontece no *Bin Packing Problem*) os arredondamentos podem ter um impacto significativo no valor da função objetivo do problema.

[Marcotte \(1985\)](#) conjecturou que o problema de corte unidimensional com objetivo de minimizar objetos tem a propriedade IRUP (*Integer Round-Up Property*), ou seja, o *gap* de integralidade, a diferença entre o valor ótimo da solução inteira e o valor ótimo da solução relaxada arredondada para o inteiro mais próximo superior é

menor ou igual a 1. Posteriormente a mesma autora [Marcotte \(1986\)](#) apresentou um exemplo em que a conjectura que ela propôs não se aplica. [Scheithauer e Terno \(1995\)](#) conjecturaram então que o problema de corte unidimensional tem a propriedade MIRUP (*Modified Integer Round-Up Property*), em que o *gap* de integralidade, a diferença entre o valor ótimo da solução inteira e o valor ótimo da solução relaxada arredondada para o inteiro superior é menor ou igual que 2. Alguns trabalhos da literatura apresentam instâncias do problema de corte unidimensional com valores do *gap* de integralidade de: 1,0333... e 1,0378... (cf [Scheithauer e Terno \(1995\)](#)).

Seja $\mathbf{x} = (x_1, x_2, \dots, x_n)$ a solução ótima do problema relaxado (2.7)-(2.9). Vamos admitir que pelo menos uma componente de \mathbf{x} é não-inteira, caso contrário, a solução inteira para o PCE já teria sido encontrada. Os procedimentos de arredondamento apresentados a seguir, encontram-se em [Wascher e Gau \(1996\)](#), que os classificaram em três diferentes tipos e serão assim referenciados: (A) Aproximação Básica de Padrões, (B) Aproximação por Problemas Residuais e (C) Aproximação Composta.

2.3.3.1 Aproximação Básica de Padrões

Os padrões básicos da solução ótima do problema relaxado (2.7)-(2.9), correspondem às variáveis básicas da solução, e são em número de m , que é a quantidade de diferentes itens envolvidos no problema. Somente padrões básicos serão considerados na geração da solução inteira, sendo portanto as variáveis não básicas fixadas em zero. Para este caso, [Wascher e Gau \(1996\)](#) propõem as seguintes técnicas de arredondamento:

2.3.3.1.1 Procedimento BRUSIM

Este procedimento consiste em arredondar as componentes não-inteiras de \mathbf{x} para o inteiro superior mais próximo, ou seja, $x_j^* = \lceil x_j \rceil$, $j \in J_B$, em que J_B é o conjunto de índices das variáveis básicas. Tal procedimento é extremamente rápido, resultando imediatamente numa solução viável para o PCE, visto que não viola as restrições de demanda (2.8). A desvantagem é que este procedimento geralmente resulta em soluções inteiras com valor da função objetivo maior do que o valor ótimo.

2.3.3.1.2 Procedimento BRURED

O arredondamento para cima das componentes não-inteiras de \mathbf{x} , pode frequentemente gerar excesso na produção de alguns itens. Pode-se evitar que isto ocorra,

reduzindo-se de uma em uma unidade as frequências de alguns padrões de corte para o menor valor inteiro possível, desde que as restrições de demanda (2.8) não sejam violadas. Assim que for identificada uma variável $x_k, (k \in J_B)$ para a qual é verificada a desigualdade:

$$a_{ik}(x_k - 1) + \sum_{\substack{j \in J_B \\ j \neq k}} a_{ij}x_j \geq d_j, i = 1, \dots, m$$

tal variável é atualizada fazendo $x_k = x_k - 1$.

2.3.3.1.3 Procedimento BRUSUC

Para este caso, [Stadtler \(1990\)](#) (cf. [Wascher e Gau \(1996\)](#)) sugere determinar uma solução para o problema inteiro (2.7)-(2.9) consistindo apenas de padrões básicos. Após fixar todas as variáveis inteiras de \mathbf{x} : $x_j^* = x_j, j \in J_B$ e x_j inteiro e arredondar para cima a variável x_k de \mathbf{x} com maior valor fracionário: $x_k^* = \lceil x_k \rceil$, um outro problema de programação linear é definido:

$$\begin{aligned} \text{minimize} : & \sum_{j \in J_B} \bar{x}_j \\ \text{sujeito a} : & \sum_{j=1}^n a_{ij} \bar{x}_j \geq d_i, \quad i = 1, \dots, m \\ & \bar{x}_j \geq 0, \quad j \in J_B \end{aligned} \tag{2.12}$$

Em que $\bar{x}_j = x_j^*$ se x_j^* inteiro, $j \in J_B$ e $\bar{x}_k = \lceil x_k \rceil$.

Note que após resolver o problema (2.12) a solução gerada não necessariamente é inteira, neste caso o processo é repetido até que todas as variáveis $x_j^*, j = 1, \dots, m$ tornem-se inteiras. Como pelo menos uma variável é fixada a cada iteração, o procedimento termina com todas as componentes da solução do problema (2.12) inteiras depois de no máximo m iterações. Devido ao número relativamente pequeno de variáveis (no máximo a quantidade de itens m), pode-se determinar diretamente uma solução inteira para o problema (2.12), resolvendo-o através de programação linear inteira.

2.3.3.2 Aproximação por Problemas Residuais

Este procedimento tem início também com a solução ótima do problema relaxado (2.7)-(2.9), entretanto todas as componentes não-inteiras de \mathbf{x} são arredondadas para baixo, o que corresponde ao vetor: $\bar{x} = (\lfloor x_1 \rfloor, \lfloor x_2 \rfloor, \dots, \lfloor x_n \rfloor)$ sendo a função objetivo correspondente dada por: $\sum_{j=1}^n \lfloor x_j \rfloor$. Note que \bar{x} não é sempre uma solução viável para o PCE, porque os arredondamentos resultantes provenientes das componentes não-inteiras de \mathbf{x} não necessariamente atendem a demanda de todos itens. Ao efetuarmos o corte de $\sum_{j=1}^n \lfloor x_j \rfloor$ objetos em estoque, serão produzidos somente $\sum_{j=1}^n a_{ij} \lfloor x_j \rfloor$ unidades do item $l_i, i = 1, \dots, m$. Desta forma, deve ser resolvido o seguinte problema residual a fim de completar a demanda remanescente de alguns itens:

$$\text{minimize : } \sum_{j=1}^n x_j \quad (2.13)$$

$$\text{sujeito a : } \sum_{j=1}^n a_{ij} x_j \geq dr_i \quad i = 1, \dots, m \quad (2.14)$$

$$x_j \geq 0; a_{ij} \geq 0 \text{ e inteiros } j = 1, \dots, n \quad i = 1, \dots, m \quad (2.15)$$

em que: $dr_i = \text{Max} \left(0, d_i - \sum_{j=1}^n a_{ij} \lfloor x_j \rfloor \right), i = 1, \dots, m$.

A solução final do problema original será então dada pela combinação das soluções do problema residual (2.13)-(2.15) com a solução obtida a partir do processo de arredondamento para baixo das componentes não inteiras da solução do problema relaxado, e a soma das funções objetivos correspondentes, fornece o total de objetos cortados para o problema (2.7)-(2.9).

Como as demandas dos itens no problema residual são bem menores que no problema original, costuma-se normalmente tratá-lo não como um PCE, mas como um *Bin Packing Problem*. A seguir são apresentados alguns métodos encontrados na literatura para resolver o problema residual.

2.3.3.2.1 Procedimento ROPT

Neste caso, o problema residual é resolvido de maneira exata. Note que devido à separação do problema original em dois subproblemas, a combinação das soluções do problema residual e do método de arredondamento, não necessariamente define

uma solução ótima do problema original. Além do mais, quando o número m de itens é grande, o problema residual tem a tendência de também ser grande (em termos do número de itens).

2.3.3.2.2 Procedimento RFFD

Ao invés de resolver o problema residual através de métodos exatos, aplicam-se métodos heurísticos, reduzindo consideravelmente o esforço computacional em solucioná-lo. Uma heurística proposta neste caso é a FFD (*First Fit Decreasing*).

2.3.3.2.3 Procedimento RSUC

Neste caso, o problema residual (2.13)-(2.15) é relaxado com respeito às restrições de integralidade das variáveis e o Método Simplex com Geração de Colunas é aplicado para resolvê-lo. Caso a solução gerada apresente componentes não-inteiras, estas são arredondadas para o maior inteiro inferior, o que dará origem a um outro problema residual, que deverá ser tratado da mesma forma descrita anteriormente. Este processo termina quando o arredondamento para baixo das componentes não inteiras da solução, resultar em todas as frequências nulas. Caso a demanda de algum item não tenha sido atendida, o problema residual correspondente é resolvido através de um algoritmo exato para o *Bin Packing Problem*.

2.3.3.3 Aproximação Composta

Este tipo de abordagem, combina as duas idéias apresentadas anteriormente de Aproximação Básica de Padrões e Aproximação por Problemas Residuais. Iniciando-se com o procedimento BRUSUC gera-se uma solução viável para o PCE. Observe que neste estágio, geralmente ocorre excesso na produção, o qual será reduzido iterativamente. Para isso, ao se identificar o padrão de corte que contém o maior número de itens com excesso na produção, reduz-se sua frequência em uma unidade. Este passo é repetido até que não haja mais excesso na produção. Como alguns itens podem ter suas demandas não completamente atendidas, um problema residual deve ser resolvido a fim de completá-las. A seguir são apresentados dois métodos de resolução para este problema, encontrados em [Wascher e Gau \(1996\)](#).

2.3.3.3.1 Procedimento CSTAOPT

Após aplicar os procedimentos iniciais descritos anteriormente, o problema residual

é resolvido através de um algoritmo exato.

2.3.3.3.2 Procedimento CSTAFD

Este procedimento é semelhante ao método CSTAOPT, só que ao invés de um algoritmo exato, o problema residual é resolvido através da heurística FFD.

2.3.4 Outras Heurísticas Residuais

Outras heurísticas de arredondamento, visando gerar uma solução inteira para o problema de corte de estoque unidimensional inteiro a partir da solução fracionária do problema (2.7)-(2.9) com atenção especial para problemas com baixa demanda foram propostas por Poldi e Arenales (2006).

As heurísticas residuais novas consistem basicamente em, a cada iteração, resolver um problema de corte de estoque relaxado e ordenar o vetor solução de forma específica. Três formas de ordenação foram consideradas. Para cada posição deste vetor, na ordem especificada, arredonda-se a frequência para o número inteiro acima do fracionário obtido e testa-se a factibilidade desta solução (no sentido de que excessos de itens não sejam gerados). Caso não seja factível (isto é, houve excesso de itens), a frequência é reduzida de uma unidade até que excessos sejam eliminados. Quando o último padrão de corte gerado for examinado, atualiza-se a demanda, resultando num problema residual, que será tratado da mesma forma. A estrutura geral do algoritmo para heurísticas do tipo residuais, reproduzido de Poldi e Arenales (2006), é a seguinte:

INÍCIO

- Passo 1:** {*Inicialização*}
- Sejam $k = 1$, $\mathbf{dr}^1 = \mathbf{d}$, os dados para o problema residual inicial.
- Passo 2:** {*Determinar uma solução ótima contínua*}
- Resolver o problema relaxado
- Seja \mathbf{x}^k a solução contínua obtida.
- Se $\mathbf{x}^k \in Z^n$, ou seja, é uma solução inteira, então PARE.
- Passo 3:** {*Determinar uma solução inteira aproximada*}
- Determine uma solução inteira para \mathbf{x}^k e denote-a por \mathbf{x}^{*k} .
- Se \mathbf{x}^{*k} for um vetor nulo, então vá para o passo final.
- Passo 4:** {*Atualizações*}
- Determine a nova demanda residual: $\mathbf{dr}^{k+1} = \mathbf{dr}^k - \mathbf{Ax}^{*k}$.

$k = k + 1$.

Repita o *Passo 2*.

Passo Final:

Resolva o problema residual final.

FIM

A seguir definimos como determinar a solução inteira aproximada \mathbf{x}^{*k} no **Passo 3** e como resolver o problema residual final no **Passo Final**. Poldi e Arenales (2006) dividiram o **Passo 3** em duas partes: O pré-processamento (ordenação dos padrões) e o arredondamento.

Passo 3.1: (pré-processamento)

Ordene o vetor solução $\mathbf{x}^k \in \mathfrak{R}^n$ obtido no **Passo 2** segundo um critério (a ser definido a seguir).

Passo 3.2: (arredondamento)

Suponha que N padrões de corte gerados na solução ótima \mathbf{x}^k no **Passo 2** tenham frequência positiva ($x_1^k > 0, \dots, x_N^k > 0$)

Para $i = 1, \dots, N$, faça:

$x_i^{*k} = \lceil x_i^k \rceil$ e $x_j^{*k} = 0$ para $j = i + 1, \dots, m$, ou seja, $\mathbf{x}^{*k} = (x_1^{*k}, \dots, x_i^{*k}, 0, \dots, 0)$.

Enquanto \mathbf{x}^{*k} for inactível (ou seja, há itens em excesso), faça: $x_i^{*k} = x_i^{*k} - 1$

Segundo Poldi e Arenales (2006), após $i = N$, todas as frequências foram "arredondadas"(algumas frequências foram arredondadas possivelmente para o inteiro superior, outras para o inteiro inferior ou valores menores, podendo-se até mesmo anular uma frequência maior que um), gerando-se uma solução aproximada inteira \mathbf{x}^{*k} .

A seguir são apresentados três critérios de ordenação do **Passo 3.1**, dando origem a três versões da Heurística proposta por Poldi e Arenales (2006).

2.3.4.1 Heurística nova 1

Na versão 1, os padrões são ordenados de forma não-decrescente da frequência de utilização, admitindo sem perda de generalidade que $x_1^k \geq x_2^k \geq \dots \geq x_N^k$.

2.3.4.2 Heurística nova 2

Na versão 2 os padrões de corte são ordenados conforme a perda que eles apresentam, de forma não-crescente. Sem perda de generalidade admitindo que $\alpha_1 \leq \alpha_2 \leq \dots \leq$

α_N .

2.3.4.3 Heurística nova 3

Na heurística nova 3 ordena-se os padrões de corte conforme a parte fracionária das frequências dos padrões de corte, em ordem não crescente. Seja $f_j = x_j - \lfloor x_j \rfloor$, $j = 1, \dots, N$. (f é a parte fracionária). Ordene o vetor solução. Sem perda de generalidade admita que $f_1 \geq f_2 \geq \dots \geq f_N$.

Um procedimento similar, em que o vetor solução é ordenado em ordem não-crescente da parte fracionária de suas componentes, foi utilizado em [Belov e Scheithauer \(2002b\)](#).

2.4 Outras Abordagens para Resolução do Problema de Corte Unidimensional

Em [Carvalho \(1999\)](#), o Problema de Corte Unidimensional é modelado como um problema de fluxo em arcos.

2.4.1 Modelo de Fluxo de Arcos

Para formular o problema de corte de estoque unidimensional como um problema de fluxo em arcos, [Carvalho \(1999\)](#) associa os padrões para o problema com caminhos num grafo G com vértices numerados de 0 a L .

Cada caminho é constituído por um conjunto de arcos que representam os itens fornecidos por cada padrão.

Seja $G = (V, A)$ um grafo acíclico direcionado (vértices inicial e final não coincidentes), sendo $V = 0, 1, 2, \dots, L$ um conjunto de vértices e $A = \{(i, j) : 0 \leq i < j \leq L\}$ um conjunto de arcos com $j - i = l_t$ para algum t com $t = 1, \dots, m$, indicando a existência de um arco (i, j) em G se existir um item de tamanho correspondente ao tamanho do arco dado. Para problemas com itens $l_i > 1$, $i = 1, \dots, m$, são consideradas perdas, arcos do grafo G de comprimento 1, que vão de um nó a outro imediatamente superior, e correspondem a porções não utilizadas do objeto (L).

Um exemplo de um grafo associado a objetos de comprimento $L = 5$ e itens de tamanhos 2 e 3 a serem cortados é mostrado na [Figura 2.2](#), reproduzido de [Carvalho \(1999\)](#). Observe que existem vários caminhos entre 0 e 5, que conduzem a diferentes

padrões de corte e que podem ser usados para satisfazer a demanda dos itens dados. Um deles é mostrado na [Figura 2.2](#), e representa o corte de dois itens de comprimento 2, com uma unidade de perda.

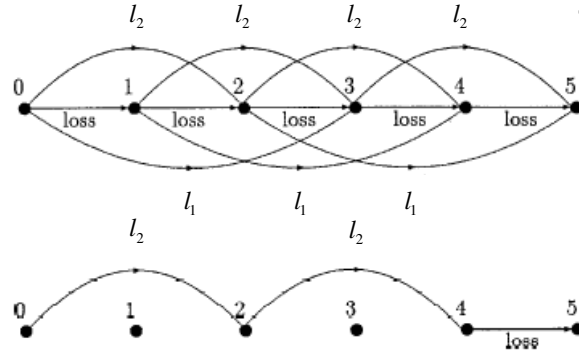


Figura 2.2 - Grafo e um caminho (padrão de corte).

Fonte: Reproduzida de [Carvalho \(1999\)](#).

Admitindo que $l_i \neq l_j \forall i \neq j, i, j = 1, \dots, m$, com as definições e observações anteriores, o problema de corte de estoque com minimização do número de objetos pode ser formulado como um problema de fluxo como segue:

$$\text{minimize : } z \tag{2.16}$$

$$\text{sujeito a : } + \sum_{(i,j) \in A} x_{ij} - \sum_{(j,k) \in A} x_{jk} = \begin{cases} -z, \text{ se } j = 0, \\ 0, \text{ se } j = 1, 2, \dots, L-1, \\ z, \text{ se } j = L \end{cases} \tag{2.17}$$

$$\sum_{(k,k+l_t) \in A} x_{k,k+l_t} \geq d_t, t = 1, 2, \dots, m \tag{2.18}$$

$$x_{ij} \geq 0 \text{ e inteiros, } \forall (i, j) \in A \tag{2.19}$$

em que:

- z : é um arco de realimentação (*feedback*);
- x_{ij} corresponde ao número de itens de tamanho $j - i$ obtidos de um objeto de tamanho L ;
- d_t é a demanda do item de tamanho $l_t, t = 1, \dots, m$.

A função objetivo (2.16) minimiza a quantidade de fluxo que é mandada da origem à fonte. Um caminho no grafo do nó 0 ao nó L corresponde a um padrão de corte.

Cada unidade de fluxo que passa por este caminho corresponde a utilização de um objeto. Veja que por um caminho podem passar zero, uma ou mais unidades de fluxo, indicando que um mesmo padrão de corte foi utilizado zero, uma ou mais vezes. (2.17) são restrições de conservação de fluxo. As restrições (2.18) garantem que a demanda de cada item seja satisfeita enquanto que (2.19) são restrições de não negatividade e integralidade, da quantidade de itens de um dado tamanho obtida do corte de algum objeto em estoque.

A dificuldade encontrada na busca por uma solução para o problema de corte utilizando modelo de fluxo de arcos, é a existência de muitas soluções alternativas equivalentes. Alguns procedimentos de eliminação de padrões simétricos ou equivalentes são aplicados, como por exemplo, a opção por padrões (caminhos) em que os itens possam ser cortados a partir de uma ordem não crescente de tamanho e a perda se encontre ao final do objeto, ou seja, que os arcos de perda se acumulem no final de cada caminho. A Figura 2.3 mostra uma aplicação destes procedimentos no corte de itens de comprimento $l_1 = 2\text{cm}$ e $l_2 = 3\text{cm}$ a partir de objetos em estoque de comprimento $L = 7\text{cm}$.

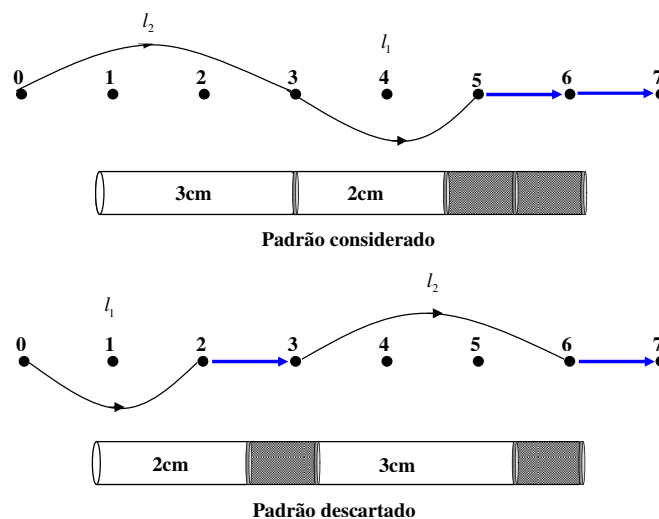


Figura 2.3 - Exemplo de padrões equivalentes

2.4.2 Modelos Onecut

Modelos *onecut* para o problema de corte unidimensional foram introduzidos por Dyckhoff (1981). Segundo Dyckhoff (1981), o modelo *onecut* explicita geralmente,

mais soluções simétricas que o modelo proposto por [Gilmore e Gomory \(1961\)](#), equações de (2.7) a (2.9), possivelmente, porque o conjunto residual resultante das operações *onecut*, pode ser composto por peças iguais, embora tenha sido formado a partir de objetos de diferentes tamanhos, como mostra [Figura 2.4](#).

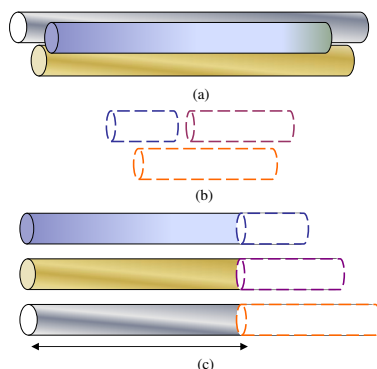


Figura 2.4 - (a) Diferentes objetos em estoque; (b) itens; (c) *onecut* (simetria).

Em modelos *onecut*, um objeto padrão em estoque é dividido em duas outras peças menores (operação com um único corte no objeto), denominadas primeira seção e segunda seção do *onecut* respectivamente, e pelo menos uma destas peças menores corresponde ao tamanho de um dos itens do problema. Modelos similares são tratados em [Ferreira et al. \(1990\)](#) e [Zak \(2002\)](#). As peças a serem divididas podem ser oriundas dos objetos em estoque ou de peças intermediárias, resultantes das operações de corte previamente realizadas. Para a formulação do modelo, sejam S o conjunto de K objetos de diferentes tamanhos $p \in \{L_1, \dots, L_k\} \subset N$ e D o conjunto de itens demandados de diferentes tamanhos $q \in \{l_1, \dots, l_m\} \subset N$ tal que $S \cap D = \emptyset$ em que N é o conjunto dos números naturais. As seguintes variáveis são consideradas:

- z_k : é o número de objetos de comprimento L_k usados;
- R é o conjunto de peças residuais;
- N_q é a demanda do item de tamanho q ;
- y_{pq} indica o número de objetos de comprimento p que são divididos em um item de comprimento q , mais uma peça residual de comprimento $p - q$;
- B_k é a disponibilidade do objeto L_k em estoque, $k = 1, \dots, K$.

O modelo *onecut* é o seguinte:

$$\text{minimize } \sum_{k=1}^K L_k z_k \quad (2.20)$$

$$\text{sujeito a : } z_k + \sum_{p \in D: p+q \in SUR} y_{p+q,p} \geq \sum_{p \in D: p < q} y_{q,p}, \quad \{\forall q = L_K\}, k = 1, \dots, K \quad (2.21)$$

$$\sum_{p \in SUR: p > q} y_{p,q} + \sum_{p \in D: p+q \in SUR} y_{p+q,p} \geq \sum_{p \in D: p < q} y_{q,p} + N_q, \quad \forall q \in (D \cup R) \setminus S, \quad (2.22)$$

$$z_k \leq B_k, k = 1, \dots, K, \quad (2.23)$$

$$y_{p,q} \geq 0, \text{ e inteiros, } p \in S \cup R, q \in D, q < p, \quad (2.24)$$

$$z_k \geq 0, \text{ e inteiros, } k = 1, \dots, K, q \in D, \quad (2.25)$$

em que ($N_q = 0$; para $q \notin D$ com $q = l_i, N_q = d_i, i = 1, \dots, m$). A função objetivo (2.20) minimiza o número de objetos de comprimento L_k a ser cortado. O conjunto de restrições (2.21) impõe que os cortes dos itens demandados de um determinado tamanho devem ser feitos de objetos em estoque ou de peças obtidas de cortes realizados. A restrição (2.22) garante que a demanda dos itens seja atendida. (2.23) são restrições de limitação da quantidade de objetos de comprimento L_k em estoque. (2.24) são restrições da não negatividade do número de peças de largura p a serem divididas em peças de largura q . (2.25) são restrições de não negatividade do número de objetos de comprimento L_k utilizados.

