Algumas extensões do problema de corte de estoque¹

Kelly Cristina Poldi

Orientador: *Prof. Dr. Marcos Nereu Arenales*

Dissertação apresentada ao Instituto de Ciências Matemáticas e de Computação - ICMC-USP, como parte dos requisitos para obtenção do título de Mestre em Ciências de Computação e Matemática Computacional.

USP - São Carlos - SP

Maio de 2003

¹Apoio FAPESP

Aos meus pais.

Agradecimentos

À Deus, por ter sempre me guiado e iluminado.

Ao meu orientador Marcos Nereu Arenales pela orientação, dedicação, amizade, e principalmente, pela confiança depositada no desenvolvimento deste trabalho e de outros que ainda deverão ser desenvolvidos.

À meus pais Arnaldo e Neiva, pelo amor, carinho, educação, apoio incondicional, incentivo e presença em todos os anos de minha vida.

Aos meus irmãos Debbye e Junior, e à toda minha família pelo incentivo e carinho durante esses anos.

Ao Maurício pelo apoio, compreensão, paciência, amizade, amor e por ter estado sempre ao meu lado.

Aos professores e funcionários do ICMC-USP que, direta ou indiretamente, contribuíram com este trabalho.

A todos os meus amigos de São Carlos por estarem sempre presentes, seja nas horas de trabalho, seja nos momentos de descontração.

À FAPESP pela credibilidade e apoio financeiro.

E finalmente a todos que colaboraram indiretamente na realização deste trabalho.

Resumo

Neste trabalho, visamos estudar o problema de corte de estoque inteiro unidimensional, o qual consiste em cortar um conjunto de barras disponíveis em estoque para a produção de itens menores tal que a demanda seja atendida e um objetivo seja otimizado (minimizar custos, perdas). Estudamos o caso no qual a quantidade de barras estão disponíveis em quantidades ilimitadas e quando há limitações nos números de barras em estoque.

Apresentamos alguns métodos heurísticos para obtenção da solução inteira, que podem ser classificados como construtivos ou residuais. Estas heurísticas são analisadas empiricamente, isto é, por meio dos experimentos computacionais realizados para um conjunto de problemas gerados aleatoriamente. Ao compararmos as abordagens residuais com as heurísticas puramente construtivas, pudemos claramente concluir que as heurísticas residuais apresentaram resultados muito melhores que as abordagens puramente construtivas, e, em particular, a heurística Nova, proposta neste trabalho, apresentou os melhores resultados.

Um subproduto importante desta análise contraria o folclore na área de corte e empacotamento de que heurísticas construtivas são as mais adequadas para a solução do problema de corte de estoque quando há baixa demanda. As heurísticas residuais, baseadas na geração de colunas, são muito superiores.

Outro assunto abordado é a redução do número de padrões de corte. Estudamos uma heurística para este tipo de problema, e analisamos seu desempenho. Notamos que a heurística apresentada funcionou muito bem para apenas certas classes de problemas, com itens pequenos em relação às barras em estoque.

Abstract

In this work we have studied the classical integer cutting stock problem, which consists of cutting a set of available bars in stock in order to produce ordered smaller items in such a way as to optimise a given objective function (e.g., minimising costs, waste, etc.). We have studied the cases in which the quantities of bars are limited or not and the demand is high or low.

Some heuristic methods are given in order to obtain an integer solution, which can be classified as constructive or residual. These heuristics are empirically analysed, i.e., by solving a set of randomly generated instances. The analysis showed that residual heuristics are much more superior than constructive ones. In particular, a new proposed residual heuristic obtained better results than the others.

An important by-product from the empirical analysis contradicts the idea that specialists believe in terms of cutting and packing, for which the constructive heuristics should be used if one is faced with low demand problems. The residual heuristics, based on a linear optimisation approach, are clearly superior.

Another topic briefly investigated in this work is the reduction of the number of cutting patterns (i.e., set up minimisation). We analysed only a recent heuristic for this kind of problem. It was possible to conclude that the heuristic works well, however only in some instances, where the items are much smaller than the bars in stock.

Conteúdo

1	Introdução	1
1.1	Classificação dos Problemas de Corte quanto à Dimensão	2
1.2	Objetivos	5
1.3	Organização do Texto	7
2	Breve Revisão Bibliográfica	9
3	Problema de Corte de Estoque Unidimensional	13
3.1	Definição e Formulação Matemática	13
3.1.1	Modelo Básico	13
3.1.2	Vários tipos de barras em estoque em quantidade limitada	18
3.2	Método Simplex com Geração de Colunas e Aplicação ao Problema de Corte	21
3.2.1	Método Simplex	22
3.2.2	Geração de Colunas	26
3.3	Problema da Mochila	30
3.4	Uma Revisão dos Métodos de Resolução para o Problema da Mochila Restrito	32
3.4.1	Programação Dinâmica	32
3.4.2	Método da Enumeração Implícita (<i>Branch-and-Bound</i>)	34
3.5	Observações Importantes para o Problema de Corte com Restrições de Estoque	36
3.5.1	Encontrando uma Solução Básica Inicial	37
4	Heurísticas de Construção para o Problema de Corte de Estoque Inteiro	43
4.1	Heurística FFD	43
4.1.1	Exemplo	45

4.2	Heurística Gulosa	47
4.2.1	Exemplo	48
5	Heurísticas Residuais	53
5.1	Problema Residual	53
5.2	Heurísticas Residuais	54
5.3	Heurísticas Residuais por Truncamento	56
5.3.1	Exemplo	56
5.3.2	Heurística Residual FFD	58
5.3.3	Exemplo	58
5.3.4	Heurística Residual Gulosa	60
5.3.5	Exemplo	60
5.4	Heurística Residual <i>Nova</i>	61
5.4.1	Exemplo	63
5.5	Heurística Residual <i>Nova</i> - versão 2	66
5.5.1	Exemplo	66
6	Resultados Computacionais - Heurísticas de Arredondamento	71
6.1	O Gerador Aleatório	71
6.2	Resultados	72
6.3	Análise dos Resultados	76
6.4	Outra análise	78
7	Alguns Exemplos Resolvidos	81
7.1	Exemplo 1	81
7.2	Exemplo 2	85
7.3	Exemplo 3	89
7.4	Exemplo 4	94
7.5	Exemplo 5	99
8	O Problema de Redução de Padrões de Corte	105
8.1	Uma Heurística para Redução dos Padrões de Corte	106
8.1.1	Exemplo	107

9 Resultados Computacionais - Redução de Padrões de Corte	113
9.1 O Gerador Aleatório	114
9.2 Resultados	115
9.3 Análise dos Resultados	117
10 Alguns Exemplos Resolvidos	119
10.1 Exemplo 1	119
10.2 Exemplo 2	121
11 Conclusões e Propostas	125
11.1 Conclusões	125
11.2 Extensões a serem investigadas	125
Referências Bibliográficas	127

Capítulo 1

Introdução

A otimização do processo de cortagem de peças maiores, disponíveis em estoque, para a produção de peças menores, em quantidades encomendadas, tem sido objeto de estudo há 4 décadas, desde os trabalhos pioneiros de Gilmore e Gomory [4], [5] e [6]. Trata-se de um problema importante (devido a várias aplicações industriais) e interessante (devido à complexidade computacional) da otimização combinatória.

Problemas de corte de estoque são essenciais no planejamento da produção em várias indústrias, tais como indústrias de papel, vidro, móveis, plástico, chapas de madeira, aço, têxtil etc. Estes problemas consistem em produzir um conjunto de itens, para atender uma certa demanda, a partir da cortagem de peças em estoque, otimizando uma função objetivo, como por exemplo, minimizar o número total de peças em estoque cortadas ou as perdas. Em geral, estes problemas são formulados como problemas de otimização linear inteira. Em 1961, Gilmore e Gomory [4] relaxaram a condição de integralidade e propuseram um método eficiente de geração de colunas para a resolução do problema linear resultante. Entretanto, devido à sua complexidade e pela inexistência de algoritmos exatos que resolvam problemas de tamanhos médios, várias heurísticas foram propostas para a solução do problema inteiro (isto é, problema com a condição de integralidade). A importância econômica das aplicações, aliada à complexidade de resolução, tem motivado muitas pesquisas na área, aumentando com isto o número de publicações na área.

Não é conhecido nenhum algoritmo exato para encontrar a solução ótima inteira de problemas de corte práticos, de tamanho médio, de modo que, então, devem ser tratados por métodos heurísticos. Várias heurísticas têm sido propostas na literatura para o problema unidimensional: Wäscher e Gau [28], Pinto [21] e Haessler [8].

Outro tópico abordado é a redução do número de padrões de corte. O alto número de padrões de corte pode tornar-se um inconveniente de ordem prática. Existem poucos trabalhos na literatura que tratam este tipo de objetivo, podemos citar: Limeira e Yanasse [12], Haessler [7] e Foerster e Wäscher [2].

1.1 Classificação dos Problemas de Corte quanto à Dimensão

Problemas de corte e empacotamento aparecem na literatura sob vários nomes diferentes, como por exemplo, problema de corte de estoque, *trim loss*, problema de empacotamento de *bins*, problema de carregamento de paletes, contêineres, etc. Estes problemas podem ser classificados de acordo com as dimensões relevantes do objeto a ser cortado.

Unidimensional

Um problema é dito *unidimensional* quando apenas uma dimensão é relevante no processo de cortagem. Sua formulação e métodos de resolução serão tratados no Capítulo 3. Veja a Figura 1.1.

Figura 1.1: (a) Objeto (barra) a ser cortado; (b) Objeto cortado produzindo 4 itens e uma perda.

Problemas de corte unidimensional têm várias aplicações industriais, como por exemplo, na indústria de papel (Gilmory e Gomory [4], [5]), onde grandes rolos têm de ser cortados em tamanhos encomendados, na indústria de aço (Marques [13] e Hoto [10]), onde bobinas de aço são cortadas em bobinas menores e, também, na indústria de alumínio (Stadtler [25]), onde um produtor de perfis de alumínio fornece aos clientes o serviço de calcular o número mínimo de peças necessárias para atender aos pedidos.

Bidimensional

O problema é dito *bidimensional* quando duas dimensões (comprimento e largura) são significativas no processo de cortagem. Resolver este tipo de problema consiste em combinar geometricamente os itens ao longo do comprimento e da largura das peças em estoque. Veja a Figura 1.1. Este problema ocorre, por exemplo, em indústrias de placas de vidro, de madeira, etc (Vianna [27] e Pinto [21]).

Figura 1.2: **(a)** *Objeto (chapa) a ser cortado*; **(b)** *objeto cortado*.

Tridimensional

No problema *tridimensional* as três dimensões são significativas no processo de corte. Este tipo de problema é aplicado, por exemplo, em indústrias de matéria-prima para a produção de colchões, travesseiros etc. Mas, suas aplicações mais interessantes são na solução de um problema contrário ao de corte, o chamado *problema de empacotamento*, que

consiste, basicamente, em empacotar unidades pequenas dentro de uma unidade grande, de tal forma que um certo objetivo seja otimizado; por exemplo, arranjar o maior volume possível de caixas dentro de um contêiner (Morabito [15]). Veja a Figura 1.1.

Figura 1.3: (a) *Contêiner*; (b) *exemplo de empacotamento*.

1.5, 2.5-dimensional e multidimensional

Considerando ainda o aspecto geométrico, podemos encontrar problemas do tipo *1.5-dimensional*, que são essencialmente problemas bidimensionais, porém uma das duas dimensões consideradas é variável. Um exemplo desse tipo de problema é o de um rolo de tecido que tem largura fixa e comprimento suficientemente grande para a produção de roupas.

Podemos encontrar, também, problemas do tipo *2.5-dimensional*, que também pode ser considerado como um problema tridimensional, ou seja, três dimensões são relevantes para o problema, porém uma delas é variável. Um exemplo seria um contêiner com largura e comprimento fixas e suficientemente alto para acomodar um volume da carga.

Quando mais de três dimensões são relevantes para a solução do problema temos um problema *multidimensional*. Por exemplo, no problema de alocação de tarefas (Morabito [15]).

1.2 Objetivos

No problema de corte de estoque, quando a demanda é suficientemente alta, uma aproximação por um modelo de otimização linear é adequado e o método de geração de colunas de Gilmore e Gomory [6] é muito eficiente. Cada coluna do modelo linear corresponde a um padrão de corte sobre um objeto (isto é, o arranjo dos itens sobre o objeto).

Dependendo da aplicação em estudo, diferentes regras são usadas para definição de um padrão de corte, tais como, cortes do tipo guilhotina, limitação de itens, estágios, etc. Na indústria de móveis, por exemplo, os cortes são guilhotinados (são utilizadas serras para cortar placas de madeira que sempre produzem novos retângulos) e é comum que o máximo de estágios seja dois. Nas indústrias papelreira ou metalúrgica, bobinas são cortadas e é comum que o número de facas disponíveis seja limitado, produzindo com isto limitações no padrão de corte (neste caso unidimensional).

Quando a demanda é baixa, há um folclore na área de problemas de corte que a aproximação por otimização linear pode não ser adequada. Neste caso, heurísticas construtivas têm sido utilizadas, as quais têm a seguinte estrutura:

- i)* Um ‘padrão eficiente’ é gerado;
- ii)* o padrão eficiente é utilizado repetidas vezes, até que a demanda de um ou mais itens seja satisfeita;
- iii)* os itens com as demandas atendidas são excluídos e as demais demandas atualizadas;
- iv)* repete-se os passos acima até que toda a demanda seja satisfeita.

Embora estas duas abordagens básicas, isto é, a abordagem por otimização linear com geração de colunas e a abordagem construtiva, sejam bastante exploradas na literatura, há algumas questões subjacentes que serão investigadas neste trabalho:

- a)* *solução inteira da abordagem por otimização linear;*
- b)* *controle no número de padrões de corte;*

c) efeito da baixa demanda.

A primeira questão decorre do fato que, a solução básica encontrada pelo método simplex na resolução do problema de corte é, em geral, fracionária e a solução executável para o problema de corte de estoque (número de vezes que um objeto é cortado segundo um padrão) deva ser inteira. Várias heurísticas têm sido propostas na literatura para o problema unidimensional (Wäscher e Gau [28]) e algumas delas foram estendidas para o problema bidimensional em Pinto [21]. Haessler [8] propôs uma pequena alteração no processo de geração de colunas (apenas uma certa porcentagem da demanda pode ser atendida por um único padrão de corte) e observou que a questão do arredondamento da solução fracionária foi bastante contornada para problemas unidimensionais.

Um inconveniente de ordem prática é o alto número de diferentes padrões de corte, sendo alguns deles utilizados um número baixo de vezes, diminuindo a produtividade da máquina de cortagem (algumas máquinas cortam simultaneamente até 30 placas com o mesmo padrão). Uma abordagem consiste em incluir na função objetivo uma espécie de penalidade, caso um padrão de corte seja utilizado e desenvolver heurísticas para sua resolução (Limeira e Yanasse [12], Foerster e Wäscher [2] e Haessler [7]).

A dificuldade com baixas demandas está também relacionada com a questão da data de entrega. Indústrias, geralmente pequenas, aceitam pedidos de clientes de poucos itens, de dimensões muito particulares, que não coincidem com pedidos de outros clientes. Este problema é crucial para pequenas indústrias, pois é significativo o número de tais clientes, podendo representar mais de 50% do faturamento e, as perdas, que decorrem de itens não padronizados, são altas. O uso da heurística construtiva, descrita anteriormente, pode levar a alguns impasses, pois a carteira de pedidos é bastante dinâmica e com prazos de entrega bastante apertados. A heurística tende a combinar certos itens de clientes distintos, gerando inicialmente excelentes padrões de corte (do ponto de vista de perda de material) e péssimos padrões no final. Os primeiros padrões seriam então executados e os finais rejeitados (aqueles que apresentarem perda acima de uma tolerância admitida). Como a carteira de pedidos é bastante dinâmica (novos pedidos chegam todos os dias, às vezes vários por dia), os itens rejeitados seriam então reconsiderados com os novos e, a

heurística aplicada novamente.

Baixa demanda surge também em indústrias de médio e grande porte, como por exemplo indústrias metalúrgicas, onde bobinas de aço são recortadas. É comum que as bobinas de aço em estoque sejam cortadas ao longo de todo seu comprimento, gerando sub-bobinas, que muitas vezes, uma única unidade atende a demanda.

Neste trabalho mostramos que as heurísticas construtivas podem levar a soluções desastrosas e as heurísticas residuais são superiores, contrariando um folclore que na área dos problemas de corte e empacotamento que a abordagem por otimização linear é inadequada para tratar problemas com demanda baixa.

Finalmente, os objetivos principais deste projeto podem ser resumidos como:

- Heurísticas de arredondamento para o problema de corte de estoque unidimensional, com vários tipos de barras disponíveis em estoque, analisando, principalmente, o efeito da baixa demanda;
- problema de redução do número de padrões de corte.

1.3 Organização do Texto

No Capítulo 2, apresentamos uma breve revisão bibliográfica de algumas publicações relacionadas ao problema em questão, em ordem cronológica.

No Capítulo 3, apresentamos um modelo básico para o problema de corte de estoque e, a seguir, o modelo com limitação no número de barras disponíveis em estoque. Apresentamos o método simplex com o processo de geração de colunas para resolver problemas lineares relaxados. O problema da mochila surge como sub-problema na geração de colunas, tratado nas seções 3.3 e 3.4. No final deste capítulo, fazemos algumas observações importantes para o problema de corte com restrições de estoque.

No Capítulo 4, apresentamos dois métodos heurísticos construtivos para determinar soluções inteiras para o problema de corte de estoque.

No Capítulo 5, apresentamos heurísticas residuais para obtenção da solução inteira para o problema de corte de estoque.

No Capítulo 6, descrevemos um gerador de exemplos aleatórios, utilizado para criar 360 exemplos-teste para analisar o desempenho das heurísticas. Alguns destes exemplos estão detalhados no Capítulo 7.

No Capítulo 8, tratamos do problema de redução do número de padrões de corte, que é resolvido por uma heurística proposta por Limeira e Yanasse [12]. No Capítulo 9, descrevemos outro gerador aleatório que foi utilizado para testar o desempenho desta heurística e no Capítulo 10 detalhamos alguns destes exemplos.

Finalmente, no Capítulo 11, apresentamos as conclusões finais e as propostas futuras para continuidade deste trabalho.

Capítulo 2

Breve Revisão Bibliográfica

Problemas de corte de estoque começaram a ser estudados por volta de 1940, embora as principais pesquisas tenham surgido apenas nos anos 60. Nota-se que as pesquisas nesta área têm caminhado no sentido de desenvolver técnicas heurísticas adequadas para a resolução de tais problemas, pois são da classe NP-completo e técnicas exatas, tais como métodos de enumeração implícita, não podem ser utilizadas para resolver problemas práticos.

Na década de 60 surgiram vários trabalhos importantes publicados na área. As modelagens e métodos de resolução de maior repercussão na literatura foram os publicados por Gilmore e Gomory [4], [5] e [6]. Em 1961, Gilmore e Gomory apresentaram o método simplex com geração de colunas para um modelo de otimização linear (uma aproximação para o problema) que, pela primeira vez, resolveu problemas grandes de corte unidimensional. Em Gilmore e Gomory [5], um novo método para o problema da mochila foi apresentado e realizado um estudo de caso no corte de papel.