Como exemplo, reproduzido de [Carvalho \(2002\)](#), considere o conjunto $S = (9, 6, 5)$ dos comprimentos dos objetos em estoque e o conjunto $D = (4, 3, 2)$ dos comprimentos dos itens com demandas 20, 10 e 20 respectivamente. Das possíveis operações *onecut* o conjunto de peças residuais é $R = (7, 6, 5, 4, 3, 2)$. Na representação matricial mostrada na [Figura 2.5](#), tem-se as restrições do modelo de programação linear para este exemplo.

	z_1	z_2	z_3	y_{92}	y_{93}	y_{94}	y_{72}	y_{74}	y_{64}	y_{63}	y_{53}	y_{42}	
9	1			-1	-1	-1							≥ 0
8													≥ 0
7				1			-1	-1					≥ 0
6		1			1				-1	-1			≥ 0
5			1			1	1				-1		≥ 0
4						1		1	1			-1	≥ 20
3					1			1		2	1		≥ 10
2				1			1		1		1	2	≥ 20
$W_k = 9$	1												$\leq B_1$
6		1											$\leq B_2$
5			1										$\leq B_3$

Figura 2.5 - Exemplo de programação linear para modelos onecut

Fonte: Reproduzida de [Carvalho \(2002\)](#).

Observe que $K = 3$, ou seja, três objetos de comprimento 9, 6 e 5 existentes em estoque. A quinta linha da matriz ([Figura 2.5](#)), representa a restrição $z_3 + y_{94} + y_{72} - y_{53} \geq 0$, indicando que a quantidade de objetos originais de comprimento $L_3 = 5$, somada à quantidade de objetos de comprimento $L_1 = 9$ usada para obter itens de tamanho 4, mais a quantidade de peças residuais de tamanho 7 usadas para obter itens de tamanho 2, subtraída do número de peças de tamanho 5 (originais e residuais) usadas para obter itens de tamanho 3 deve ser positiva. De um modo geral, esta restrição simplesmente garante a existência de peças de tamanho 5 em quantidade suficiente para produzir itens de tamanho 3 indicando também que, ou são geradas peças residuais de tamanho 5, ou são cortadas peças de tamanho 5 para produzir itens menores.

A restrição de demanda $y_{94} + y_{74} + y_{64} - y_{42} \geq 20$ garante que a quantidade de itens de comprimento 4 cortada de objetos de tamanho 9, 7, 6 excluindo-se as peças residuais de tamanho 4 que são usadas para cortar itens de tamanho 2, deve pelo menos atender a demanda do item $l_1 = 4$. As discussões e abordagens apresentadas nesta seção para modelos onecut, basearam-se em [Carvalho \(2002\)](#).

2.4.3 Modelo do Bin Packing Problem

O *Bin Packing Problem* (BPP) consiste em alocar peças de tamanhos l_j e de demandas $d_j = 1$, $j = 1, \dots, m$, em peças maiores (*bins*), de forma a não exceder sua capacidade máxima L e minimizar o número de *bins* usados. Este modelo tem sido utilizado geralmente em PCEU com demandas pequenas. A formulação matemática para este problema para o caso unidimensional, é a seguinte:

$$\text{minimize } z = \sum_{i=1}^n y_i \quad (2.26)$$

$$\text{sujeito a : } \sum_{j=1}^m l_j x_{ij} \leq L y_i, \quad i = 1, \dots, n, \quad (2.27)$$

$$\sum_{i=1}^n x_{ij} = 1, \quad j = 1, \dots, m \quad (2.28)$$

$$y_i = 0 \text{ ou } 1, \quad i = 1, \dots, n \quad (2.29)$$

$$x_{ij} = 0 \text{ ou } 1, \quad i = 1, \dots, n, \quad j = 1, \dots, m \quad (2.30)$$

em que:

- y_i é igual a 1 se o objeto i é utilizado, e 0 caso contrário $i = 1, \dots, n$;
- x_{ij} é igual a 1 se o item de comprimento l_j é alocado no objeto i e 0 caso contrário, para $i = 1, \dots, n$; $j = 1, \dots, m$;
- n é o número máximo de objetos;
- m é o número total de itens (podem ser iguais).

A função objetivo (2.26) minimiza o número de recipientes (*bins*). (2.27) são restrições de capacidade do *bin*. As restrições (2.28), garantem que cada item seja alocado a apenas um e somente um *bin*. As restrições (2.29) e (2.30) indicam que as variáveis de decisão são binárias.

Veja que no modelo clássico, temos uma variável para cada tipo de padrão, enquanto que no BPP temos uma variável para cada item. Assim, se um item j tem demanda d_j , tem-se d_j variáveis para estes itens se quisermos manter apenas variáveis binárias. Usualmente, quanto maior o número de variáveis, maior o esforço computacional necessário para determinar uma solução para o modelo. Portanto, a aplicação deste

modelo com variáveis 0-1 é mais apropriada a problemas práticos em que há poucos itens de cada tipo a produzir.

No caso de permitirmos variáveis inteiras, não somente binárias, a seguinte formulação pode ser utilizada:

$$\text{minimize } z = \sum_{i=1}^n y_i \quad (2.31)$$

$$\text{sujeito a : } \sum_{j=1}^m l_j x_{ij} \leq L y_i, \quad i = 1, \dots, n, \quad (2.32)$$

$$\sum_{i=1}^n x_{ij} = d_j, \quad j = 1, \dots, m \quad (2.33)$$

$$y_i = 0 \text{ ou } 1, \quad i = 1, \dots, n \quad (2.34)$$

$$x_{ij} \geq 0 \text{ e inteiro, } \quad i = 1, \dots, n, \quad j = 1, \dots, m \quad (2.35)$$

neste caso m é a quantidade de tipos de itens diferentes.

2.5 Extensão dos Modelos para Diferentes Objetos

Os modelos clássico e do *bin packing* apresentados anteriormente, para um único tipo de objeto em estoque, podem ser facilmente estendidos para o caso em que os objetos em estoque tem tamanhos diferentes, (ver [Morabito \(1994\)](#), [Belov e Scheithauer \(2002a\)](#)). Observe que o modelo *onecut* da [Subseção 2.4.2](#), foi formulado para objetos variados em estoque e em quantidade limitada. O modelo de fluxo de arcos também pode ser estendido para o caso de múltiplos tipos de objetos. Inicialmente os objetos em estoque de comprimento L_k e quantidade $B_k > 0$ são agrupados em K diferentes classes para $k = 1, \dots, K$. Os itens são agrupados em m classes, com demandas d_i e comprimento l_i para $i = 1, \dots, m$. Vamos supor que tais classes são indexadas em ordem decrescente de comprimento dos objetos e dos itens. Sem perda de generalidade tem-se $L_1 > L_2 > \dots > L_K$ e $l_1 > l_2 > \dots > l_m$. Seja $L_{max} = \max \{L_k\} = L_1$ para $k = 1, \dots, K$. O conjunto de arcos A , apresentado na [Subseção 2.4.1](#), é assim redefinido: $A = \{(i, j) : 0 \leq i < j \leq L_{max}, j - i = l_t \forall 1 \leq i \leq m\}$, em que $(i, i + 1)$, $i = 0, 1, \dots, L_{max} - 1$ são arcos de perda, admitindo-se $l_i > 1$ para $i = 1, \dots, m$

A existência de um padrão de corte para o objeto de comprimento L_k , está condicionada à existência de um caminho no grafo $G = (V, A)$, entre os vértices 0 e L_k . A seleção do objeto a ser utilizado num determinado padrão de corte, é feita por meio

de arcos de retorno, que são arcos direcionados do vértice L_k ao vértice 0. Os arcos que compõem o caminho definem os itens selecionados para o padrão. O modelo matemático é o seguinte:

$$\text{minimize } \sum_{k=1}^K L_k z_k \quad (2.36)$$

$$\text{sujeito a : } - \sum_{(i,j) \in A} x_{ij} + \sum_{(j,f) \in A} x_{jf} = \begin{cases} \sum_{k=1}^K z_k \text{ se } j = 0, \\ -z_k \text{ se } j = L_k, k = 1, \dots, K, \\ 0 \text{ caso contrário,} \end{cases} \quad (2.37)$$

$$\sum_{(i,i+l_t) \in A} x_{i,i+l_t} \geq b_t, t = 1, \dots, m \quad (2.38)$$

$$z_k \leq B_k, k = 1, \dots, K, \quad (2.39)$$

$$x_{ij} \geq 0 \text{ e inteiros } \forall (i, j) \in A, \quad (2.40)$$

$$z_k \geq 0 \text{ e inteiros, } k = 1, \dots, K. \quad (2.41)$$

em que:

- z_k : é o número de objetos do tipo L_k usados;
- x_{ij} corresponde ao número de itens de tamanho $j - i$ obtidos de algum objeto L_k , $k = 1, \dots, K$;
- A é o conjunto de arcos.

A função objetivo (2.36) minimiza a quantidade de objetos de comprimento L_k usados. (2.37) são restrições de conservação de fluxo. (2.38) garantem que as demandas dos itens sejam atendidas. (2.39) exigem que o número de objetos utilizados não ultrapasse a quantidade máxima permitida, (2.40) e (2.41) são restrições de integralidade e não negatividade das variáveis de decisão.

Como exemplo, consideremos objetos em estoque de comprimentos $L_1 = 9, L_2 = 6$ e $L_3 = 5$ em quantidades B_1, B_2 e B_3 , respectivamente, e itens de comprimento $l_1 = 4, l_2 = 3$ e $l_3 = 2$ com demandas $d_1 = 20, d_2 = 10$ e $d_3 = 20$. O grafo para este exemplo, reproduzido de [Carvalho \(2002\)](#), é mostrado na [Figura 2.6](#).

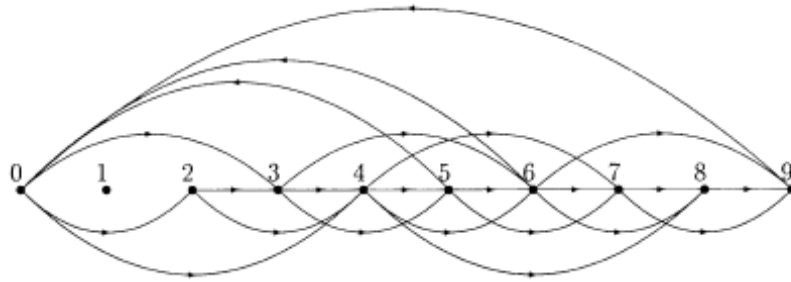


Figura 2.6 - Grafo para diferentes objetos em estoque.

Fonte: Reproduzida de [Carvalho \(2002\)](#).

A soma total de objetos disponíveis é calculada através da relação $\sum_k L_k B_k$. As restrições do modelo de *PL* reproduzida de [Carvalho \(2002\)](#), são mostradas na [Figura 2.7](#).

	z_1	z_2	z_3	x_{04}	x_{48}	x_{03}	x_{36}	x_{47}	x_{69}	x_{02}	x_{24}	x_{35}	x_{46}	x_{57}	x_{68}	x_{79}	x_{23}	x_{34}	x_{45}	x_{56}	x_{67}	x_{78}	x_{89}		
node 0	1	1	1	-1		-1				-1														= 0	
1																									= 0
2										1	-1														= 0
3						1	-1						-1					1	-1						= 0
4				1	-1			-1			1		-1					1	-1						= 0
5		-1										1		-1					1	-1					= 0
6		-1					1	-1					1		-1					1	-1				= 0
7								1							1	-1						1	-1		= 0
8					1											1							1	-1	= 0
9		-1							1								1							1	= 0
$w_i = 4$				1	1																				≥ 20
3						1	1	1	1																≥ 10
2										1	1	1	1	1	1	1									≥ 20
$W_k = 9$	1																								$\leq B_1$
6		1																							$\leq B_2$
5			1																						$\leq B_3$

Figura 2.7 - Modelo de *PL* para diferentes objetos em estoque.

Fonte: Reproduzida de [Carvalho \(2002\)](#).

Apresentamos a seguir uma extensão do modelo proposto por [Gilmore e Gomory \(1961\)](#), considerando-se agora o caso em que tem-se vários tipos de objetos em estoque, em quantidades de cada tipo suficientes para atender toda a demanda. Utilizamos como referência neste caso o trabalho de [Arenales et al. \(36.,2004\)](#). Para formulação do modelo (2.7)-(2.8) com um único tipo de objeto, consideramos somente os dados de demanda, passamos agora a considerar também os dados de estoque:

- K número de tipos de objetos em estoque;
- L_k comprimento do objeto do tipo k , $k = 1, \dots, K$;
- c_k custo do objeto do tipo k , $k = 1, \dots, K$.

O modelo matemático neste caso é definido da mesma forma como no modelo clássico para um tipo de objeto em estoque, mas os padrões de corte devem ser definidos para cada objeto em estoque, portanto satisfazem as seguintes condições de várias mochilas:

$$\begin{aligned} l_1 a_{1jk} + l_2 a_{2jk} + \dots + l_m a_{mjk} &\leq L_k \\ a_{ijk} &\geq 0, \text{ e inteiros } i = 1, \dots, m; \quad j = 1, \dots, n_k; \quad k = 1, \dots, K. \end{aligned} \quad (2.42)$$

Suponha que o sistema (2.42) tenha n_k soluções dadas por:

$$\mathbf{a}_{1k} = \begin{pmatrix} a_{11k} \\ a_{21k} \\ a_{31k} \\ \cdot \\ \cdot \\ \cdot \\ a_{m1k} \end{pmatrix}; \quad \mathbf{a}_{2k} = \begin{pmatrix} a_{12k} \\ a_{22k} \\ a_{32k} \\ \cdot \\ \cdot \\ \cdot \\ a_{m2k} \end{pmatrix}; \quad \mathbf{a}_{3k} = \begin{pmatrix} a_{13k} \\ a_{23k} \\ a_{33k} \\ \cdot \\ \cdot \\ \cdot \\ a_{m3k} \end{pmatrix}; \quad \dots; \quad \mathbf{a}_{n_k k} = \begin{pmatrix} a_{1n_k k} \\ a_{2n_k k} \\ a_{3n_k k} \\ \cdot \\ \cdot \\ \cdot \\ a_{mn_k k} \end{pmatrix}.$$

Em que a_{ijk} é o número de itens do tipo i no padrão de corte j para o objeto k , $i = 1, \dots, m$, $j = 1, \dots, n_k$, $k = 1, \dots, K$ e \mathbf{a}_{jk} corresponde ao j -ésimo padrão de corte viável obtido do objeto k . O modelo matemático neste caso é o seguinte:

$$\text{minimizar } \sum_{j=1}^{n_1} c_1 x_{j1} + \sum_{j=1}^{n_2} c_2 x_{j2} + \dots + \sum_{j=1}^{n_K} c_K x_{jK} \quad (2.43)$$

$$\text{sujeito a : } \sum_{j=1}^{n_1} a_{ij1} x_{j1} + \sum_{j=1}^{n_2} a_{ij2} x_{j2} + \dots + \sum_{j=1}^{n_K} a_{ijK} x_{jK} \geq \mathbf{d}, \quad i = 1, \dots, m \quad (2.44)$$

$$x_{jk} \geq 0, \quad j = 1, \dots, n_k, \quad k = 1, \dots, K, \quad (2.45)$$

em que x_{jk} é o número de vezes que o objeto do tipo k é cortado usando o padrão j .

Veja que o modelo (2.7)-(2.8) com um único tipo de objeto em estoque é um caso particular de (2.43)-(2.45) com $K = 1$. Lembramos que as colunas da matriz de

restrições são os vetores associados aos padrões de corte para cada objeto.

Para a extensão do *Bin Packing Problem* para objetos de diferentes tamanhos, suponha que existem em estoque K classes de *bins* de capacidade L_k em quantidade n_k , tal que $L_k \neq L_t$ se $k \neq t$. Tem-se neste caso, o seguinte modelo de programação linear:

$$\text{minimize } z = \sum_{k=1}^K \sum_{i=1}^{n_k} y_{ik} \quad (2.46)$$

$$\text{sujeito a : } \sum_{j=1}^m l_j x_{ijk} \leq L_k y_{ik}, \quad i = 1, \dots, n_k, \quad k = 1, \dots, K \quad (2.47)$$

$$\sum_{k=1}^K \sum_{i=1}^{n_k} x_{ijk} = 1, \quad j = 1, \dots, m \quad (2.48)$$

$$y_{ik} = 0 \text{ ou } 1, \quad i = 1, \dots, n_k, \quad k = 1, \dots, K \quad (2.49)$$

$$x_{ijk} = 0 \text{ ou } 1, \quad i = 1, \dots, n_k, \quad j = 1, \dots, m, \quad k = 1, \dots, K \quad (2.50)$$

em que:

- y_{ik} é igual a 1 se o objeto i de capacidade L_k é utilizado, e 0 caso contrário $i = 1, \dots, n_k, k = 1, \dots, K$;
- x_{ijk} é igual a 1 se o item de comprimento l_j é alocado no objeto i de capacidade L_k e 0 caso contrário, para $i = 1, \dots, n_k; j = 1, \dots, m; k = 1, \dots, K$.
- n_k é o número máximo de *bins* de capacidade $L_k, k = 1, \dots, K$;
- m é o número de itens (podem ser iguais).

A função objetivo (2.46) minimiza o número de recipientes (*bins*) de capacidade L_k . (2.47) são restrições de capacidade do *bin*. As restrições (2.48), garantem que cada item seja alocado a apenas um e somente um *bin*. As restrições (2.49) e (2.50) indicam que as variáveis de decisão são binárias.

3 MINIMIZAÇÃO DE PADRÕES DE CORTE

Apresentamos a seguir uma revisão bibliográfica de trabalhos encontrados na literatura que versam sobre a minimização de padrões de corte.

3.1 Revisão Bibliográfica

Este problema é NP-hard (McDiarmid (1999)), e o único algoritmo exato encontrado na literatura para resolvê-lo foi proposto por Vanderbeck (2000). Nos demais trabalhos encontrados, heurísticas são propostas ou utilizadas.

Haessler (1975) propôs o seguinte modelo para o problema de corte de estoque unidimensional, com custos associados à mudança de padrões e perdas. O modelo proposto por Haessler (1975) para o problema é:

$$\text{Min } c_1 \sum_{j=1}^n t_j x_j + c_2 \sum_{j=1}^n \delta(x_j) \quad (3.1)$$

$$\text{s.a } d_l^i \leq \sum_{j=1}^n a_{ij} x_j \leq d_u^i \quad i = 1, \dots, m \quad (3.2)$$

$$x_j \geq 0 \text{ e inteiro } \quad i = 1, \dots, m \quad j = 1, \dots, n. \quad (3.3)$$

em que:

- $a_{ij} \geq 0$ é um parâmetro que indica a quantidade de itens do tipo i produzida pelo padrão j , $i = 1, \dots, m$, $j = 1, \dots, n$;
- c_1 é o custo unitário da perda;
- c_2 é o custo do *setup*;
- d_l^i e d_u^i é o limitante inferior e superior da demanda do item i , $i = 1, \dots, m$;
- t_j é a quantidade de perda gerada pelo padrão j , dada por $t_j = L - \sum_{i=1}^m a_{ij} l_i$,

$i = 1, \dots, m$, veja que a restrição (3.3) garante que $t_j \geq 0$, $j = 1, \dots, n$;

$$\delta(x_j) = \left\{ \begin{array}{l} 1 \text{ se } x_j > 0 \\ 0 \text{ se } x_j = 0 \end{array} \right\}.$$

Considerando a_{ij} como parâmetro, $i = 1, \dots, m$, $j = 1, \dots, n$, neste modelo temos um problema não linear devido à função objetivo (3.1), segundo Haessler (1975), devido à grande quantidade de padrões existentes, é muito difícil resolvê-lo com o uso dos algoritmos existentes. Por isso, ele propôs uma heurística SHP (*Sequential Heuristic Procedure*) capaz de gerar boas soluções (não necessariamente ótimas) num tempo computacional aceitável. A chave para isso é tentar gerar padrões de corte com perda mínima e alto nível de aplicação. A heurística sugerida por Haessler (1975) inicia analisando a demanda a ser atendida. Para isso são calculados o número de objetos (NO) necessários para atender a demanda restante, ou seja, $NO = (\sum_{i=1}^m dr_i l_i) / L$ em que dr_i é a demanda residual do item i (veja logo adiante o que chamamos de demanda residual), e o número médio de itens (NM) fornecido pelo objeto L é dado por $NM = \sum_{i=1}^m dr_i / [(\sum_{i=1}^m dr_i l_i) / L]$.

A seguir, uma busca é executada para determinar se existe um padrão de corte viável que satisfaz os níveis de aspiração desejáveis, ou seja um padrão satisfazendo as seguintes condições:

- 1- $L - \sum_{i=1}^m a_{ij} l_i \leq MAXTL$ (a perda não excede o máximo permitido)
- 2- $MINR \leq \sum_{i=1}^m a_{ij} l_i \leq MAXR$ (o número de itens no padrão é limitado)
- 3- $dr_i / (a_{ij} MINU) \geq 1$ (a quantidade de itens por padrão é limitada)

em que:

MAXTL: máxima perda permissível.

MAXR: número máximo de itens permitido no padrão.

MINR: número mínimo de itens permitido no padrão.

MINU: número mínimo de objetos a serem cortados de acordo com o padrão.

Os valores destes parâmetros dependem da demanda remanescente de cada item e podem variar ao longo do processo de busca por um bom padrão. Os valores sugeridos por Haessler (1975) são: o parâmetro MAXTL assume valores entre $0,006L$ e $0,03L$, $MINR = NM - 1$, MAXR assume um valor igual à capacidade máxima do equipamento de corte e MINU assume valores entre $0,5NO$ e $0,9NO$.

A busca por um padrão que satisfaz os níveis de aspiração corrente tem início com ordenação dos itens com demanda residual não-nula em ordem decrescente de tamanho. Inicialmente, $dr_i = d_i$ para $i = 1, \dots, m$. Os padrões são gerados em ordem lexicográfica, com prioridade para o item com maior demanda residual. Note que a busca deve ser efetuada apenas sobre aqueles padrões viáveis com relação à meta de uso, $MINU$, sendo então calculado o limitante superior para a quantidade de itens permitida por padrão, $\lfloor dr_i/MINU \rfloor$ para $i = 1, \dots, m$.

Se nenhum padrão viável é encontrado, o nível de aspiração é reduzido diminuindo o valor de $MINU$. Isto faz com que o subconjunto de padrões sobre o qual a busca é realizada seja ampliado. Se nenhum padrão com os valores de $MINU > 1$ é determinado, então o padrão com menor perda é aceito e $MINU$ é fixado em 1. Isto garante que o procedimento termina com uma solução viável para o problema.

O padrão selecionado entrará na solução com a maior frequência possível sem que haja superprodução de algum item e, após atualizar a demanda, o processo é repetido até que toda demanda seja atendida. Segundo [Haessler \(1975\)](#) a vantagem deste procedimento é que não há necessidade de arredondamento da solução, sendo possível cortar a quantidade exata de cada item. O fluxograma da heurística SHP é mostrado na [Figura 3.1](#).

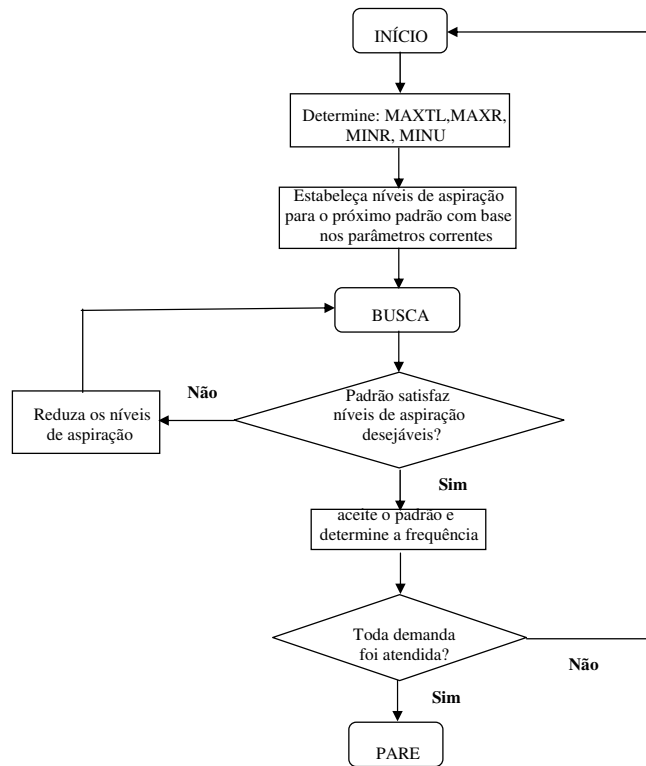


Figura 3.1 - Algoritmo SHP

Umetani et al. (2003) propuseram uma metaheurística que incorpora uma técnica de geração de colunas para resolver o problema de redução de padrões de corte. Eles sugerem um modelo matemático para o problema de corte de estoque unidimensional que fixa o número de diferentes tipos de padrões na solução. Com isso, eles buscam uma solução cujo desvio quadrático da produção em relação à demanda se mantenha dentro de uma margem aceitável. A minimização do número de diferentes padrões é obtida através da aplicação iterativa de um algoritmo de busca local, considerando diferentes valores para o número de padrões distintos. O modelo matemático proposto cujo objetivo é minimizar a quantidade de padrões distintos é o seguinte:

$$\begin{aligned}
 & \text{Minimize } |\Pi| \\
 & \text{Sujeito a : } \sum_{i=1}^m \left(\sum_{j \in \Pi} a_{ij} x_j - d_i \right)^2 \leq D, \\
 & x_j \in Z_+
 \end{aligned} \tag{3.4}$$

em que:

- $|\Pi|$ é a cardinalidade do conjunto de padrões de corte;
- m é o número de diferentes tipos de itens;
- a_{ij} é o numero de itens do tipo i no padrão do tipo j ;
- x_j é frequência do padrão j ;
- d_i é a demanda do item i ;
- D é um limitante aceitável do desvio de produção.

O problema anterior apresenta dificuldades para a sua resolução geralmente devido à grande quantidade de padrões existentes. Para resolvê-lo, [Umetani et al. \(2003\)](#) reformularam o modelo proposto fixando o número de padrões diferentes. Seja N uma constante que representa a quantidade de padrões de corte, o problema (3.4) é então reformulado como segue, tendo agora como objetivo minimizar o desvio quadrático da produção em relação à demanda:

$$\begin{aligned}
 \text{Minimize } f(\Pi, x) &= \sum_{i=1}^m \left(\sum_{j \in \Pi} a_{ij} x_j - d_i \right)^2 \\
 \text{Sujeito a : } &|\Pi| = N, \Pi \subseteq S \\
 &x_j \in Z_+
 \end{aligned} \tag{3.5}$$

em que S é o conjunto de todos os possíveis padrões.

Para resolver o problema os autores aplicaram o *ILS* (*Iterated Local Search*) um algoritmo de busca local associado a uma técnica adaptada de geração de colunas. O algoritmo de busca local parte de alguma solução inicial do problema. Tal solução é substituída por uma outra melhor na sua vizinhança, até que nenhuma outra solução melhor seja encontrada. A vizinhança é um conjunto de soluções obtidas da solução corrente através de pequenas perturbações, e é obtida através da remoção de um padrão da solução corrente e introdução de um outro potencialmente melhor.

A solução resultante, para a qual não existe uma solução melhor na vizinhança é chamada de *ótimo local*. Segundo [Umetani et al. \(2003\)](#) o algoritmo *ILS* encontra boas soluções com um número menor de padrões diferentes para a maioria dos exemplos testados, quando comparado com os resultados obtidos por outros algoritmos encontrados na literatura.

Vanderbeck (2000) propôs um algoritmo exato para resolver o problema de minimizar o número de diferentes padrões no problema de corte de estoque unidimensional, formulado como um problema de programação quadrática inteira que envolve um número exponencial de variáveis binárias e colunas associadas (padrões de corte). O modelo proposto envolve uma quantidade limitada de objetos em estoque, e é resolvido usando uma técnica que combina geração de colunas e *branch-and-bound*. O subproblema gerado é um problema de programação inteira não linear que pode ser decomposto em vários problemas limitados da mochila. O modelo matemático proposto por Vanderbeck (2000) é o seguinte:

$$Z = \text{Minimize} \sum_{k=1}^K y_k, \quad (3.6)$$

$$\text{Sujeito a : } \sum_{k=1}^K x_k a_{ik} = d_i \quad i = 1, \dots, m, \quad (3.7)$$

$$\sum_{k=1}^K x_k \leq K, \quad (3.8)$$

$$x_k \leq K y_k \quad k = 1, \dots, K \quad (3.9)$$

$$\sum_{i=1}^n l_i a_{ik} \leq L y_k \quad k = 1, \dots, K \quad (3.10)$$

$$a_{ik} \in N \quad i = 1, \dots, m, \quad k = 1, \dots, K \quad (3.11)$$

$$y_k \in \{0, 1\} \quad k = 1, \dots, K \quad (3.12)$$

$$x_k \in N \quad k = 1, \dots, K \quad (3.13)$$

em que:

- a_{ik} representa o número de itens do tipo i produzido pelo padrão k , $i = 1, \dots, m$
- K é o número máximo de objetos em estoque;
- y_k é uma variável binária que assume o valor 1 se o padrão de corte k é usado e zero caso contrário;
- d_i é a demanda do item i , $i = 1, \dots, m$;
- x_k é o número de vezes que o padrão de corte k é usado;
- m é o número de diferentes tipos de itens;
- l_i é o tamanho do item i , $i = 1, \dots, m$
- L é o comprimento do objeto em estoque.

O objetivo é minimizar o número de diferentes padrões de corte a serem usados.

O conjunto de restrições não lineares (3.7), asseguram que a demanda dos itens seja atendida na quantidade exata. A restrição (3.8) limita a quantidade de objetos que podem ser utilizados. (3.9) relaciona logicamente variáveis indicadoras dos padrões utilizados, enquanto que a viabilidade dos padrões de corte é garantida pelas restrições (3.10).

Entre os trabalhos encontrados na literatura para a resolução do PCEU e que levam em consideração também os custos relativos ao número de diferentes padrões, o de Vanderbeck (2000) é o único que apresenta um método exato de resolução. Uma das possíveis explicações para isso é devida à dificuldade de resolução deste problema.

Cui et al. (2008) apresentaram o SHPC (*SHP-Sequential Heuristic Procedure C*), uma heurística para o problema de corte unidimensional tendo como meta dois objetivos: a minimização da perda e do número de diferentes padrões na solução. Segundo os autores, o algoritmo proposto é baseado nas idéias de Haessler (1975), com níveis de aspiração para o desperdício e número de itens diferentes por padrão. A diferença entre eles é que enquanto a heurística SHP proposta por Haessler (1975), definida anteriormente, utiliza todos os itens para gerar o padrão corrente, SHPC utiliza um subconjunto de itens selecionados dentre aqueles com demandas não-nulas. Para $i = 1, \dots, m$, seja $B = \{b_1, \dots, b_m\}$ em que b_i é o limitante superior da quantidade de itens do tipo i no padrão e $\mathbf{d} = \{d_1, \dots, d_m\}$ o vetor de demandas. Segundo Cui et al. (2008) quanto mais itens com demandas não-nulas forem selecionados e quanto maior for a soma dos comprimentos destes itens, maior é a possibilidade em se gerar um padrão com pouco desperdício. Para determinar uma solução melhor para o problema, são utilizados os chamados parâmetros de controle: n_b que controla o número mínimo de diferentes tipos de itens por padrão e r_b que estabelece a taxa tolerável de desperdício permitida por padrão. $r_b L$ é a soma mínima do comprimento dos itens do conjunto B . Seja k_0 a frequência mínima para o próximo padrão. Seja n_I o número de tipos de itens e L_{sum} a soma total dos comprimentos dos itens no conjunto B . A frequência mínima k_0 para o padrão a ser gerado é um valor compreendido entre k_{max} e 1, em que $k_{max} = \max \{d_i\}$ para $i = 1, \dots, m$. O procedimento (SELECT) que determina o conjunto B e k_0 é mostrado a seguir:

ALGORITMO SELECT

INÍCIO

Entrada: Problema de Corte Unidimensional

Saída: Conjunto de itens candidatos ao próximo padrão

1-Seja $k_{max} = \max \{d_i\}$, $i = 1, \dots, m$

2-Para $k = k_{max}, \dots, 2$

3- Seja $n_I = L_{sum} = 0$

4- Para $i = 1, \dots, m$

5- Seja $b_i = \lfloor d_i/k \rfloor$

6- **Se** $b_i > 0$ **então** seja $n_I = n_I + 1$ e $L_{sum} = L_{sum} + b_i l_i$

7- **Se** $n_I \geq n_b$ e $L_{sum} \geq r_b L$ **então** faça $k_0 = k$ e saia do procedimento

8-Passe para o próximo valor de k

9-**Se** nenhum valor de k for encontrado **então** seja $b_i = d_i$ para $i = 1, \dots, m$ e $k_0 = 1$

FIM DO ALGORITMO

Para gerar o padrão usando os itens em B , [Cui et al. \(2008\)](#) estabelecem um valor de utilidade do item i , definido como $v_i = l_i$ se $l_i \leq \alpha_b L$ e $v_i = \beta_b l_i$ caso contrário, para $i = 1, \dots, m$, sendo α_b e β_b dois números reais com $0 \leq \alpha_b \leq 1$ e $\beta_b \geq 1$, assim, os itens maiores terão prioridade sobre os itens menores na geração do padrão. O seguinte problema da mochila limitado é resolvido para obter o padrão de corte:

$$\begin{aligned} & \text{Maximize } \left\{ \sum_{i=1}^m v_i y_i \right\} \\ & \text{Sujeito a : } \sum_{i=1}^m l_i y_i \leq L \\ & 0 \leq y_i \leq b_i, \quad y_i \text{ inteiros} \quad i = 1, \dots, m \end{aligned} \tag{3.14}$$

Toda vez que um padrão é gerado, determina-se sua frequência, atualiza-se a demanda e repete-se o processo de geração de padrões, até que toda a demanda tenha sido atendida. O algoritmo GEN que gera uma solução inteira para o problema de corte, chama o procedimento SELECT e é mostrado a seguir.

ALGORITMO GEN

INÍCIO

Entrada: Problema de Corte Unidimensional

Saída: Plano de corte para o problema

- 1-**Enquanto** (houver demanda não-nula) faça:
- 2- Chame o procedimento SELECT e determine o conjunto B
- 3- Resolva o problema da mochila limitada para obter o padrão corrente
- 4- Adicione o padrão corrente ao plano de corte e determine a frequência
- 5- Atualize a demanda dos itens usando o padrão gerado e volte ao passo 1
- 6- **Fim do Enquanto**

FIM DO ALGORITMO

Segundo Cui et al. (2008), a combinação de diferentes níveis de aspiração para o desperdício e quantidade de diferentes itens permitida por padrão, conduz a diferentes soluções para o problema de corte, o que possibilita escolher dentre os planos de corte gerados aquele que apresente menor percentual de desperdício e menor número de padrões distintos.

Cada par distinto de parâmetros (n_b, r_b) , conduz a uma solução para o problema que pode ser melhor ou pior que a solução corrente, segundo os objetivos almejados. Cui et al. (2008) consideram que uma solução é melhor que outra quando a perda gerada é reduzida, ou quando a perda permanece a mesma e o número de padrões distintos é menor. O algoritmo SHPC completo é como segue:

ALGORITMO SHPC

INÍCIO

Entrada: Problema de Corte Unidimensional

Saída: Melhor Plano de corte obtido para o problema

- 1-Seja $T_{best} = 1$ e $N_{best} = M$ (M um número grande)
- 2-Para $n_b = n_b^{\min}, \dots, n_b^{\max}$
- 3- Para $r_b = r_b^{\min}, \dots, r_b^{\max}$ (*passo* = Δr)
- 4- Entre com os dados do problema.
- 5- Chame o procedimento GEN para gerar o plano de corte corrente
- 6- **Se** $T_c < T_{best}$ ou $T_c = T_{best}$ e $N_c < N_{best}$ **então**
- 7- $T_{best} = T_c$, $N_{best} = N_c$ e o plano de corte corrente é a melhor solução
- 8-Considere o próximo par de parâmetros (n_b, r_b)

FIM DO ALGORITMO

em que:

T_{best} é a perda gerada pelo melhor plano de corte obtido.

T_c é a perda gerada pelo plano de corte corrente.

N_{best} é o número de diferentes padrões do melhor plano de corte.

N_c é o número de diferentes padrões do plano de corte corrente obtido.

Δr é o tamanho do passo.

Segundo os autores, o algoritmo proposto apresenta vantagens em relação a outros encontrados na literatura, como a simplicidade de implementação do código e a independência de planos de corte gerados por outros algoritmos. Para os experimentos computacionais Cui et al. (2008) atribuíram os seguintes valores aos parâmetros: $\alpha_b = 0, 4$, $\beta_b = 1, 15$, $n_b = 1, \dots, 4$ e $r_b = 0, 8; \dots; 5, 6$ com passo $\Delta r = 0, 3$.

Yanasse e Limeira (2006) e Limeira (2003) propuseram um procedimento híbrido para obter um número reduzido de padrões para o Problema de Corte de Estoque Unidimensional. A heurística proposta gera padrões de corte com desperdício limitado, em que cada padrão quando repetido num número máximo de vezes, sem que haja excesso na produção dos itens, deve completar a demanda de pelo menos dois dos itens contidos nele. Se nenhum dos padrões gerados não mais satisfaz os níveis de aspiração desejados, o processo de geração de padrões é interrompido e, se ainda houver itens com demandas não-nulas, um problema residual é resolvido e técnicas de redução de padrões são aplicadas à solução encontrada.

A idéia da heurística é relativamente simples. Do problema original (P), um problema P_1 é definido com os mesmos itens do problema original P mas com as demandas ajustadas de acordo com um fator F_1 . O problema (P_1) é obtido dividindo-se d_1, d_2, \dots, d_m por um fator F_1 e considerando-se apenas a parte inteira. Qualquer padrão de corte viável para o problema (P_1) é também viável para (P) e, além disso, este padrão pode ser cortado F_1 vezes sem que tenhamos excesso de produção de qualquer item.

Para o problema (P_1) um padrão é aceito se possui um desperdício aceitável e completa simultaneamente as demandas de mais de um item contido nele (níveis de aspiração). Este padrão fará parte da solução global do problema (P) tendo uma frequência F_1 . Se nenhum padrão foi escolhido, nenhuma redução é realizada. Um novo fator F_2 é definido e o processo é repetido. O processo termina quando o fator 1 é alcançado e nenhum outro padrão alcança os níveis de aspiração desejados.

Visto que os padrões desejados, quando cortados, são aqueles que finalizam simultaneamente as demandas de pelo menos dois itens, os fatores $F_1, F_2, \dots, 1$ são definidos como submúltiplos das demandas dos itens da seguinte maneira: para cada $j, j = 1, \dots, m, k \in \{1, 2, \dots, d_j\}$, que satisfaz $d_j \equiv 0(\text{mod } k)$. Estes valores são ordenados em ordem decrescente (exclui-se os valores iguais) e são testados um a um.

Um dos níveis de aspiração para os padrões gerados é que o desperdício precisa estar em torno de um valor α que é a meta desejada de desperdício global da solução procurada. O valor de α é obtido do desperdício gerado na solução do problema de corte de estoque. A seguir é apresentada a heurística proposta por [Yanasse e Limeira \(2006\)](#).

HEURÍSTICA

INÍCIO

Entrada: Problema de Corte de Estoque P

Saída: Solução para o Problema de Corte de Estoque com número reduzido de padrões.

(Fase 1)

1. Compute α para o problema (P)
3. Compute os fatores F para a demanda corrente
4. Defina um Problema ($P1$) para o fator corrente F_i e monte uma lista em ordem decrescente. Pegue F_i o primeiro da lista.
5. Gere um Padrão de Corte
6. **Se** o padrão possui desperdício menor ou igual a α e completa pelo menos dois itens

então

armazenar o padrão gerado e sua frequência F_i

atualize a demanda

se toda demanda não foi satisfeita

então volte para o passo 3

senão aplicar técnicas de redução de padrões

senão

se $F_i = 1$

então

resolver o problema residual
montar a solução geral
aplicar técnicas de redução de padrões

senão

atualizar fator F_i (pegue o próximo da lista)
volte para o passo 4

FIM

Diegel et al. (1993) propuseram combinar os padrões para tentar reduzir o número de padrões de corte distintos. Eles apresentaram um procedimento que tenta identificar um par de padrões num plano de corte que pode ser substituído por um único padrão (combinação 2 para 1). Dados dois padrões quaisquer 1 e 2 com suas respectivas frequências x_1 e x_2 , para cada item de comprimento l_i a soma $s_i = a_{i1}x_1 + a_{i2}x_2$, $i = 1, \dots, m$, representa o número de itens do tipo i fornecido pelos dois padrões, e $x_* = x_1 + x_2$ é o número total de objetos usados pelos padrões 1 e 2. Os dois padrões podem ser substituídos por um único padrão $*$ $= (a_{1*}, \dots, a_{m*})$ com $a_{i*} = s_i/x_*$, $i = 1, \dots, m$, se todas as razões s_i/x_* são números inteiros e o padrão $*$ é viável. Aplicando o novo padrão x_* vezes teremos o mesmo número de itens de cada tamanho conforme os dois padrões originais.

Foerster e Wascher (1999) estenderam o método de Diegel et al. (1993) para uma combinação maior de padrões que foi denominado genericamente de KOMBI. Tal procedimento também mantém constante o número de objetos cortados. Os procedimentos extraídos do artigo de Foerster e Wascher (1999) são os seguintes :

- Combinações 3 para 2 e 3 para 1

De acordo com este procedimento as frequências x_* e x_{**} de dois padrões (denotados por $*$ e $**$) resultantes da combinação de três padrões arbitrários 1, 2 e 3 (combinação 3 para 2) são escolhidas de três diferentes maneiras:

$$\begin{aligned} x_* &= x_1 \text{ e } x_{**} = x_2 + x_3 \\ x_* &= x_2 \text{ e } x_{**} = x_1 + x_3 \\ x_* &= x_3 \text{ e } x_{**} = x_1 + x_2 \end{aligned} \tag{3.15}$$

Para identificar dois padrões $*$ $= (a_{1*}, \dots, a_{m*})$ e $**$ $= (a_{1**}, \dots, a_{m**})$ relacionados às

freqüências x_* e x_{**} procede-se da seguinte forma: a soma $s_i = a_{i1}x_1 + a_{i2}x_2 + a_{i3}x_3$, $i = 1, \dots, m$ são computadas, representando o número de itens do tipo i fornecido pelos três padrões originais. Então para todas as freqüências x_* e x_{**} , os padrões $*$ e $**$ são construídos componente a componente. Para todo $i = 1, \dots, m$, são determinadas recursivamente todas as possíveis decomposições inteiras de s_i , isto é, todos os pares $(a_{i*}, a_{i**}), a_{i*}, a_{i**}$ inteiros, que satisfazem $s_i = a_{i*}x_* + a_{i**}x_{**}$.

Toda vez que o procedimento recursivo é chamado, a viabilidade dos padrões determinados (a_{1*}, \dots, a_{m*}) e $(a_{1**}, \dots, a_{m**})$ é verificada. Se a viabilidade não é mantida ($\sum_{i=1}^m l_i a_{i*} > L$ ou $\sum_{i=1}^m l_i a_{i**} > L$) o procedimento recursivo é abandonado. Note que o procedimento de combinação 2 para 1 proposto por [Diegel et al. \(1993\)](#) é um caso especial do procedimento de combinação 3 para 2, em que um dos três padrões iniciais e a freqüência correspondente é mantida inalterada. É também óbvio que o procedimento de combinação 3 para 1 obedece o mesmo princípio de construção da combinação 2 para 1.

- Combinações 4 para 3, 4 para 2 e 4 para 1

Seguindo a mesma metodologia do caso anterior, combinações de quatro padrões (1, 2, 3 e 4) em três $*$ = (a_{1*}, \dots, a_{m*}) , $**$ = $(a_{1**}, \dots, a_{m**})$ e $***$ = $(a_{1***}, \dots, a_{m***})$ (combinação 4 para 3) são consideradas com freqüências assim definidas:

$$\begin{aligned} x_* &= x_{j_1}, \quad x_{**} = x_{j_2} \text{ e } x_{***} = x_{j_3} + x_{j_4} \\ (j_1, j_2, j_3, j_4) &\in \{1, 2, 3, 4\} \end{aligned} \tag{3.16}$$

No máximo existem seis diferentes combinações de freqüências desse tipo. Analogamente todas as decomposições inteiras $(a_{i*}, a_{i**}, a_{i***})$ de s_i satisfazendo $s_i = a_{i*}x_* + a_{i**}x_{**} + a_{i***}x_{***}$ e os padrões de corte viáveis podem ser determinados recursivamente.

Para a combinação de quatro padrões em dois (combinação 4 para 2) existem no máximo sete conjuntos de freqüências x_* e x_{**} para os padrões resultantes $*$ =

(a_{1*}, \dots, a_{m*}) e $** = (a_{1**}, \dots, a_{m**})$:

$$\begin{aligned} x_* &= x_{j_1} \text{ e } x_{**} = x_{j_2} + x_{j_3} + x_{j_4} \text{ ou} \\ x_* &= x_{j_1} + x_{j_2} \text{ e } x_{**} = x_{j_3} + x_{j_4} \\ (j_1, j_2, j_3, j_4) &\in \{1, 2, 3, 4\} \end{aligned} \tag{3.17}$$

Note que o procedimento recursivo identifica os dois padrões $*$ e $**$. A combinação de 4 padrões para 1 pode ser desenvolvida da mesma maneira aplicada ao caso anterior.

- Combinações p para q

Baseado na restrição de que as frequências dos padrões resultantes são a soma das frequências dos padrões iniciais, é possível estabelecer uma generalização da combinação de p padrões para q, em que $p > q$, p e q inteiros com $q > 2$. Em cada caso, por meio de um procedimento recursivo, todas as decomposições $(a_{i*}, \dots, a_{i**}, \dots)$ de s_i ($i = 1, \dots, m$) as quais satisfazem $s_i = a_{i*}x_* + a_{i**}x_{**} + \dots$ podem ser analisadas e os correspondentes padrões de corte viáveis, determinados.

O tempo computacional para a determinação de decomposições inteiras e a identificação de padrões de corte viáveis aumenta rapidamente com o número de padrões originais a serem combinados. Portanto, em sua implementação do KOMBI, [Foerster e Wascher \(1999\)](#) limitaram as combinações a no máximo quatro padrões e somente reduções do tipo p para (p-1) foram consideradas. O pseudocódigo do algoritmo é apresentado a seguir:

Procedimento KOMBI1234

INÍCIO

Enquanto combinações 2 para 1 são encontradas **faça**

para todos os pares de padrões (j_1, j_2) **faça**

combinação 2 para 1

fim

fim

Enquanto combinações 3 para 2 são encontradas **faça**

para todas as triplas de padrões (j_1, j_2, j_3) **faça**

combinação 3 para 2

fim

fim

Enquanto combinações 4 para 3 são encontradas **faça**

para todas as quádruplas de padrões (j_1, j_2, j_3, j_4) **faça**

combinação 4 para 3

fim

fim

FIM

[Walker \(1976\)](#) apresentou um algoritmo para encontrar uma solução aproximada para o Problema da Carga Fixa, considerando uma modificação no método simplex para a regra de seleção da variável a entrar na base. O método é aplicado na resolução do problema de corte de estoque, com custos associados a troca de padrões e, segundo relata, resultaram em soluções com 20% menos padrões que aqueles obtidos usando a formulação de Programação Linear proposta na literatura.

O algoritmo começa com uma solução básica inicial e um novo padrão é aceito na base se melhora o valor da função objetivo, ou seja, reduz a quantidade de objetos processados. Após encontrar a melhor solução, há um esforço da heurística em melhorá-la ainda mais, promovendo axaustivamente trocas simples e duplas entre os padrões de corte básicos e não básicos. Toda vez que uma troca é realizada com sucesso, uma nova solução com melhor valor da função objetivo é encontrada.

[Farley e Richardson \(1984\)](#) consideraram o problema da carga fixa proposto por [Walker \(1976\)](#) e aplicaram ao problema de redução do número de diferentes padrões no problema de corte de estoque. Lembrando que para qualquer problema de programação linear com m restrições e n variáveis de decisão, o número total de

variáveis de folga mais as variáveis de decisão da base é sempre m . Eles sugeriram um procedimento no qual ao invés de buscar uma solução de custo mínimo, explora o *trade-off* entre o número de objetos e o número de diferentes padrões. O método inicia com uma solução inicial obtida por meio de programação linear e busca reduzir o número de variáveis básicas com relação aos padrões de corte através de sucessivas iterações simplex. O processo promove as trocas entre as variáveis estruturais e as variáveis de folga.