Em 1965, Gilmore e Gomory [6] apresentam um método para resolução do problema de corte de estoque bidimensional. Foram impostas algumas restrições; tais como o corte guilhotinado, estagiado e irrestrito. Tais restrições são encontradas na prática com frequência. O modelo principal deste trabalho com limitação de estoque (veja seção (3.1.2)) foi formulado em [6], porém visando o balanceamento de máquinas.

Em 1975, Haessler [7] apresenta um procedimento heurístico, computacionalmente

eficiente, para resolver problemas de corte de estoque unidimensional com carga fixa associado à troca de padrões de corte. Esta heurística utiliza *níveis de aspiração* para determinar se um padrão de corte encontrado deve ou não ser usado.

No início da década de 80, foram propostas algumas modificações no algoritmo de Gilmore e Gomory [4] e [5] por Haessler [8], onde o objetivo de minimizar a perda de material utilizado foi ampliado. Tais modificações se referem aos procedimentos que geram a solução inicial e na geração dos padrões a entrar na base. Segundo o autor, controlar a geração dos padrões a entrar na base e usar uma formulação mais restrita, porém menos eficiente, do problema da mochila, ajuda a reduzir problemas de arredondamento e mudanças de padrões de corte.

Hinxman [9], em 1980, faz uma *revisão* dos problemas de corte em suas várias dimensões e aponta estrita ligação entre o problema da mochila e o problema de corte de estoque. O autor também classifica os métodos para solução do problema de corte em categorias: *branch-and-bound*, programação dinâmica e repetição exaustiva (que aqui chamamos de heurísticas construtivas).

Iniciando os anos 90, Stadler [25] realizou um estudo de caso em uma indústria de alumínio, onde o cálculo do número mínimo de peças em estoque necessárias para atender os pedidos dos clientes deveria ser determinado. Neste caso, os resultados obtidos pela heurística FFD (caso particular de heurísticas construtivas, veja seção (4.1)) foram considerados insatisfatórios então apresentou-se um novo método, baseado no processo de Geração de Colunas proposto por Gilmore e Gomory [4], [6], acrescido de um processo de arredondamento para obtenção de uma solução inteira (heurística residual).

Considerando ainda os trabalhos da década de 90, citamos a abordagem em grafo E/OU para solucionar o problema de corte bidimensional guilhotinado não-estagiado e irrestrito proposta por Morabito [14].

Wäscher e Gau [28], em 1995, desenvolveram um gerador para o problema de corte de estoque unidimensional padrão. Também em 1995, Scheithauer e Terno publicaram um artigo conjecturando que o problema de corte de estoque unidimensional tem a pro-

priedade MIRUP. Esta propriedade ocorre quando o valor ótimo do problema de corte de estoque inteiro unidimensional não é maior do que o valor ótimo do problema relaxado por programação linear correspondente, arredondado para o inteiro superior mais um, quando o número de objetos cortados define a função objetivo. Em 1995, Morabito e Arenales [18] analisaram o desempenho de duas heurísticas para resolver problemas grandes de corte guilhotinado bidimensional. Os autores estenderam a abordagem feita em Morabito [14], onde agora problemas restritos e estagiados são tratados, utilizando ainda a representação do problema em grafo E/OU.

Vahrenkamp [26], em 1996, faz um estudo comparativo entre o algoritmo de Gilmore e Gomory para a solução do problema de corte de estoque unidimensional com uma heurística baseada em algoritmos genéticos. A estrutura randômica dos algoritmos genéticos leva a uma escolha randômica dos padrões de corte, com o objetivo de minimizar a perda de material. Segundo o autor, esta abordagem randômica para seleção dos padrões ainda não tinha sido considerada na literatura.

Em 1996, Wäscher e Gau [28] reuniram em um artigo vários métodos heurísticos para a resolução do problema de corte de estoque inteiro unidimensional, apresentando um estudo computacional. Alguns deles estão descritos no Capítulo 5.

Foerster e Wäscher [2], em 2000, estudando o planejamento da produção com o objetivo de minimizar a perda de material e também de diminuir o número de padrões de corte, apresentaram um novo método para o segundo problema, que pode ser considerado como uma generalização dos métodos propostos anteriormente na literatura. A qualidade das soluções obtidas por esse novo método se mostraram claramente superior aos métodos existentes, mas houve um aumento no tempo computacional para algumas classes de problemas. Concluiu-se que, com relação a redução dos padrões de corte, esse algoritmo provou ser superior ao proposto por Haessler, para uma grande classe de problemas.

Em 2001, Limeira e Yanasse [12] propuseram uma heurística para redução do número de padrões de corte para problemas de corte de estoque. Esta heurística está descrita no Capítulo 8 e baseia-se em níveis de aspirações, tal como proposto por Haessler [7].

Em processos industriais, pode ocorrer também casos em que se deseja determinar a “melhor” seqüência na qual os padrões serão cortados. O problema de seqüenciamento de padrões de corte depende do problema de corte e vice-versa. Frequentemente, uma boa solução para o problema de corte não resulta numa boa solução para o problema de seqüenciamento, e vice-versa. Existe, portanto, um *tradeoff* entre os objetivos dos dois problemas. Este foi o objeto de estudo, em 2002, de Pileggi [20].

Capítulo 3

Problema de Corte de Estoque Unidimensional

3.1 Definição e Formulação Matemática

O problema de corte de estoque unidimensional pode ser definido da seguinte forma:

Considere que temos em estoque um número suficientemente grande de *objetos* (barras, bobinas, etc) de um determinado comprimento L e um conjunto de pedidos de barras menores de comprimentos $l_i, i = 1, \dots, m$, os quais chamaremos de *itens*. Cada item i deve ser produzido na quantidade $d_i, i = 1, \dots, m$. O problema então consiste em produzir os itens a partir do corte das peças em estoque de modo a atender a demanda, otimizando uma certa função, por exemplo, minimizar o número total de peças a serem cortadas, ou minimizar a perda, ou maximizar o lucro, etc. Veja um exemplo na Figura 3.1.

3.1.1 Modelo Básico

Por simplicidade, considere inicialmente apenas um tipo de barra em estoque. Assim, a seguir, o problema de corte de estoque unidimensional é modelado.

Figura 3.1: (a) *Peça em estoque*; (b) *Tipos de itens demandados*; (c) *Exemplos de possíveis padrões de corte*.

Definição 3.1 Chamamos de **padrão de corte** a maneira como um objeto em estoque é cortado para a produção dos itens demandados. A um padrão de corte associamos um vetor m -dimensional que contabiliza os itens produzidos:

$$\mathbf{a} = (\alpha_1, \alpha_2, \dots, \alpha_m)$$

onde α_i é a quantidade de itens do tipo i no padrão de corte. Veja Figura 3.1(c). Além disso, dois padrões de corte que tenham o mesmo vetor associado são chamados *equivalentes*.

Observe que um vetor $\mathbf{a} = (\alpha_1, \alpha_2, \dots, \alpha_m)$ corresponde a um padrão de corte se e somente se satisfizer as restrições do Problema da Mochila (considerando apenas as restrições físicas):

$$\begin{aligned} l_1\alpha_1 + l_2\alpha_2 + \dots + l_m\alpha_m &\leq L \\ \alpha_i &\geq 0, i = 1, \dots, m \text{ e inteiro.} \end{aligned} \tag{3.1}$$

Outras restrições de processo (por exemplo, limitação no número de facas) podem ser incluídas em (3.1).

Por exemplo, considere o problema de corte unidimensional com objetos em estoque de comprimento $L = 11$ a serem cortados para a produção de três tipos de itens ($m = 3$), com comprimentos l_i e demandas $d_i, i = 1, \dots, 3$ descritos na Tabela 3.1. A Figura 3.2 apresenta alguns possíveis padrões de corte unidimensionais com seus respectivos vetores $(\mathbf{a}_j)$ e perda (c_j) associados.

Item	Comprimento	Demanda
1	3	4
2	4	3
3	5	2

Tabela 3.1: *Dados do exemplo do problema de corte.*

Figura 3.2: *Exemplos de padrões de corte e seus respectivos vetores e perdas associados.*

Definição 3.2 Um padrão de corte que produza apenas um tipo de item é chamado padrão de corte homogêneo.

Em outras palavras, um padrão de corte é homogêneo se o vetor associado tem apenas uma coordenada não-nula: $(0, \dots, \alpha_i, \dots, 0), \alpha_i \neq 0$. Os padrões 1, 2 e 3 da

Figura 3.2 são exemplos de padrões homogêneos. Note que sempre teremos m padrões homogêneos, cujos vetores associados definem uma matriz diagonal. A hipótese de que $l_i \leq L$, para todo i , está implícita, caso contrário o problema seria infactível. Os valores α_i são determinados por $\alpha_i = \left\lfloor \frac{L}{l_i} \right\rfloor, i = 1, \dots, m$.

Após definirmos os padrões de corte, o próximo passo será determinar o número de vezes que cada padrão será utilizado para resolver o problema, ou seja, a modelagem matemática de um problema de corte de estoque é feita em duas etapas:

1. Definir todos os possíveis padrões de corte;
2. Definir quantas vezes cada padrão de corte será utilizado para atender a demanda, que deverá ser um número inteiro e não-negativo.

Suponha que temos um estoque composto por barras de comprimento L , em quantidade ilimitada e, conforme introduzido anteriormente, faremos a modelagem matemática em duas etapas:

- Defina todos os possíveis padrões de corte, isto é, determine todas as possíveis soluções de (3.1). Suponha que sejam n as possíveis soluções:

$$\mathbf{a}_1 = \begin{bmatrix} \alpha_{11} \\ \alpha_{21} \\ \vdots \\ \alpha_{m1} \end{bmatrix}, \quad \mathbf{a}_2 = \begin{bmatrix} \alpha_{12} \\ \alpha_{22} \\ \vdots \\ \alpha_{m2} \end{bmatrix}, \quad \dots \quad \mathbf{a}_n = \begin{bmatrix} \alpha_{1n} \\ \alpha_{2n} \\ \vdots \\ \alpha_{mn} \end{bmatrix}$$

onde α_{ij} é o número itens do tipo i no padrão de corte j , $i = 1, \dots, m, j = 1, \dots, n$.

- Seja $x_j = \text{número de vezes que o objeto é cortado usando o padrão } j$.

O problema de corte de estoque, considerando como objetivo minimizar o número de objetos em estoque a serem cortados, pode ser formulado como um problema de otimização linear inteiro:

$$\begin{aligned} &\text{minimizar} && f(\mathbf{x}) = x_1 + x_2 + \dots + x_n \\ &\text{sujeito a:} && \begin{cases} \mathbf{a}_1 x_1 + \mathbf{a}_2 x_2 + \dots + \mathbf{a}_n x_n = \mathbf{d} \\ x_j \geq 0, j = 1, \dots, n \text{ e inteiro.} \end{cases} \end{aligned} \tag{3.2}$$

Em notação matricial o problema (3.2) é escrito como segue:

$$\begin{aligned} \text{minimizar } f(\mathbf{x}) &= \sum_{j=1}^n x_j \\ \text{sujeito a: } \mathbf{Ax} &= \mathbf{d}, \quad \mathbf{x} \geq \mathbf{0} \quad \text{e inteiro} \end{aligned} \tag{3.3}$$

onde cada coluna da matriz $\mathbf{A} \in \Re^{m \times n}$ é um vetor associado a um padrão de corte, conforme a definição 3.1.

A função objetivo em (3.3) minimiza o total de objetos a serem cortados. As primeiras restrições em (3.3) garantem que a quantidade de itens produzidos seja exatamente igual à demanda, e as últimas restrições em (3.3) garantem que a repetição de cada padrão de corte j seja um número inteiro não-negativo.

Podemos, ainda considerar outros objetivos como, por exemplo, minimizar a perda total. Neste caso, a função objetivo em (3.3) seria substituída pela função *perda total*, que é definida como:

$$f(\mathbf{x}) = \sum_{j=1}^n c_j x_j \tag{3.4}$$

onde

$$c_j = L - (\alpha_{1j}l_1 + \alpha_{2j}l_2 + \dots + \alpha_{mj}l_m)$$

define a perda no padrão de corte j , $j = 1, \dots, n$ (veja Figura 3.2).

A minimização da função $f(\mathbf{x})$ em (3.4) resulta no conjunto de padrões de corte com perda mínima. Observamos que, neste caso simples, onde há apenas um tipo de objeto em estoque, são equivalentes: minimizar o total de objetos cortados e minimizar a perda total. Segue a prova desta afirmação.

Considere o seguinte problema (que minimiza a perda):

$$\begin{aligned} \text{minimizar } f(\mathbf{x}) &= \sum_{j=1}^n c_j x_j \\ \text{sujeito a: } \mathbf{Ax} &= \mathbf{d}, \quad \mathbf{x} \geq \mathbf{0} \quad \text{e inteiro.} \end{aligned} \tag{3.5}$$

Como $c_j = L - (\alpha_{1j}l_1 + \alpha_{2j}l_2 + \dots + \alpha_{mj}l_m)$, podemos reescrever o problema (3.5) como:

$$\begin{aligned}
& \text{minimizar} & f(\mathbf{x}) &= \sum_{j=1}^n (L - \sum_{i=1}^m l_i \alpha_{ij}) x_j \\
& \text{sujeito a:} & \begin{cases} \sum_{j=1}^n \alpha_{ij} x_j = d_i, & i = 1, \dots, m \\ \mathbf{x} \geq \mathbf{0} \text{ e inteiro.} \end{cases}
\end{aligned} \tag{3.6}$$

A função objetivo de (3.6) pode ser reescrita como:

$$f(\mathbf{x}) = L \sum_{j=1}^n x_j - \sum_{j=1}^n \sum_{i=1}^m l_i \alpha_{ij} x_j. \tag{3.7}$$

Pelas propriedades comutativa da adição e distributiva, podemos reescrever (3.7) como:

$$f(\mathbf{x}) = L \sum_{j=1}^n x_j - \sum_{i=1}^m l_i \sum_{j=1}^n \alpha_{ij} x_j. \tag{3.8}$$

Vemos que o último somatório em (3.8) é igual ao lado esquerdo da restrição em (3.6): $\sum_{j=1}^n \alpha_{ij} x_j = d_i$. Então, fazendo a substituição teremos a seguinte função objetivo:

$$\text{minimizar} \quad L \sum_{j=1}^n x_j - \sum_{i=1}^m l_i d_i. \tag{3.9}$$

Como $L > 0$ e $\sum_{i=1}^m l_i d_i$ são constantes, temos que a função objetivo original (minimizar perda) é equivalente a minimizar o total de barras a serem cortadas: $\sum_{j=1}^n x_j$.

Portanto, minimizar o total de barras cortadas e minimizar a perda total são objetivos equivalentes quando temos apenas um tipo de barra disponível em estoque e restrições de igualdade.

3.1.2 Vários tipos de barras em estoque em quantidade limitada

No modelo anterior foram considerados apenas os dados dos itens (comprimento e demanda) e tínhamos apenas um tipo de barra em estoque. Apresentamos agora um modelo no qual temos vários tipos de barras em estoque e em quantidade limitada. Devemos considerar os seguintes dados:

Dados de demanda:

- m : número de tipos de itens;
- l_i : dimensões do item i , $i = 1, \dots, m$;
- d_i : demanda do item tipo i , $i = 1, \dots, m$.

Dados de estoque:

- K : número de tipos de objetos em estoque;
- L_k : dimensões do objeto k , $k = 1, \dots, K$;
- e_k : disponibilidade em estoque do objeto k , $k = 1, \dots, K$.

O modelo matemático é análogo ao caso anterior. Entretanto, os padrões de corte devem ser definidos para cada tipo de barra em estoque, isto é, devem satisfazer:

$$\begin{aligned} l_1\alpha_{1k} + l_2\alpha_{2k} + \dots + l_m\alpha_{mk} &\leq L_k \\ \alpha_{ik} &\geq 0, i = 1, \dots, m \text{ e inteiro, } k = 1, \dots, K. \end{aligned} \tag{3.10}$$

Suponha que o sistema acima (3.10) tenha N_k soluções, dadas por:

$$\mathbf{a}_{1k} = \begin{bmatrix} \alpha_{11k} \\ \alpha_{21k} \\ \vdots \\ \alpha_{m1k} \end{bmatrix}, \quad \mathbf{a}_{2k} = \begin{bmatrix} \alpha_{12k} \\ \alpha_{22k} \\ \vdots \\ \alpha_{m2k} \end{bmatrix}, \quad \dots \quad \mathbf{a}_{N_k k} = \begin{bmatrix} \alpha_{1N_k k} \\ \alpha_{2N_k k} \\ \vdots \\ \alpha_{mN_k k} \end{bmatrix}, \quad k = 1, \dots, K$$

onde α_{ijk} é o número de itens do tipo i no padrão de corte j para o objeto k , $i = 1, \dots, m, j = 1, \dots, N_k, k = 1, \dots, K$ ($\mathbf{a}_{jk}$ é o vetor correspondente ao j -ésimo padrão de corte sobre o objeto k).

Consideramos um dado adicional para a descrição do modelo matemático:

- c_{jk} : custo do objeto k , segundo o padrão de corte j , $j = 1, \dots, N_k, k = 1, \dots, K$.

Observe que c_{jk} pode ser a perda: $c_{jk} = L_k - \sum_{i=1}^m l_i \alpha_{ijk}$ ou simplesmente um custo do objeto k (que independe do padrão de corte): $c_{jk} = c_k$.

Algumas vezes, minimizar o custo é mais interessante do que minimizar a perda, pois os objetos podem ser comprados com descontos (por ter dimensões fora dos padrões) ou são retalhados de cortes anteriores que retornaram ao estoque e deseja-se que estes sejam utilizados antes que os outros objetos.

Variáveis de decisão:

- x_{jk} : número de vezes que o objeto do tipo k é cortado usando o padrão j , $j = 1, \dots, N_k$, $k = 1, \dots, K$.

O problema pode então ser formulado por:

$$\begin{aligned}
 \text{minimizar} \quad & f(x_{11}, x_{12}, \dots) = \sum_{j=1}^{N_1} c_{j1} x_{j1} + \sum_{j=1}^{N_2} c_{j2} x_{j2} + \dots + \sum_{j=1}^{N_K} c_{jK} x_{jK} \\
 \text{sujeito a:} \quad & \left\{ \begin{array}{ll} \sum_{j=1}^{N_1} \mathbf{a}_{j1} x_{j1} + \sum_{j=1}^{N_2} \mathbf{a}_{j2} x_{j2} + \dots + \sum_{j=1}^{N_K} \mathbf{a}_{jK} x_{jK} & = \mathbf{d} \\ \sum_{j=1}^{N_1} x_{j1} & \leq e_1 \\ \sum_{j=1}^{N_2} x_{j2} & \leq e_2 \\ & \vdots \\ \sum_{j=1}^{N_K} x_{jK} & \leq e_K \\ x_{jk} \geq 0, \text{ e inteiro } j = 1, \dots, N_k, \quad k = 1, \dots, K. \end{array} \right. \quad (3.11)
 \end{aligned}$$

Note que o modelo básico (3.3), com um único tipo de objeto em estoque, é apenas um caso particular deste modelo (3.11) com $K = 1$ e e_1 suficientemente grande. Observamos que as colunas da matriz de restrições são os vetores associados aos padrões de corte para cada objeto, agora complementado com K componentes com 1 na posição $(m + k)$ e 0 nas demais. Assim, uma coluna j do modelo (3.11) pode ser escrita como em (3.12):

$$\begin{bmatrix} \alpha_{1jk} \\ \vdots \\ \alpha_{mjk} \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix} \quad (3.12)$$

A primeira linha de restrições garante que a quantidade de itens produzidos seja exatamente igual à demanda. As outras restrições (de $\leq$) garantem que a quantidade de cada barra disponível em estoque não seja violada. E a última restrição garante que a repetição de cada padrão de corte j seja um número inteiro não-negativo.