Dikili et al. (2007) aplicaram o problema de corte de estoque unidimensional à construção naval e propuseram um método de resolução para o problema considerando dois objetivos: (1) a minimização do desperdício e (2) a minimização do número de diferentes padrões. Eles estabeleceram um conjunto de busca, formado por padrões viáveis para o problema de corte e desenvolveram um método de eliminação sucessiva de padrões, em que planos de corte são determinados diretamente sem a necessidade de estabelecer um modelo matemático. O conjunto de padrões para iniciar o processo de eliminação, é obtido por métodos analíticos (apenas considerando-se a viabilidade dos padrões gerados). Do conjunto inicial, um padrão é selecionado, considerando-se algumas prioridades (descritas a seguir). Após determinar a frequência e atualizar a demanda dos itens, o padrão escolhido, juntamente com aqueles padrões que produzem excesso de algum item são excluídos do conjunto de busca e o processo é repetido até que uma solução seja encontrada. Para isso, inicialmente os itens são reordenados em ordem decrescente de tamanho e um conjunto de K padrões viáveis subdividido em m diferentes subconjuntos é construído seguindo a seguinte metodologia: no primeiro subconjunto de padrões o item 1 deve, sempre que possível estar presente, no segundo, o item 2 e, assim sucessivamente. A quantidade do item i no padrão de corte j ($a_{j,i}$) pode ser calculada usando a seguinte relação recursiva:

$$a_{j,i+1} = \left\lfloor (L - \sum_{p=1}^i a_{j,p} l_p) / l_{i+1} \right\rfloor, \\ a_{1,1} = \lfloor L / l_1 \rfloor. \quad (3.18)$$

os valores de $Waste_j$, o desperdício gerado pelo padrão j , SM_j , a frequência do padrão j e PU_j , o total de itens cortados pelo padrão j podem ser calculados usando

as seguintes relações:

$$\begin{aligned}Waste_j &= L - \sum_{i=1}^m a_{j,i} l_i \\SM_j &= \min(\lfloor d_i/a_{j,i} \rfloor) \\PU_j &= SM_j \sum_{i=1}^m a_{j,i}\end{aligned}\tag{3.19}$$

A Heurística completa é a seguinte:

HEURÍSTICA

INÍCIO

Entrada: problema de corte de estoque unidimensional

Saída: solução para o problema

Passo 1: ordene os itens em ordem decrescente de tamanho: para $i = 1, \dots, m$,
admite-se sem perda de generalidade $l_i \geq l_{i+1}$

Passo 2: Defina o conjunto de busca formado por padrões viáveis, conforme descrito anteriormente.

Passo 2.1: Para cada padrão determine o desperdício gerado, a frequência máxima possível sem que haja excesso na demanda dos itens e a quantidade total de itens cortados.

Passo 2.2: Determine o melhor padrão de acordo com as seguintes prioridades:

- (i) Menor desperdício, $Min(Waste_j)$
- (ii) Maior frequência, $Max(SM_j)$
- (iii) Menor quantidade de itens cortados, $Min(PU_j)$

Passo 2.3: Adicione o padrão à solução do problema

Passo 2.4: Atualize a demanda para o próximo procedimento de eliminação

Passo 2.5: Atualize o conjunto de busca, eliminando todos os padrões com excesso

na produção dos itens

Passo 3: **Se** toda demanda foi atendida **então** estabeleça a solução do problema

Senão

volte para o *Passo 2.2*

FIM

Segundo [Dikili et al. \(2007\)](#) uma das vantagens do método proposto é a garantia de uma solução inteira ao final do processo, mas a necessidade de enumerar um número de padrões iniciais para iniciar o processo de eliminação e seleção dos padrões da solução, torna-o computacionalmente custoso.

[SallesNeto \(2005\)](#) apresentou um novo método para minimizar o número de objetos cortados e o número de padrões distintos num problema de corte unidimensional. Ele considerou o seguinte modelo matemático:

$$\text{Min } c_1 \sum_{j=1}^n x_j + c_2 \sum_{j=1}^n \delta(x_j) \quad (3.20)$$

$$\sum_{j=1}^n a_{ij}x_j \geq d_i \quad i = 1, \dots, m. \quad (3.21)$$

$$x_j \in N \quad j = 1, \dots, n. \quad (3.22)$$

$$\delta(x_j) = \left\{ \begin{array}{l} 1 \text{ se } x_j > 0, \\ 0 \text{ se } x_j = 0. \end{array} \right\}. \quad (3.23)$$

em que:

$a_{ij} \geq 0$ é um parâmetro que indica a quantidade de itens do tipo i produzida pelo padrão j , $i = 1, \dots, m$, $j = 1, \dots, n$;

c_1 é o custo de cada objeto em estoque;

c_2 o custo de *setup*.

A dificuldade em se resolver este problema resulta da descontinuidade da função objetivo. Não se pode relaxar as condições de integralidade da solução e usar o Método de Geração de Colunas, pois não temos linearidade e continuidade na função objetivo. Técnicas de programação não linear encontradas na literatura, aplicadas a funções diferenciáveis que aproximam funções descontínuas foram utilizadas. Para

gerar uma solução inicial para o problema, SallesNeto (2005) utilizou a heurística SHP, desenvolvida por Haessler (1975) e posteriormente adaptou o método de geração de colunas desenvolvido por Gilmore e Gomory (1961) para resolver o modelo não-linear considerado, utilizando heurísticas sugeridas na literatura.

Umetani et al. (2006) melhoraram a heurística proposta em Umetani et al. (2003), incorporando ao modelo (3.5) o número máximo de diferentes padrões de corte podendo este valor ser menor, e denominaram este problema de PRP (Pattern Restricted Problem). Nesta nova heurística a busca por uma melhor solução é feita por dois procedimentos diferentes de busca local: o *1-add neighborhood* e o *shift neighborhood*. O *1-add neighborhood* é definido como sendo o conjunto de soluções obtidas por aumento do número de um item tipo i e diminuição de outros itens do tipo j , $i \neq j$ de algum padrão de corte na solução para $i = 1, \dots, m$, $j = 1, \dots, m$. O *shift neighborhood* é definido como sendo o conjunto de soluções obtidas por substituição de um item de um padrão de corte por algum outro item em um outro padrão, de modo a melhorar a eficiência dos padrões. Segundo os autores, como o tamanho da vizinhança é importante para a eficiência da busca local, técnicas de PL são utilizadas para tentar reduzir o tamanho da vizinhança.

No geral, a heurística consiste em repetir o algoritmo de busca local para diferentes soluções iniciais e tentar encontrar uma solução melhor. Escolhendo-se o procedimento de busca, da solução inicial do problema de corte, tem-se um conjunto com n -padrões distintos, $\Pi = \{p_1, p_2, \dots, p_n\}$ e suas respectivas frequências $X = \{x_1, x_2, \dots, x_n\}$. O algoritmo de busca local substitui o conjunto de padrões corrente Π por um conjunto melhor de padrões Π^* na vizinhança $NB(\Pi)$, ou seja, um conjunto com menos padrões, ou com o mesmo número padrões, mas com perdas menores. Para cada conjunto de padrões Π calcula-se o valor da função objetivo $f(\Pi, X)$. Se $f(\Pi^*, X^*) < f(\Pi, X)$, o algoritmo de busca local substitui então o conjunto de padrões (Π, X) pelo conjunto (Π^*, X^*) , em que $X^* = (x_{1^*}, x_{2^*}, \dots, x_{n^*})$ é a frequência do novo conjunto de padrões Π^* .

Para determinar a frequência dos padrões e avaliar a qualidade das soluções encontradas, resolve-se heurísticamente um problema auxiliar de programação inteira (PI). Para reduzir o número de soluções candidatas na vizinhança e acelerar a resolução dos problemas auxiliares, algumas técnicas de programação linear são utilizadas. Segundo Umetani et al. (2006), o algoritmo de busca local obtém soluções de melhor qualidade que as obtidas por outros métodos da literatura.

Alves et al. (2009) estabeleceram limitantes inferiores para o PMP (Pattern Minimization Problem) no PCEU explorando diferentes modelos de programação inteira (PI). São propostas estratégias diferentes para melhorar os limitantes inferiores para o problema baseadas em programação linear (PL). Um limitante trivial utilizado para o PMP é obtido resolvendo um problema de Bin Packing com o conjunto de itens, cada um deles com demanda 1. Denotamos por \underline{z} este limitante a ser melhorado. Uma proposta de melhoramento é aumentar o limitante de uma em uma unidade (fazendo $\underline{z} = \underline{z} + 1$) e utilizar técnicas de programação linear para encontrar uma solução viável para o problema. Segundo Alves et al. (2009), os testes computacionais realizados utilizando instâncias da literatura, mostram uma clara melhora na qualidade dos limitantes para o PMP.

4 NOVAS HEURÍSTICAS PARA O PROBLEMA DE REDUÇÃO DE PADRÕES DE CORTE

Neste capítulo, apresentamos novas heurísticas de redução de padrões para o problema de corte de estoque unidimensional. As 3 primeiras heurísticas são compostas de 3 fases e diferem entre si apenas na fase 1. A quarta heurística utiliza idéias da heurística de [Cui et al. \(2008\)](#), combinada com o procedimento KOMBI de redução de padrões.

4.1 Heurística 1

Como já mencionamos, a heurística 1 é composta de três fases:

Na primeira fase geram-se padrões levando-se em consideração a demanda média dos itens remanescentes. Apenas itens cujas demandas são superiores a esta média podem aparecer uma ou mais de uma vez no padrão. Além disso, um padrão é construído valorizando os itens maiores, começando com itens maiores que $L/2$ e, caso o mesmo não seja aceito como parte da solução por não apresentar os níveis toleráveis de desperdício, estes valores são reduzidos, permitindo que sejam colocados itens menores de modo a favorecer a aceitação de um padrão de corte gerado.

A frequência do padrão é determinada de maneira que o mesmo possa ser utilizado o máximo possível sem que nenhum item tenha sua demanda ultrapassada. Nesta primeira fase do algoritmo, o processo é repetido até que os padrões de corte gerados não sejam mais aceitos por não apresentarem o nível aceitável de desperdício estabelecido previamente. Na segunda fase, o problema residual (problema com itens cujas demandas não foram ainda atendidas) é resolvido e, na terceira fase, aplica-se a técnica de redução de padrões proposta por [Foerster e Wascher \(1999\)](#) à solução obtida. A seguir as três fases da heurística 1 são apresentadas em detalhes.

4.1.1 Fase 1: Processo Inicial de Geração de Padrões

Seja $d_M = (\sum_{i \in I_K} d_i)/K$ a demanda média dos itens com demandas não-nulas em que K é a quantidade desses itens e I_k seu conjunto de índices com $I_K \subset I$, $|I_K| = K$ e $I = \{1, \dots, m\}$. Inicialmente $K = m$. Um item i é candidato a estar no próximo padrão se $l_i > L/\tau$, iniciando-se com um valor para τ igual a 2. Caso o valor corrente de τ leve à geração de um padrão que não satisfaça os níveis de aspiração para o desperdício, seu valor é atualizado para $\tau = \tau + 1$ e um novo padrão de corte é

determinado, com maior possibilidade de ser aceito, visto que quanto maior o valor de τ , maior deve ser o número de itens candidatos.

Almejamos uma solução com um número reduzido de padrões, assim, impomos um limitante para a quantidade máxima de um mesmo item tipo i permitida no padrão calculada através da relação $q_i = \lceil d_i/d_M \rceil \forall i \in I_K, l_i > \frac{L}{\tau}$ com q_i igual a zero para os demais itens. Para que os itens maiores que $L/2$ sejam preferidos em comparação aos menores na determinação do padrão, define-se o valor de utilidade do item i da seguinte forma:

$$v_i = \begin{cases} l_i, & \text{se } l_i \leq L/2, \quad i = 1, \dots, m \\ l_i^2, & \text{caso contrário.} \end{cases} \quad (4.1)$$

Sem perda de generalidade, admite-se que $l_i \geq 1$ para todo i . Se $l_i < 1$, para algum i uma mudança de escala pode ser realizada para que todos os l_i sejam maiores ou iguais a 1.

O padrão de corte é obtido resolvendo o seguinte problema da mochila limitada:

$$\text{Maximize } \left\{ \sum_{i=1}^m v_i y_i \right\} \quad (4.2)$$

$$\text{Sujeito a : } \sum_{i=1}^m l_i y_i \leq L \quad (4.3)$$

$$0 \leq y_i \leq \min\{q_i, L/l_i\} \quad y_i \in N \quad i = 1, \dots, m \quad (4.4)$$

em que N é o conjunto dos números naturais.

Na heurística proposta tem-se este cuidado especial para itens maiores que $L/2$ pois itens deste tamanho aparecem necessariamente uma única vez nos padrões que os contêm e, geralmente, são difíceis de serem combinados com os demais itens na formação de padrões com perdas pequenas. Para tentar manter um número reduzido de padrões e gerar bons padrões que os contêm é interessante considerar estes itens no início, quando ainda se tem um número maior de itens a combinar.

Após gerar o padrão, o mesmo será aceito se o seu desperdício é limitado, ou seja, se $\Delta L = (L - \sum_{i \in I_K} l_i y_i) \leq \mu L$ em que μ é uma taxa de desperdício tolerável para o problema, previamente estabelecida. Caso o padrão gerado seja aceito, sua frequência é dada por $f_P = \text{Min} [d_i/y_i]$ com $0 < y_i \leq d_i, \forall i \in I_K$, em que y_i é a quantidade de

itens do tipo i fornecida pelo padrão.

Dada a restrição imposta ao tamanho dos itens, este processo de geração de padrões pode levar a sobras reaproveitáveis do objeto cortado. Considera-se *sobra* todo pedaço cortado que não seja um item demandado (ver [Cherri e Arenales \(37.,2005\)](#)) e um pedaço cortado de comprimento suficientemente grande para ser reaproveitado é chamado *retalho*. Veja que o processo de seleção de itens adotado para a primeira fase da heurística, com prioridade para os itens maiores, pode deixar de fora do padrão itens cujos comprimentos são menores ou iguais a um *retalho*. Isto pode ser verificado após a resolução do problema da mochila (4.2)-(4.4) que determina o padrão de corte.

Enquanto existirem itens com demandas maiores ou iguais à frequência do padrão corrente, ou seja, $d_i \geq f_p$ para algum $i \in I_k$ tal que $\Delta L \geq l_i$ e $l_i < \frac{L}{\tau}$, então para completar o padrão corrente faz-se $y_i = 1$ e $\Delta L = \Delta L - l_i$. Finalmente, a demanda dos itens é atualizada fazendo-se $d_i = d_i - f_p y_i$, $i = 1, \dots, m$, e o processo de geração de padrões é repetido enquanto existirem itens com demandas não nulas, ou padrões com desperdício limitado possam ser determinados.

4.1.2 Fase 2: Resolução do Problema Residual

Nesta fase resolvemos um problema residual para completar as demandas dos itens que não foram completamente atendidas na Fase 1. O processo de geração de padrões nesta fase é baseado nas idéias de [Cui et al. \(2008\)](#) com algumas modificações. Um conjunto de itens é selecionado previamente, antes de se determinar o padrão gerado, mas ao invés de utilizarmos dois níveis de aspiração por padrão como eles fizeram, para o número mínimo de diferentes itens e para o desperdício, optamos apenas pelo segundo. A taxa tolerável de desperdício é obtida a partir da perda gerada pela solução do problema de corte aplicando-se o procedimento sugerido por [Gilmore e Gomory \(1961\)](#).

A solução inteira para o problema de corte é obtida arredondando para baixo as variáveis não inteiras da solução do problema relaxado (veja [Subseção 2.3.3](#)) e o problema residual é resolvido com a heurística FFD. O plano de corte encontrado fornece uma quantidade de objetos N_{obj} de uma solução referência para o problema, correspondendo à taxa μ de desperdício.

Para tentar gerar padrões de qualidade, com o nível de aspiração estabelecido, a

frequência do próximo padrão é determinada a partir da maior demanda existente para o problema residual. Seja dr_i a demanda residual do item i e $D = \text{Max} \{dr_i\}$ para $i = 1, \dots, m$. Determinamos o maior valor inteiro k , $k \in [1, \dots, D]$ tal que $(L - \sum_{i=1}^m l_i q_i) \leq \mu L$ em que $q_i = \lfloor dr_i/k \rfloor$ é a quantidade máxima de itens tipo i permitida em cada padrão. Resolvemos o problema da mochila limitada, e o padrão $P = [y_1, \dots, y_m]$ gerado é aceito. Sua frequência é dada por $f_p = k$. Este processo é repetido até terminar com toda demanda.

Diferentes porcentagens de perda μ são utilizadas como níveis de aspiração para o desperdício, variando-se para mais e para menos a quantidade de objetos N_{obj} obtida. São considerados alguns valores diferentes, variando-se o número de objetos no intervalo $[[0, 95N_{obj}], [1, 05N_{obj}]]$, em que $[a, b]$ denota o intervalo fechado de extremos a, b . Para minimizar o esforço computacional na aplicação da heurística, se N_{obj} for pequeno ($N_{obj} \leq 5$), estes valores são tomados de um em um dentro do intervalo, mas se N_{obj} for grande são tomados no máximo 11 valores neste intervalo, ou seja, se $[0, 05N_{obj}]$ for maior do que 5, considera-se um número de objetos neste intervalo espaçados de aproximadamente $\Delta s = \lfloor 0,05N_{obj}/5 \rfloor$, caso contrário $\Delta s = 1$. Ao considerarmos estes diferentes valores do número de objetos na heurística, diferentes soluções são geradas. A melhor delas é guardada.

4.1.3 Fase 3: Aplicação de um método de redução de padrões

Nesta fase, aplicamos o método KOMBI de redução de padrões sugerido por [Foerster e Wascher \(1999\)](#) à solução corrente. A Heurística 1 completa pode ser esquematizada como segue:

HEURÍSTICA 1

INÍCIO

Entrada: Problema de Corte de Estoque Unidimensional

Saída: solução do problema de corte com número reduzido de padrões

(Primeira Fase)

Resolver o Problema de Corte e determinar N_{obj} , a quantidade de objetos de referência para o desperdício.

Faça: $Per_{otm} = 1$ (a perda da melhor solução) e $Pad_{otm} = M$ (o número de padrões da melhor solução) (M é um número grande)

Se $[0, 05N_{obj}] > 3$ **então** $\Delta s = [0, 05N_{obj}/5]$, **senão** $\Delta s = 1$

Passo 1: para $\beta = - [0, 05N_{obj}], \dots, + [0, 05N_{obj}]$, com passo Δs faça

$$\mu = \frac{(N_{obj} + \beta)L - \sum_{i=1}^m l_i d_i}{(N_{obj} + \beta)L} \text{ (percentual de referência para a perda máxima)}$$

Faça: $dr_i = d_i$ (dr_i é a demanda residual do item $i, i = 1, \dots, m$) e $I_k = I$

Faça PARE=Falso

Faça SAIR=Falso

Passo 2: **Enquanto** (PARE=Falso e SAIR=Falso) faça $\tau = 2$:

Compute K para os itens $l_i \forall i \in I_K$. Inicie com $K = m$

Compute a demanda média $d_m = (\sum_{i \in I_K} dr_i) / K$.

Passo 3: **Se** $\tau \geq 20$ **então** SAIR=Verdadeiro

Faça $q_i = 0$ para $i = 1, \dots, m$.

Para os itens l_i tal que $l_i > L/\tau, \forall i \in I_K$, faça $q_i = \lceil dr_i/d_m \rceil$

Gere o padrão de corte resolvendo o problema da mochila limitada

Se o padrão obedece ao nível de aspiração **então** aceite-o,

senão faça $\tau = \tau + 1$ e volte ao *Passo 3*.

Determine a frequência máxima $f_P = \text{Min} \lceil dr_i/y_i \rceil$ sem que ocorra superprodução de nenhum item, em que y_i é a quantidade de itens tipo i no padrão $\forall i \in I_K$.

Faça $\Delta L = (L - \sum_{i \in I_K} l_i y_i)$.

Para $i \in I_K$ tal que $l_i < \frac{L}{\tau}, d_i \geq f_P, \Delta L \geq l_i$, faça $y_i = 1$ e $\Delta L = \Delta L - l_i$

Atualize a demanda fazendo: $dr_i = dr_i - f_P y_i, \forall i \in I_k$.

Se $dr_i = 0, i = 1, \dots, m$ então PARE=Verdadeiro

(Fim do Enquanto)

Segunda Fase

Passo 4: Se SAIR=Verdadeiro **então**

Compute $D = \text{Max} \{dr_i\} \forall i \in I_k$

Determine o maior valor inteiro $k, k \in [1, \dots, D]$ tal que $(L - \sum_{i \in I_k} l_i q_i) \leq \mu L$
e $q_i = \lfloor dr_i/k \rfloor$. Se k não existe, faça $q_i = dr_i, \forall i \in I_k$

Determine o padrão $P = [y_1, \dots, y_m]$, resolvendo o problema da mochila limitada e aceite-o.

Atualize a demanda dos itens

Passo 5: **Se** toda demanda foi atendida **então**

Estabeleça a solução do problema original.

(Terceira Fase)

Aplique o método KOMBI de redução de padrões.

Compute a perda (Per) e o número de padrões (Pad) da solução corrente

Se $Per \leq Per_{otm}$ e $Pad \leq Pad_{otm}$ **então**

$Per_{otm} = Per, Pad_{otm} = Pad$ e o parâmetro corrente β fornece a melhor solução para o problema

Atualize β e volte ao *Passo 1*.

Senão volte ao *Passo 4*

FIM

τ é um parâmetro que controla o tamanho do itens a serem considerados para a

geração do padrão e assume no máximo o valor 20, pois itens pequenos (menores que 5% do tamanho do objeto em estoque) e com demandas pequenas são geralmente mais fáceis de serem arranjados nos padrões. A seguir apresenta-se um exemplo em que é aplicada a heurística 1. São considerados 10 tipos de itens ($m = 10$) e tamanho do objeto em estoque $L = 1000$. Os tamanhos e as demandas dos itens são mostrados na [Tabela 4.1](#)

Tabela 4.1 - Exemplo 1: dados de um problema de corte

Item	Comprimento	Demanda
1	164	126
2	158	129
3	139	170
4	135	168
5	125	108
6	114	40
7	108	49
8	74	63
9	64	95
10	12	52

Resolvendo o problema de corte através do Método Simplex com Geração de Colunas com o pacote CPLEX obtém-se uma solução com 10 padrões distintos e 122,074 objetos. Portanto, um limitante inferior para o PCEU é de 123 objetos. Para determinar uma solução inteira, arredondam-se para baixo as componentes não inteiras da solução relaxada (veja [Subseção 2.3.3](#)) e utiliza-se a heurística FFD para completar a demanda residual. Com isso obtém-se uma solução inteira com 123 objetos e 15 padrões distintos (dados na [Tabela 4.2](#)). O gap de otimalidade neste caso é zero, ou seja, esta solução é ótima.

A porcentagem de desperdício desta solução é $\alpha = \frac{N_{obj}L - \sum_{i=1}^m l_i d_i}{N_{obj}L} = \frac{123000 - 122074}{123000} = 0,0075 = 0,75\%$. Ao resolver este exemplo aplicando a Heurística 1, tem-se que o tamanho do passo é $\Delta s = \lfloor 0,05N_{obj}/5 \rfloor = \lfloor 6,15/5 \rfloor = 1$, e β , a quantidade de objetos a ser acrescentada, varia entre -6 e $+6$. As quantidades de objetos testadas têm seus valores no intervalo $\lfloor \lfloor 0,95N_{obj} \rfloor, \lfloor 1,05N_{obj} \rfloor \rfloor = \lfloor \lfloor 116,775 \rfloor, \lfloor 129,15 \rfloor \rfloor = \lfloor 116, 129 \rfloor$.

A solução obtida com a heurística fornece 123 objetos e 7 padrões distintos e é

Tabela 4.2 - Solução Inteira do Problema de Corte

Frequência	Padrões									
12	0	0	0	0	8	0	0	0	0	0
2	0	0	0	0	0	8	0	0	1	2
18	0	4	0	2	0	0	0	1	0	2
3	0	0	0	0	0	0	0	4	11	0
10	0	0	4	0	0	0	4	0	0	1
5	0	4	0	1	1	0	1	0	0	0
31	4	0	0	2	0	0	0	1	0	0
20	0	0	3	3	0	1	0	0	1	0
16	0	2	4	0	0	0	0	0	2	0
1	2	4	0	0	0	0	0	0	0	2
1	0	1	6	0	0	0	0	0	0	0
1	0	0	0	5	2	0	0	1	0	0
1	0	0	0	0	5	3	0	0	0	0
1	0	0	0	0	0	1	4	1	5	0
1	0	0	0	0	0	0	0	0	3	0

mostrada na [Tabela 4.3](#). A aplicação do procedimento KOMBI conseguiu reduzir o número de padrões na solução de 7 para 5 como mostrado na [Tabela 4.4](#). A melhor solução ocorre quando $\beta = 0$.

Tabela 4.3 - Solução da Heurística

Frequência	Padrão									
54	1	2	0	2	2	0	0	0	0	0
36	2	0	3	1	0	1	0	0	0	0
4	0	1	2	1	0	1	2	1	0	1
17	0	1	2	0	0	0	1	2	2	1
10	0	0	2	0	0	0	2	2	5	3
1	0	0	0	3	0	0	4	2	0	1
1	0	0	0	0	0	0	0	3	11	0

Tabela 4.4 - Solução da Heurística Após Aplicar Método KOMBI

Frequência	Padrão									
54	1	2	0	2	2	0	0	0	0	0
36	2	0	3	1	0	1	0	0	0	0
10	0	0	2	2	0	0	2	0	3	1
21	0	1	2	0	0	0	1	3	3	2
2	0	0	0	2	0	2	4	0	1	0

4.2 Heurística 2

A heurística 2 é também composta de três fases, mas a metodologia aplicada no procedimento de geração de padrões da primeira fase, difere em relação ao que foi desenvolvido na heurística 1. Na heurística 2 os padrões são gerados a partir de uma seleção feita do agrupamento dos itens em dois conjuntos disjuntos de acordo com suas demandas. Este procedimento permite gerar padrões que contenham itens cujas demandas estejam próximas, possibilitando assim, produzir uma quantidade "razoável" destes itens, favorecendo a busca por soluções que apresentem um número reduzido de padrões distintos. Para o problema residual, adotamos o mesmo procedimento aplicado à heurística 1.

A heurística inicia determinando (D_{max}) e (d_{min}) a maior e menor demanda, respectivamente, do problema de corte de estoque original, os quais são mantidos constantes até o final do processo. Isto permite definir dois intervalos disjuntos $I_1 = [d_{min}, \lfloor D_{max}/2 \rfloor]$ e $I_2 = (\lfloor D_{max}/2 \rfloor, D_{max}]$ ou $I_1 = [\lfloor D_{max}/2 \rfloor, d_{min}]$ e $I_2 = (d_{min}, D_{max}]$ se $d_{min} > D_{max}/2$. Se (d_{min}) < 3 então faça (d_{min}) = 3, com isso, demandas menores que 3 serão atendidas através da resolução do problema residual.

Seja $dr_i > 0$ a demanda residual do item i e r_d a menor das demandas residuais, iniciando com $dr_i = d_i$ para $i = 1, \dots, m$. Inicialmente, somente itens com demandas grandes são consideradas, ou seja, no começo serão gerados padrões com itens i tal que $dr_i \in I_2$. Se nenhum padrão com demandas dos itens no intervalo I_2 é mais aceito, o processo de geração de padrões é interrompido, passando-se então para o conjunto I_1 . Nesta fase os padrões gerados são aceitos se obedecem os níveis de aspiração para o desperdício e cada vez que um é aceito, a demanda residual dos itens é atualizada e o processo de geração de padrões é repetido. Caso padrões com desperdício limitado não possam mais ser gerados, o processo de geração de padrões é interrompido e um problema residual é resolvido para completar a demanda dos

itens. Neste caso todo padrão gerado é aceito, ou seja, nenhum nível de aspiração é considerado.

O nível de aspiração para a aceitação de um padrão na solução é determinado como na heurística 1. O plano de corte encontrado fornece a taxa de desperdício α , referência para o problema. O nível de aspiração para a eficiência do padrão é $naep = 1 - \mu$. São testados três diferentes valores para μ : $\mu = \alpha - 0.05\alpha$, $\mu = \alpha$ e $\mu = \alpha + 0.05\alpha$. Estes valores representam uma variação com relação à perda gerada ao se resolver o problema de corte através do CPLEX de -5% , 0% e $+5\%$, respectivamente. A solução que apresenta menor média de desperdício e menor número médio de padrões é mantida. O pseudocódigo da heurística 2 é apresentado a seguir.

HEURÍSTICA 2

INÍCIO

Entrada: Problema de Corte de Estoque Unidimensional

Saída: solução do problema de corte com número reduzido de padrões

(Primeira fase)

Faça $dr_i = d_i$ para $i = 1, \dots, m$

Passo 1: resolva o problema de corte usando o pacote CPLEX e determine o desperdício médio (α) da solução inteira gerada.

Passo 2: compute a maior (D_{max}) e menor (d_{min}) demandas do problema original.
se $(d_{min}) < 3$ **então** faça $(d_{min}) = 3$.

Passo 3: defina os intervalos de demanda $I_1 = [d_{min}, \lfloor D_{max}/2 \rfloor]$ e
 $I_2 = (\lfloor D_{max}/2 \rfloor, D_{max}]$

Passo 4: para $\mu = \alpha - 0.05\alpha$, $\mu = \alpha$, $\mu = \alpha + 0.05\alpha$, faça $naep = 1 - \mu$.
(considere o problema original)

Faça: $Per_{otm} = 1$ (a perda da melhor solução) e $Pad_{otm} = M$
o número de padrões da melhor solução) (M é um número grande)

Passo 5: **se** $dr_i > 3 \forall i = 1, \dots, m$ compute a menor demanda residual r_d **senão**
passe para o *Passo 13*

Passo 6: para $i = 1, \dots, m$, **se** $dr_i \in I_2$ compute a quantidade máxima de itens permitida por padrão: $q_i = \lfloor dr_i/r_d \rfloor$ **senão** $q_i = 0$

Passo 7: resolva o problema da mochila limitada com $i \in I_2$ e obtenha o padrão corrente: $P_j = [y_1, \dots, y_m]$

Passo 8: **se** (padrão satisfaz nível de aspiração: $\sum_{i=1}^m y_i l_i \geq L \cdot naep$) **então** determine a frequência $f_p = \min \lfloor dr_i/y_i \rfloor$, $dr_i \geq y_i > 0$. Atualize as demandas e retorne ao *Passo 5*

Passo 9: para $i = 1, \dots, m$, **se** $dr_i \in I_1$ compute a quantidade máxima de itens permitida por padrão: $q_i = \lfloor dr_i/r_d \rfloor$ **senão** $q_i = 0$.

Passo 10: resolva o problema da mochila limitada com $i \in I_1$ e obtenha o padrão corrente: $P_j = [y_1, \dots, y_m]$

Passo 11: **se** (padrão satisfaz nível de aspiração) **então** determine a frequência e atualize as demandas dos itens.

Passo 12: **se** $dr_i > 3 \forall i = 1, \dots, m$ determine $r_d = \min \{dr_i\}$, $i = 1, \dots, m$, e retorne ao *Passo 9* **senão** vá para o *Passo 13*.

(Segunda fase)

Passo 13: **se** ($dr_i > 0$) para algum $i = 1, \dots, m$ **então** chame o procedimento para resolver o problema residual **senão** vá para o *Passo 14*

(Terceira fase)

Passo 14: estabeleça a solução do problema

Passo 15: Aplique a técnica de redução de padrões (KOMBI) à solução. Compute a perda (Per) e o número de padrões (Pad) da solução corrente.

se $Per \leq Per_{otm}$ e $Pad \leq Pad_{otm}$ **então**

$Per_{otm} = Per$, $Pad_{otm} = Pad$ e o parâmetro corrente μ fornece a melhor solução para o problema

Atualize μ e volte ao *Passo 4*.

FIM

A seguir apresenta-se um exemplo com aplicação da heurística 2. Os dados do exemplo (exemplo 2) são mostrados na [Tabela 4.5](#), para $m = 10$ e $L = 1000$. Resolvendo este problema de corte com o Método Simplex com Geração de Colunas, são utilizados 49,5 objetos (o limitante inferior é de 50 objetos). Aplicando o método de arredondamento com solução do problema residual utilizando FFD, obém-se 50 objetos com 11 padrões distintos na solução (dados na [Tabela 4.6](#)). O gap de otimalidade neste caso também é zero.

Tabela 4.5 - Exemplo 2: dados do problema de corte

Item	Comprimento	Demanda
1	720	10
2	619	7
3	563	13
4	519	11
5	462	11
6	456	9
7	432	7
8	402	9
9	337	12
10	82	11

Tabela 4.6 - Solução do Problema de Corte

Frequência	Padrão									
8	1	0	0	0	0	0	0	0	0	0
5	0	0	0	0	2	0	0	0	0	0
3	0	0	0	0	0	2	0	0	0	0
5	0	0	1	0	0	0	0	0	1	1
1	0	0	0	1	0	0	0	1	0	0
7	0	0	0	1	0	0	1	0	0	0
8	0	0	1	0	0	0	0	1	0	0
7	0	1	0	0	0	0	0	0	1	0
3	0	0	0	1	0	1	0	0	0	0
2	1	0	0	0	0	0	0	0	0	3
1	0	0	0	0	1	0	0	0	0	0

A maior demanda do problema original é $D_{max} = 13$ e a menor $d_{min} = 7$, os intervalos de busca são $I_1 = [6, 7]$ e $I_2 = (7, 13]$. Na busca por padrões, cujos itens tenham suas demandas no intervalo $I_2 = (7, 13]$, dois foram aceitos: $P_1 = [0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0]$ cortado 9 vezes e $P_2 = [0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0]$ também cortado 9 vezes. Outros três padrões foram obtidos para os itens cujas demandas encontram-se no intervalo $I_1 = [6, 7]$: $P_3 = [0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1]$ cortado 6 vezes, $P_4 = [0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0]$ cortado também 6 vezes, e $P_5 = [0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0]$ cortado 7 vezes.

Como o corte destes 5 padrões não atende toda a demanda, um problema residual, cujos dados são mostrados na [Tabela 4.7](#) é resolvido. A solução do problema original, com os padrões gerados na primeira fase da heurística e os padrões obtidos através da resolução do problema residual é mostrada na [Tabela 4.8](#). Para esta solução foram utilizados 9 padrões distintos e 50 objetos.

Tabela 4.7 - Dados do problema residual obtido do exemplo 2

Item	Comprimento	Demanda
1	720	10
2	619	1
3	563	0
4	519	2
5	462	2
6	456	0
7	432	0
8	402	0
9	337	0
10	82	5

Tentamos reduzir ainda mais o número de padrões encontrado utilizando o KOMBI (fase 3 da heurística). Este procedimento aplicado à solução obtida, reduziu o número de padrões de 9 para 7, como mostrado na [Tabela 4.9](#). Comparando a solução final obtida pela heurística 2 com a obtida através do procedimento de [Gilmore e Gomory \(1961\)](#) e arredondamento, nota-se uma redução no número de padrões distintos de 11 para 7, sem que nenhuma variação tenha ocorrido no número de objetos cortados.

Tabela 4.8 - Solução da Heurística

Frequência	Padrão									
9	0	0	0	1	0	0	0	1	0	0
9	0	0	0	0	1	1	0	0	0	0
7	0	0	1	0	0	0	1	0	0	0
6	0	0	1	0	0	0	0	0	1	1
6	0	1	0	0	0	0	0	0	1	0
5	1	0	0	0	0	0	0	0	0	1
2	0	0	0	1	1	0	0	0	0	0
5	1	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0

Tabela 4.9 - Solução da Heurística pós KOMBI

Frequência	Padrão									
9	0	0	0	0	0	1	0	1	0	0
7	0	0	1	0	0	0	1	0	0	0
6	0	0	1	0	0	0	0	0	1	0
6	0	1	0	0	0	0	0	0	1	0
11	0	0	0	1	1	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	1
10	1	0	0	0	0	0	0	0	0	1

4.3 Heurística 3

A heurística 3 é uma variante da heurística 2. Na primeira fase da heurística 2, a geração dos padrões acontece a partir da seleção dos itens cujas demandas encontram-se em dois intervalos disjuntos obtidos através da maior e menor demandas do problema original. Estes intervalos são os mesmos até o fim do procedimento e dificulta em alguns casos, o processo de seleção de itens para o padrão, devido à limitação de dois intervalos fixos, principalmente quando não se consegue mais gerar padrões que atendam os níveis de aspiração desejáveis.

Nesta variação, ao invés de considerarmos dois intervalos fixos, temos uma quantidade de intervalos variável, definida de acordo com a demanda residual dos itens a cada iteração. Isto é feito inicialmente ordenando os itens em ordem não crescente de demanda residual. Supondo sem perda de generalidade que $dr_1 \geq dr_2 \geq \dots \geq dr_m$. No início tem-se a demanda residual $dr_i = d_i$ para $i = 1, \dots, m$. Determina-se um

número inteiro k tal que $dr_{k+1} > 0$ e $\sum_{i=1}^{k+1} l_i > L$. Isto permite definir um intervalo $[dr_k, dr_1]$ a cada iteração para a seleção dos itens. A quantidade máxima permitida de cada item no padrão é obtida da relação $q_i = \lfloor dr_i/dr_k \rfloor$ para $i = 1, \dots, k$ e $q_i = 0$ caso contrário.

Para aceitar um padrão, mantemos o esquema de níveis de aspiração estabelecidos para a heurística 2, bem como a prioridade dos itens maiores sobre os menores. Os níveis de aspiração para a aceitação de um padrão na solução, são os mesmos estabelecidos para a heurística 2.

HEURÍSTICA 3

INÍCIO

Entrada: Problema de Corte de Estoque Unidimensional

Saída: solução do problema de corte com número reduzido de padrões

(Primeira fase)

Faça $dr_i = d_i$ para $i = 1, \dots, m$

Passo 1: resolva o problema de corte usando o pacote CPLEX e determine o desperdício médio (α) da solução inteira obtida.

Passo 2: para $\mu = \alpha - 0,05\alpha$, $\mu = \alpha$, $\mu = \alpha + 0,05\alpha$, faça $naep = 1 - \mu$.
(considere o problema original)

Faça: $Per_{otm} = 1$ (a perda da melhor solução) e $Pad_{otm} = M$
(o número de padrões da melhor solução) (M é um número grande)

Passo 3: reordene os itens em ordem não crescente de demanda. Sem perda de generalidade, seja $dr_1 \geq dr_2 \geq \dots \geq dr_m$.

Passo 4: determine o menor inteiro positivo k tal que $dr_{k+1} > 0$ e $\sum_{i=1}^{k+1} l_i > L$. Se k não existe, faça $k = m$

Passo 5: para $i = 1, \dots, k$, compute $q_i = \lfloor dr_i/dr_k \rfloor$, a quantidade máxima de itens tipo i no padrão, $q_i = 0$, $i = k + 1, \dots, m$.

Passo 6: resolva o problema da mochila limitada e obtenha o padrão de corte

$$P_j = [y_1, \dots, y_m].$$

Passo 7: **se** (padrão satisfaz nível de aspiração: $\sum_{i=1}^n y_i l_i \geq L \cdot naep$) **então** determine a frequência $f_p = \min \lfloor dr_i / y_i \rfloor$, $dr_i \geq y_i > 0$. Atualize as demandas e retorne ao *Passo 3* **senão** passe ao passo 8

(Segunda fase)

Passo 8: **se** ($dr_i > 0$) para algum $i = 1, \dots, m$ **então** chame o procedimento para resolver o problema residual **senão** vá para o *Passo 9*

(Terceira fase)

Passo 9: componha a solução do problema

Passo 10: aplique a técnica de redução de padrões (KOMBI) à solução.

compute a perda (Per) e o número de padrões (Pad) da solução corrente.

se $Per \leq Per_{otm}$ e $Pad \leq Pad_{otm}$ **então**

$Per_{otm} = Per$, $Pad_{otm} = Pad$ e o parâmetro corrente μ fornece a melhor solução para o problema

Atualize μ e volte ao *Passo 2*.

FIM

Ilustramos a aplicação da heurística 3 no exemplo 3 cujos dados são mostrados na [Tabela 4.10](#). Resolvendo este problema de corte através do Método Simplex com Geração de Colunas são utilizados 126,011 objetos (limitante inferior: 127 objetos). Após o método de arredondamento com resolução do problema residual via FFD, obtém-se uma solução inteira com 127 objetos e 14 padrões distintos (mostrado na [Tabela 4.11](#)). A percentagem de desperdício desta solução é $\alpha = \frac{127000 - 126011}{127000} = 0,00778 = 0,78\%$. O gap de otimalidade neste caso é também zero.

Tabela 4.10 - Exemplo 3: dados do problema de corte

Item	Comprimento	Demanda
1	200	161
2	198	140
3	179	55
4	164	6
5	157	97
6	129	120
7	105	74
8	77	119
9	45	63
10	29	165

Tabela 4.11 - Exemplo 3: Solução do Problema de Corte

Frequência	Padrão									
32	5	0	0	0	0	0	0	0	0	0
4	0	0	5	0	0	0	1	0	0	0
15	0	0	0	0	6	0	0	0	0	2
1	0	0	0	0	5	0	0	0	4	0
34	0	4	1	0	0	0	0	0	0	1
2	0	0	0	0	0	1	0	1	17	1
14	0	0	0	0	0	1	2	8	1	0
19	0	0	0	0	0	5	2	0	0	5
1	0	1	0	0	1	5	0	0	0	0
1	1	3	1	0	0	0	0	0	0	0
1	0	0	0	1	5	0	0	0	1	0
1	0	0	0	0	1	4	3	0	0	0
1	0	0	0	0	0	0	1	5	10	2
1	0	0	0	0	0	0	0	0	0	2

Resolvendo o exemplo 3 com a heurística 3, ao reordenarmos os itens em ordem não crescente de demanda residual, $k = 8$ itens são selecionados para o próximo padrão, uma vez que $\sum_{i=1}^9 l_i > L$. Portanto o item com a oitava maior demanda é $l_k = 45$ com $d_k = 63$ e o item de maior demanda é $l_1 = 29$ com $d_1 = 165$ (ver Tabela 4.10). Assim o primeiro intervalo de busca é definido por $I_1 = [dr_k, dr_1] = [63, 165]$. O padrão aceito com itens cujas demandas encontram-se neste intervalo é $P_1 = [2 \ 1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 1 \ 0 \ 2]$ cortado 74 vezes. Atualizando a demanda dos itens, temos $dr = [13 \ 66 \ 55 \ 6 \ 23 \ 120 \ 0 \ 45 \ 63 \ 17]$. Na segunda iteração, $l_k = 29$

com $d_k = 17$ e $l_1 = 129$ com $d_1 = 120$ e o padrão aceito com itens com demanda no intervalo $I_2 = [17, 120]$ é $P_2 = [0 \ 3 \ 2 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0]$, cortado 22 vezes. Repetindo o processo, outros três padrões são aceitos: $P_3 = [2 \ 0 \ 1 \ 0 \ 2 \ 0 \ 0 \ 1 \ 0 \ 1]$ cortado 6 vezes com busca no intervalo $I_3 = [6, 120]$, $P_4 = [0 \ 0 \ 2 \ 2 \ 2 \ 0 \ 0 \ 0 \ 0 \ 0]$ cortado 2 vezes com busca no intervalo $I_4 = [1, 120]$, e $P_5 = [1 \ 0 \ 0 \ 2 \ 2 \ 1 \ 0 \ 0 \ 0 \ 1]$ cortado 1 vez com busca no intervalo $I_5 = I_4$. Este padrões não completam as demandas de todos os itens do problema, e um problema residual, (dados na [Tabela 4.12](#)) deve ser resolvido.

Tabela 4.12 - Exemplo 3: dados do problema residual

Item	Comprimento	Demanda
1	200	0
2	198	0
3	179	1
4	164	0
5	157	5
6	129	119
7	105	0
8	77	39
9	45	41
10	29	10

A solução do problema original é mostrada na [Tabela 4.13](#). Para esta solução foram utilizados 8 padrões distintos e 127 objetos. A técnica de redução de padrões (KOMBI) aplicada à solução obtida, reduziu o número de padrões de 8 para 6, como pode ser observado na [Tabela 4.14](#). A solução obtida pela heurística 3 apresenta o mesmo número de objetos obtidos ao se resolver o problema de corte através do CPLEX, mas com uma redução no número de padrões distintos de 14 para 6.

Tabela 4.13 - Solução da Heurística

Frequência	Padrão									
74	2	1	0	0	1	0	1	1	0	2
22	0	3	2	0	0	0	0	0	1	0
6	2	0	1	0	2	0	0	1	0	1
2	0	0	2	2	2	0	0	0	0	0
1	1	0	0	2	2	1	0	0	0	1
19	0	0	0	0	0	6	0	2	1	0
2	0	0	0	0	1	2	0	0	11	3
1	0	0	1	0	3	1	0	1	0	4

Tabela 4.14 - Solução da Heurística pós KOMBI

Frequência	Padrão									
74	2	1	0	0	1	0	1	1	0	2
22	0	3	2	0	0	0	0	0	1	0
6	2	0	1	0	2	0	0	1	0	1
1	1	0	0	1	1	1	0	1	2	6
5	0	0	1	1	2	1	0	0	4	1
19	0	0	0	0	0	6	0	2	1	0

4.4 Heurística 4

A heurística 4 é uma variante da heurística SHPC proposta por [Cui et al. \(2008\)](#) em que diferentes frequências para os padrões são utilizadas, em ordem não crescente, iniciando-se com o valor da maior demanda. Para cada uma destas frequências, padrões são gerados se existir uma quantidade mínima de tipos diferentes de itens que podem ser incluídos nestes padrões e, se a combinação dos itens que podem ser incluídos nestes padrões supera, em comprimento, um percentual pré-fixado do tamanho do objeto. Utilizamos diferentes valores para os parâmetros de controle (n_b, r_b) ($n_b = 1, \dots, 4, r_b = 0, 6; \dots; 5, 4$, com passo $\Delta r = 0, 2$) (ver heurística SHPC na [Seção 3.1](#)), que levam a diferentes soluções para o problema de corte, não utilizamos os parâmetros α_b e β_b , em seu lugar definimos o valor de utilidade do item i , $i = 1, \dots, m$ como $v_i = l_i$ se $l_i \leq L/2$ e $v_i = l_i^2$ caso contrário. Sem perda de generalidade, admite-se que $l_i \geq 1$ para todo i . Se $l_i < 1$, para algum i uma mudança de escala pode ser realizada para que todos os l_i sejam maiores ou iguais a 1.

Finalizando, um procedimento de redução de padrões é aplicado à solução.

HEURÍSTICA 4

INÍCIO

Entrada: Problema de Corte de Estoque Unidimensional

Saída: solução do problema de corte com número reduzido de padrões

Heurística SHPC modificada

Aplique o KOMBI à melhor solução.

FIM

Ilustra-se a seguir uma aplicação da heurística 4 com os dados do exemplo 4 que são mostrados na tabela [Tabela 4.15](#). Resolvendo o problema através do Método Simplex com Geração de Colunas são utilizados 12,35 objetos (limitante inferior: 13 objetos). Com o procedimento de arredondamento obtém-se uma solução com 13 objetos e 9 padrões distintos (ver [Tabela 4.16](#)). A percentagem de desperdício desta solução é $\alpha = \frac{13000-12347}{13000} = 0,0502 = 5,02\%$. O gap de otimalidade neste caso é zero.

Tabela 4.15 - Exemplo 4: dados do problema de corte

Item	Comprimento	Demanda
1	164	13
2	158	13
3	139	17
4	135	17
5	125	11
6	114	4
7	108	5
8	74	6
9	64	10
10	12	4

Tabela 4.16 - Exemplo 4: Solução do Problema de Corte

Frequência	Padrão									
1	0	0	0	0	8	0	0	0	0	0
1	0	4	0	2	0	0	0	1	0	2
3	4	0	0	2	0	0	0	1	0	0
2	0	0	3	3	0	1	0	0	1	0
2	0	2	4	0	0	0	0	0	2	0
1	1	5	0	0	0	0	0	0	0	2
1	0	0	3	3	1	0	0	0	0	0
1	0	0	0	0	2	2	4	1	0	0
1	0	0	0	0	0	0	1	1	4	0

Aplicando-se a heurística 4, obtemos uma solução com 4 padrões distintos e 13 objetos (ver [Tabela 4.17](#)), mantendo-se a taxa percentual de desperdício de 5,02% obtida pelo CPLEX e utilizando a heurística FFD para completar a demanda residual. Esta foi a melhor solução obtida, quando os parâmetros de controle assumem os valores $(n_b, r_b) = (1; 0, 8)$, ou seja, os níveis de aspiração para a aceitação de um padrão na solução, exige utilização de pelo menos 80% do comprimento do objeto, mas não há restrições quanto a quantidade mínima de itens no padrão. O procedimento de redução de padrões (KOMBI) foi aplicado a esta solução, mas nenhuma redução foi verificada.

Tabela 4.17 - Exemplo 4: Solução da Heurística

Frequência	Padrão									
8	1	1	2	2	1	0	0	0	0	0
3	1	1	0	0	1	1	1	2	2	1
1	2	2	0	0	0	0	2	0	2	1
1	0	0	1	1	0	1	0	0	2	0

4.4.1 Exemplo Comum às 4 heurísticas

Os exemplos apresentados para ilustração das heurísticas 1, 2, 3 e 4 nas subseções anteriores, foram selecionados por conveniência, pois ilustram melhor os impactos das mudanças propostas nas heurísticas. A título de ilustração, resolvemos o exemplo 1, dados na [Tabela 4.1](#), com as heurísticas propostas. As soluções obtidas são apresentadas a seguir.

Na [Tabela 4.18](#) tem-se a solução obtida ao resolver o exemplo 1 aplicando a heurística 2. Obtemos uma solução com 124 objetos e 6 padrões distintos. Com a aplicação do KOMBI na terceira fase, nenhuma redução no número de padrões é verificada. O gap de otimalidade neste caso é 0,81%.