3.2 Método Simplex com Geração de Colunas e Aplicação ao Problema de Corte

A condição de integralidade sobre as variáveis x_j torna os problemas de corte (3.3) e (3.11) difíceis, senão impossíveis, de serem resolvidos computacionalmente. Para contornar este problema relaxamos a condição de integralidade sobre as variáveis x_j e resolvemos o problema de otimização linear resultante pelo método simplex. Em problemas práticos, m (que é a quantidade de tipos de itens) é da ordem de dezenas, enquanto que n (o qual depende de m, L e $l_i, i = 1, \dots, m$) pode ser da ordem de milhões, o que inviabiliza a resolução direta do problema. Para contornar esta dificuldade, utilizamos o processo de *geração de colunas* (seção 3.2.2) na resolução da relaxação por otimização linear de (3.3) e (3.11) e também alguns métodos heurísticos para arredondamento da solução, na seção 5, que serão importantes na resolução do problema de corte de estoque inteiro.

3.2.1 Método Simplex

Por simplicidade, considere o problema de otimização (3.3) escrito em notação matricial (a extensão para o problema (3.11) é simples):

$$\begin{aligned} &\text{minimizar} && f(\mathbf{x}) = \mathbf{c}^T \mathbf{x} \\ &\text{sujeito a:} && \mathbf{A}\mathbf{x} = \mathbf{d}, \quad \mathbf{x} \geq \mathbf{0} \end{aligned} \tag{3.13}$$

onde $\mathbf{A} \in \mathbb{R}^{m \times n}$ e posto $(\mathbf{A}) = m$. Note que, para o problema de corte, sempre existirão m colunas linearmente independentes na matriz $\mathbf{A}$, formadas pelos padrões homogêneos, o que garante posto $(\mathbf{A}) = m$.

A solução geral do sistema de equações em (3.13) pode ser descrita da seguinte maneira. Admita a seguinte partição básica: $\mathbf{A} = [\mathbf{B}, \mathbf{N}]$. A matriz $\mathbf{B}$, onde $\mathbf{B} \in \mathbb{R}^{m \times m}$, é chamada de *matriz básica* e é composta por m colunas linearmente independentes da matriz $\mathbf{A}$ (é inversível) e a matriz $\mathbf{N}$ é chamada de *matriz não-básica*.

Esta partição nas colunas da matriz $\mathbf{A}$ define também uma partição nos vetores $\mathbf{x}$ e $\mathbf{c}$: $\mathbf{x} = [\mathbf{x}_B, \mathbf{x}_N]$ e $\mathbf{c} = [\mathbf{c}_B, \mathbf{c}_N]$; $\mathbf{x}_B$ é chamado *vetor das variáveis básicas* e $\mathbf{x}_N$ *vetor das variáveis não-básicas (ou variáveis livres)*. Reescrevendo o sistema (3.13) com as partições definidas acima obteremos:

$$\begin{aligned} &\text{minimizar} && f(\mathbf{x}) = \mathbf{c}_B^T \mathbf{x}_B + \mathbf{c}_N^T \mathbf{x}_N \\ &\text{sujeito a:} && \begin{cases} \mathbf{B}\mathbf{x}_B + \mathbf{N}\mathbf{x}_N = \mathbf{d} \\ \mathbf{x}_B \geq \mathbf{0}, \quad \mathbf{x}_N \geq \mathbf{0}. \end{cases} \end{aligned}$$

A solução geral do sistema de equações lineares que definem as restrições pode ser escrita da seguinte forma:

$$\mathbf{B}\mathbf{x}_B + \mathbf{N}\mathbf{x}_N = \mathbf{d} \Leftrightarrow \mathbf{x}_B = \mathbf{B}^{-1}\mathbf{d} - \mathbf{B}^{-1}\mathbf{N}\mathbf{x}_N.$$

A atribuição de valores às variáveis não-básicas $\mathbf{x}_N$ faz com que as variáveis básicas $\mathbf{x}_B$ fiquem bem determinadas.

A solução particular obtida por $\mathbf{x}_B^0 = \mathbf{B}^{-1}\mathbf{d}$ e $\mathbf{x}_N^0 = \mathbf{0}$ é chamada *solução básica*. Se $\mathbf{x}_B^0 = \mathbf{B}^{-1}\mathbf{d} \geq \mathbf{0}$ então a solução básica é dita *factível* e, neste caso, diz-se que a partição básica é factível. Demonstra-se que se o problema (3.13) tem solução ótima, então existe

uma solução básica ótima. Reescrevendo a função objetivo restrita às soluções do sistema linear, teremos:

$$\begin{aligned} f(\mathbf{x}) &= \mathbf{c}_B^T (\mathbf{B}^{-1} \mathbf{d} - \mathbf{B}^{-1} \mathbf{N} \mathbf{x}_N) + \mathbf{c}_N^T \mathbf{x}_N \\ &= \mathbf{c}_B^T \mathbf{B}^{-1} \mathbf{d} + (\mathbf{c}_N^T - \mathbf{c}_B^T \mathbf{B}^{-1} \mathbf{N}) \mathbf{x}_N. \end{aligned}$$

Observe que $f(\mathbf{x}^0) = \mathbf{c}_B^T \mathbf{B}^{-1} \mathbf{d} = \mathbf{c}_B^T \mathbf{x}_B^0$ é o valor da função objetivo para a solução básica. Chamamos o vetor $\pi \in \Re^m$, dado por $\pi = \mathbf{c}_B^T \mathbf{B}^{-1}$ o vetor das *variáveis duais* ou *vetor multiplicador simplex*. Se

$$(c_j - \pi^T a_j) \geq 0, \quad i = 1, \dots, n$$

então π é uma *solução básica dual-factível*. Neste caso, dizemos que a partição é *dual-factível*. Com isso, a função objetivo $f(\mathbf{x})$ (em termos das variáveis não-básicas) é dada por:

$$f(\mathbf{x}) = f(\mathbf{x}^0) + \sum_{j \in \mathcal{N}} (c_j - \pi^T a_j) x_j$$

onde a_j corresponde à coluna j de $\mathbf{A}$ e $\mathcal{N}$ é o conjunto dos índices das colunas da matriz $\mathbf{N}$ (colunas não-básicas da matriz $\mathbf{A}$). Isto mostra como a função objetivo se altera quando alterações são feitas nas variáveis não-básicas. Os coeficientes das variáveis não-básicas $x_j : c_j - \pi^T a_j$ são chamados *custos reduzidos*.

A *estratégia simplex* é definida como uma alteração da solução básica, através da perturbação de apenas uma componente de $\mathbf{x}_N$. Como queremos minimizar $f(\mathbf{x})$ e $x_j \geq 0$ para todo $j \in \mathcal{N}$, a componente x_k a ser perturbada deve ser tal que $(c_k - \pi^T a_k) < 0$.

Fazendo $x_k = \varepsilon \geq 0$ e $x_j = 0, \forall j \in \mathcal{N} - \{k\}$, as variáveis básicas são alteradas resultando em uma nova solução dada por:

$$\begin{aligned} \mathbf{x}_B &= \mathbf{x}_B^0 + \varepsilon \mathbf{y}, \\ \mathbf{x}_N &= \varepsilon \mathbf{e}_k \end{aligned} \tag{3.14}$$

e o valor da função objetivo é dado por:

$$f(\mathbf{x}) = f(\mathbf{x}^0) + (c_k - \pi^T a_k) \varepsilon \tag{3.15}$$

onde $\mathbf{y} = -\mathbf{B}^{-1} \mathbf{a}_k$ e $\mathbf{e}_k = (0, \dots, 1, \dots, 0)^T \in \Re^{n-m}$, com 1 na k -ésima componente.

Note que a direção $\mathbf{s} \in \mathbb{R}^n$, dada por: $\mathbf{s} = (\mathbf{s}_B, \mathbf{s}_N)^T = (\mathbf{y}, \mathbf{e}_k)^T$, define uma perturbação da solução básica e é chamada *direção simplex*. Se a solução básica for não-degenerada, isto é, $\mathbf{x}_B^0 > 0$, de (3.14) e (3.15) segue que $\mathbf{s}$ é uma direção factível e de descida. Note ainda, que o produto escalar da direção simplex $\mathbf{s}$ e o gradiente da função objetivo é $\mathbf{c}^T \mathbf{s} = (c_k - \pi^T a_k) < 0$.

De (3.14) podemos determinar o maior valor de ε , impondo $\mathbf{x}_B \geq 0$, já que as demais restrições são satisfeitas $\forall \varepsilon \geq 0$:

$$\varepsilon^0 = -\frac{x_{B_i}^0}{y_i} = \min \left\{ -\frac{x_{B_i}^0}{y_i} \mid y_i < 0, \quad i = 1, \dots, m \right\}$$

onde $x_{B_i}^0$ é a i -ésima componente de $\mathbf{x}_B^0$. Naturalmente, se $y_i \geq 0, i = 1, \dots, m$, então não há limitante para ε , $f(\mathbf{x}) \rightarrow \infty$ (a direção $\mathbf{s}$ é um raio de descida) e o problema (3.13) não tem solução ótima.

Para esta escolha de ε , a l -ésima componente de $\mathbf{x}_B$ se anula, enquanto que apenas uma variável de $\mathbf{x}_N$ tornou-se positiva: $x_k = \varepsilon^0 > 0$ (a menos de degeneração: $\varepsilon^0 = 0$). Isto sugere uma nova partição básica $\mathbf{A} = [\mathbf{B}', \mathbf{N}']$, na qual a l -ésima coluna de $\mathbf{B}'$ é dada por $\mathbf{a}_k \in \mathbf{N}$. Esta nova partição é básica primal-factível, cuja solução básica associada é dada por: $\mathbf{x}^1 = \mathbf{x}^0 + \varepsilon^0 \mathbf{s}$.

Podemos escolher qualquer k tal que $(c_k - \pi^T a_k) < 0$. Uma *heurística gulosa*, chamada *regra de Dantzig*, consiste em escolher a direção simplex com o menor custo reduzido, isto é:

$$(c_k - \pi^T a_k) = \min \{ (c_j - \pi^T a_j), j = 1, \dots, n \}.$$

Para determinar o menor custo reduzido é preciso examinar todas as n colunas da matriz $\mathbf{A}$. Note que, caso todos os custos reduzidos sejam não-negativos: $(c_j - \pi^T a_j) \geq 0, \forall j \in \mathcal{N}$, a solução básica é ótima.

Para ilustrar, um pequeno problema de corte unidimensional será resolvido utilizando o método simplex considerando a matriz $\mathbf{A}$ formada pelos padrões de corte da Figura 3.2. Observe que o conjunto de padrões da Figura 3.2 formam a matriz $\mathbf{A}$. Os dados da demanda estão na Tabela 3.1 e objetos em estoque são de comprimentos $L = 11$.

Suponha que o objetivo seja a minimização da perda.

A matriz $\mathbf{A}$ é dada por:

$$\mathbf{A} = \begin{bmatrix} 3 & 0 & 0 & 2 & 1 & 0 \\ 0 & 2 & 0 & 0 & 2 & 1 \\ 0 & 0 & 2 & 1 & 0 & 1 \end{bmatrix}$$

onde a coluna 1 corresponde ao padrão de corte 1, a coluna 2 ao padrão de corte 2, e assim sucessivamente, veja Figura 3.2. Seja a matriz $\mathbf{B}$ formada pelas colunas $[a_1, a_2, a_3]$,

$$\mathbf{B} = \begin{bmatrix} 3 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix}$$

A solução básica $\mathbf{x}_B = [x_1, x_2, x_3]^T$ é dada por:

$$\mathbf{x}_B = \mathbf{B}^{-1}\mathbf{d} = \begin{bmatrix} \frac{1}{3} & 0 & 0 \\ 0 & \frac{1}{2} & 0 \\ 0 & 0 & \frac{1}{2} \end{bmatrix} \cdot \begin{bmatrix} 4 \\ 3 \\ 2 \end{bmatrix} = \begin{bmatrix} \frac{4}{3} \\ \frac{3}{2} \\ 1 \end{bmatrix}$$

e o vetor multiplicador simplex:

$$\pi = \mathbf{c}_B \mathbf{B}^{-1} = \begin{bmatrix} 2 & 3 & 1 \end{bmatrix} \cdot \begin{bmatrix} \frac{1}{3} & 0 & 0 \\ 0 & \frac{1}{2} & 0 \\ 0 & 0 & \frac{1}{2} \end{bmatrix} = \begin{bmatrix} \frac{2}{3} \\ \frac{3}{2} \\ \frac{1}{2} \end{bmatrix}$$

Os custos reduzidos das variáveis não-básicas devem ser calculados para testar a otimalidade.

$$\begin{aligned} \hat{c}_4 = c_4 - \pi^T a_4 = 0 & - \begin{bmatrix} \frac{2}{3} & \frac{3}{2} & \frac{1}{2} \end{bmatrix} \cdot \begin{bmatrix} 2 \\ 0 \\ 1 \end{bmatrix} = -\frac{11}{6} \\ \hat{c}_5 = c_5 - \pi^T a_5 = 0 & - \begin{bmatrix} \frac{2}{3} & \frac{3}{2} & \frac{1}{2} \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 2 \\ 0 \end{bmatrix} = -\frac{11}{3} \\ \hat{c}_6 = c_6 - \pi^T a_6 = 2 & - \begin{bmatrix} \frac{2}{3} & \frac{3}{2} & \frac{1}{2} \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} = 0 \end{aligned}$$

Como as colunas a_4 e a_5 apresentam custos relativos negativos, qualquer uma delas poderia ser escolhida para entrar na base, pois isso faria com que o valor da função objetivo diminuísse. O menor custo relativo, neste caso, define a variável x_5 para ser aumentada. A coluna a_5 entrará na base no lugar da coluna a_2 . Continuando este procedimento mais uma mudança na base é realizada: a coluna a_4 entra no lugar da coluna a_1 . A solução ótima obtida é $x_4^* = 1,25$, $x_5^* = 1,5$ e $x_3^* = 0,375$ e com perda $f(\mathbf{x}^*) = 0,375$.

No exemplo acima, em cada iteração simplex foi calculado todos os custos reduzidos (necessitando-se explicitamente de todas as colunas da matriz $\mathbf{A}$) para se determinar o menor deles. Como em problemas reais este número é extremamente grande, a avaliação computacional de todos os custos relativos torna-se inviável. Gilmore e Gomory [4] propuseram um método (conhecido por *geração de colunas*) que, em cada iteração simplex, procura implicitamente (ou seja, sem a necessidade de percorrer explicitamente todas as colunas da matriz $\mathbf{N}$) uma coluna não-básica candidata a substituir uma coluna básica. O procedimento consiste em gerar a coluna j (isto é, um padrão de corte) cujo valor $c_j - \pi a_j$ seja mínimo.

3.2.2 Geração de Colunas

O método *simplex* pode ser usado na resolução de problemas de corte de estoque. Em problemas práticos, o número de variáveis (*colunas*) n é muito maior que o número de restrições (*linhas*) m . Cada coluna representa um padrão de corte. Entretanto, durante as iterações do método Simplex, temos de determinar uma nova coluna para entrar na base. Este problema torna-se impraticável devido ao grande número de colunas que devem ser investigadas. Para contornar este problema, Gilmore e Gomory [4], em 1961, propuseram uma técnica de geração de colunas que é bastante eficiente para resolver o problema.

Este procedimento, consiste em gerar uma coluna k , isto é, um novo padrão de corte, utilizando o critério de Dantzig, que procura a variável x_k com o menor custo relativo, o que sugere o seguinte sub-problema:

$$\mathbf{c}_k - \pi^T \mathbf{a}_k = \min \left\{ \mathbf{c}_j - \pi^T \mathbf{a}_j, \quad j = 1, 2, \dots \right\} \quad (3.16)$$

onde π é o vetor multiplicador Simplex de uma determinada iteração.

A formulação do problema (3.20) corresponde ao *problema da mochila restrito*, ou seja, deve ser levado em conta a limitação da demanda de cada item. Para resolver o problema da mochila, duas técnicas são mais utilizadas, a saber: programação dinâmica, na seção 3.4.1 e enumeração implícita, na seção 3.4.2. Mas, antes disso, veremos o algoritmo primal-simplex com geração de colunas e faremos uma breve discussão sobre alguns pontos importantes a serem considerados no método simplex com geração de colunas.

No caso do problema de corte de estoque com baixa demanda, a escolha da matriz básica inicial com os padrões homogêneos, como usados no modelo básico, pode não ser adequada, pois pode ser que o corte de uma única barra conforme o padrão homogêneo já ultrapasse a demanda de algum item. Então, a matriz $\mathbf{B}$ será uma matriz diagonal (de padrões homogêneos), mas a quantidade do item no padrão deve ser:

$$b_{ii} = \min \left\{ \left\lfloor \frac{L}{l_i} \right\rfloor, d_i \right\}, \quad i = 1, \dots, m \quad (3.17)$$

onde $\lfloor x \rfloor$ é o maior inteiro menor ou igual a x . Observe que se a demanda d_i fosse grande não seria necessário utilizar a função *mínimo* em (3.17).

No caso do problema com restrições de estoque, modelo (3.11), a matriz básica inicial é construída com os padrões homogêneos da maior barra disponível em estoque, pois pode ocorrer de algum item a ser produzido ser maior que algum dos tipos de barras disponíveis.

Dependendo da função que desejamos minimizar, que pode ser o total de barras cortadas (caso 1), ou a perda total (caso 2), teremos algumas modificações no método Simplex com geração de colunas. Estas diferenças são descritas a seguir.

Caso 1: Quando o objetivo é minimizar o total de barras a serem cortadas (modelo (3.3)), temos o vetor custo $\mathbf{c}$ igual a $(1, 1, \dots, 1)$. Assim,

$$\min \{ \mathbf{c}_j - \pi^T \mathbf{a}_j \} = 1 - \max \{ \pi^T \mathbf{a}_j \}.$$

Com isso, encontrar uma coluna não-básica, que denotamos genericamente por $(\alpha_1, \alpha_2, \dots, \alpha_m)^T$, que substituirá uma coluna básica, corresponde a resolver o problema:

$$\begin{aligned}
&\text{maximizar} && \pi_1\alpha_1 + \pi_2\alpha_2 + \dots + \pi_m\alpha_m \\
&\text{sujeito a que} && (\alpha_1, \alpha_2, \dots, \alpha_m)^T \text{corresponda a um padrão de corte.}
\end{aligned} \tag{3.18}$$

Para que a coluna $\mathbf{a}_j = (\alpha_1, \alpha_2, \dots, \alpha_m)^T$ corresponda a um padrão de corte, basta que satisfaça as restrições:

$$\begin{aligned}
&l_1\alpha_1 + l_2\alpha_2 + \dots + l_m\alpha_m \leq L \\
&0 \leq \alpha_i \leq d_i, i = 1, \dots, m \quad \text{e inteiro.}
\end{aligned} \tag{3.19}$$

Assim, o problema (3.18) poderá ser reescrito como:

$$\begin{aligned}
&g(\mathbf{a}) = \text{máximo } \pi_1\alpha_1 + \pi_2\alpha_2 + \dots + \pi_m\alpha_m \\
&\text{sujeito a: } \begin{cases} l_1\alpha_1 + l_2\alpha_2 + \dots + l_m\alpha_m \leq L \\ 0 \leq \alpha_i \leq d_i, i = 1, \dots, m \quad \text{e inteiro,} \end{cases}
\end{aligned} \tag{3.20}$$

que é um problema da mochila. Desta forma, se o menor dos custos relativos (veja (3.16)) são não-negativos, isto é, se $1 - g(\mathbf{a}) \geq 0$, podemos garantir que a solução do problema (3.3) relaxado (contínua) é ótima.