Tabela 4.18 - Solução do exemplo 1 com a Heurística 2

Frequência	Padrão									
84	1	1	2	2	1	0	0	0	0	0
24	1	1	0	0	1	1	2	0	3	2
10	1	2	0	0	0	1	0	5	0	0
4	2	0	0	0	0	1	0	3	5	1
1	0	1	2	0	0	2	1	1	2	0
1	0	0	0	0	0	0	0	0	1	0

Na [Tabela 4.19](#) mostra-se solução obtida ao resolver o exemplo 1 aplicando a heurística 3. Obtemos uma solução com 124 objetos e 7 padrões distintos. A aplicação do KOMBI na terceira fase, reduziu o número de padrões de 7 para 6 ([Tabela 4.20](#)). Novamente, o gap de otimalidade neste caso é de 0,81%.

Tabela 4.19 - Solução do exemplo 1 com a Heurística 3

Frequência	Padrão									
52	2	2	0	2	0	0	0	1	0	1
27	0	0	2	0	4	1	1	0	0	0
29	0	0	4	2	0	0	0	0	2	0
11	2	2	0	0	0	0	2	1	1	0
3	0	1	0	2	0	4	0	0	1	0
1	0	0	0	0	0	0	0	0	13	0
1	0	0	0	0	0	0	0	0	10	0

Tabela 4.20 - Solução do exemplo 1 com a Heurística 3, após KOMBI

Frequência	Padrão									
52	2	2	0	2	0	0	0	1	0	1
27	0	0	2	0	4	1	1	0	0	0
29	0	0	4	2	0	0	0	0	2	0
11	2	2	0	0	0	0	2	1	1	0
3	0	1	0	0	0	3	0	0	6	0
2	0	0	0	3	0	2	0	0	4	0

Na [Tabela 4.21](#) mostra-se solução obtida ao se aplicar a heurística 4. Obtemos uma solução com 123 objetos e 6 padrões distintos. A aplicação do KOMBI na terceira fase, reduziu o número de padrões de 6 para 5 ([Tabela 4.22](#)). O gap de otimalidade desta solução é 0.

Tabela 4.21 - Solução do exemplo 1 com a Heurística 4

Frequência	Padrão									
84	1	1	2	2	1	0	0	0	0	0
24	1	1	0	0	1	1	2	0	3	2
10	1	2	0	0	0	0	0	6	1	0
3	2	0	0	0	0	3	0	1	4	0
1	2	1	0	0	0	3	1	0	1	0
1	0	0	2	0	0	4	0	0	0	4

Tabela 4.22 - Solução do exemplo 1 com a Heurística 4 após KOMBI

Frequência	Padrão									
84	1	1	2	2	1	0	0	0	0	0
24	1	1	0	0	1	1	2	0	3	2
10	1	2	0	0	0	0	0	6	1	0
3	0	1	2	0	0	0	0	3	1	4
1	2	0	0	0	0	4	1	0	3	0

Para este exemplo 1, as melhores soluções foram obtidas com a aplicação das heurísticas 1 e 4, com 123 objetos e 5 padrões. Estas soluções são melhores que as soluções encontradas com as heurísticas 2 e 3.

No próximo capítulo apresentamos testes computacionais realizados com as heurísticas aplicadas a um conjunto de instâncias gerados aleatoriamente.

5 IMPLEMENTAÇÕES E RESULTADOS COMPUTACIONAIS

As heurísticas 1, 2, 3 e 4 e todos os algoritmos utilizados foram implementadas em C++ e todos os testes computacionais para este trabalho foram realizados em um computador Intel Core 2 Duo, 1.50GHz com 2GB de memória (RAM). O CPLEX 10 foi utilizado para resolução do problema de corte com a técnica de geração de colunas e determinação de α , a taxa de referência para o desperdício na solução. A solução inteira é obtida arredondando-se para baixo as componentes não inteiras da solução relaxada e aplicando-se o procedimento **RFFD** comentado na [Subsubseção 2.3.3.2](#) para resolver o problema residual.

Os problemas testes foram obtidos através do CUTGEN1, um gerador aleatório de problemas de corte de estoque unidimensional desenvolvido por [Gau e Wascher \(1995\)](#). Consideramos as mesmas 18 classes testadas por [Foerster e Wascher \(1999\)](#), [Yanasse e Limeira \(2006\)](#) e [Cui et al. \(2008\)](#), cada uma com 100 instâncias e adotamos no CUTGEN1 a mesma semente, 1994. As classes estão divididas conforme a quantidade de itens m , a média d_M das demandas d_i para $i = 1, \dots, m$ e combinações diferentes de valores v_1 e v_2 para a determinação do comprimento dos itens que são gerados no intervalo $[v_1L, v_2L]$, como mostrado na [Tabela 5.1](#). O comprimento do objeto em estoque é $L = 1000$ para todas as instâncias testadas.

Tabela 5.1 - Problemas Testes Utilizados

Classe	m	v_1	v_2	d_M
1	10	0,01	0,2	10
2	10	0,01	0,2	100
3	20	0,01	0,2	10
4	20	0,01	0,2	100
5	40	0,01	0,2	10
6	40	0,01	0,2	100
7	10	0,01	0,8	10
8	10	0,01	0,8	100
9	20	0,01	0,8	10
10	20	0,01	0,8	100
11	40	0,01	0,8	10
12	40	0,01	0,8	100
13	10	0,2	0,8	10
14	10	0,2	0,8	100
15	20	0,2	0,8	10
16	20	0,2	0,8	100
17	40	0,2	0,8	10
18	40	0,2	0,8	100

Os resultados dos testes computacionais obtidos com as quatro heurísticas propostas são comparados com os resultados de outros três trabalhos encontrados na literatura propostos por Foerster e Wascher (1999), Yanasse e Limeira (2006) e Cui et al. (2008), pois de nosso conhecimento, são os que apresentam os melhores resultados da literatura. O trabalho de SallesNeto (2005) não foi utilizado nas comparações pois relativamente, valoriza mais o objetivo de redução de padrões comparativamente com a minimização de objetos.

Nos trabalhos de Foerster e Wascher (1999) e Yanasse e Limeira (2006) os resultados dos testes computacionais realizados foram apresentados em função da média do número de objetos e padrões, lembrando que estes resultados correspondem à mé-

dia de 100 instâncias para cada classe. Cui et al. (2008) apresentam suas soluções, não em função da média do número de objetos, mas do percentual de desperdício. Eles converteram, de alguma forma, as soluções de Foerster e Wascher (1999) e Yanasse e Limeira (2006) dadas em termos dos objetos, colocando-as em termos da taxa percentual de desperdício. Os nossos resultados computacionais são apresentados em função da perda média, para que possamos compará-los com os resultados obtidos por Cui et al. (2008). Observamos que as soluções de Foerster e Wascher (1999) e Yanasse e Limeira (2006) apresentadas neste trabalho foram extraídas da tabela fornecida em Cui et al. (2008). Apresentamos a seguir resultados dos testes computacionais obtidos com as quatro heurísticas propostas.

5.1 Resultados dos Testes Computacionais

Apresentamos nas tabelas seguintes os resultados dos testes computacionais realizados com as 4 heurísticas. As colunas H_1 , H_2 , H_3 e H_4 correspondem às soluções obtidas com as heurísticas 1, 2, 3, e 4 respectivamente. Nestas tabelas incluímos os resultados de Cui et al. (2008), colunas **Cui 1** e **Cui 2**, e também os obtidos por Yanasse e Limeira (2006) e Foerster e Wascher (1999), colunas **YL** e **FW**, respectivamente. Nestas tabelas e nas seguintes, a coluna (perd) indica a taxa média (em porcentagem) de desperdício na solução, enquanto que a coluna (pad) é a média do número de padrões na solução.

Em seu trabalho, Cui et al. (2008) apresentaram duas soluções diferentes para o problema. Na primeira (coluna Cui 1) tem-se seu melhor resultado para a perda média que foi comparada apenas com o procedimento de Foerster e Wascher (1999), que priorizaram a minimização do desperdício. Na segunda (coluna Cui 2), são soluções com vistas a reduzir a média do número de padrões, que foram comparadas com o procedimento de Yanasse e Limeira (2006). Os valores em destaque das tabelas indicam soluções não dominadas em cada linha.

Neste trabalho, o par (perda média, média do número de padrões) das soluções obtidas pela aplicação de cada uma das heurísticas são comparadas. Dizemos que a média de soluções de uma heurística domina a de outra se ao comparar os valores médios obtidos com a média de soluções da outra heurística, em pelo menos um deles, na perda média ou na média do número de objetos, o valor médio é estritamente menor e no outro o valor médio é igual ou menor. Para os casos em que a dominância não existe, dizemos que as soluções são não dominadas.

Cabe observar que neste trabalho, as heurísticas não foram desenvolvidas com o intuito de estabelecer a fronteira de Pareto (vide [Lopes e Araujo \(2009\)](#)).

Apresentamos a seguir em quatro diferentes tabelas, os resultados dos testes computacionais realizados com as heurísticas. São avaliados os desempenhos individuais de cada uma das 4 heurísticas comparativamente a outros procedimentos da literatura.

Tabela 5.2 - Desempenho da heurística 1

<i>Classe</i>	YL		FW		Cui 1		Cui 2		H_1	
	perd	pad	perd	pad	perd	pad	perd	pad	perd	pad
1	5,04	3,31	4,47	3,40	4,55	3,83	4,55	3,83	5,03	3,26
2	0,61	6,95	0,47	7,81	0,52	6,53	0,52	6,53	1,04	5,35
3	2,70	4,96	2,52	5,89	2,56	5,75	2,56	5,75	2,56	5,47
4	0,27	10,32	0,25	14,26	0,25	9,50	0,25	9,50	0,45	9,73
5	1,17	7,63	1,10	10,75	1,08	8,59	1,08	8,59	1,10	7,57
6	0,10	13,31	0,12	25,44	0,11	13,76	0,11	13,76	0,34	13,29
7	17,83	7,66	15,41	7,90	13,64	8,33	15,11	7,67	14,67	7,73
8	15,46	9,62	15,00	9,96	13,14	9,90	13,91	9,23	14,39	9,51
9	16,20	13,64	11,00	15,03	9,92	14,90	13,47	13,08	11,98	13,65
10	12,24	18,21	10,72	19,28	9,63	18,46	11,24	16,58	12,11	17,64
11	16,19	24,60	7,33	28,74	7,35	27,37	13,28	22,60	10,59	24,66
12	11,37	33,23	7,29	37,31	7,12	34,45	12,08	28,56	10,77	30,38
13	19,77	8,93	19,17	8,97	17,49	9,20	18,16	8,79	17,98	8,89
14	18,70	10,51	18,55	10,32	17,15	10,36	17,53	10,00	17,89	10,31
15	17,83	16,28	14,76	16,88	13,83	16,93	16,32	15,52	15,90	16,13
16	15,09	19,89	14,67	19,91	13,60	19,51	14,20	18,65	15,91	19,16
17	17,41	29,76	10,30	31,46	10,22	30,78	14,94	27,28	14,12	28,93
18	11,25	37,90	10,22	38,28	10,08	36,69	10,79	35,09	14,75	35,24

Tabela 5.3 - Desempenho da heurística 2

<i>Classe</i>	YL		FW		Cui 1		Cui 2		H₂	
	perd	pad	perd	pad	perd	pad	perd	pad	perd	pad
1	5,04	3,31	4,47	3,40	4,55	3,83	4,55	3,83	5,28	3,36
2	0,61	6,95	0,47	7,81	0,52	6,53	0,52	6,53	1,79	5,31
3	2,70	4,96	2,52	5,89	2,56	5,75	2,56	5,75	2,79	5,25
4	0,27	10,32	0,25	14,26	0,25	9,50	0,25	9,50	1,01	8,02
5	1,17	7,63	1,10	10,75	1,08	8,59	1,08	8,59	1,33	8,38
6	0,10	13,31	0,12	25,44	0,11	13,76	0,11	13,76	0,73	12,78
7	17,83	7,66	15,41	7,90	13,64	8,33	15,11	7,67	16,23	7,63
8	15,46	9,62	15,00	9,96	13,14	9,90	13,91	9,23	16,01	9,69
9	16,20	13,64	11,00	15,03	9,92	14,90	13,47	13,08	13,57	13,53
10	12,24	18,21	10,72	19,28	9,63	18,46	11,24	16,58	14,66	17,05
11	16,19	24,60	7,33	28,74	7,35	27,37	13,28	22,60	12,40	23,89
12	11,37	33,23	7,29	37,31	7,12	34,45	12,08	28,56	13,68	30,31
13	19,77	8,93	19,17	8,97	17,49	9,20	18,16	8,79	19,40	8,82
14	18,70	10,51	18,55	10,32	17,15	10,36	17,53	10,00	20,08	10,36
15	17,83	16,28	14,76	16,88	13,83	16,93	16,32	15,52	17,30	16,01
16	15,09	19,89	14,67	19,91	13,60	19,51	14,20	18,65	17,80	19,39
17	17,41	29,76	10,30	31,46	10,22	30,78	14,94	27,28	14,94	28,65
18	11,25	37,90	10,22	38,28	10,08	36,69	10,79	35,09	16,55	35,64

Nota-se que individualmente o melhor desempenho foi obtido pela heurística 4 (ver [Tabela 5.5](#)), com somente uma de suas soluções dominadas, na classe 16; o segundo melhor desempenho foi o da heurística 1 (ver [Tabela 5.2](#)), que teve suas soluções dominadas em 6 das 18 classes a saber, classes 4, 8, 10, 14, 16 e 18; a seguir aparecem as heurísticas 2 e 3 com desempenhos bem parecidos, obtendo soluções não dominadas em torno da metade das 18 classes testadas (ver [Tabela 5.3](#) e [Tabela 5.4](#)).

Tabela 5.4 - Desempenho da heurística 3

Classe	YL		FW		Cui 1		Cui 2		H_3	
	perd	pad	perd	pad	perd	pad	perd	pad	perd	pad
1	5,04	3,31	4,47	3,40	4,55	3,83	4,55	3,83	5,06	3,35
2	0,61	6,95	0,47	7,81	0,52	6,53	0,52	6,53	1,26	6,06
3	2,70	4,96	2,52	5,89	2,56	5,75	2,56	5,75	2,60	5,48
4	0,27	10,32	0,25	14,26	0,25	9,50	0,25	9,50	0,52	9,32
5	1,17	7,63	1,10	10,75	1,08	8,59	1,08	8,59	1,12	8,52
6	0,10	13,31	0,12	25,44	0,11	13,76	0,11	13,76	0,21	13,63
7	17,83	7,66	15,41	7,90	13,64	8,33	15,11	7,67	15,22	7,86
8	15,46	9,62	15,00	9,96	13,14	9,90	13,91	9,23	14,95	9,89
9	16,20	13,64	11,00	15,03	9,92	14,90	13,47	13,08	12,66	14,14
10	12,24	18,21	10,72	19,28	9,63	18,46	11,24	16,58	13,29	18,07
11	16,19	24,60	7,33	28,74	7,35	27,37	13,28	22,60	11,44	24,96
12	11,37	33,23	7,29	37,31	7,12	34,45	12,08	28,56	12,12	32,30
13	19,77	8,93	19,17	8,97	17,49	9,20	18,16	8,79	19,00	8,87
14	18,70	10,51	18,55	10,32	17,15	10,36	17,53	10,00	18,74	10,62
15	17,83	16,28	14,76	16,88	13,83	16,93	16,32	15,52	16,29	15,92
16	15,09	19,89	14,67	19,91	13,60	19,51	14,20	18,65	16,80	19,38
17	17,41	29,76	10,30	31,46	10,22	30,78	14,94	27,28	14,69	28,74
18	11,25	37,90	10,22	38,28	10,08	36,69	10,79	35,09	15,91	35,65

Na [Tabela 5.6](#) comparamos todas as heurísticas numa única tabela. A heurística 4 é a que apresenta maior número de médias de soluções não dominadas, 16 no total. Apenas nas classes 1 e 18 suas médias de soluções foram dominadas pelos procedimentos de [Foerster e Wascher \(1999\)](#) e [Cui et al. \(2008\)](#), respectivamente. Em seguida, com 12 classes com médias de soluções não dominadas vem a heurística de [Cui et al. \(2008\)](#). Suas médias de soluções foram dominadas nas classes de 1 a 6, que são de instâncias com itens pequenos. A heurística 1 vem em seguida tendo 8 classes com médias de soluções dominadas. A heurística 3 e o método de [Yanasse e Limeira \(2006\)](#), tiveram suas médias de soluções dominadas em quase todas as

Tabela 5.5 - Desempenho da heurística 4

<i>Classe</i>	YL		FW		Cui 1		Cui 2		H₄	
	perd	pad	perd	pad	perd	pad	perd	pad	perd	pad
1	5,04	3,31	4,47	3,40	4,55	3,83	4,55	3,83	4,47	3,42
2	0,61	6,95	0,47	7,81	0,52	6,53	0,52	6,53	0,48	5,76
3	2,70	4,96	2,52	5,89	2,56	5,75	2,56	5,75	2,56	4,95
4	0,27	10,32	0,25	14,26	0,25	9,50	0,25	9,50	0,25	8,49
5	1,17	7,63	1,10	10,75	1,08	8,59	1,08	8,59	1,07	8,18
6	0,10	13,31	0,12	25,44	0,11	13,76	0,11	13,76	0,11	13,28
7	17,83	7,66	15,41	7,90	13,64	8,33	15,11	7,67	13,56	7,82
8	15,46	9,62	15,00	9,96	13,14	9,90	13,91	9,23	12,99	9,99
9	16,20	13,64	11,00	15,03	9,92	14,90	13,47	13,08	9,78	14,25
10	12,24	18,21	10,72	19,28	9,63	18,46	11,24	16,58	9,43	19,29
11	16,19	24,60	7,33	28,74	7,35	27,37	13,28	22,60	7,09	25,92
12	11,37	33,23	7,29	37,31	7,12	34,45	12,08	28,56	7,08	36,21
13	19,77	8,93	19,17	8,97	17,49	9,20	18,16	8,79	17,41	8,92
14	18,70	10,51	18,55	10,32	17,15	10,36	17,53	10,00	17,08	10,54
15	17,83	16,28	14,76	16,88	13,83	16,93	16,32	15,52	13,75	16,47
16	15,09	19,89	14,67	19,91	13,60	19,51	14,20	18,65	13,56	20,68
17	17,41	29,76	10,30	31,46	10,22	30,78	14,94	27,28	10,09	29,89
18	11,25	37,90	10,22	38,28	10,08	36,69	10,79	35,09	10,10	39,05

classes, exceções feitas às classes 15 e 17 para a Heurística 3, e à classe 6 para Yanasse e Limeira (2006). A heurística 2 fornece médias de soluções não dominadas em 5 classes, e a heurística de Foerster e Wascher (1999) em 3.

Tabela 5.6 - Testes Computacionais Realizados com as Heurísticas (KOMBI)

Classe	YL		FW		Cui 1		Cui 2		H ₁		H ₂		H ₃		H ₄	
	perd	pad	perd	pad	perd	pad	perd	pad	perd	pad	perd	pad	perd	pad	perd	pad
1	5,04	3,31	4,47	3,40	4,55	3,83	4,55	3,83	5,03	3,26	5,28	3,36	5,06	3,35	4,47	3,42
2	0,61	6,95	0,47	7,81	0,52	6,53	0,52	6,53	1,04	5,35	1,79	5,31	1,26	6,06	0,48	5,76
3	2,70	4,96	2,52	5,89	2,56	5,75	2,56	5,75	2,56	5,47	2,79	5,25	2,60	5,48	2,56	4,95
4	0,27	10,32	0,25	14,26	0,25	9,50	0,25	9,50	0,45	9,73	1,01	8,02	0,52	9,32	0,25	8,49
5	1,17	7,63	1,10	10,75	1,08	8,59	1,08	8,59	1,10	7,57	1,33	8,38	1,12	8,52	1,07	8,18
6	0,10	13,31	0,12	25,44	0,11	13,76	0,11	13,76	0,34	13,29	0,73	12,78	0,21	13,63	0,11	13,28
7	17,83	7,66	15,41	7,90	13,64	8,33	15,11	7,67	14,67	7,73	16,23	7,63	15,22	7,86	13,56	7,82
8	15,46	9,62	15,00	9,96	13,14	9,90	13,91	9,23	14,39	9,51	16,01	9,69	14,95	9,89	12,99	9,99
9	16,20	13,64	11,00	15,03	9,92	14,90	13,47	13,08	11,98	13,65	13,57	13,53	12,66	14,14	9,78	14,25
10	12,24	18,21	10,72	19,28	9,63	18,46	11,24	16,58	12,11	17,64	14,66	17,05	13,29	18,07	9,43	19,29
11	16,19	24,60	7,33	28,74	7,35	27,37	13,28	22,60	10,59	24,66	12,40	23,89	11,44	24,96	7,09	25,92
12	11,37	33,23	7,29	37,31	7,12	34,45	12,08	28,56	10,77	30,38	13,68	30,31	12,12	32,30	7,08	36,21
13	19,77	8,93	19,17	8,97	17,49	9,20	18,16	8,79	17,98	8,89	19,40	8,82	19,00	8,87	17,41	8,92
14	18,70	10,51	18,55	10,32	17,15	10,36	17,53	10,00	17,89	10,31	20,08	10,36	18,74	10,62	17,08	10,54
15	17,83	16,28	14,76	16,88	13,83	16,93	16,32	15,52	15,90	16,13	17,30	16,01	16,29	15,92	13,75	16,47
16	15,09	19,89	14,67	19,91	13,60	19,51	14,20	18,65	15,91	19,16	17,80	19,39	16,80	19,38	13,56	20,68
17	17,41	29,76	10,30	31,46	10,22	30,78	14,94	27,28	14,12	28,93	14,94	28,65	14,69	28,74	10,09	29,89
18	11,25	37,90	10,22	38,28	10,08	36,69	10,79	35,09	14,75	35,24	16,55	35,64	15,91	35,65	10,10	39,05

6 KOMBI MODIFICADO - KOMBI-M

Considere um exemplar do problema de corte apresentado na [Tabela 6.1](#). São considerados 10 diferentes itens e o tamanho do objeto em estoque é $L = 1000$. Resolvendo o problema com aplicação da primeira e segunda fases da heurística 2, obtém-se uma solução com 17 objetos e 4 diferentes padrões, apresentado na [Tabela 6.2](#).

Tabela 6.1 - Exemplo 5: dados do problema de corte

Item	Comprimento	Demanda
1	200	24
2	195	2
3	183	21
4	178	3
5	143	24
6	142	1
7	121	14
8	108	6
9	83	3
10	15	2

Tabela 6.2 - Solução obtida com a Heurística 2

Frequência	Padrão									
12	2	0	1	0	2	0	1	0	0	0
3	0	0	3	1	0	0	0	2	0	0
1	0	2	0	0	0	1	2	0	2	2
1	0	0	0	0	0	0	0	0	1	0

Aplicamos à solução o KOMBI12 que tenta combinar dois padrões em um. Considere os dois últimos padrões da solução gerada, $p_1 = (0 \ 2 \ 0 \ 0 \ 0 \ 1 \ 2 \ 0 \ 2 \ 2)$ e $p_2 = (0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0)$, com frequências $x_1 = 1$ e $x_2 = 1$, respectivamente. O número de itens do tipo i fornecido pelos dois padrões é obtido através da soma $S_i = a_{i1}x_1 + a_{i2}x_2$ para $i = 1, \dots, m$. Note que os dois padrões juntos produzem somente itens do tipo 2, 6, 7, 9 e 10, em quantidades $s_2 = 2$, $s_6 = 1$, $s_7 = 2$, $s_9 = 2$ e $s_{10} = 2$, respectivamente.

A frequência do novo padrão $p = (a_{1p}, \dots, a_{np})$ é a soma das frequências dos padrões

combinados $x_p = x_1 + x_2 = 2$. A quantidade de itens do tipo i no padrão p é dada por $a_{ip} = s_i/x_p$ para $i = 1, \dots, m$. Portanto $a_{2p} = 1$, $a_{6p} = 0,5$, $a_{7p} = 1$, $a_{9p} = 1,5$ e $a_{10p} = 1$. Como dois dos valores obtidos não são inteiros o KOMBI12 proposto por Foerster e Wascher (1999) não os agrupa, e nenhuma redução no número de padrões é possível. No entanto podemos combinar estes dois padrões pois o padrão $p = (0 \ 1 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0 \ 2 \ 1)$ obtido neste caso é viável, embora quando cortado 2 vezes, produza número de itens a mais que os dois padrões originais. Esta combinação reduz o número de padrões na solução de 4 para 3 mantendo constante o número de objetos como mostra a Tabela 6.3.