O valor de utilidade de cada item é o multiplicador simplex: $v_i = \pi_i, i = 1, \dots, m$.

Caso 2: Quando o objetivo torna-se minimizar a perda total, como definido em (3.4), teremos o seguinte custo $\mathbf{c}$ associado ao padrão correspondente à coluna $\mathbf{a} = (\alpha_1, \dots, \alpha_m)^T$: $\mathbf{c} = L - \sum_{i=1}^m l_i\alpha_i$. Portanto, a função objetivo do sub-problema (3.16) é dada por:

$$\mathbf{c} - \pi^T \mathbf{a} = (L - \sum_{i=1}^m l_i\alpha_i) - \sum_{i=1}^m \pi_i\alpha_i = L - \sum_{i=1}^m (l_i + \pi_i)\alpha_i.$$

O sub-problema (3.16) é descrito, então, pelo seguinte problema da mochila:

$$\begin{aligned}
&\text{maximizar} && \sum_{i=1}^m (l_i + \pi_i)\alpha_i \\
&\text{sujeito a: } && \begin{cases} \sum_{i=1}^m l_i\alpha_i \leq L \\ 0 \leq \alpha_i \leq d_i, i = 1, \dots, m \quad \text{e inteiro.} \end{cases}
\end{aligned} \tag{3.21}$$

Apresentamos agora o algoritmo do método primal-simplex com geração de colunas utilizado na resolução dos problemas de corte de estoque.

Algoritmo

Início

FASE 1:

- Gere a matriz básica inicial $\mathbf{B}$, conforme (3.17):

$$\mathbf{B} = \begin{bmatrix} b_{11} & 0 & \dots & 0 \\ 0 & b_{22} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & b_{mm} \end{bmatrix}$$

- Faça: $\text{PARE} = \textit{Falso}$ e $\text{IT} = 1$.

{ A variável lógica PARE terá valor *Falso* até que a condição de otimalidade seja verificada e IT é o número da iteração atual. }

FASE 2:

Enquanto $\text{PARE} = \textit{Falso}$, faça:

{ Início da iteração Simplex }

2.1 Determine a solução básica corrente: $\mathbf{B}\mathbf{x} = \mathbf{d}$.

2.2 Determine a solução dual: $\mathbf{B}^T\pi = \mathbf{c}_B$.

2.3 Resolva o Problema da Mochila: { Gera uma coluna para entrar na base. }

$$g(\mathbf{a}) = \text{máximo } v_1\alpha_1 + v_2\alpha_2 + \dots + v_m\alpha_m$$

$$\text{sujeito a: } \begin{cases} l_1\alpha_1 + l_2\alpha_2 + \dots + l_m\alpha_m \leq L \\ 0 \leq \alpha_i \leq d_i, i = 1, \dots, m \text{ e inteiro.} \end{cases}$$

Obs: $v_i = \pi_i$ no caso 1 e $v_i = \pi_i + l_i$ no caso 2.

Seja $\mathbf{a} = (\alpha_1, \alpha_2, \dots, \alpha_m)^T$ a solução do problema da mochila.

2.4 { Teste de Otimalidade }

Se $1 - g(\mathbf{a}) \geq 0$ no caso (1), ou $L - g(\mathbf{a}) \geq 0$ no caso (2)

então PARE = *Verdadeiro* { Solução atual é ótima e foi obtida em IT iterações. }

senão **Início**

2.4.1 { Determine $\mathbf{y}$ as coordenadas básicas da direção simplex: }

$$\mathbf{B}\mathbf{y} = -\mathbf{a}$$

2.4.2 { Determinação do Tamanho do Passo }

Encontre l tal que:

$$-\frac{x_l}{y_l} = \text{mínimo} \left\{ -\frac{x_i}{y_i} \mid y_i < 0, \quad i = 1, \dots, m \right\}.$$

2.4.3 { Atualização }

Atualize a base, substituindo a l -ésima coluna da matriz $\mathbf{B}$ por $\mathbf{a}$ (obtida no passo 2.3).

Faça: $\text{IT} = \text{IT} + 1$.

Fim-senão

Fim

3.3 Problema da Mochila

O problema descrito a seguir é conhecido na literatura como *problema da mochila* (*knap-sack problem*). Associando-o ao problema de corte unidimensional, temos a situação

onde uma barra deve ser cortada ao longo de seu comprimento L em itens de tamanhos $l_1, l_2, \dots, l_m$. Considerando a limitação na quantidade de itens a serem selecionados, (essa situação surge, especialmente, em problemas de corte com baixa demanda), o problema é dito *problema da mochila restrito ou limitado*. Para este problema o limite máximo para cada item: o vetor $\mathbf{d}$, deve ser respeitado.

Variável de decisão:

- α_i : quantidade de itens do tipo i selecionados, $i = 1, \dots, m$.

Modelagem Matemática:

$$\begin{aligned} g(\mathbf{a}) = & \text{máximo } v_1\alpha_1 + v_2\alpha_2 + \dots + v_m\alpha_m \\ \text{sujeito a: } & \begin{cases} l_1\alpha_1 + l_2\alpha_2 + \dots + l_m\alpha_m \leq L \\ 0 \leq \alpha_i \leq d_i, i = 1, \dots, m \text{ e inteiro} \end{cases} \end{aligned} \quad (3.22)$$

onde:

L = comprimento da peça em estoque;

m = número de tipos de itens;

l_i = comprimento do item i , $i = 1, \dots, m$;

v_i = valor de utilidade do item i , $i = 1, \dots, m$;

d_i = quantidade máxima de itens do tipo i que pode ser cortado, $i = 1, \dots, m$.

O problema da mochila surge como sub-problema na solução de um problema de corte de estoque usando-se o método simplex. No passo 2.3 do algoritmo simplex, temos de determinar uma coluna a entrar na base, esta coluna é a solução de um problema da mochila, para o qual o valor de utilidade do item i é o i -ésimo multiplicador simplex, no caso (1) ou $\pi_i + l_i$, no caso (2), como já foi comentado na seção anterior.

3.4 Uma Revisão dos Métodos de Resolução para o Problema da Mochila Restrito

O problema da mochila pode ser resolvido por uma sequência de decisões que consiste em incluir um item por vez, dentre os m tipos de itens disponíveis. Inicialmente, espaço disponível na mochila é L . Ao tomarmos a decisão de colocarmos o item i , de comprimento l_i , na quantidade α_i , restará um espaço de $L - \alpha_i l_i$ a ser preenchido. Esse espaço resultante poderá ser preenchido escolhendo-se outro item do tipo j na quantidade α_j , restará, então, um espaço de $L - \alpha_i l_i - \alpha_j l_j$. Este processo é repetido até que o espaço restante não seja suficiente para incluir nenhum tipo de item. Quando forem investigadas todas as possibilidades de inclusão, teremos percorrido todas as possíveis soluções do problema da mochila.

Existem na literatura vários métodos para a resolução deste problema, sendo eles: programação dinâmica, algoritmos de enumeração implícita, métodos lagrangianos e heurísticos e abordagens em redes.

Descrevemos agora dois métodos computacionais conhecidos na literatura que foram utilizados para a resolução do *problema da mochila* (3.22). Os métodos são *programação dinâmica* e *enumeração implícita*. Ambos foram implementados, mas somente o segundo é utilizado nos resultados computacionais.

3.4.1 Programação Dinâmica

A *programação dinâmica* caracteriza-se por um processo de mudanças entre os possíveis *estados* do problema, cada estado definindo uma configuração diferente do problema. Escolhas devem ser feitas para mudar de um estado para outro e, por isto, este processo é chamado de *processo de decisão*. Se apenas uma decisão deve ser tomada, então o processo ocorre em um simples *estágio*; se uma sequência de decisões devem ser tomadas, então ele é chamado *processo de decisão multi-estágio*.

A programação dinâmica é um procedimento de otimização aplicável a problemas

solucionáveis usando-se esta sequência de decisões, por sua vez, resultando numa sequência de estados ou caminho e, num valor associado a cada caminho que deve ser minimizado ou maximizado. Duas características são comuns nestes procedimentos:

- (i) um dado problema pode ser resolvido se as melhores soluções para certos estados deste problema forem determinadas;
- (ii) os estados finais deste problema devem ser tão simples a ponto de terem soluções triviais.

Um método de programação dinâmica para a resolução do problema da mochila consiste numa fórmula recursiva utilizando uma estratégia de percorrer todas as possíveis decisões, ou seja, a solução de um nó representando um espaço livre X , cujo valor da função objetivo é denotado por $G_i(X)$, é determinada assim que os nós predecessores estiverem resolvidos.

Considere inicialmente o problema restrito (3.22). Seja $G_i(X)$ o máximo valor obtido ao combinar os comprimentos $l_1, l_2, \dots, l_i, i = 1, \dots, m$ ao longo do comprimento $X, X = 0, 1, \dots, L$. Ou seja,

$$G_i(X) = \text{máximo} \sum_{k=1}^i v_k \alpha_k$$

$$\text{sujeito a: } \begin{cases} \sum_{k=1}^i l_k \alpha_k \leq X \\ 0 \leq \alpha_k \leq d_k, k = 1, \dots, i \text{ e inteiro.} \end{cases}$$

Os valores $X = 0, 1, \dots, L$ são chamados *estados* e representam diferentes configurações do problema durante o processo de decisão. Seja

$$l_0 = \min \{ l_1, l_2, \dots, l_m \}$$

definido como o menor comprimento dentre todos os itens. O comprimento X deve ser preenchido (ou cortado) pelos comprimentos $l_1, l_2, \dots, l_m$ até restar, no máximo, o comprimento $l_0 - 1$ de L . Definem-se $X = L$ como o estado inicial e $X = 0, 1, \dots, l_0 - 1$ como os estados finais.

Note que $G_i(X) = 0, i = 1, \dots, m$, para todos os estados finais. Para os estados $X = l_0, l_0 + 1, \dots, L$, este valor pode ser obtido com

$$G_i(X) = \max_{\alpha_i} \{ v_i \alpha_i + G_{i-1}(X - l_i \alpha_i) \}, \quad i = 1, \dots, m \quad (3.23)$$

$$0 \leq \alpha_k \leq d_k, \quad k = 1, \dots, i \quad \text{e inteiro}$$

onde $G_0(y) = 0$, para qualquer y .

Cada valor $G_i(X)$ pode ser encontrado *recursivamente* e, por isto, (3.23) é chamada *fórmula recursiva*. $G_m(X)$ é o valor máximo obtido para o estado X e depende de uma escolha de $\alpha_i, i = 1, \dots, m$, o que corresponde a um estágio do processo de decisão. Este valor é resultado da seqüência de decisões (*multi-estágios*) tomadas até o estágio X . Note que $G_m(L)$ corresponde à solução ótima do problema da mochila (3.22).

3.4.2 Método da Enumeração Implícita (*Branch-and-Bound*)

O seguinte método da *enumeração implícita* foi proposto por Gilmore e Gomory [5], em 1963, para o problema da mochila. Este método é implementado utilizando uma busca em profundidade primeiro e consiste em enumerar implicitamente todas as soluções do problema da mochila. Para isto, utiliza-se *limitantes* para o problema que permitem que soluções “piores” que a atual sejam descartadas, sem perder a otimalidade. Este método é importante, pois gerar explicitamente todas as possíveis soluções do problema da mochila, para exemplos de tamanhos moderados, se torna um trabalho quase impossível de ser computacionalmente resolvido. O método de enumeração implícita para a resolução do problema da mochila restrito está descrito a seguir. Mais detalhes veja Marques [13].

Algoritmo

Início

Passo 1: { Defina o problema segundo as variáveis mais valiosas. }

Faça: $\theta_i = \frac{v_i}{l_i}, i = 1, \dots, m$, e reordene as variáveis, tal que:

$$\theta_1 \geq \theta_2 \geq \dots \geq \theta_m.$$

Faça: $v_{m+1} = 0$, $l_{m+1} = 1$ e $g(\mathbf{a}) = 0$.

Passo 2: { Determine a solução inicial, utilizando busca em profundidade primeiro. }

Determine a solução $\mathbf{a} = (\alpha_1, \alpha_2, \dots, \alpha_m)$, tal que:

$$\alpha_1 = \min \left\{ \left\lfloor \frac{L}{l_1} \right\rfloor, d_1 \right\} \Rightarrow \text{sobra : } L - \alpha_1 l_1$$

$$\alpha_2 = \min \left\{ \left\lfloor \frac{L - \alpha_1 l_1}{l_2} \right\rfloor, d_2 \right\} \Rightarrow \text{sobra : } L - \alpha_1 l_1 - \alpha_2 l_2$$

$\vdots$

$$\alpha_k = \min \left\{ \left\lfloor \frac{L - \sum_{j=1}^{k-1} l_j \alpha_j}{l_k} \right\rfloor, d_k \right\} \quad k = 2, \dots, m$$

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

Determine:

$$g(\mathbf{a}) = \sum_{i=1}^m v_i \alpha_i.$$

(Considere inicialmente $\underline{G} = 0$). **Se** $\underline{G} < g(\mathbf{a})$ **então** faça:

$$\underline{G} = g(\mathbf{a})$$

e guarde a solução correspondente: $\mathbf{a}^* = \mathbf{a}$

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

Determine k o maior índice tal que: $\alpha_k \neq 0$.

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

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

$$\bar{G}(\mathbf{a}) = v_1 \alpha_1 + v_2 \alpha_2 + \dots + v_k (\alpha_k - 1) + \frac{v_{k+1}}{l_{k+1}} (L - l_1 \alpha_1 - l_2 \alpha_2 - \dots - l_k (\alpha_k - 1))$$

Passo 5: { *Backtracking* }

5.1: { Retorno longo }

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

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

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

$$\alpha_j = \min \left\{ \left\lfloor \frac{L - \sum_{i=1}^{j-1} l_i \alpha_i}{l_j} \right\rfloor, d_j \right\}, \quad j = k + 1, \dots, m$$

e volte ao Passo 3.

{ A quantidade de itens do tipo $k + 1, k + 2, \dots, m$ que é possível colocar no espaço que sobrou com a retirada de uma unidade do item tipo k . }

Fim

Ao final do algoritmo, $g(\mathbf{a})$ guarda o valor máximo para o problema (3.22). Ou seja, está determinado o padrão de corte $(\alpha_1^*, \alpha_2^*, \dots, \alpha_m^*)^T$ mais valioso, uma vez que o algoritmo examina implicitamente todos os possíveis padrões de corte e determina o melhor deles.

3.5 Observações Importantes para o Problema de Corte com Restrições de Estoque

O procedimento de geração de colunas discutido na seção 3.2 para o problema (3.3) pode ser facilmente estendido para o problema (3.11), quando novas restrições devido à limitação de estoque devem ser consideradas.

3.5.1 Encontrando uma Solução Básica Inicial

Para encontrarmos uma matriz básica inicial adequada ao problema de corte com restrições de estoque (3.11) é preciso observar alguns pontos que descreveremos a seguir.

De início, variáveis de folga $\mu_k \geq 0, k = 1, \dots, K$, serão necessárias em cada uma das restrições de disponibilidade de estoque e o sistema (3.11) é descrito por:

$$\begin{aligned}
 \sum_{j=1}^{N_1} \mathbf{a}_{j1} x_{j1} + \sum_{j=1}^{N_2} \mathbf{a}_{j2} x_{j2} + \dots + \sum_{j=1}^{N_K} \mathbf{a}_{jK} x_{jK} &= \mathbf{d} \\
 \sum_{j=1}^{N_1} x_{j1} + \mu_1 &= e_1 \\
 \sum_{j=1}^{N_2} x_{j2} + \mu_2 &= e_2 \\
 &\vdots \\
 \sum_{j=1}^{N_K} x_{jK} + \mu_K &= e_K.
 \end{aligned} \tag{3.24}$$

Sem perda de generalidade, construímos uma base considerando padrões de corte apenas para as barras do tipo 1, que supomos ser a barra de maior comprimento entre todas as disponíveis, isto é, $L_1 \geq L_k, k = 2, \dots, K$. Além disso, supomos que $l_i \leq L_1, i = 1, \dots, m$, caso contrário, o problema seria trivialmente infactível. Assim, considere m colunas associadas a padrões de corte homogêneos, que agora têm $m + K$ posições, veja (3.12):

$$\mathbf{a}_{j1} = (0, \dots, b_{jj}, 0, \dots, 1, 0, \dots, 0)^T \tag{3.25}$$

onde $b_{jj} = \min \left\{ \left\lfloor \frac{L_1}{l_j} \right\rfloor, d_j \right\}, \quad j = 1, \dots, m.$

Observe agora que uma base é formada por $m + K$ colunas, de modo que necessitamos de mais K colunas, que são correspondentes às variáveis de folga. Observe que se considerarmos o sistema (3.24) apenas com as colunas em (3.25), isto é, fazendo-se $x_{jk} = 0$ para as demais variáveis (variáveis não básicas), as m primeiras equações determinam unicamente o valor de $x_{j1}, j = 1, \dots, m$:

$$x_{j1} = \frac{d_j}{b_{jj}}, \quad j = 1, \dots, m.$$

Devemos analisar se esta solução é factível (caso 1) ou infactível (caso 2).

Caso 1: Se $\sum_{j=1}^m x_{j1} \leq e_1$, então temos uma solução factível e a matriz básica inicial $\mathbf{B}$, de ordem $(m + K)$, pode ser escrita como:

$$\mathbf{B} = \begin{bmatrix} b_{11} & \dots & 0 & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & b_{mm} & 0 & \dots & 0 \\ 1 & \dots & 1 & 1 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & 0 & \dots & 1 \end{bmatrix} \quad (3.26)$$

Observe que a coluna associada a variável de folga μ_k tem a seguinte forma:

$$(0, \dots, 0, 0, \dots, 1, \dots, 0)^T$$

onde o valor 1 ocupa a posição $(m + k)$.

Observe, também, que as barras $2, 3, \dots, K$ não foram cortadas e, portanto, as restrições em (3.24) são satisfeitas com:

$$\begin{aligned} \mu_1 &= e_1 - \sum_{j=1}^m x_{j1} \quad \text{e} \\ \mu_j &= e_j, \quad j = 2, \dots, K, \end{aligned}$$

sendo todas as variáveis de folga consideradas básicas.