Tabela 6.3 - Solução obtida com KOMBI-M

Frequência	Padrão									
12	2	0	1	0	2	0	1	0	0	0
3	0	0	3	1	0	0	0	2	0	0
2	0	1	0	0	0	1	1	0	2	1

Baseado nesta observação sugerimos a modificação a seguir. No procedimento KOMBI de Foerster e Wascher (1999) exige-se que a quantidade de itens produzidos pelos padrões combinados seja igual à quantidade dos itens produzidos pelos padrões originais e que o total de objetos utilizados em ambos os casos sejam iguais. Entretanto, acreditamos que se a solução preserva o número total de objetos e satisfaz a demanda, não existe problema em produzir itens em excesso. Itens eventualmente produzidos a mais podem ser estocados (se houver interesse) ou serem simplesmente descartados, como aparentemente são descartadas as sobras das soluções, no caso do KOMBI original.

Seja a combinação de 2 padrões em 1 inicialmente proposta por Diegel et al. (1993) e mais tarde estendida por Foerster e Wascher (1999) para combinações com um conjunto maior de padrões.

O padrão $*$ = (a_{1*}, \dots, a_{n*}) , resultante da combinação de dois padrões quaisquer 1 e 2, substitui os dois padrões originais somente se $a_{i*} = \lceil s_i/x_* \rceil$ e o padrão $*$ = (a_{1*}, \dots, a_{n*}) é viável.

Esta modificação favorece um número maior de combinações comparativamente ao KOMBI, aumentando as chances de se reduzir ainda mais a quantidade de padrões

na solução. A mesma idéia pode ser aplicada para a combinação de p padrões em q , $p > q$, p e q inteiros positivos. A este procedimento, denominamos KOMBI MODIFICADO ou KOMBI-M.

Resolvemos os exemplos do [Capítulo 4](#) aplicando o KOMBI-M ao invés do KOMBI original como procedimento de redução de padrões. Para os exemplos 1 ([Tabela 4.1](#)) e 3 ([Tabela 4.10](#)) em que são aplicadas as heurísticas 1 e 3 respectivamente, nenhuma alteração foi verificada em relação às soluções obtidas com aplicação do KOMBI original (ver [Tabela 4.4](#) e [Tabela 4.14](#)).

Para o exemplo 2 ([Tabela 4.5](#)) em que é aplicada a heurística 2, a aplicação do KOMBI-M reduziu o número de padrões na solução de 9 para 6 como mostra a [Tabela 6.4](#), portanto mais eficiente que o KOMBI original, que conseguiu reduzir de 9 para 7 (ver [Tabela 4.9](#)).

Tabela 6.4 - Eemplo 2: solução da Heurística pós KOMBI-M

Frequência	Padrão									
9	0	0	0	0	0	1	0	1	0	0
7	0	0	1	0	0	0	1	0	0	0
6	0	0	1	0	0	0	0	0	1	1
7	0	1	0	0	0	0	0	0	1	0
11	0	0	0	1	1	0	0	0	0	0
10	1	0	0	0	0	0	0	0	0	1

Para o exemplo 4 ([Tabela 4.15](#)) em que é aplicada a heurística 4, a aplicação do KOMBI-M reduziu o número de padrões na solução de 4 para 3 como mostra a [Tabela 6.5](#), com a aplicação do KOMBI original, nenhuma redução de padrões foi verificada (ver [Tabela 4.17](#)).

Tabela 6.5 - Exemplo 4: solução da Heurística pós KOMBI-M

Frequência	Padrão									
8	1	1	2	2	1	0	0	0	0	0
3	1	1	0	0	1	1	1	2	2	1
2	1	1	1	1	0	1	1	0	2	1

6.1 Experimentos Computacionais com o KOMBI-M

Resolvemos o problema de corte utilizando Programação Linear com Geração de Colunas, após relaxar a integralidade do problema, similarmente ao que foi feito por [Stadtler \(1990\)](#) e por [Foerster e Wascher \(1999\)](#) em seus testes computacionais. Para iniciar o procedimento resolvemos o problema de corte através da heurística FFD e utilizamos os padrões gerados por esta heurística e mais os padrões homogêneos.

A [Tabela 6.6](#) mostra a solução reproduzida de [Foerster e Wascher \(1999\)](#) obtida com a técnica de geração de colunas (coluna FW(GC)) e a solução obtida com a nossa implementação (coluna CY(GC)). Note que obtivemos ligeiras diferenças de valores, pois não seguimos fielmente os mesmos procedimentos de arredondamento usados por eles. Os dados das colunas KB23 e KB234, respectivamente, foram reproduzidas do trabalho de [Foerster e Wascher \(1999\)](#), e correspondem às médias da quantidade de padrões obtidas ao aplicar à solução obtida por [Foerster e Wascher \(1999\)](#) o procedimento KOMBI23 (que tenta combinar 3 padrões em 2 e 2 padrões em 1) e procedimento KOMBI234 (que tenta combinar 4 padrões em 3, 3 padrões em 2 e 2 padrões em 1), respectivamente.

Tabela 6.6 - Kombi Modificado

Classe	FW(GC)		KB23	KB234	CY(GC)		KB23*	KB234*	KBM23	KBM234
	objs	pads	pads	pads	objs	pads	pads	pads	pads	pads
1	11,49	8,07	4,36	3,40	11,53	8,86	4,14	3,32	3,72	3,27
2	110,25	11,86	9,23	7,81	110,39	13,50	9,28	7,64	8,77	7,58
3	22,13	15,49	7,72	5,89	22,15	16,59	7,73	5,91	7,65	5,78
4	215,93	23,21	17,00	14,26	216,20	25,64	17,24	14,16	16,65	13,75
5	42,96	29,22	14,40	10,75	43,04	30,75	14,64	10,88	14,17	10,46
6	424,71	44,48	31,01	25,44	425,24	49,01	30,79	24,71	30,25	24,14
7	50,21	10,33	8,44	7,90	50,24	8,68	7,74	7,43	7,25	7,13
8	499,52	11,67	10,39	9,96	499,62	9,14	8,96	8,84	8,22	8,17
9	93,67	19,92	16,25	15,03	93,66	17,84	15,64	14,53	14,64	14,13
10	932,32	22,51	20,19	19,28	932,27	18,99	18,23	17,81	16,95	16,85
11	176,97	38,34	32,01	28,74	176,92	35,59	30,76	28,10	29,17	27,72
12	1766,20	43,22	39,11	37,31	1763,51	38,29	36,74	35,51	34,63	34,17
13	63,27	10,04	9,33	8,97	63,47	9,14	8,56	8,24	8,11	7,97
14	623,12	10,53	10,42	10,32	632,36	9,58	9,45	9,34	8,97	8,85
15	119,43	19,71	17,74	16,88	119,59	17,98	16,53	15,89	15,93	15,50
16	1191,80	20,57	20,17	19,91	1192,00	18,76	18,41	18,22	17,92	17,77
17	224,68	38,12	33,53	31,46	224,85	35,19	31,67	30,20	30,70	29,69
18	2242,40	39,90	39,12	38,28	2242,60	36,88	36,11	35,62	35,28	34,84

Nas colunas KB23* e KB234* apresentam-se as médias da quantidade de padrões obtidas ao aplicarmos o KOMBI à nossa solução e, nas colunas KBM23 e KBM234, as médias da quantidade de padrões obtidas ao aplicarmos o KOMBI-M. Como esperado, houve redução em todas as classes. A redução obtida em termos percentuais é apresentada na [Tabela 6.7](#).

Tabela 6.7 - Redução de Padrões em Porcentagem

Classe	FW(GC)	KB23		KB234		CY(GC)	KB23*		KB234*		KBM23		KBM234	
	pads	pads	red(%)	pads	red(%)	pads	pads	red(%)	pads	red(%)	pads	red(%)	pads	red(%)
1	8,07	4,36	45,97	3,40	57,87	8,86	4,14	53,27	3,32	62,52	3,72	58,01	3,27	63,09
2	11,86	9,23	22,18	7,81	34,15	13,50	9,28	31,26	7,64	43,41	8,77	35,04	7,58	43,86
3	15,49	7,72	50,16	5,89	61,98	16,59	7,73	53,40	5,91	64,38	7,65	53,89	5,78	64,51
4	23,21	17,00	26,76	14,26	38,56	25,64	17,24	32,76	14,16	44,77	16,65	35,06	13,75	46,37
5	29,22	14,40	50,72	10,75	63,21	30,75	14,64	52,39	10,88	64,62	14,17	53,92	10,46	65,98
6	44,48	31,01	30,28	25,44	42,81	49,01	30,79	37,18	24,71	49,58	30,25	38,28	24,14	50,74
7	10,33	8,44	18,30	7,90	23,52	8,68	7,74	10,83	7,43	14,40	7,25	16,47	7,13	17,86
8	11,67	10,39	10,97	9,96	14,65	9,14	8,96	1,97	8,84	3,28	8,22	10,07	8,17	10,61
9	19,92	16,25	18,42	15,03	24,55	17,84	15,64	12,33	14,53	18,55	14,64	17,94	14,13	20,80
10	22,51	20,19	10,31	19,28	14,35	18,99	18,23	4,00	17,81	6,21	16,95	10,74	16,85	11,27
11	38,34	32,01	16,51	28,74	25,04	35,59	30,76	13,57	28,10	21,05	29,17	18,04	27,72	22,11
12	43,22	39,11	9,51	37,31	13,67	38,29	36,74	4,05	35,51	7,26	34,63	9,56	34,17	10,76
13	10,04	9,33	7,07	8,97	10,66	9,14	8,56	6,35	8,24	9,85	8,11	11,27	7,97	12,80
14	10,53	10,42	1,04	10,32	1,99	9,58	9,45	1,36	9,34	2,51	8,97	6,37	8,85	7,62
15	19,71	17,74	9,99	16,88	14,36	17,98	16,53	8,06	15,89	11,62	15,93	11,40	15,50	13,79
16	20,57	20,17	1,94	19,91	3,21	18,76	18,41	1,87	18,22	2,88	17,92	4,48	17,77	5,28
17	38,12	33,53	12,04	31,46	17,47	35,19	31,67	10,00	30,20	14,18	30,70	12,76	29,69	15,63
18	39,90	39,12	1,95	38,28	4,06	36,88	36,11	2,09	35,62	3,42	35,28	4,34	34,84	5,53

Na [Tabela 6.8](#) mostra-se o tempo computacional médio para as 18 classes testadas. Tem-se nas colunas $KB23_t$ e $KB234_t$ o tempo médio obtido por [Foerster e Wascher \(1999\)](#) ao aplicar às soluções o KOMBI23 e KOMBI234 respectivamente. Nas colunas $KB23^*_t$ e $KB234^*_t$ tem o tempo médio obtido com a nossa implementação, aplicando-se às soluções o KOMBI23 e KOMBI234. Finalmente, tem-se nas colunas $KBM23_t$ e $KBM234_t$ o tempo médio obtido aplicando-se o KOMBI-M. Nota-se que em geral o tempo computacional médio com o KOMBI-M é maior quando comparado com a nossa implementação do KOMBI.

Tabela 6.8 - Média do Tempo Computacional (s)

Classe	KB23 _t	KB234 _t	KB23* _t	KB234* _t	KBM23 _t	KBM234 _t
1	0,31	0,35	0,11	0,14	1,27	3,16
2	0,87	1,26	0,14	0,36	3,33	7,16
3	0,87	2,10	0,16	2,08	5,15	15,18
4	9,79	16,41	0,60	5,60	3,75	18,63
5	33,21	40,03	1,12	13,14	19,12	51,54
6	188,47	383,30	16,76	27,14	45,16	61,91
7	0,08	0,11	0,10	0,13	0,08	0,13
8	0,10	0,24	0,11	0,26	0,12	0,22
9	0,52	1,47	0,17	0,53	0,65	1,04
10	0,63	3,40	0,30	1,64	2,41	3,89
11	7,03	36,98	1,29	11,55	17,84	29,16
12	7,28	77,41	2,23	33,54	36,91	71,90
13	0,07	0,13	0,08	0,13	0,08	0,11
14	0,08	0,18	0,11	0,22	0,11	0,20
15	0,45	1,92	0,13	0,42	0,18	0,62
16	0,42	2,71	0,29	1,28	0,34	1,65
17	7,18	51,31	1,02	8,95	3,91	14,89
18	4,50	71,31	1,87	23,36	3,40	32,95

6.2 Resultados dos Testes Computacionais com KOMBI-M

Apresentamos a seguir em quatro diferentes tabelas, os resultados dos testes computacionais realizados com as heurísticas (colunas H_1^* , H_2^* , H_3^* e H_4^*) em que é aplicada como técnica de redução de padrões o KOMBI-M. Nota-se que neste caso, todas as 4 heurísticas têm um desempenho individual satisfatório com nenhuma de suas médias de soluções dominada pelos demais procedimentos da literatura (ver [Tabela 6.9](#), [Tabela 6.10](#), [Tabela 6.11](#) e [Tabela 6.12](#)), exceções feitas à classe 3 com a heurística 2 e à classe 12 com a heurística 3.

É interessante ressaltar que a heurística proposta por [Yanasse e Limeira \(2006\)](#) teve

suas médias de soluções dominadas em praticamente todas as classes testadas, única exceção feita à classe 6. O mesmo ocorre com a heurística de Foerster e Wascher (1999), sendo exceção neste caso a classe 2, que tem soluções com a menor taxa de desperdício entre todas as demais soluções das heurísticas consideradas.

Tabela 6.9 - Desempenho da heurística 1 com KOMBI-M

Classe	YL		FW		Cui 1		Cui 2		H_1^*	
	perd	pad	perd	pad	perd	pad	perd	pad	perd	pad
1	5,04	3,31	4,47	3,40	4,55	3,83	4,55	3,83	5,03	3,19
2	0,61	6,95	0,47	7,81	0,52	6,53	0,52	6,53	1,04	5,34
3	2,70	4,96	2,52	5,89	2,56	5,75	2,56	5,75	2,56	5,05
4	0,27	10,32	0,25	14,26	0,25	9,50	0,25	9,50	0,45	8,97
5	1,17	7,63	1,10	10,75	1,08	8,59	1,08	8,59	1,10	7,07
6	0,10	13,31	0,12	25,44	0,11	13,76	0,11	13,76	0,34	12,33
7	17,83	7,66	15,41	7,90	13,64	8,33	15,11	7,67	14,67	6,89
8	15,46	9,62	15,00	9,96	13,14	9,90	13,91	9,23	14,39	7,91
9	16,20	13,64	11,00	15,03	9,92	14,90	13,47	13,08	11,98	12,92
10	12,24	18,21	10,72	19,28	9,63	18,46	11,24	16,58	12,11	15,84
11	16,19	24,60	7,33	28,74	7,35	27,37	13,28	22,60	10,59	23,43
12	11,37	33,23	7,29	37,31	7,12	34,45	12,08	28,56	10,77	28,64
13	19,77	8,93	19,17	8,97	17,49	9,20	18,16	8,79	17,98	7,89
14	18,70	10,51	18,55	10,32	17,15	10,36	17,53	10,00	17,89	8,97
15	17,83	16,28	14,76	16,88	13,83	16,93	16,32	15,52	15,90	14,93
16	15,09	19,89	14,67	19,91	13,60	19,51	14,20	18,65	15,91	16,98
17	17,41	29,76	10,30	31,46	10,22	30,78	14,94	27,28	14,12	27,00
18	11,25	37,90	10,22	38,28	10,08	36,69	10,79	35,09	14,75	32,63

Tabela 6.10 - Desempenho da heurística 2 com KOMBI-M

<i>Classe</i>	YL		FW		Cui 1		Cui 2		<i>H₂*</i>	
	perd	pad	perd	pad	perd	pad	perd	pad	perd	pad
1	5,04	3,31	4,47	3,40	4,55	3,83	4,55	3,83	5,28	3,06
2	0,61	6,95	0,47	7,81	0,52	6,53	0,52	6,53	1,79	5,05
3	2,70	4,96	2,52	5,89	2,56	5,75	2,56	5,75	2,79	4,97
4	0,27	10,32	0,25	14,26	0,25	9,50	0,25	9,50	1,01	7,73
5	1,17	7,63	1,10	10,75	1,08	8,59	1,08	8,59	1,33	7,87
6	0,10	13,31	0,12	25,44	0,11	13,76	0,11	13,76	0,73	12,26
7	17,83	7,66	15,41	7,90	13,64	8,33	15,11	7,67	16,23	6,67
8	15,46	9,62	15,00	9,96	13,14	9,90	13,91	9,23	16,01	7,92
9	16,20	13,64	11,00	15,03	9,92	14,90	13,47	13,08	13,57	12,47
10	12,24	18,21	10,72	19,28	9,63	18,46	11,24	16,58	14,66	15,09
11	16,19	24,60	7,33	28,74	7,35	27,37	13,28	22,60	12,40	22,49
12	11,37	33,23	7,29	37,31	7,12	34,45	12,08	28,56	13,68	27,95
13	19,77	8,93	19,17	8,97	17,49	9,20	18,16	8,79	19,40	7,84
14	18,70	10,51	18,55	10,32	17,15	10,36	17,53	10,00	20,08	8,81
15	17,83	16,28	14,76	16,88	13,83	16,93	16,32	15,52	17,30	14,71
16	15,09	19,89	14,67	19,91	13,60	19,51	14,20	18,65	17,80	17,23
17	17,41	29,76	10,30	31,46	10,22	30,78	14,94	27,28	14,94	26,84
18	11,25	37,90	10,22	38,28	10,08	36,69	10,79	35,09	16,55	32,63

Na [Tabela 6.15](#) apresentam-se os tempos computacionais obtidos com as 4 heurísticas e pelos demais procedimentos da literatura. Estes valores não são diretamente comparáveis pois foram executados em máquinas diferentes e por versões diferentes de otimizadores. [Foerster e Wascher \(1999\)](#) realizaram seus testes computacionais num computador pessoal 486/66 compatível com um IBM, utilizando o MODULA-2 como linguagem de programação. [Yanasse e Limeira \(2006\)](#) realizaram suas implementações em C++ e os testes computacionais em um microcomputador Intel Celeron, 266 MHz com 128MB de RAM e em uma estação de trabalho Sun Ultra 30, 296 MHz com 384 MB de RAM. [Cui et al. \(2008\)](#) utilizaram um computador com 2,80

Tabela 6.11 - Desempenho da heurística 3 com KOMBI-M

Classe	YL		FW		Cui 1		Cui 2		H_3^*	
	perd	pad	perd	pad	perd	pad	perd	pad	perd	pad
1	5,04	3,31	4,47	3,40	4,55	3,83	4,55	3,83	5,06	3,14
2	0,61	6,95	0,47	7,81	0,52	6,53	0,52	6,53	1,26	5,87
3	2,70	4,96	2,52	5,89	2,56	5,75	2,56	5,75	2,60	5,25
4	0,27	10,32	0,25	14,26	0,25	9,50	0,25	9,50	0,52	8,93
5	1,17	7,63	1,10	10,75	1,08	8,59	1,08	8,59	1,12	7,87
6	0,10	13,31	0,12	25,44	0,11	13,76	0,11	13,76	0,21	13,07
7	17,83	7,66	15,41	7,90	13,64	8,33	15,11	7,67	15,22	6,77
8	15,46	9,62	15,00	9,96	13,14	9,90	13,91	9,23	14,95	7,92
9	16,20	13,64	11,00	15,03	9,92	14,90	13,47	13,08	12,66	12,91
10	12,24	18,21	10,72	19,28	9,63	18,46	11,24	16,58	13,29	15,56
11	16,19	24,60	7,33	28,74	7,35	27,37	13,28	22,60	11,44	23,20
12	11,37	33,23	7,29	37,31	7,12	34,45	12,08	28,56	12,12	29,08
13	19,77	8,93	19,17	8,97	17,49	9,20	18,16	8,79	19,00	7,83
14	18,70	10,51	18,55	10,32	17,15	10,36	17,53	10,00	18,74	8,93
15	17,83	16,28	14,76	16,88	13,83	16,93	16,32	15,52	16,29	14,36
16	15,09	19,89	14,67	19,91	13,60	19,51	14,20	18,65	16,80	16,97
17	17,41	29,76	10,30	31,46	10,22	30,78	14,94	27,28	14,69	26,72
18	11,25	37,90	10,22	38,28	10,08	36,69	10,79	35,09	15,91	32,11

GHz de CPU e 512 MB de RAM. Podemos apenas dizer que os tempos de execução das heurísticas são aceitáveis para sua aplicação prática. É importante ressaltar que em algumas classes o tempo computacional da heurística 1 é alto, devido à aplicação do método KOMBI a todas as soluções testadas podendo chegar a até 11 diferentes soluções.

Apresentamos, ver tabelas 6.13 a 6.15, os resultados dos testes computacionais das 4 heurísticas com a aplicação do KOMBI-M na terceira fase, colunas H_1^* , H_2^* , H_3^* e H_4^* . Na Tabela 6.13 testamos o desempenho das heurísticas 1, 2 e 3 excluindo-se a heurística 4, pois esta é um variante da heurística SHPC proposta por Cui et al.

Tabela 6.12 - Desempenho da heurística 4 com KOMBI-M

Classe	YL		FW		Cui 1		Cui 2		H_4^*	
	perd	pad	perd	pad	perd	pad	perd	pad	perd	pad
1	5,04	3,31	4,47	3,40	4,55	3,83	4,55	3,83	4,47	3,13
2	0,61	6,95	0,47	7,81	0,52	6,53	0,52	6,53	0,48	5,72
3	2,70	4,96	2,52	5,89	2,56	5,75	2,56	5,75	2,56	4,76
4	0,27	10,32	0,25	14,26	0,25	9,50	0,25	9,50	0,25	7,73
5	1,17	7,63	1,10	10,75	1,08	8,59	1,08	8,59	1,07	7,29
6	0,10	13,31	0,12	25,44	0,11	13,76	0,11	13,76	0,11	12,33
7	17,83	7,66	15,41	7,90	13,64	8,33	15,11	7,67	13,56	6,84
8	15,46	9,62	15,00	9,96	13,14	9,90	13,91	9,23	12,99	8,28
9	16,20	13,64	11,00	15,03	9,92	14,90	13,47	13,08	9,78	13,25
10	12,24	18,21	10,72	19,28	9,63	18,46	11,24	16,58	9,43	17,15
11	16,19	24,60	7,33	28,74	7,35	27,37	13,28	22,60	7,09	24,56
12	11,37	33,23	7,29	37,31	7,12	34,45	12,08	28,56	7,08	33,14
13	19,77	8,93	19,17	8,97	17,49	9,20	18,16	8,79	17,41	7,89
14	18,70	10,51	18,55	10,32	17,15	10,36	17,53	10,00	17,08	9,11
15	17,83	16,28	14,76	16,88	13,83	16,93	16,32	15,52	13,75	15,18
16	15,09	19,89	14,67	19,91	13,60	19,51	14,20	18,65	13,56	18,38
17	17,41	29,76	10,30	31,46	10,22	30,78	14,94	27,28	10,09	28,13
18	11,25	37,90	10,22	38,28	10,08	36,69	10,79	35,09	10,10	35,63

(2008). Nota-se neste cenário um bom desempenho heurística 1 com nenhuma de suas médias de soluções dominadas e da heurística 3, dominada pelos demais procedimentos em 4 classes apenas, quase o mesmo número de classes (3) da heurística de Cui et al. (2008).

Tabela 6.13 - Testes Computacionais sem H_4^*

Classe	YL		FW		Cui 1		Cui 2		H_1^*		H_2^*		H_3^*	
	perd	pad	perd	pad	perd	pad	perd	pad	perd	pad	perd	pad	perd	pad
1	5,04	3,31	4,47	3,40	4,55	3,83	4,55	3,83	5,03	3,19	5,28	3,06	5,06	3,14
2	0,61	6,95	0,47	7,81	0,52	6,53	0,52	6,53	1,04	5,34	1,79	5,05	1,26	5,87
3	2,70	4,96	2,52	5,89	2,56	5,75	2,56	5,75	2,56	5,05	2,79	4,97	2,60	5,25
4	0,27	10,32	0,25	14,26	0,25	9,50	0,25	9,50	0,45	8,97	1,01	7,73	0,52	8,93
5	1,17	7,63	1,10	10,75	1,08	8,59	1,08	8,59	1,10	7,07	1,33	7,87	1,12	7,87
6	0,10	13,31	0,12	25,44	0,11	13,76	0,11	13,76	0,34	12,33	0,73	12,26	0,21	13,07
7	17,83	7,66	15,41	7,90	13,64	8,33	15,11	7,67	14,67	6,89	16,23	6,67	15,22	6,77
8	15,46	9,62	15,00	9,96	13,14	9,90	13,91	9,23	14,39	7,91	16,01	7,92	14,95	7,92
9	16,20	13,64	11,00	15,03	9,92	14,90	13,47	13,08	11,98	12,92	13,57	12,47	12,66	12,91
10	12,24	18,21	10,72	19,28	9,63	18,46	11,24	16,58	12,11	15,84	14,66	15,09	13,29	15,56
11	16,19	24,60	7,33	28,74	7,35	27,37	13,28	22,60	10,59	23,43	12,40	22,49	11,42	23,20
12	11,37	33,23	7,29	37,31	7,12	34,45	12,08	28,56	10,77	28,64	13,68	27,95	12,12	29,08
13	19,77	8,93	19,17	8,97	17,49	9,20	18,16	8,79	17,98	7,89	19,40	7,84	19,00	7,83
14	18,70	10,51	18,55	10,32	17,15	10,36	17,53	10,00	17,89	8,97	20,08	8,81	18,74	8,93
15	17,83	16,28	14,76	16,88	13,83	16,93	16,32	15,52	15,90	14,93	17,30	14,71	16,29	14,36
16	15,09	19,89	14,67	19,91	13,60	19,51	14,20	18,65	15,91	16,98	17,80	17,23	16,80	16,97
17	17,41	29,76	10,30	31,46	10,22	30,78	14,94	27,28	14,12	27,00	14,94	26,84	14,69	26,72
18	11,25	37,90	10,22	38,28	10,08	36,69	10,79	35,09	14,75	32,63	16,55	32,63	15,91	32,11

Incluimos agora as 4 heurísticas numa única tabela (ver [Tabela 6.14](#)). Pelos resultados obtidos ao compararmos com os demais procedimentos da literatura, conclui-se que o desempenho obtido foi satisfatório, com várias médias de soluções não dominadas.

Tabela 6.14 - Testes Computacionais (KOMBI-M)

Classe	YL		FW		Cui 1		Cui 2		H ₁ *		H ₂ *		H ₃ *		H ₄ *	
	perd	pad	perd	pad	perd	pad	perd	pad	perd	pad	perd	pad	perd	pad	perd	pad
1	5,04	3,31	4,47	3,40	4,55	3,83	4,55	3,83	5,03	3,19	5,28	3,06	5,06	3,14	4,47	3,13
2	0,61	6,95	0,47	7,81	0,52	6,53	0,52	6,53	1,04	5,34	1,79	5,05	1,26	5,87	0,48	5,72
3	2,70	4,96	2,52	5,89	2,56	5,75	2,56	5,75	2,56	5,05	2,79	4,97	2,60	5,25	2,56	4,76
4	0,27	10,32	0,25	14,26	0,25	9,50	0,25	9,50	0,45	8,97	1,01	7,73	0,52	8,93	0,25	7,73
5	1,17	7,63	1,10	10,75	1,08	8,59	1,08	8,59	1,10	7,07	1,33	7,87	1,12	7,87	1,07	7,29
6	0,10	13,31	0,12	25,44	0,11	13,76	0,11	13,76	0,34	12,33	0,73	12,26	0,21	13,07	0,11	12,33
7	17,83	7,66	15,41	7,90	13,64	8,33	15,11	7,67	14,67	6,89	16,23	6,67	15,22	6,77	13,56	6,84
8	15,46	9,62	15,00	9,96	13,14	9,90	13,91	9,23	14,39	7,91	16,01	7,92	14,95	7,92	12,99	8,28
9	16,20	13,64	11,00	15,03	9,92	14,90	13,47	13,08	11,98	12,92	13,57	12,47	12,66	12,91	9,78	13,25
10	12,24	18,21	10,72	19,28	9,63	18,46	11,24	16,58	12,11	15,84	14,66	15,09	13,29	15,56	9,43	17,15
11	16,19	24,60	7,33	28,74	7,35	27,37	13,28	22,60	10,59	23,43	12,40	22,49	11,42	23,20	7,09	24,56
12	11,37	33,23	7,29	37,31	7,12	34,45	12,08	28,56	10,77	28,64	13,68	27,95	12,12	29,08	7,08	33,14
13	19,77	8,93	19,17	8,97	17,49	9,20	18,16	8,79	17,98	7,79	19,40	7,84	19,00	7,83	17,41	7,89
14	18,70	10,51	18,55	10,32	17,15	10,36	17,53	10,00	17,89	8,97	20,08	8,81	18,74	8,93	17,08	9,11
15	17,83	16,28	14,76	16,88	13,83	16,93	16,32	15,52	15,90	14,93	17,30	14,71	16,29	14,36	13,75	15,18
16	15,09	19,89	14,67	19,91	13,60	19,51	14,20	18,65	15,91	16,98	17,80	17,23	16,80	16,97	13,56	18,38
17	17,41	29,76	10,30	31,46	10,22	30,78	14,94	26,84	14,12	27,00	14,94	26,84	14,69	26,72	10,09	28,13
18	11,25	37,90	10,22	38,28	10,08	36,69	10,79	35,09	14,75	32,63	16,55	32,63	15,91	32,11	10,10	35,63

Tabela 6.15 - Tempo Médio das 4 Heurísticas

Classe	$Y L_t$	FW_t	C^{ult}_t	H_{1t}	H_{2t}	H_{3t}	H_{4t}	H_{1t}^*	H_{2t}^*	H_{3t}^*	H_{4t}^*
1	0,23	0,35	0,50	0,14	0,15	0,15	0,07	0,88	0,54	3,11	0,52
2	0,48	1,26	0,79	1,25	0,39	0,42	0,26	14,45	7,47	6,68	8,65
3	0,12	2,10	0,72	10,10	2,21	3,54	2,18	12,19	7,10	3,71	5,76
4	2,75	16,41	1,02	29,70	5,47	6,24	2,61	52,08	9,05	10,76	9,22
5	3,43	40,03	0,98	15,32	7,05	8,52	7,09	34,01	16,19	18,23	14,04
6	7,81	383,3	1,42	421,00	28,06	31,06	9,60	457,07	56,09	61,28	45,01
7	0,11	0,11	0,19	0,39	0,18	0,21	0,06	0,50	0,19	0,21	0,66
8	0,60	0,24	0,50	1,74	0,54	0,54	0,29	2,03	0,56	0,57	0,30
9	0,49	1,47	0,33	3,12	0,66	0,87	0,35	3,78	0,68	0,21	0,62
10	3,36	3,40	0,60	21,83	4,92	6,27	3,62	25,14	5,27	8,01	4,73
11	7,17	36,98	0,59	53,23	9,33	12,73	5,98	73,47	30,23	22,84	12,99
12	44,62	77,41	0,85	302,50	62,51	82,92	77,32	347,7	95,47	94,64	89,47
13	0,13	0,13	0,16	0,49	0,21	0,21	0,07	0,58	0,21	0,21	0,07
14	0,25	0,18	0,46	1,73	0,48	0,54	0,29	1,80	0,56	0,45	0,29
15	0,97	1,92	0,28	4,50	0,65	0,96	0,47	4,87	1,14	0,97	0,56
16	2,46	2,71	0,55	21,07	6,39	4,82	3,19	22,08	6,89	4,80	3,47
17	15,46	51,31	0,52	86,18	14,67	15,24	8,49	73,23	16,06	17,28	9,51
18	50,61	71,34	0,77	352,4	91,55	93,69	46,90	395,10	99,39	109,13	62,57

Compara-se a seguir na [Tabela 6.16](#), a heurística 4 com o KOMBI modificado com o método proposto por [Umetani et al. \(2006\)](#). Os resultados de [Cui et al. \(2008\)](#) não são comparados porque os resultados de [Tabela 6.16](#) são dados em termos do número de objetos e não em termos da perda média. Em seu trabalho, [Umetani et al. \(2006\)](#) apresentam três resultados computacionais diferentes para o problema. No primeiro, coluna $UM_{5\%}$, utiliza-se como referência para o desperdício o valor obtido com a resolução do problema de corte com o método de geração de colunas considerando-se até 5% deste valor para cima, no segundo até 3% deste valor para cima, coluna $UM_{3\%}$ e no terceiro até 1% deste valor para cima, coluna $UM_{1\%}$. Nota-se da tabela que apenas uma das médias de soluções de H_4^* foi dominada (classe 14). Destacamos o bom desempenho da heurística $UM_{5\%}$ de [Umetani et al. \(2006\)](#) com nenhuma de suas médias de soluções dominadas e da heurística de [Foerster e Wascher \(1999\)](#), dominada em apenas 5 das 18 classes. Todas as médias de soluções de [Yanasse e Limeira \(2006\)](#) foram dominadas.

Tabela 6.16 – Testes Computacionais Adicionais

Classe	YL		FW		UM _{5%}		UM _{3%}		UM _{1%}		H ₄ *	
	objs	pad	objs	pad	objs	pad	objs	pad	objs	pad	objs	pad
1	11,56	3,31	11,49	3,40	12,24	2,43	12,24	2,43	12,24	2,43	11,48	3,13
2	110,40	6,95	110,25	7,81	114,49	3,18	113,02	3,68	111,60	4,57	110,26	5,72
3	22,17	4,96	22,13	5,89	23,53	3,98	23,08	4,42	23,08	4,42	22,13	4,76
4	215,98	10,32	215,93	14,26	223,92	4,89	221,04	5,57	218,44	7,36	215,93	7,73
5	42,99	7,63	42,96	10,75	45,29	6,68	44,76	7,35	43,95	9,32	42,95	7,29
6	424,89	13,31	424,71	25,44	442,00	7,25	435,44	8,49	429,10	12,48	424,68	12,33
7	51,69	7,66	50,21	7,90	51,35	5,52	50,98	5,72	50,62	5,92	50,26	6,84
8	502,23	9,62	499,52	9,96	507,92	5,75	503,47	6,04	500,77	6,42	499,94	8,28
9	99,49	13,64	93,67	15,03	96,35	10,05	95,49	10,44	94,40	11,38	93,89	13,25
10	948,41	18,21	932,32	19,28	954,55	10,30	946,09	10,83	936,18	12,11	934,26	17,15
11	195,67	24,60	176,97	28,74	182,88	18,32	180,93	19,30	178,34	22,08	177,62	24,56
12	1847,42	33,23	1766,20	37,31	1818,35	18,40	1799,14	19,58	1773,74	22,66	1770,87	33,14
13	64,20	8,93	63,27	8,97	64,16	6,68	63,80	6,84	63,53	7,01	63,48	7,89
14	633,26	10,51	632,12	10,32	637,03	6,85	634,50	7,02	630,50	7,43	632,39	9,11
15	123,90	16,28	119,43	16,88	122,44	12,34	121,34	12,70	120,53	13,26	119,70	15,18
16	1197,66	19,89	1191,80	19,91	1217,91	12,46	1207,63	12,91	1196,57	13,80	1193,33	18,38
17	244,02	29,76	224,68	31,46	231,41	22,44	229,14	23,09	226,62	24,49	225,44	28,13
18	2268,30	37,90	2242,40	38,28	2307,70	22,50	2282,51	23,31	2255,12	25,36	2248,16	35,63

7 CONCLUSÕES

Neste trabalho focalizamos o problema de corte de estoque unidimensional com um número reduzido de padrões. Apresentamos quatro novas heurísticas para resolução deste problema que utilizam em sua parte final uma técnica de redução de padrões da literatura denominada genericamente de KOMBI. Propomos também uma modificação deste procedimento KOMBI.

No geral os resultados computacionais obtidos com as heurísticas propostas foram bons e se mostraram competitivos com os resultados de procedimentos considerados os melhores da literatura.

O tempo computacional observado é em geral aceitável. Em algumas classes a heurística 1 requer um tempo maior de execução e isso se deve à grande quantidade de vezes que o KOMBI é aplicado.

Caso a redução do número de padrões seja prioritário, como fizeram [Cui et al. \(2008\)](#) e [SallesNeto \(2005\)](#) em seus trabalhos, outras abordagens podem ser desenvolvidas.

Alguns resultados desenvolvidos neste trabalho já foram apresentados em eventos e publicados em periódicos ([Cerqueira e Yanasse \(2008\)](#), [Cerqueira e Yanasse \(2009\)](#)), ([Yanasse e Cerqueira \(2009\)](#)). Outros trabalhos estão em fase de preparação.

Para trabalhos futuros pode-se efetuar experimentos computacionais com variações das heurísticas desenvolvidas, por exemplo aplicando-se outras formas de arredondamento para obter uma solução inteira do problema de corte, valorizando mais a redução de padrões ao invés do número de objetos.

Uma outra proposta seria procurar explorar o modelo de fluxo de arcos, para tentar reduzir a quantidade de padrões no problema de corte de estoque unidimensional, conforme propôs [Carvalho \(1999\)](#), ou seja, procurar por soluções utilizando um número menor de caminhos.

Uma outra proposta seria procurar explorar a idéia utilizada em [Umetani et al. \(2003\)](#) e tentar resolver o problema de se achar a melhor solução em termos de minimização do desperdício tendo fixado o número de diferentes padrões da solução.

Também merece ser investigada a possibilidade de se determinar a fronteira ótima de Pareto para o problema.

REFERÊNCIAS BIBLIOGRÁFICAS

- ALVES, C.; MACEDO, R.; CARVALHO, J. V. New lower bounds based on column generation and constraint programming for the pattern minimization problem. **Computers & Operations Research**, v. 36, p. 2944–2954, 2009. [82](#)
- ARENALES, M. N.; MORABITO, R.; YANASSE, H. H. Problemas de corte e empacotamento. In: SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL. **Anais...** São João del-Rei: Rio de Janeiro: SOBRAPO, 36.,2004. [25](#), [59](#)
- BARNHART, C.; JOHNSON, E.; NENHAUSER, G. L.; VANCE, P. H. Solving binary cutting stock problems by column generation and branch and bound. **Computational Optimization and Applications**, v. 3, p. 111–130, 1994. [43](#)
- BELOV, G.; SCHEITHAUER, G. Cutting plane algorithm for the one-dimensional cutting stock problem with multiple stock lengths. **European Journal of Operational Research**, v. 141, p. 274–294, 2002. [57](#)
- _____. Decomposition approaches for solving the integer one-dimensional cutting stock problem with different types of standard lengths. **European Journal of Operational Research**, v. 141, p. 295–312, 2002. [50](#)
- CARVALHO, J. M. V. Exact solution of cutting stock problem using column generation and branch and bound. **Annals of Operations Research**, v. 86, p. 629–659, 1999. [43](#), [50](#), [51](#), [133](#)
- _____. Lp models for bin packing and cutting stock problems. **European Journal of Operational Research**, v. 141, p. 253–273, 2002. [31](#), [54](#), [55](#), [58](#), [59](#)
- _____. Using extra dual cuts to accelerate column generation. **INFORMS Journal on Computing**, v. 17, p. 175–182, 2005. [43](#)
- CERQUEIRA, G. R. L.; YANASSE, H. H. Uma heurística para resolução do problema de corte de estoque unidimensional com um número reduzido de padrões distintos. In: WORKSHOP DOS CURSOS DE COMPUTAÇÃO APLICADA DO INPE, 8. **Anais do WORCAP**. São José dos Campos: INPE, 2008. p. 97–103. [133](#)
- _____. A pattern reduction procedure in a one-dimensional cutting stock problem by grouping items according to their demands. **Journal of Computational Interdisciplinary Sciences**, v. 2, p. 159–164, 2009. [133](#)

CHERRI, A. C.; ARENALES, M. N. O problema de corte de estoque com reaproveitamento das sobras de material - heurística ffd modificada. In: SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL. **Anais...** Gramado: Rio de Janeiro: SOBRAPO, 37.,2005. [85](#)

CUI, Y.; YANG, Y.; YU, P. A heuristic for the one-dimensional cutting stock problem with pattern reduction. **Engineering Manufacture**, v. 222, 2008. [69](#), [70](#), [71](#), [72](#), [83](#), [85](#), [101](#), [107](#), [108](#), [109](#), [112](#), [123](#), [125](#), [130](#), [133](#)

DIEGEL, A.; CHETTY, M.; VANSCHALKWYCK, S.; NAIDOO, S. **Setup combining in the trim loss problem 3 to 2 and 2 to 1**. Durban: Business Administration, University of Natal. Working Paper. First Draft, 1993. [74](#), [75](#), [116](#)

DIKILI, A. C.; SARIOZ, E.; PEK, N. A. A sucessive elimination method for one-dimensional stock cutting problems in ship production. **Ocean Engineering**, v. 34, p. 1841–1849, 2007. [78](#), [80](#)

DYCKHOFF, H. A new linear programming approach to the cutting stock problem. **Operations Research**, v. 29, p. 1092–1104, 1981. [52](#)

_____. A typology of cutting and packing problems. **European Journal of Operational Research**, v. 44, p. 145–159, 1990. [21](#), [26](#), [27](#), [29](#)

FARLEY, A. A.; RICHARDSON, K. V. Fixed charge problems with identical fixed charges. **European Journal of Operational Research**, v. 18, p. 245–249, 1984. [77](#)

FERREIRA, J. S.; NEVES, M. A.; CASTRO, P. F. A two-phase roll cutting problem. **European Journal of Operational Research**, v. 44, p. 185–196, 1990. [53](#)

FOERSTER, H.; WASCHER, G. Pattern reduction in one-dimensional cutting stock problems. **International Journal of Production Research**, v. 38, p. 1657–1676, 1999. [74](#), [76](#), [83](#), [86](#), [107](#), [108](#), [109](#), [112](#), [113](#), [116](#), [118](#), [120](#), [122](#), [123](#), [130](#)

GAU, T.; WASCHER, G. Cutgen1: a problem generator for one-dimensional cutting stock problem. **European Journal of Operational Research**, v. 84, p. 572–579, 1995. [107](#)

GILMORE, P.; GOMORY, R. A linear programming approach to the cutting-stock problem. **Operations Research**, v. 9, p. 849–859, 1961. [38](#), [40](#), [41](#), [53](#), [59](#), [81](#), [85](#), [95](#)

_____. A linear programming approach to the cutting-stock problem ii. **Operations Research**, v. 6, p. 863–888, 1963. [34](#), [38](#), [41](#)

GRAMANI, M. C. N. **Problema de corte bidimensional guilhotinado e restrito em 2-estágios**. Tese (Doutorado) — USP, 1997. [34](#)

HAESSLER, R. W. Controlling cutting pattern changes in one-dimensional trim problems. **Operations Research**, v. 23, p. 483–493, 1975. [63](#), [64](#), [65](#), [69](#), [81](#)

HINXMAN, A. I. The trim-loss and assortment problems: a survey. **European Journal of Operational Research**, v. 5, p. 8–18, 1980. [35](#)

KANTOROVICH, L. V. Mathematical methods of organizing and planning production. **Management Science**, v. 6, p. 366–422, 1960. [31](#)

LIMEIRA, M. S. **Redução do número de padrões em problemas de corte de estoque**. Tese (Doutorado) — INPE, 2003. [72](#)

LOPES, A. M.; ARAUJO, S. A. **Uma abordagem multiobjetivo para o problema de corte de estoque unidimensional**. São José do Rio Preto: UNESP, 2009. 88 p. Disponível em: <<http://www.otm.icmc.usp.br/webfacil/publico/File/oficina/s6/andre-malvezzi.pdf>>. Acesso em: 24 ago. 2009. [25](#), [110](#)

MARCOTTE, O. The cutting stock problem and integer rounding. **Mathematical Programming**, v. 33, p. 82–92, 1985. [43](#)

_____. An instance of the cutting stock problem for which the rounding property does not hold. **Operations Research Letters**, v. 4, p. 239–243, 1986. [44](#)

MARQUES, F. P.; ARENALES, M. N. O problema da mochila compartimentada e aplicações. **Pesquisa Operacional**, v. 22, p. 285–304, 2002. [31](#)

MARTELLO, S.; TOTH, P. **Knapsack problems: algorithms and computer implementations**. [S.l.: s.n.], 1990. [33](#), [35](#)

MCDIARMID, C. Pattern minimisation in cutting stock problems. **Discrete Applied Mathematics**, v. 98, p. 121–130, 1999. [63](#)

MORABITO, R. Modelos de otimização para o problema de corte nas indústrias de papel e papelão e de móveis. **Gestão e Produção**, v. 1, p. 59–76, 1994. [57](#)

MORETTI, A. C.; NETO, L. L. S. Nonlinear cutting stock problem model to minimize the number of different patterns and objects. **Computational and Applied Mathematics**, v. 27, p. 61–78, 2008. [25](#)

PINTO, M. J. **Algumas contribuições à resolução do problema de corte integrado ao problema de sequenciamento dos padrões**. Tese (Doutorado) — INPE, 2004. [34](#)

PISINGER, D. **Algorithms for knapsack problems**. Thesis-DIKU, University of Copenhagen, Copenhagen. 200p: [s.n.], 1993. [33](#)

POLDI, K. C.; ARENALES, M. N. Heurísticas para o problema de corte de estoque unidimensional inteiro. **Pesquisa Operacional**, v. 26, p. 473–492, 2006. [35](#), [48](#), [49](#)

PREVIERO, W. D. **A técnica de geração de colunas e aplicações**. Dissertação (Mestrado) — USP, 2001. [43](#)

RESPICIO, A. L. C. C. **Apoio à tomada de decisão no planejamento e escalonamento da produção**. Tese (Doutorado) — Universidade de Lisboa, 2003. [31](#)

SALLESNETO, L. L. **Modelo não linear para mimizar o número de objetos processados e o setup num problema de corte unidimensional**. Tese (Doutorado) — UNICAMP, 2005. [80](#), [81](#), [108](#), [133](#)

SCHEITHAUER, G.; TERNÓ, J. The modified integer round-up property of the one-dimensional cutting stock problem. **European Journal of Operational Research**, v. 84, p. 562–571, 1995. [44](#)

STADTLER, H. A one-dimensional cutting stock problem in the aluminium industry and its solution. **European Journal of Operational Research**, v. 44, p. 209–223, 1990. [45](#), [118](#)

SWEENEY, P. E.; HAESSLER, R. W. One-dimensional cutting stock for rolls with multiple quality grades. **European Journal of Operational Research**, v. 44, p. 224–231, 1990. [25](#)

- SWEENEY, P. E.; PATERNOSTER, E. R. Cutting and packing problems: a categorized, application-orientated research bibliography. **Journal of the Operations Research Society**, v. 43, p. 691–706, 1992. [28](#)
- UMETANI, S.; YAGIURA, M.; IBARAKI, T. One-dimensional cutting stock problem to minimize the number of different patterns. **European Journal of Operational Research**, v. 146, p. 388–402, 2003. [66](#), [67](#), [81](#), [133](#)
- _____. One-dimensional cutting stock problem with a given number of setups: a hybrid approach of metaheuristics and linear programming. **Journal of Mathematical Modelling and Algorithms**, v. 5, p. 43–64, 2006. [81](#), [130](#)
- VANCE, P. Branch and price algorithm for the one-dimensional cutting stock problem. **Computational Optimization and Applications**, v. 9, p. 211–228, 1998. [43](#)
- VANDERBECK, F. Computacional study of a colum generation algorithm for bin packing and stocking problems. **Mathematical Programming**, v. 86, p. 565–594, 1999. [43](#)
- _____. Exact algorithm for minimising the number of setups in the one-dimensional cutting stock problem. **Operations Research**, v. 48, p. 915–926, 2000. [63](#), [68](#), [69](#)
- WALKER, W. E. A heuristic adjacent extreme point algorithm for the fixed charge problem. **Management Science**, v. 22, p. 587–596, 1976. [77](#)
- WASCHER, G. An lp-based approach to cutting stock problems with multiple objectives. **European Journal of Operational Research**, v. 44, p. 175–184, 1990. [25](#)
- WASCHER, G.; GAU, T. Heuristics for the integer one-dimensional cutting stock problem: a computational study. **OR Spektrum**, v. 18, p. 131–144, 1996. [44](#), [45](#), [47](#)
- WASCHER, G.; HAUSSNER, H.; SCHUMANN, H. An improved typology of cutting and packing problems. **European Journal of Operational Research**, v. 183, p. 1109–1130, 2007. [29](#)

YANASSE, H. H.; CERQUEIRA, G. R. L. Uma heurística baseada em geração sequencial de padrões para o problema de corte de estoque unidimensional com um número reduzido de padrões. **Gestão e Produção**, v. 16, p. 200–208, 2009. [133](#)

YANASSE, H. H.; LIMEIRA, M. S. A hybrid heuristic to reduce the number of different patterns in cutting stock problems. **Computers & Operations Research**, v. 33, p. 2744–2756, 2006. [72](#), [73](#), [107](#), [108](#), [109](#), [112](#), [113](#), [121](#), [123](#), [130](#)

YANASSE, H. H.; SOMA, N. Y. **Finding solutions to Value independent knapsack problems**. São José dos Campos: INPE, 1992. 30 p. Disponível em: <http://urlib.net/sid.inpe.br/mtc-5390-TAE/011/1992/14.08.17.47>. Acesso em: 09 jun. 2009. [35](#)

YANASSE, H. H.; SOMA, N. Y.; MACULAN, N. An algorithm for determining the k-best solutions of the one-dimensional knapsack problem. **Pesquisa Operacional**, v. 20, p. 117–134, 2000. [35](#)

ZAK, E. J. Modeling multistage cutting stock problems. **European Journal of Operational Research**, v. 141, p. 313–327, 2002. [53](#)