Caso 2: Se $\sum_{j=1}^m x_{j1} > e_1$, então a solução é infactível e, neste caso, será necessário resolver um Problema Artificial, penalizando-se o uso excessivo da barra 1 (Fase 1 do método simplex). O problema artificial terá apenas uma variável artificial μ_1^a , relativa à restrição violada $m + 1$ (primeira restrição de estoque):

$$\begin{aligned}
& \text{minimizar} \quad f_a = \sum_{j=1}^{N_1} c_{j1}x_{j1} + \dots + \sum_{j=1}^{N_K} c_{jK}x_{jK} + M\mu_1^a \\
& \text{sujeito a:} \quad \left\{ \begin{array}{ll} \sum_{j=1}^{N_1} \mathbf{a}_{j1}x_{j1} + \sum_{j=1}^{N_2} \mathbf{a}_{j2}x_{j2} + \dots + \sum_{j=1}^{N_K} \mathbf{a}_{jK}x_{jK} & = \mathbf{d} \\ \sum_{j=1}^{N_1} x_{j1} & + \mu_1 - \mu_1^a = e_1 \\ & \sum_{j=1}^{N_2} x_{j2} + \mu_2 = e_2 \\ & \ddots \quad \quad \quad \vdots \\ & \sum_{j=1}^{N_K} x_{jK} + \mu_K = e_K \\ x_{jk} \geq 0 \text{ e inteiro, } \mu_j \geq 0, \mu_1^a \geq 0, \quad j = 1, \dots, N_k, \quad k = 1, \dots, K. \end{array} \right. \quad (3.27)
\end{aligned}$$

Tomamos a mesma base para o problema artificial como em (3.26), porém a variável de folga $\mu_1 < 0$ é substituída pela variável artificial μ_1^a . A base inicial do problema artificial tem a seguinte forma:

$$\mathbf{B} = \begin{bmatrix} b_{11} & \dots & 0 & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & b_{mm} & 0 & \dots & 0 \\ 1 & \dots & 1 & -1 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & 0 & \dots & 1 \end{bmatrix} \quad (3.28)$$

Observamos que $\mu_1^a = -e_1 - \sum_{j=1}^m x_{j1} = -\mu_1$ e as demais variáveis têm os mesmos valores anteriores.

Se na solução ótima do problema artificial (3.27), $\mu_1^a > 0$ significa que não existe uma solução básica factível para o problema de corte de estoque. Se, por outro lado, $\mu_1^a = 0$ em alguma iteração do método simplex, então a variável artificial μ_1^a é abandonada.

O custo relativo de uma variável associada a um padrão de corte para a barra k é dado por (a função perda é considerada):

$$\begin{aligned}
c_{jk} - \pi^T \mathbf{a}_{jk} &= \\
&= \left(L_k - \sum_{i=1}^m l_i \alpha_i \right) - \begin{bmatrix} \pi_1 & \dots & \pi_m & \pi_{m+1} & \dots & \pi_{m+K} \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_m \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix} = \\
&= L_k - \sum_{i=1}^m l_i \alpha_i - \sum_{i=1}^m \pi_i \alpha_i - \pi_{m+k} = L_k - \sum_{i=1}^m (l_i + \pi_i) \alpha_i - \pi_{m+k}
\end{aligned}$$

Para determinar o menor custo relativo, resolve-se o seguinte sub-problema:

$$\begin{aligned}
&\text{minimizar} \quad L_k - \sum_{i=1}^m (l_i + \pi_i) \alpha_i - \pi_{m+k} \\
&\text{sujeito a que } (\alpha_1, \dots, \alpha_m)^T \text{ corresponda a um padrão de corte.}
\end{aligned}$$

Assim, o menor custo relativo será dado por:

$$L_k - g(\mathbf{a}) - \pi_{m+k}$$

onde $\mathbf{a} = (\alpha_1, \dots, \alpha_m)^T$ é obtido por:

$$\begin{aligned}
g(\mathbf{a}) &= \text{máximo} \sum_{i=1}^m (l_i + \pi_i) \alpha_i \\
\text{sujeito a: } &\begin{cases} \sum_{i=1}^m l_i \alpha_i \leq L_k \\ 0 \leq \alpha_i \leq d_i, i = 1, \dots, m \text{ e inteiro.} \end{cases}
\end{aligned}$$

O custo relativo associado à variável de folga é mais simples que aquele associado a uma coluna que representa um padrão de corte:

$$0 - \begin{bmatrix} \pi_1 & \dots & \pi_m & \pi_{m+1} & \dots & \pi_{m+K} \end{bmatrix} \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix} = -\pi_{m+k}$$

Assim, sempre que $\pi_{m+k} > 0$, a variável μ_k é escolhida a entrar na base, sem necessidade de se resolver um problema da mochila.

Para efeito de uma implementação computacional criamos um vetor de identificação das variáveis básicas, as quais estão associadas às barras em estoque, seja um padrão de corte ou uma variável de folga. Em outras palavras, cada coluna da matriz básica é um representante de um padrão de corte, ou uma coluna da matriz identidade associada à uma variável de folga, ou a coluna da variável artificial.

$$\text{id}(j) = \begin{cases} k & \text{se a } j\text{-ésima coluna básica é um padrão de corte associado à barra } k; \\ -k & \text{se a } j\text{-ésima coluna básica corresponde à uma variável de folga associada à restrição de estoque da barra } k; \\ 0 & \text{se a } j\text{-ésima coluna básica corresponde à variável artificial.} \end{cases}$$

Desta forma, fica fácil a recuperação da solução do problema a partir da matriz básica e deste vetor de identificação.

Capítulo 4

Heurísticas de Construção para o Problema de Corte de Estoque Inteiro

Apresentaremos neste capítulo duas heurísticas clássicas, bem conhecidas na literatura (Wäscher e Gau[28], Pinto [21], Hoto [10]) que são as heurísticas FFD (*First-Fit-Decreasing*) e Gulosa. Estas heurísticas têm a seguinte estrutura geral:

Enquanto houver demanda faça:

1. Construa um bom padrão de corte;
2. Use o padrão do passo 1 tanto quanto for possível, sem gerar excessos de itens cortados;
3. Atualize a demanda e o estoque.

4.1 Heurística FFD

Apresentaremos agora um algoritmo para a heurística FFD (*First-Fit-Decreasing*), que de forma geral, consiste em colocar o item maior num padrão de corte tantas vezes quanto for possível, ou seja, até que não haja mais espaço para colocar este item ou, até que sua

demanda já tenha sido atendida (os itens maiores são alocados em primeiro lugar pois são os mais difíceis de serem combinados). Quando não for mais possível colocar o item maior, passar para o segundo maior item, e assim por diante.

Esta heurística encontra sempre uma solução inteira para o problema de corte. Vejamos um algoritmo.

Algoritmo

Início

Passo 1: Ordene os itens demandados de acordo com o seu tamanho, em ordem decrescente. Por simplicidade de notação, suponha que:

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

Faça $r_i = d_i, i = 1, \dots, m$ { r_i é a demanda residual, para todo $i = 1, \dots, m$. }

Considere $k = 1$ { Primeiro padrão de corte. }

Faça $D = \sum_{i=1}^m r_i$ { Demanda residual total. }

Passo 2:

Enquanto $D > 0$ (isto é, enquanto existir demanda a ser atendida) faça:

{ início enquanto1 }

Faça: $RESTO = L; \quad i = 1; \quad \alpha_j = 0, j = 1, \dots, m.$

($RESTO$ representa o espaço restante para alocação dos itens que, inicialmente será igual ao comprimento total da barra.)

Enquanto $\left(i \leq m \quad \text{e} \quad RESTO \geq l_m \right)$

(isto é, enquanto houver item e espaço), faça:

{ início enquanto2 }

$$\alpha_i = \min \left\{ \left\lfloor \frac{RESTO}{l_i} \right\rfloor, \quad r_i \right\}.$$

(o vetor $\mathbf{a} = (\alpha_1, \dots, \alpha_m)^T$ contabiliza como os itens são alocados no padrão de corte)

{ atualização }

- $RESTO = RESTO - (\alpha_i l_i)$
- $r_i = r_i - \alpha_i$
- $D = D - \alpha_i$
- $i = i + 1$

{ fim enquanto2 }

Faça: $\alpha_{ik} = \alpha_i, \quad i = 1, \dots, m.$

(Ou seja, o padrão de corte será montado com o auxílio do vetor $\mathbf{a}$ já obtido.)

Faça: $x_k = \min \left\{ \left\lfloor \frac{r_i}{\alpha_i} \right\rfloor, \quad r_i \neq 0, d_i \neq 0, \quad i = 1, \dots, m \right\}.$

(x_k guarda a quantidade de vezes que o padrão de corte será utilizado.)

Faça: $k = k + 1$ (Passa a construção para o próximo padrão.)

{ fim enquanto1 }

Fim

Para o problema de corte com restrições de estoque (3.11) geramos um padrão, seguindo o algoritmo acima, para cada tipo de barra disponível em estoque. Como o objetivo é minimizar a perda, escolhemos, dentre os padrões gerados, aquele que apresentar a *menor perda*. Atualizamos o estoque e a demanda e repetimos o procedimento, até que toda a demanda seja completada.

4.1.1 Exemplo

Considere o seguinte exemplo de problema de corte de estoque. Suponha que temos disponível, objetos de comprimento $L = 11$ a serem cortados para a produção de três

tipos de itens ($m = 3$), com comprimentos l_i e demandas $d_i, i = 1, \dots, 3$ descritos na Tabela 4.1.

Item	Comprimento	Demanda
1	3	3
2	4	3
3	5	2

Tabela 4.1: *Dados do exemplo do problema de corte.*

Resolveremos este exemplo com a heurística FFD. Primeiramente, ordenamos os itens em ordem decrescente de comprimento. Teremos:

$$l_3 = 5 \geq l_2 = 4 \geq l_1 = 3.$$

Em uma barra de comprimento 11, começaremos a alocar os itens, na ordem descrita acima, até que não caiba aquele item ou até que a sua demanda seja completada. Colocaremos 2 itens 3. Com isso, a demanda do item 3 é satisfeita e resta um espaço de 1 na barra. Não cabe mais nenhum item nesta barra. Assim, temos o primeiro padrão gerado: $\mathbf{a}_1 = (0, 0, 2)^T$, com perda = 1, que pode ser visto na Figura 4.1.

Figura 4.1: *Primeiro padrão de corte gerado.*

Passamos para a construção do segundo padrão de corte, agora com $r_3 = 0$. Colocamos o item 2 duas vezes, não é possível colocar este item pela terceira vez pois excede a capacidade da barra. Tentamos o próximo item, que é o item 1. Ele é colocado uma vez e a capacidade da barra é completada. Na Figura 4.2 temos, então, o segundo padrão gerado: $\mathbf{a}_2 = (1, 2, 0)^T$, com perda zero.

Figura 4.2: *Segundo padrão de corte gerado.*

Começamos, agora, a construção do terceiro padrão, com $r_1 = 2, r_2 = 1$ e $r_3 = 0$. Colocamos um item 2 e com isso, temos a sua demanda completada. Sobra um espaço de 7 unidades na barra e é possível colocar dois itens 1, que tem sua demanda atendida. Finalmente, na Figura 4.3, temos o terceiro e último padrão: $\mathbf{a}_3 = (2, 1, 0)^T$, com perda igual a 1. Os três padrões são utilizados com frequência 1.

Figura 4.3: *Terceiro padrão de corte gerado.*

Na solução final temos 3 padrões de corte, e a perda total é igual a 2. Vejamos a solução final na tabela 4.2

Quantidade de barras	Padrão de corte			Perda no padrão
$\mathbf{x}$	item 1	item 2	item 3	
1	0	0	2	1
1	1	2	0	0
1	2	1	0	1

Tabela 4.2: *Solução obtida pela heurística FFD.*

Observe que esta é uma solução inteira e factível.

4.2 Heurística Gulosa

Outra maneira de se construir soluções inteiras para um problema de corte de estoque é através da *heurística gulosa*, que é também uma heurística de repetição exaustiva (Hinxman [9]). A idéia desta heurística é gerar um bom padrão de corte e utilizá-lo à exaustão (sem que haja excessos). A geração do padrão consiste na resolução de um Problema da Mochila (3.22), o qual tem as demandas atualizadas após o uso.

Cada padrão gerado: $\mathbf{a} = (\alpha_1, \alpha_2, \dots, \alpha_m)^T$ é utilizado o maior número possível de vezes. Este número é dado por: $\min \{ \lfloor r_i / \alpha_i \rfloor, r_i \neq 0, \alpha_i \neq 0, i = 1, \dots, m \}$.

Algoritmo

Início

Passo 1: { Inicialização }

Faça: $r_i = d_i, \quad i = 1, \dots, m$ { r_i é a demanda residual, para $i = 1, \dots, m$ }

Faça: $k = 1$ { primeiro padrão de corte }

Passo 2: { Gera o padrão de corte mais valioso }

Resolva o problema da mochila:

$$\begin{aligned} & \text{maximizar} && l_1\alpha_1 + l_2\alpha_2 + \dots + l_m\alpha_m \\ & \text{sujeito a:} && \begin{cases} l_1\alpha_1 + l_2\alpha_2 + \dots + l_m\alpha_m \leq L \\ 0 \leq \alpha_i \leq r_i, i = 1, \dots, m \text{ e inteiro.} \end{cases} \end{aligned} \quad (4.1)$$

Passo 3: { Determine o número de vezes que este padrão pode ser repetido }

$$x_k = \min \left\{ \left\lfloor \frac{r_i}{\alpha_i} \right\rfloor, r_i \neq 0, \alpha_i \neq 0, i = 1, \dots, m \right\}.$$

Passo 4: { Atualize a demanda }

$$r_i = r_i - x_k\alpha_i, \quad i = 1, \dots, m.$$

Passo 5: { Critério de parada }

Se a demanda de todos os itens estiver satisfeita, ou seja, $\mathbf{r} = \mathbf{0}$, então PARE.

Senão, faça $k = k + 1$ e volte ao Passo 2.

Fim

4.2.1 Exemplo

Considere o mesmo exemplo dado na seção anterior, cujos dados estão dados na Tabela 4.1.

Resolveremos este exemplo com a heurística Gulosa. O primeiro passo é resolver o seguinte problema da mochila restrito:

$$\begin{array}{ll} \text{maximizar} & g(\mathbf{a}) = 3\alpha_1 + 4\alpha_2 + 5\alpha_3 \\ \text{sujeito a:} & \left\{ \begin{array}{l} 3\alpha_1 + 4\alpha_2 + 5\alpha_3 \leq 11 \\ 0 \leq \alpha_1 \leq 3, \quad \text{e inteiro} \\ 0 \leq \alpha_2 \leq 3, \quad \text{e inteiro} \\ 0 \leq \alpha_3 \leq 2, \quad \text{e inteiro.} \end{array} \right. \end{array}$$

Obtemos como solução, o primeiro padrão: $\mathbf{a}_1 = (2, 0, 1)^T$. O número de vezes que este padrão será utilizado é dado por:

$$x_1 = \min \left\{ \left\lfloor \frac{3}{2} \right\rfloor, \left\lfloor \frac{2}{1} \right\rfloor \right\} = 1.$$

Assim, $x_1 = 1$. Calculamos a demanda residual: $\mathbf{r} = \mathbf{d} - \mathbf{Ax} = (1, 3, 1)^T$. Agora, resolvemos o problema:

$$\begin{array}{ll} \text{maximizar} & g(\mathbf{a}) = 3\alpha_1 + 4\alpha_2 + 5\alpha_3 \\ \text{sujeito a:} & \left\{ \begin{array}{l} 3\alpha_1 + 4\alpha_2 + 5\alpha_3 \leq 11 \\ 0 \leq \alpha_1 \leq 1, \quad \text{e inteiro} \\ 0 \leq \alpha_2 \leq 3, \quad \text{e inteiro} \\ 0 \leq \alpha_3 \leq 1, \quad \text{e inteiro.} \end{array} \right. \end{array}$$

Obtemos como solução, o segundo padrão: $\mathbf{a}_2 = (1, 2, 0)^T$. O número de vezes que ele é utilizado é dado por:

$$x_2 = \min \left\{ \left\lfloor \frac{1}{1} \right\rfloor, \left\lfloor \frac{3}{2} \right\rfloor \right\} = 1.$$

Assim, este padrão é utilizado uma vez ($x_2 = 1$). Calculamos a demanda residual: $\mathbf{r} = \mathbf{d} - \mathbf{Ax} = (0, 1, 1)^T$, e resolvemos o problema:

$$\begin{array}{ll} \text{maximizar} & g(\mathbf{a}) = 3\alpha_1 + 4\alpha_2 + 5\alpha_3 \\ \text{sujeito a:} & \left\{ \begin{array}{l} 3\alpha_1 + 4\alpha_2 + 5\alpha_3 \leq 11 \\ \alpha_1 = 0 \\ 0 \leq \alpha_2 \leq 1, \quad \text{e inteiro} \\ 0 \leq \alpha_3 \leq 1, \quad \text{e inteiro.} \end{array} \right. \end{array}$$

Obtemos como solução, o terceiro padrão: $\mathbf{a}_3 = (0, 1, 1)^T$. Este padrão é utilizado

$$x_3 = \min \left\{ \left\lfloor \frac{1}{1} \right\rfloor, \left\lfloor \frac{1}{1} \right\rfloor \right\} = 1.$$

Assim, $x_3 = 1$. Calculamos a demanda residual: $\mathbf{r} = \mathbf{d} - \mathbf{A}\mathbf{x} = (0, 0, 0)^T$. Como a demanda foi completada, terminamos o procedimento. Temos a solução final com três padrões de corte e a perda total igual a 2. A solução final é a seguinte, dada na Tabela 4.3:

Quantidade de barras	Padrão de corte			Perda no padrão
$\mathbf{x}$	item 1	item 2	item 3	
1	2	0	1	0
1	1	2	0	0
1	0	1	1	2

Tabela 4.3: *Solução obtida pela heurística Gulosa.*

Uma das dificuldades, quando se utiliza heurísticas construtivas gulosas com objetivo de minimizar a perda, é a concentração de itens mais desfavoráveis (itens difíceis de serem combinados, isto é, itens cujas dimensões não formam uma boa combinação com os demais itens do problema, considerando as dimensões do objeto) nos padrões finais, o que pode resultar em padrões com perdas elevadas. No início do algoritmo, bons padrões, isto é, com perdas pequenas, são gerados e na medida em que o procedimento avança, começa a haver uma perda de qualidade dos padrões, ou seja, padrões com grandes perdas passam a ser gerados, o que pode comprometer a solução geral do problema. Em geral isso ocorre, pois no início existe uma variedade maior de tipos de itens e também muitas vezes existem itens com dimensões pequenas, que são fáceis de serem alocados em pequenos espaços.

Esta desvantagem pode se tornar uma vantagem se novos itens são incorporados, de modo que os padrões de baixa qualidade não precisam ser utilizados. Caso nenhuma interferência seja feita, esse procedimento resulta, em geral, em padrões finais com altas perdas (quando comparados com as perdas dos primeiros padrões). Para se evitar

isso, é preciso interferir de alguma forma no processo de geração, a fim de que os itens mais desfavoráveis possam ser utilizados nos primeiros padrões (quando ainda existe uma variedade maior de tipos de itens para serem combinados). Pileggi [20] discute como atribuir valores para os itens, de forma a se obter uma solução global mais satisfatória, apresentando um procedimento iterativo, que procura calibrar os valores dos itens, com o objetivo de encontrar uma solução geral melhor.

Adaptar esta heurística para o caso com vários tipos de barras disponíveis em quantidades limitadas é uma tarefa fácil. A cada iteração resolve-se um problema da mochila (4.1) para cada tipo de barra disponível e escolhe-se aquele que apresentar a menor perda. Atualiza-se a demanda e o estoque e repete-se o procedimento.

Observação

Observamos que o exemplo apresentado nesse capítulo e os que serão apresentados no próximo capítulo são exemplos com apenas um tipo de barra disponível em estoque para simplificar e facilitar a explicação do funcionamento das heurísticas, porém são facilmente aplicadas a problemas com vários tipos de barras, como veremos no Capítulo 7.

Capítulo 5

Heurísticas Residuais

Os procedimentos deste capítulo geram uma solução inteira para o problema de corte de estoque unidimensional inteiro a partir da solução ótima fracionária do problema (3.11) relaxado, que, por sua vez, é obtida pelo método Simplex. Supomos que pelo menos uma componente da solução ótima seja não-inteira, pois, caso contrário, a solução ótima inteira já teria sido obtida. Os procedimentos estudados para gerar a solução inteira são basicamente de dois tipos: *residuais* e *gulosos*. Os procedimentos puramente gulosos foram apresentados no Capítulo 4; aqui, apresentaremos procedimentos residuais. Alguns destes procedimentos e vários outros estão em Wäscher e Gau [28].

5.1 Problema Residual

Considere, novamente, o seguinte problema de corte de estoque:

$$\begin{aligned} \text{minimizar} \quad & f(\mathbf{x}) = \sum_{j=1}^n c_j x_j \\ \text{sujeito a:} \quad & \mathbf{Ax} = \mathbf{d}, \quad \mathbf{x} \geq \mathbf{0} \quad \text{e inteiro} \end{aligned} \tag{5.1}$$

e seja $\tilde{\mathbf{x}}$ sua solução ótima fracionária.

Definição 5.1 *Seja $\tilde{\mathbf{y}}$ uma aproximação inteira para $\tilde{\mathbf{x}}$ tal que:*

$$\mathbf{A}\tilde{\mathbf{y}} \leq \mathbf{d} \tag{5.2}$$

chamamos $\tilde{\mathbf{y}}$ de solução inteira aproximada.

Note que $\tilde{\mathbf{y}}$ não é necessariamente uma solução factível para o problema (5.1). Uma maneira usual de obter $\tilde{\mathbf{y}}$ é arredondar por truncamento $\tilde{\mathbf{x}}$:

$$\tilde{\mathbf{y}} = (\lfloor \tilde{x}_1 \rfloor, \lfloor \tilde{x}_2 \rfloor, \dots, \lfloor \tilde{x}_n \rfloor)$$

o qual satisfaz (5.2) uma vez que todos os coeficientes de $\mathbf{A}$ são não-negativos e $\tilde{\mathbf{x}}$ satisfaz $\mathbf{A}\tilde{\mathbf{x}} = \mathbf{d}$. Veremos na seção 5.4 outra maneira de se obter uma solução inteira aproximada.

Definição 5.2 *Considere $\tilde{\mathbf{y}}$ uma solução inteira aproximada. Seja $\mathbf{r}$ dado por:*

$$\mathbf{r} = \mathbf{d} - \mathbf{A}\tilde{\mathbf{y}}. \quad (5.3)$$

O vetor $\mathbf{r}$ é chamado de demanda residual. O problema residual é dado por:

$$\begin{aligned} \text{minimizar} \quad & f(\mathbf{x}) = \sum_{j=1}^n c_j x_j \\ \text{sujeito a:} \quad & \mathbf{A}\mathbf{x} = \mathbf{r}, \quad \mathbf{x} \geq \mathbf{0} \quad \text{e inteiro.} \end{aligned} \quad (5.4)$$

A matriz $\mathbf{A}$ dos padrões de corte em (5.4) difere da matriz $\mathbf{A}$ em (5.1), pois na geração dos padrões de corte em (5.4) o problema da mochila que gera cada coluna de $\mathbf{A}$ deve considerar a demanda residual:

$$\begin{aligned} \text{maximizar} \quad & g(\mathbf{a}) = v_1\alpha_1 + v_2\alpha_2 + \dots + v_m\alpha_m \\ \text{sujeito a:} \quad & \begin{cases} l_1\alpha_1 + l_2\alpha_2 + \dots + l_m\alpha_m \leq L \\ 0 \leq \alpha_i \leq r_i, \quad i = 1, \dots, m \quad \text{e inteiro.} \end{cases} \end{aligned} \quad (5.5)$$

5.2 Heurísticas Residuais

Estas heurísticas consistem em resolver o problema original relaxado, obter uma solução inteira aproximada, resolver o problema residual relaxado resultante, obter uma solução inteira aproximada e assim por diante.

A seguir, damos a estrutura geral das heurísticas residuais.

Algoritmo

Início

1. { Passo inicial: } Faça: $k \leftarrow 0$ e $\mathbf{r}^k \leftarrow \mathbf{d}$. { $\mathbf{r}^k$ armazena a demanda residual }

2. { Passo principal: }

Enquanto PARE = FALSO, faça:

2.1. Resolva o seguinte problema de otimização linear (condição de integralidade relaxada), por geração de colunas, usando gerador de colunas restrito ($0 \leq \alpha_i \leq r_i^k$):

$$\begin{aligned} &\text{minimizar} && f(\mathbf{x}) = \sum_{j=1}^n c_j x_j \\ &\text{sujeito a:} && \begin{cases} \mathbf{A}\mathbf{x} = \mathbf{r}^k \\ \mathbf{x} \geq \mathbf{0} \end{cases} \end{aligned} \tag{5.6}$$

e obtenha $\tilde{\mathbf{x}}^k = (x_1^k, \dots, x_m^k)$ e a base correspondente: $\mathbf{B}^k = [\mathbf{a}_1^k, \dots, \mathbf{a}_m^k]$, onde $\mathbf{a}_i^k, i = 1, \dots, m$, são os padrões de corte gerados na resolução do problema (5.6).

2.2. Determine $\mathbf{y}^k$: solução inteira aproximada (veja definição 5.1).

2.3. Guarde $\mathbf{B}^k$ e $\mathbf{y}^k$.

2.4. Faça: $\mathbf{r}^{k+1} \leftarrow \mathbf{r}^k - \mathbf{B}^k \mathbf{y}^k$ e $k \leftarrow k + 1$.

2.5. { Teste de Parada: }

Se algum critério for satisfeito: PARE = VERDADEIRO.

3. { Passo final: } Se ainda houver demanda residual não-nula, aplique uma heurística para satisfazer esta demanda.

Fim

Nas próximas seções, heurísticas particulares são obtidas por especificar os passos 2.2, 2.5 e 3 do algoritmo anterior.

5.3 Heurísticas Residuais por Truncamento

Estas heurísticas são definidas ao estabelecermos os passos 2.2, 2.5 e 3 do algoritmo da seção 5.2.

No passo 2.2 utiliza-se o arredondamento por truncamento trivial:

$$\mathbf{y} = (\lfloor \tilde{x}_1 \rfloor, \lfloor \tilde{x}_2 \rfloor, \dots, \lfloor \tilde{x}_n \rfloor).$$

O critério de parada, passo 2.5, é quando o arredondamento tal como descrito acima não mais alterar a solução, ou seja, $\tilde{\mathbf{y}} = 0$.

Mas, pode ocorrer de ainda existir demanda residual não atendida (passo 3). Neste caso, a demanda residual pode ser completada com a aplicação de alguma heurística de construção: *FFD* ou *Gulosa*. Apresentamos agora um exemplo de aplicação da heurística residual por truncamento e, nas próximas seções, trataremos o passo 3 para completar a demanda residual.

5.3.1 Exemplo

Considere o seguinte exemplo de problema de corte de estoque. Suponha que temos disponível, objetos de comprimento $L = 11$ a serem cortados para a produção de três tipos de itens ($m = 3$), com comprimentos l_i e demandas $d_i, i = 1, \dots, 3$ descritos na Tabela 5.1.

Item	Comprimento	Demanda
1	3	4
2	4	3
3	5	2

Tabela 5.1: *Dados do exemplo do problema de corte.*

Primeiramente, obtemos a solução relaxada usando o *método simplex*. Esta é apresentada na Tabela 5.2.

Quantidade de barras	Padrão de corte		
$\tilde{\mathbf{x}}^0$	item 1	item 2	item 3
1,25	2	0	1
1,5	1	2	0
0,375	0	0	2

Tabela 5.2: *Solução relaxada.*

Fazemos o arredondamento desta solução e obtemos a solução inteira dada na Tabela 5.3 a seguir:

Quantidade de barras	Padrão de corte		
$\tilde{\mathbf{y}}^0$	item 1	item 2	item 3
1	2	0	1
1	1	2	0
0	0	0	2

Tabela 5.3: *Solução inteira.*

Atualizamos a demanda: $\mathbf{r}^0 = \mathbf{d} - \mathbf{A}\tilde{\mathbf{y}}^0$ e vemos que ainda restam itens com demandas não atendidas. Assim, temos um problema residual a ser resolvido, dado na Tabela 5.4.

Item	Comprimento	Demanda Residual: $\mathbf{r}^0$
1	3	1
2	4	1
3	5	1

Tabela 5.4: *Dados do problema residual.*

2ª iteração:

A segunda iteração consiste em resolver o problema residual resultante da primeira iteração que está descrito na Tabela 5.4. A sua solução relaxada está na Tabela 5.5:

Quantidade de barras	Padrão de corte		
$\tilde{x}^1$	item 1	item 2	item 3
0,5	1	1	0
0,5	1	0	1
0,5	0	1	1

Tabela 5.5: *Solução relaxada - segunda iteração.*

Fazendo o arredondamento obteremos o vetor solução $\tilde{x}^1$ todo nulo. Então, paramos este procedimento e temos aqui o *problema residual*, que deve ser resolvido por uma das heurísticas (*Gulosa ou FFD*). Estas heurísticas, cada uma a sua maneira, fornecerão padrões de corte, que deverão ser acrescentados ao que aqui foram gerados, para atender a demanda residual.

5.3.2 Heurística Residual FFD

A heurística residual FFD consiste em aplicar a heurística residual descrita na seção 5.2 e ao final, no passo 3, se ainda houver demanda residual, aplica-se a heurística FFD (seção 4.1) para completar a demanda residual.

5.3.3 Exemplo

A heurística residual FFD parte do problema residual final encontrado pelo algoritmo da seção 5.2. Assim, considerando o mesmo exemplo anterior, iniciamos com o problema residual da Tabela 5.4; lembrando que os padrões da Tabela 5.3 já fazem parte da solução heurística final do problema de corte de estoque original.

Ordenamos os itens em ordem decrescente de comprimento. Teremos:

$$l_3 = 5 \geq l_2 = 4 \geq l_1 = 3.$$

Em uma barra de comprimento 11, alocamos os itens, na ordem descrita acima, até que não caiba aquele item ou até que a sua demanda seja completada. Colocamos um item 3 e, com isso, a demanda deste item é completada. Partimos para o segundo item, o item 2 é colocado uma vez e temos sua demanda completada. Resta um espaço de 2 na barra. Não cabe mais nenhum item nesta barra. Assim, temos o primeiro padrão gerado: $\mathbf{a}_1 = (0, 1, 1)^T$, com perda = 2, que pode ser visto na Figura 5.1.

Figura 5.1: *Primeiro padrão de corte gerado.*

Passamos para a construção do segundo padrão de corte. Colocamos um item do tipo 1 e completamos a sua demanda e, com isso, a demanda de todo o problema. Assim, temos o segundo e último padrão de corte gerado: $\mathbf{a}_2 = (1, 0, 0)^T$, com perda 8.

Figura 5.2: *Segundo padrão de corte gerado.*

Na solução final temos 4 padrões de corte, dois que foram determinados na solução do problema residual e dois que foram criados agora pela Heurística FFD. Vejamos a solução final na Tabela 5.6.

Quantidade de barras	Padrão de corte			Perda no padrão
$\mathbf{x}$	item 1	item 2	item 3	
1	2	0	1	0
1	1	2	0	0
1	0	1	1	2
1	1	0	0	8

Tabela 5.6: *Solução obtida pela heurística residual FFD.*

5.3.4 Heurística Residual Gulosa

A heurística residual Gulosa consiste, também, em aplicar a heurística residual descrita na seção 5.2 e ao final, no passo 3, se ainda houver demanda residual, aplica-se a heurística Gulosa (seção 4.2) para completar a demanda residual.

5.3.5 Exemplo

A heurística residual Gulosa parte do problema residual final encontrado pelo algoritmo da seção 5.2. Assim, considerando o mesmo exemplo anterior, iniciamos com o problema residual da Tabela 5.4; lembrando que os padrões $\mathbf{a}_1$ e $\mathbf{a}_2$ da Tabela 5.3 já fazem parte da solução final do problema de corte de estoque original.

O primeiro passo é resolver o seguinte problema da mochila restrito:

$$\begin{array}{ll} \text{maximizar} & g(\mathbf{a}) = 3\alpha_1 + 4\alpha_2 + 5\alpha_3 \\ \text{sujeito a:} & \left\{ \begin{array}{l} 3\alpha_1 + 4\alpha_2 + 5\alpha_3 \leq 11 \\ 0 \leq \alpha_1 \leq 1, \text{ e inteiro} \\ 0 \leq \alpha_2 \leq 1, \text{ e inteiro} \\ 0 \leq \alpha_3 \leq 1, \text{ e inteiro.} \end{array} \right. \end{array}$$

Obtemos como solução, o primeiro padrão: $\mathbf{a}_3 = (0, 1, 1)^T$. O número de vezes que este padrão será utilizado é dado por:

$$x_3 = \min \left\{ \left\lfloor \frac{1}{1} \right\rfloor, \left\lfloor \frac{1}{1} \right\rfloor \right\} = 1.$$

Assim, $x_3 = 1$. Calculamos a demanda residual: $\mathbf{r}^1 = \mathbf{r}^0 - \mathbf{a}_3 x_3 = (1, 0, 0)^T$. Agora, resolvemos o problema:

$$\begin{array}{ll} \text{maximizar} & g(\mathbf{a}) = 3\alpha_1 + 4\alpha_2 + 5\alpha_3 \\ \text{sujeito a:} & \left\{ \begin{array}{l} 3\alpha_1 + 4\alpha_2 + 5\alpha_3 \leq 11 \\ 0 \leq \alpha_1 \leq 1, \text{ e inteiro} \\ 0 \leq \alpha_2 \leq 0, \text{ e inteiro} \\ 0 \leq \alpha_3 \leq 0, \text{ e inteiro.} \end{array} \right. \end{array}$$

Obtemos como solução, o segundo padrão: $\mathbf{a}_4 = (1, 0, 0)^T$. O número de vezes que ele é utilizado é dado por:

$$x_4 = \min \left\{ \left\lfloor \frac{1}{1} \right\rfloor \right\} = 1.$$

Assim, este padrão é utilizado uma vez ($x_4 = 1$). Calculamos a demanda residual: $\mathbf{r}^2 = \mathbf{r}^1 - \mathbf{a}_4 x_4 = (0, 0, 0)^T$. Como a demanda foi completada, terminamos o procedimento. Temos a solução final com dois padrão que já tinham sido encontrados pelo algoritmo do problema residual e mais dois criados agora com a heurística Gulosa. A solução final é a seguinte, dada na Tabela 5.7.

Quantidade de barras	Padrão de corte			Perda no padrão
$\mathbf{x}$	item 1	item 2	item 3	
1	2	0	1	0
1	1	2	0	0
1	0	1	1	2
1	1	0	0	8

Tabela 5.7: *Solução obtida pela heurística residual Gulosa.*

Embora para este exemplo simples ambas as heurísticas tenham obtido as mesmas soluções, veremos nos experimentos computacionais no Capítulo 6 que a heurística residual Gulosa comportou-se melhor que a heurística residual FFD.

5.4 Heurística Residual Nova

Apresentaremos agora uma outra heurística residual. Sua estratégia é apostar na qualidade dos padrões gerados pelo modelo de otimização linear, onde toda a demanda é levada em conta na construção dos padrões de corte.

A cada iteração, resolve-se um problema de corte de estoque relaxado e ordena-se o vetor solução de forma não-crescente, ou seja, por simplicidade de notação consideramos:

$\tilde{x}_1 \geq \tilde{x}_2 \geq \dots \geq \tilde{x}_m$. A heurística aposta que o padrão 1 (o mais utilizado) mereça maior atenção. Iniciando-se com o padrão 1, o valor de $\tilde{\mathbf{x}}_1$ é arredondado para o inteiro superior e testa-se a factibilidade da seguinte solução (no sentido de que excessos de itens não foram gerados): $(\lceil \tilde{x}_1 \rceil, 0, \dots, 0)$.

Caso não seja factível (isto é, houve excesso de itens), a solução é reduzida de uma unidade até que excessos sejam eliminados. Repete-se para o padrão 2 e assim por diante, até que o último padrão de corte gerado seja examinado. Atualiza-se a demanda, gerando um problema residual. Este novo problema é resolvido relaxando-se a condição de integralidade e o procedimento anterior é repetido até que a toda a demanda seja satisfeita. Deve-se observar que a geração de colunas neste procedimento, bem como nas heurísticas residuais, deve ser restrita, pois cada padrão de corte gerado tem de poder ser utilizado pelo menos uma vez.

Para esta heurística basta especificar o passo 2.2 do algoritmo geral dado na seção 5.2.

2.2.1 Ordene os padrões tais que: $x_1^k \geq \dots \geq x_m^k$ (isto é, dê prioridade aos que são mais usados. Uma versão 2 desta heurística utiliza outro critério.)

2.2.2 Para $i = 1, \dots, m$, faça:

Início

$$y_i^k \leftarrow \lfloor x_i^k \rfloor + 1 \quad \{ \mathbf{y}^k = (y_1^k, \dots, y_{i-1}^k, y_i^k, 0, \dots, 0) \}$$

$$\text{Enquanto } \mathbf{y}^k \text{ for infactível, faça: } \{ \mathbf{y}^k \text{ é infactível se } \mathbf{B}^k \mathbf{y}^k \not\leq \mathbf{d}^k \}$$

$$y_i^k \leftarrow y_i^k - 1$$

Fim

Considere a Tabela 5.5. Observe que, para esta solução, lembrando que a demanda é $\mathbf{d} = (1, 1, 1)^T$, a frequência de qualquer um dos padrões pode ser arredondada para o inteiro superior: 1, e as demais para 0. Assim, cada iteração da heurística Nova vai necessariamente diminuir a demanda, portanto $\mathbf{y}^k$ será sempre diferente de zero. Como a demanda é finita (e no caso residual é pequena) com um número finito de passos teremos

$\mathbf{r} = \mathbf{0}$. Então, o critério de parada (passo 2.5) neste caso é quando $\mathbf{r} = \mathbf{0}$. Com isso, vemos que o procedimento termina quando todas as demandas forem completas e, não há o passo 3 descrito no algoritmo geral da seção 5.2. Vejamos o exemplo a seguir.

5.4.1 Exemplo

Considere o seguinte problema de corte de estoque. Suponha que temos disponível, objetos de comprimento $L = 194$ a serem cortados para a produção de três tipos de itens ($m = 3$), com comprimentos l_i e demandas $d_i, i = 1, \dots, 3$ descritos na Tabela 5.8.

Item	Comprimento	Demanda
1	108	4
2	13	8
3	90	7

Tabela 5.8: *Dados do exemplo do problema de corte.*

Resolveremos este exemplo com a heurística *Nova*. O primeiro passo é resolver o problema relaxado através do método *simplex* com geração de colunas. A solução obtida está na Tabela 5.9.

Quantidade de barras	Padrão de corte		
$\tilde{\mathbf{x}}^0$	item 1	item 2	item 3
1,333 ...	1	6	0
2,666 ...	1	0	0
3,5	0	0	2

Tabela 5.9: *Solução relaxada.*

Agora, ordena-se o vetor solução $\tilde{\mathbf{x}}^0$ de forma não-crescente, teremos:

$$x_3^0 = 3,5 \geq x_2^0 = 2,666 \dots \geq x_1^0 = 1,333 \dots$$

Inicia-se o arredondamento da primeira componente do vetor:

$$y_3 = \lceil 3, 5 \rceil = 4.$$

Testa-se a factibilidade. Com $y_3 = 4$ (e $y_2 = y_1 = 0$), teremos o padrão $(0, 0, 2)^T$ executado 4 vezes, produzindo 8 itens do tipo 3, mas a sua demanda é de 7. Assim, teremos um excesso na produção, o que não deve ocorrer. Portanto não aceitamos $y_3 = 4$ e então fazemos:

$$y_3 = 3$$

e testamos a factibilidade. Com $y_3 = 3$ teremos 6 itens do tipo 3 o que não excede a demanda original. Então passamos para a segunda maior componente do vetor $\mathbf{x}$. Fazemos:

$$y_2 = \lceil 2, 666 \dots \rceil = 3.$$

Testa-se a factibilidade. Com $y_2 = 3$ (e $y_3 = 3, y_1 = 0$), teremos o padrão $(1, 0, 0)^T$ executado 3 vezes e o padrão $(0, 0, 2)^T$ executado 3 vezes, produzindo 3 itens do tipo 1 e 6 itens do tipo 3, o que não excede a produção de nenhum item. Assim, partiremos para a terceira e última componente do vetor solução $\tilde{\mathbf{x}}$: Fazemos:

$$y_1 = \lceil 1, 333 \dots \rceil = 2.$$

Testa-se a factibilidade. Com $y_1 = 2$ (e $y_2 = 3, y_3 = 3$), teremos o padrão $(1, 6, 0)^T$ executado 2 vezes, o padrão $(1, 0, 0)^T$ executado 3 vezes e o padrão $(0, 0, 2)^T$ executado 3 vezes, produzindo 5 itens do tipo 1, 12 itens do tipo 2 e 6 itens do tipo 3. Nota-se que temos um excesso na produção dos itens 1 e 2, o que não deve ocorrer. Portanto não aceitamos $y_1 = 2$ e então fazemos:

$$y_1 = 1$$

e testamos a factibilidade. Com $y_1 = 1$ (e $y_2 = 3, y_3 = 3$), teremos o padrão $(1, 6, 0)^T$ executado 1 vez, o padrão $(1, 0, 0)^T$ executado 3 vezes e o padrão $(0, 0, 2)^T$ executado 3 vezes, produzindo 4 itens do tipo 1, 6 itens do tipo 2 e 6 itens do tipo 3. Com isso, vemos que a produção $[(4, 6, 6)^T]$ não está excedendo a demanda original $[(4, 8, 7)^T]$.

Como já analisamos a última componente do vetor solução $\tilde{\mathbf{x}}$, temos uma solução inteira aproximada para o problema original, dado na Tabela 5.8.

Quantidade de barras	Padrão de corte		
y^0	item 1	item 2	item 3
1	1	6	0
3	1	0	0
3	0	0	2

Tabela 5.10: *Solução inteira.*

Mas esta solução (dada na Tabela 5.10) não atende toda a demanda original. Então, guardamos esta solução, atualizamos a demanda (subtraindo os itens que já foram produzidos com esta primeira solução) e resolvemos novamente um problema com a demanda residual. Este problema é dado na Tabela 5.11.

Item	Comprimento	Demanda Residual: r^0
1	194	0
2	13	2
3	95	1

Tabela 5.11: *Novo problema a ser resolvido.*

Resolvemos este problema (relaxado) através do método *simplex* com geração de colunas. A solução obtida está na Tabela 5.12.

Quantidade de barras	Padrão de corte		
x	item 1	item 2	item 3
1	0	2	1

Tabela 5.12: *Solução relaxada.*

Como a solução é inteira, resolve otimamente o problema residual. Portanto, fim da heurística. A solução final é dada na Tabela 5.13 a seguir.

Quantidade de barras	Padrão de corte		
x	item 1	item 2	item 3
1	1	6	0
3	1	0	0
3	0	0	2
1	0	2	1

Tabela 5.13: *Solução Final - heurística Nova.*

5.5 Heurística Residual *Nova* - versão 2

Esta heurística difere da versão original (seção 5.4) apenas no passo 2.2 onde é feita a ordenação dos padrões de corte. Aqui, um outro critério é utilizado. Ordenamos o vetor solução em ordem não-crescente de perda no padrão. Caso haja empate, o critério de desempate é o mesmo da versão original, colocamos primeiro o padrão com maior frequência x_j .

5.5.1 Exemplo

Considere o seguinte problema de corte de estoque. Suponha que temos disponível, objetos de comprimento $L = 171$ a serem cortados para a produção de três tipos de itens ($m = 3$), com comprimentos l_i e demandas $d_i, i = 1, \dots, 3$ descritos na Tabela 5.14.

Item	Comprimento	Demanda
1	29	7
2	83	5
3	19	7

Tabela 5.14: *Dados do exemplo do problema de corte.*

Resolvemos este exemplo com a heurística *Nova* - Versão 2. O primeiro passo é resolver o problema relaxado através do método *simplex* com geração de colunas. A

solução obtida está na Tabela 5.15.

Quantidade de barras	Padrão de corte			Perda no padrão
	item 1	item 2	item 3	
1,555 ...	3	1	0	1
2,333 ...	1	1	3	2
0,555 ...	0	2	0	5

Tabela 5.15: *Solução relaxada.*

Agora, ordena-se o vetor solução $\tilde{\mathbf{x}}^0$. Esta ordem deve ser de acordo com a perda em cada padrão, de forma não-decrescente, assim, teremos:

$$\tilde{x}_1^0 \text{ (perda = 1), } \tilde{x}_2^0 \text{ (perda = 2), } \tilde{x}_3^0 \text{ (perda = 5).}$$

Inicia-se o arredondamento da primeira componente do vetor:

$$y_1 = \lceil 1,555 \dots \rceil = 2.$$

Testa-se a factibilidade. Com $y_1 = 2$ (e $y_2 = y_3 = 0$), teremos o padrão $(3, 1, 0)^T$ executado 2 vezes, produzindo 6 itens do tipo 1 e 2 itens do tipo 2, o que não excede a demanda original. Então passamos para a segunda componente do vetor $\tilde{\mathbf{x}}^0$. Fazemos:

$$y_2 = \lceil 2,333 \dots \rceil = 3$$

e testamos a factibilidade. Com $y_2 = 3$ (e $y_1 = 2, y_3 = 0$), serão produzidos 9 itens do tipo 1, 5 itens do tipo 2 e 9 itens do tipo 3. Nota-se que temos um excesso na produção dos itens 1 e 3, o que não deve ocorrer. Portanto não aceitamos $y_2 = 3$ e então fazemos:

$$y_2 = 2.$$

Testa-se a factibilidade. Com $y_2 = 2$ (e $y_1 = 2, y_3 = 0$), teremos o padrão $(1, 1, 3)^T$ executado 2 vezes e o padrão $(3, 1, 0)^T$ executado 2 vezes, produzindo 8 itens do tipo 1, 4 itens do tipo 2 e 6 itens do tipo 3. Mas, isto excede a demanda do item 1. Então, fazemos:

$$y_2 = 1.$$

Testa-se a factibilidade. Com $y_2 = 1$. Teremos 7 itens do tipo 1, 3 itens do tipo 2 e 3 itens do tipo 3 produzidos, o que não excede a demanda original. Passamos para a terceira componente:

$$y_3 = \lceil 0,555\dots \rceil = 1$$

e testamos a factibilidade. Com $y_3 = 1$ (e $y_1 = 2, y_2 = 1$), teremos o padrão $(0, 2, 0)^T$ executado 1 vez, o padrão $(3, 1, 0)^T$ executado 2 vezes e o padrão $(1, 1, 3)^T$ executado 1 vez, produzindo 7 itens do tipo 1, 5 itens do tipo 2 e 3 itens do tipo 3. Com isso, vemos que a produção $[(7, 5, 3)^T]$ não está excedendo a demanda original $[(7, 5, 7)^T]$.

Como já analisamos a última componente do vetor solução $\tilde{\mathbf{x}}^0$, temos uma solução inteira para o problema original, dado na Tabela 5.14.

Quantidade de barras	Padrão de corte			Perda no padrão
$\mathbf{y}^0$	item 1	item 2	item 3	
2	3	1	0	1
1	1	1	3	2
1	0	2	0	5

Tabela 5.16: *Solução inteira.*

Mas esta solução (dada na Tabela 5.16) não atende toda a demanda original. Então, guardamos esta solução, atualizamos a demanda (subtraindo os itens que já foram produzidos com esta primeira solução) e resolvemos novamente um problema com a demanda reduzida. Este problema é dado na Tabela 5.17.

Item	Comprimento	Demanda Residual: $\mathbf{r}^0$
1	29	0
2	83	0
3	19	4

Tabela 5.17: *Novo problema a ser resolvido.*

Resolvemos este problema (relaxado) através do método *simplex* com geração de colunas. A solução obtida está na Tabela 5.18.

Quantidade de barras	Padrão de corte			Perda no padrão
x	item 1	item 2	item 3	
1	0	0	4	95

Tabela 5.18: *Solução relaxada.*

A solução relaxada do problema residual é inteira e, portanto, ótima. A solução final é dada na Tabela 5.19 a seguir.

Quantidade de barras	Padrão de corte			Perda no padrão
x	item 1	item 2	item 3	
2	3	1	0	1
1	1	1	3	2
1	0	2	0	7
1	0	0	4	95

Tabela 5.19: *Solução final - heurística Nova - versão 2 - Perda total = 106.*

Observação

Há uma observação importante quanto à implementação destes procedimentos heurísticos. A cada iteração, um novo padrão é determinado pela heurística que fará parte da solução do problema, de modo que deve ser armazenado numa matriz, digamos, $\mathbf{A}$ (cada padrão é uma coluna de $\mathbf{A}$). Pode ocorrer da heurística gerar um padrão já existente na matriz $\mathbf{A}$, assim, estabelecemos uma ordem lexicográfica (isto é, a k -ésima coluna de $\mathbf{A}$, $\mathbf{a}^k$ é tal que: $\mathbf{a}_i^k = \mathbf{a}_i^{k+1}, i = 1, \dots, r$ e $\mathbf{a}_{r+1}^k > \mathbf{a}_{r+1}^{k+1}, i = r + 1, \dots, m$) para manter as colunas da matriz ordenadas de forma que, quando uma coluna tiver de ser inserida na matriz,

saberemos onde inseri-la, caso seja ‘nova’, ou caso já exista, apenas atualizamos a sua frequência; sem ter de percorrer toda a matriz.

Capítulo 6

Resultados Computacionais - Heurísticas de Arredondamento

Os testes computacionais aqui apresentados referem-se ao modelo (3.11), que trata o problema de corte com vários tipos de barras disponíveis em estoque, em quantidades limitadas, com o objetivo de minimizar a perda total. Alguns resultados para o modelo básico (3.3), no qual não há limitação no número de barras disponíveis em estoque pode ser visto em [23]. A análise dos resultados computacionais para os modelos (3.3) e (3.11) levou às mesmas conclusões.

As implementações foram desenvolvidas em Delphi 5 e todos os testes foram realizados em um computador Pentium III (866MHz) com 256 MB de memória RAM.

Realizamos alguns testes computacionais para comparar os resultados obtidos pelas quatro heurísticas residuais e também as heurísticas puramente construtivas. Para isto, geramos, aleatoriamente, 18 classes com 20 exemplos em cada classe, totalizando 360 exemplos. A seguir, mostraremos em detalhes os critérios usados para gerar os dados e os resultados obtidos.

6.1 O Gerador Aleatório

Definimos o seguinte gerador aleatório, baseado em Foerster e Wäscher [2]:

- *Número de barras em estoque:* $K = 3, 5$ e 7 ;

- *Dimensão das barras em estoque:* Os valores de $L_k, k = 1, \dots, K$ foram gerados aleatoriamente no intervalo $[1, 100]$;
- *Estoque disponível:* Os valores de $e_k, k = 1, \dots, K$ foram gerados aleatoriamente no intervalo $[1, 100\frac{m}{2}]$;
- *Número de tipos de itens:* Foram utilizados diferentes valores de $m = 5, 10, 20$ e 40 ;
- *Dimensão dos itens:* Os comprimentos dos itens l_i foram gerados aleatoriamente nos intervalos $[v_1L, v_2L]$, onde L é a média entre os valores $L_k, k = 1, \dots, K$, e v_1 e v_2 definem o menor e o maior valor de $l_i, i = 1, \dots, m$. Utilizou-se: $v_1 = 0,01$ e $v_2 = 0,2$ e $0,8$. Combinando estes valores, gerou-se classes com problemas com *itens pequenos* ($v_2 = 0,2$) e *itens médios* ($v_2 = 0,8$);
- *Demanda:* $d_i, i = 1, \dots, m$ no intervalo $[1, 10]$.

Foram criadas 18 classes de problemas combinando (K, m, l_i) e gerados 20 exemplos para cada classe.

6.2 Resultados

A Tabela 6.1 mostra as 18 classes de problemas com o número médio de objetos cortados e, a Tabela 6.2, o número médio de padrões de corte para cada uma das classes. O problema de determinar o menor número de padrões de cortes vem sendo analisado na literatura e o tratamos no Capítulo 8. Aqui mostramos o número de padrões que cada uma das heurísticas geram, sem que esforços tenham sido feitos para reduzi-los. A Tabela 6.3 mostra a perda total, que é o objetivo a ser minimizado. Em cada uma destas tabelas usamos **negrito** para a melhor solução e *itálico* para a pior.

O tempo computacional médio dos 360 problemas estão na Tabela 6.4.

	Dados			Número de objetos cortados					
				<i>Heurísticas Construtivas</i>		<i>Heurísticas Residuais</i>			
	<i>K</i>	<i>m</i>	Itens	<i>FFD</i>	<i>Gulosa</i>	<i>FFD</i>	<i>Gulosa</i>	<i>Nova</i>	<i>Nova v2</i>
C1	3	5	P	6,35	5,40	6,25	5,30	4,20	4,30
C2	3	5	M	12,20	10,85	10,75	10,50	10,45	10,45
C3	3	20	P	15,75	17,30	14,10	17,30	9,20	9,40
C4	3	20	M	48,85	26,90	35,85	34,80	31,55	31,45
C5	3	40	P	24,60	25,50	21,60	22,60	14,35	14,35
C6	3	40	M	72,75	56,45	52,35	48,35	46,10	46,10
C7	5	10	P	8,85	9,25	8,85	9,25	5,35	5,55
C8	5	10	M	27,35	24,65	20,25	19,60	18,60	19,20
C9	5	20	P	16,80	17,75	16,45	17,25	8,50	8,50
C10	5	20	M	42,55	35,25	33,95	31,70	30,10	30,15
C11	5	40	P	27,45	36,20	25,30	32,55	14,45	14,50
C12	5	40	M	82,40	68,95	58,55	57,20	54,55	54,45
C13	7	10	P	11,25	12,65	11,20	12,60	5,90	6,20
C14	7	10	M	22,05	19,95	17,80	16,95	16,30	16,15
C15	7	20	P	15,70	18,05	15,35	18,00	8,45	8,45
C16	7	20	M	35,55	29,45	26,90	24,80	22,15	22,20
C17	7	40	P	30,30	39,15	26,65	34,35	15,35	15,35
C18	7	40	M	87,00	74,45	62,40	60,40	57,75	57,70

Tabela 6.1: Média dos resultados dos 20 problemas-teste para cada uma das 18 classes.

	Dados			Número de padrões de corte					
				<i>Heurísticas Construtivas</i>		<i>Heurísticas Residuais</i>			
	<i>K</i>	<i>m</i>	Itens	<i>FFD</i>	<i>Gulosa</i>	<i>FFD</i>	<i>Gulosa</i>	<i>Nova</i>	<i>Nova v2</i>
C1	3	5	P	4,40	4,50	4,35	4,40	3,95	4,00
C2	3	5	M	6,45	5,90	5,95	5,80	5,10	5,10
C3	3	20	P	12,65	13,05	12,15	13,25	9,15	9,20
C4	3	20	M	20,70	19,35	20,90	20,25	16,65	16,50
C5	3	40	P	21,95	21,95	20,50	21,30	14,30	14,30
C6	3	40	M	38,10	34,20	38,20	34,90	29,30	29,30
C7	5	10	P	7,10	7,45	7,10	7,45	5,25	5,45
C8	5	10	M	11,10	10,90	11,05	10,75	8,85	9,15
C9	5	20	P	13,35	14,85	12,85	14,45	8,45	8,45
C10	5	20	M	19,90	18,35	21,40	19,85	15,30	15,40
C11	5	40	P	24,30	29,30	23,50	27,70	14,45	14,50
C12	5	40	M	39,10	36,45	38,45	37,60	30,30	30,25
C13	7	10	P	8,20	8,60	8,15	8,55	5,45	5,75
C14	7	10	M	10,85	10,85	10,90	10,50	8,65	9,10
C15	7	20	P	13,45	13,50	13,25	13,60	8,30	8,30
C16	7	20	M	19,70	18,60	19,10	18,10	14,60	14,60
C17	7	40	P	23,20	27,70	22,90	27,55	15,20	15,20
C18	7	40	M	39,35	35,50	40,05	38,70	30,50	30,60

Tabela 6.2: Média dos resultados dos 20 problemas-teste para cada uma das 18 classes.

	Dados			Perda Total (média)					
				<i>Heurísticas Construtivas</i>		<i>Heurísticas Residuais</i>			
	<i>K</i>	<i>m</i>	Itens	<i>FFD</i>	<i>Gulosa</i>	<i>FFD</i>	<i>Gulosa</i>	<i>Nova</i>	<i>Nova v2</i>
C1	3	5	P	207	177	185	176	166	151
C2	3	5	M	692	524	407	395	438	409
C3	3	20	P	167	187	142	185	129	133
C4	3	20	M	897	842	219	276	131	136
C5	3	40	P	189	115	152	120	154	155
C6	3	40	M	835	852	297	373	191	133
C7	5	10	P	121	125	121	125	90	95
C8	5	10	M	672	1034	383	428	331	364
C9	5	20	P	115	125	116	127	98	100
C10	5	20	M	559	846	226	290	168	159
C11	5	40	P	132	127	134	134	106	114
C12	5	40	M	478	780	174	214	116	110
C13	7	10	P	47	81	49	80	56	59
C14	7	10	M	292	466	131	172	112	118
C15	7	20	P	109	97	94	98	96	95
C16	7	20	M	138	274	114	160	85	61
C17	7	40	P	94	91	107	82	100	102
C18	7	40	M	410	1159	139	220	93	89

Tabela 6.3: Média dos resultados dos 20 problemas-teste para cada uma das 18 classes.

	<i>FFD</i>	<i>Gulosa</i>	<i>FFD</i>	<i>Gulosa</i>	<i>Nova</i>	<i>Nova</i>
	<i>Pura</i>	<i>Pura</i>	<i>Residual</i>	<i>Residual</i>		<i>versão 2</i>
Tempo (média) (seg)	0,07	0,94	9,3	9,35	6,86	6,88

Tabela 6.4: Tempo computacional médio dos 360 problemas.

6.3 Análise dos Resultados

Podemos observar que as heurísticas Nova e Nova-versão 2, seguidas das Residual FFD e Residual Gulosa foram as que melhor se comportaram, enquanto que as heurísticas construtivas FFD e Gulosa Puras, apenas para poucas classes obtiveram bons resultados. Como a heurística Nova é também do tipo residual, os resultados mostram claramente que a abordagem de resolver o problema original relaxado da condição de integralidade (usando o método simplex com geração de colunas) e procurar uma estratégia de arredondamento é bem eficiente, independentemente da demanda ser baixa ou alta. Isto será reforçado nos problemas da seção 6.4.

A heurística que mais obteve soluções com perdas menores foi a heurística Nova, que apresentou-se melhor em 7 das 18 classes de exemplos, seguida da heurística Nova-versão 2, que ganhou em 6 das 18 classes. A heurística Gulosa Pura foi a melhor em 2 classes, enquanto que as outras três: FFD Residual, FFD Pura e Gulosa Residual foram as melhores em apenas uma classe cada uma.

Quanto ao tempo computacional, a heurística mais lenta é a Gulosa Residual, pois ela consiste em resolver o problema da mochila muitas vezes, sendo este o passo ‘mais caro’. A heurística FFD Pura foi a mais rápida (o problema da mochila é resolvido heurísticamente), mas podemos notar que não se comporta muito bem quanto à qualidade das soluções. Apesar de mais lentas, as heurísticas residuais são viáveis para a resolução de problemas práticos.

Quanto ao número de padrões de corte, podemos notar na Tabela 6.2 que a heurística Nova e também a versão 2 obtiveram em média o menor número de padrões de corte. Este não foi o objetivo proposto (sobre este assunto, trataremos a seguir, na seção 8.1), mas é um sub-produto interessante destas heurísticas.

Observe que as heurísticas novas, para as classes em que tiveram os resultados piores, não foram tão ruins, veja por exemplo, as classes 11 e 13. Por outro lado, as heurísticas puras, para as classes em que tiveram os resultados piores, produziram soluções catastróficas, por exemplo, a classe 18, com perda 13 vezes superior à melhor solução encontrada.

Para uma melhor análise comparativa entre heurísticas construtivas e residuais, foi feita a média dos valores dados nas tabelas anteriores para as heurísticas construtivas e

residuais. Os três gráficos a seguir referem-se a estas médias.

Figura 6.1: *Número médio de objetos cortados.*

A Figura 6.1 mostra que as heurísticas residuais utilizaram, em média, um número menor de objetos em estoque para produzir todos os itens demandados.

Figura 6.2: *Número médio de padrões de corte.*

A Figura 6.2 mostra que as heurísticas residuais utilizaram, em média, um número menor de padrões de corte que as heurísticas puramente construtivas.

Figura 6.3: *Perda total (média).*

A Figura 6.3 mostra que as heurísticas residuais obtiveram as menores perdas. Notamos claramente que as heurísticas construtivas foram ainda piores nas classes de números pares que são classes onde os itens considerados são de tamanho médio, mais difíceis de serem combinados que itens pequenos.

Assim, concluímos que não se deve usar heurísticas construtivas, sob pena de produzir soluções desastrosas. As heurísticas novas são as mais recomendadas.

6.4 Outra análise

Realizamos uma outra análise do comportamento das heurísticas quando alguns itens têm demanda baixa (considerou-se $d_i = 1$ ou 2) e outros têm demanda alta.

Criamos 5 classes com 20 exemplos em cada uma. Consideramos:

- número de tipos de barras em estoque: $K = 3$;
- número de tipos de itens: $m = 20$;
- itens com demanda baixa: $d_i = 1$ ou 2 ;
- itens com demanda alta: $d_i \in \left[5 \left\lfloor \frac{L}{l_i} \right\rfloor, 10 \left\lfloor \frac{L}{l_i} \right\rfloor \right]$, com L sendo a média dos $L_k, k = 1, \dots, K$.

Os demais dados foram gerados tal como descrito na seção 6.1. Na segunda e terceira colunas da Tabela 6.5 tem-se o número de itens que, dentre os 20, têm demanda baixa e alta, respectivamente. Por exemplo, a classe D2 tem 5 itens com demanda pequena e 15 itens com demanda grande.

	Dados		Número de objetos cortados					
			<i>Construtivas</i>		<i>Residuais</i>			
	<i>Demanda pequena</i>	<i>Demanda grande</i>	<i>FFD</i>	<i>Gulosa</i>	<i>FFD</i>	<i>Gulosa</i>	<i>Nova</i>	<i>Nova versão 2</i>
D1	0	20	192,90	178,05	100,75	101,30	96,10	96,15
D2	5	15	140,35	116,90	71,65	72,65	68,90	68,90
D3	10	10	101,55	95,30	54,05	55,45	50,80	50,90
D4	15	5	60,65	52,05	29,30	30,10	26,80	26,75
D5	20	0	5,90	6,95	5,9	6,95	4,25	4,30

Tabela 6.5: *Exemplos com demandas pequena e grande.*

	Dados		Número de padrões de cortes					
			<i>Construtivas</i>		<i>Residuais</i>			
	<i>Demanda pequena</i>	<i>Demanda grande</i>	<i>FFD</i>	<i>Gulosa</i>	<i>FFD</i>	<i>Gulosa</i>	<i>Nova</i>	<i>Nova versão 2</i>
D1	0	20	39,10	36,25	32,50	34,40	28,90	28,95
D2	5	15	32,65	30,95	28,90	28,70	24,25	24,25
D3	10	10	25,25	23,60	25	25,35	20,75	20,85
D4	15	5	17,05	16,60	17	17,45	14	14
D5	20	0	5,65	6,40	5,65	6,40	4,20	4,25

Tabela 6.6: *Exemplos com demandas pequena e grande.*

	Dados		Perda Total (média)					
			<i>Construtivas</i>		<i>Residuais</i>			
	<i>Demanda pequena</i>	<i>Demanda grande</i>	<i>FFD</i>	<i>Gulosa</i>	<i>FFD</i>	<i>Gulosa</i>	<i>Nova</i>	<i>Nova versão 2</i>
D1	0	20	352,85	225,10	195,10	152,35	114,55	121,55
D2	5	15	309,75	259	187,40	144,50	113,35	113,35
D3	10	10	299,50	182,30	154,50	137,95	137,80	131,85
D4	15	5	389,05	387,45	179,95	169,20	131,05	131,65
D5	20	0	139,45	201,75	139,45	201,75	154,20	134,80

Tabela 6.7: *Exemplos com demandas pequena e grande.*

Concluimos, mais uma vez, que as heurísticas residuais mostraram-se superior às puras em número de objetos cortados, em perda, e também, quanto ao número de padrões de corte. Dentre as residuais podemos claramente perceber que as heurísticas novas foram as melhores em todos os critérios analisados. Veja por exemplo na Tabela 6.7, a classe D5, com todos os itens com demanda baixa. A solução obtida pela heurística gulosa (normalmente seguida pelo folclore da área de problemas de corte) é 66% pior que a solução obtida pela heurística residual Nova-versão 2. Em outras classes a diferença percentual é ainda maior.

Capítulo 7

Alguns Exemplos Resolvidos

Neste capítulo apresentamos alguns exemplos de problemas de corte de estoque. Esses exemplos foram escolhidos aleatoriamente nas classes apresentadas no Capítulo 6 e resolvidos usando cada um dos métodos apresentados anteriormente.

7.1 Exemplo 1

Exemplo da Classe 2, com número de tipos de barras: $K = 3$ e número de itens: $m = 5$.

Id Barra	Comprimento	Estoque
1	205	20
2	519	2
3	833	10

Tabela 7.1: *Dados do Exemplo 1 - Barras.*

Item	Comprimento	Demanda
1	239	2
2	402	5
3	45	2
4	208	4
5	371	9

Tabela 7.2: *Dados do Exemplo 1 - Itens.*

Solução do Método Simplex com Geração de Colunas:

Id Barra	$\tilde{x}$	Padrão					
3	2	1	0	0	1	1	
3	2,5	0	2	0	0	0	
3	1	0	0	2	0	2	
3	0,5	0	0	0	4	0	
3	2,5	0	0	0	0	2	

Tabela 7.3: *Solução relaxada.*

Solução da Heurística FFD Pura:

Id Barra	x	Padrão					
2	1	1	0	0	1	0	
2	1	0	1	2	0	0	
3	1	1	0	0	1	1	
3	2	0	2	0	0	0	
3	1	0	0	0	2	0	
3	4	0	0	0	0	2	

Tabela 7.4: *Solução da heurística FFD pura.*

Solução da Heurística Gulosa Pura:

Id Barra	x	Padrão					
2	1	2	0	0	0	0	0
3	2	0	2	0	0	0	0
3	1	0	1	0	0	0	1
3	1	0	0	2	0	0	2
3	1	0	0	0	4	0	0
3	3	0	0	0	0	0	2

Tabela 7.5: *Solução da heurística Gulosa pura.*

Solução da Heurística Residual FFD:

Id Barra	x	Padrão					
2	1	0	0	0	2	0	0
3	2	1	0	0	1	1	0
3	2	0	2	0	0	0	0
3	1	0	1	0	0	0	1
3	1	0	0	2	0	0	2
3	2	0	0	0	0	0	2

Tabela 7.6: *Solução da heurística residual FFD.*

Solução da Heurística Residual Gulosa:

Id Barra	x	Padrão					
2	1	0	0	0	0	1	
3	2	1	0	0	1	1	
3	2	0	2	0	0	0	
3	1	0	1	0	2	0	
3	1	0	0	2	0	2	
3	2	0	0	0	0	2	

Tabela 7.7: *Solução da heurística residual Gulosa.*

Solução da Heurística Residual Nova:

Id Barra	x	Padrão					
2	1	0	1	1	0	0	
3	2	1	0	0	1	1	
3	2	0	2	0	0	0	
3	1	0	0	1	2	1	
3	3	0	0	0	0	2	

Tabela 7.8: *Solução da heurística residual Nova.*

Solução da Heurística Residual Nova - versão 2:

A solução encontrada pela *heurística residual Nova - versão 2* foi a mesma encontrada pela *heurística residual Gulosa* apresentada anteriormente.

	Número de Objetos	Número de Padrões	Perda Total	Tempo (s)
Relaxada	8,5	5	331,5	0,06
FFD Pura	10	6	953	0
Gulosa Pura	9	6	434	0
FFD Residual	9	6	434	0,11
Gulosa Residual	9	6	434	0,05
Nova	9	5	434	0,11
Nova - versão 2	9	6	434	0,06

Tabela 7.9: *Solução do Exemplo 1.*

7.2 Exemplo 2

Exemplo da Classe 7, com número de tipos de barras: $K = 5$ e número de itens: $m = 10$.

Id Barra	Comprimento	Estoque
1	654	25
2	223	5
3	106	30
4	675	25
5	558	5

Tabela 7.10: *Dados do Exemplo 2 - Barras.*

Item	Comprimento	Demanda
1	65	7
2	13	8
3	22	7
4	85	4
5	43	6
6	52	7
7	9	5
8	72	8
9	81	7
10	29	1

Tabela 7.11: *Dados do Exemplo 2 - Itens.*

Solução do Método Simplex com Geração de Colunas:

Id Barra	$\tilde{x}$	Padrão										
1	0,332	0	8	7	0	0	0	0	1	4	0	
1	0,245	0	7	0	0	6	5	5	0	0	0	
1	0,624	0	2	0	4	0	0	0	4	0	0	
1	0,223	0	1	0	0	0	0	4	8	0	1	
4	0,82	6	2	0	0	1	0	0	3	0	0	
4	0,519	4	0	1	0	0	7	0	0	0	1	
4	0,258	0	2	0	0	5	0	0	0	5	1	
4	0,376	0	0	7	4	4	0	1	0	0	0	
4	0,305	0	0	5	0	3	7	0	1	0	0	
4	0,626	0	0	0	0	0	0	4	1	7	0	

Tabela 7.12: *Solução relaxada.*

Solução da Heurística FFD Pura:

Id Barra	x	Padrão											
2	2	0	0	0	2	0	1	0	0	0	0	0	0
2	3	0	0	0	0	0	1	1	0	2	0	0	0
3	1	0	3	3	0	0	0	0	0	0	0	0	0
3	1	0	3	0	0	0	0	0	0	0	0	0	0
3	2	0	1	0	0	2	0	0	0	0	0	0	0
3	1	0	0	3	0	0	0	1	0	0	0	1	0
3	1	0	0	0	0	1	1	1	0	0	0	0	0
4	1	7	0	1	0	0	1	0	2	0	0	0	0
5	1	0	0	0	0	1	0	0	6	1	0	0	0

Tabela 7.13: *Solução da heurística FFD pura.*

Solução da Heurística Gulosa Pura:

Id Barra	x	Padrão											
1	1	7	8	3	0	0	0	0	0	0	0	1	0
1	1	0	0	4	4	3	1	5	0	0	0	0	0
1	1	0	0	0	0	3	3	0	4	1	0	0	0
2	1	0	0	0	0	0	0	0	3	0	0	0	0
3	1	0	0	0	0	0	2	0	0	0	0	0	0
3	1	0	0	0	0	0	1	0	0	0	0	0	0
5	1	0	0	0	0	0	0	0	1	6	0	0	0

Tabela 7.14: *Solução da heurística Gulosa pura.*

Solução da Heurística Residual FFD:

A solução obtida através da *heurística residual FFD* é exatamente a mesma encontrada com a heurística FFD, apresentada anteriormente. Observe que todos os componentes de $\tilde{\mathbf{x}}$ são menores que 1.

Solução da Heurística Residual Gulosa:

A solução obtida pela *heurística residual Gulosa* foi a mesma encontrada pela heurística Gulosa pura, apresentada anteriormente.

Solução da Heurística Residual Nova:

Id Barra	x	Padrão										
1	1	0	2	0	4	0	0	0	4	0	0	
2	1	1	0	0	0	0	3	0	0	0	0	
2	1	0	2	7	0	1	0	0	0	0	0	
2	1	0	1	0	0	4	0	1	0	0	1	
2	1	0	1	0	0	0	4	0	0	0	0	
4	1	6	2	0	0	1	0	0	3	0	0	
4	1	0	0	0	0	0	0	4	1	7	0	

Tabela 7.15: *Solução da heurística residual Nova.*

Solução da Heurística Residual Nova - versão 2:

A solução obtida pela *heurística residual Nova - versão 2* foi a mesma encontrada pela heurística residual Nova, apresentada anteriormente.

Observe que as escolhas dos objetos em estoque foram completamente diferentes. As heurísticas Nova não escolheram o objeto 3.

	Número de Objetos	Número de Padrões	Perda Total	Tempo (s)
Relaxada	4,328	10	0	0,22
FFD Pura	13	9	92	0
Gulosa Pura	7	7	63	0
FFD Residual	13	9	92	0,22
Gulosa Residual	7	7	63	0,17
Nova	7	7	4	0,44
Nova - versão 2	7	7	4	0,49

Tabela 7.16: *Solução do Exemplo 2.*

7.3 Exemplo 3

Exemplo da Classe 8, com número de tipos de barras: $K = 5$ e número de itens: $m = 10$.

Id Barra	Comprimento	Estoque
1	219	85
2	402	25
3	671	20
4	553	45
5	322	20

Tabela 7.17: *Dados do Exemplo 3 - Barras.*

Item	Comprimento	Demanda
1	159	9
2	120	7
3	127	3
4	88	8
5	238	2
6	95	7
7	56	9
8	245	9
9	206	9
10	62	9

Tabela 7.18: *Dados do Exemplo 3 - Itens.*

Solução do Método Simplex com Geração de Colunas:

Id Barra	$\tilde{x}$	Padrão										
3	1,3	2	2	0	0	0	0	2	0	0	0	0
3	0,208	2	0	0	4	0	0	0	0	0	0	0
3	0,939	2	0	0	0	1	0	2	0	0	0	0
3	1,061	1	0	0	1	1	0	0	0	0	0	3
3	3,044	1	0	0	0	0	1	0	0	2	0	0
3	1,466	0	3	2	0	0	0	1	0	0	0	0
3	0,034	0	0	2	0	0	0	0	0	2	0	0
3	1,527	0	0	0	4	0	0	2	0	1	0	0
4	1,319	0	0	0	0	0	3	0	0	1	1	0
4	4,5	0	0	0	0	0	0	0	2	0	1	0

Tabela 7.19: *Solução relaxada.*

Solução da Heurística FFD Pura:

Id Barra	x	Padrão										
1	7	1	0	0	0	0	0	1	0	0	0	0
1	1	0	0	1	1	0	0	0	0	0	0	0
1	7	0	0	0	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	0	0	0	1
2	1	1	0	0	0	1	0	0	0	0	0	0
2	1	0	0	0	0	0	4	0	0	0	0	0
3	1	1	0	0	0	0	1	0	0	2	0	0
3	1	0	4	0	0	0	2	0	0	0	0	0
3	1	0	3	2	0	0	0	1	0	0	0	0
4	1	0	0	0	4	0	0	0	0	0	0	3
4	1	0	0	0	0	1	0	0	1	0	0	1
4	4	0	0	0	0	0	0	0	2	0	0	1
5	1	0	0	0	3	0	0	1	0	0	0	0

Tabela 7.20: *Solução da heurística FFD pura.*

Solução da Heurística Gulosa Pura:

Id Barra	x	Padrão										
1	4	0	0	0	0	0	1	0	0	0	0	2
1	7	0	0	0	0	0	0	0	0	1	0	0
2	1	0	0	1	0	0	0	0	1	0	0	0
2	1	0	0	0	0	0	1	0	1	0	0	1
3	1	2	2	0	0	0	0	2	0	0	0	0
3	2	2	0	0	4	0	0	0	0	0	0	0
3	1	1	1	0	0	0	0	7	0	0	0	0
3	1	0	4	0	0	0	2	0	0	0	0	0
3	1	0	0	2	0	0	0	0	0	2	0	0
4	1	0	0	0	0	1	0	0	1	0	0	0
4	3	0	0	0	0	0	0	0	2	0	0	0
5	1	2	0	0	0	0	0	0	0	0	0	0
5	1	0	0	0	0	1	0	0	0	0	0	0

Tabela 7.21: *Solução da heurística Gulosa pura.*

Solução da Heurística Residual FFD:

Id Barra	x	Padrão										
1	3	1	0	0	0	0	0	1	0	0	0	0
1	1	0	0	0	2	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	1	0	0	0	0
1	1	0	0	0	0	0	0	0	0	1	0	0
3	1	2	2	0	0	0	0	2	0	0	0	0
3	1	1	0	0	1	1	0	0	0	0	3	0
3	3	1	0	0	0	0	1	0	0	2	0	0
3	1	0	3	2	0	0	0	1	0	0	0	0
3	1	0	0	0	4	0	0	2	0	1	0	0
4	1	0	2	1	1	0	1	0	0	0	0	0
4	1	0	0	0	0	1	0	0	1	0	1	0
4	1	0	0	0	0	0	3	0	0	1	1	0
4	4	0	0	0	0	0	0	0	2	0	1	0

Tabela 7.22: *Solução da heurística residual FFD.*

Solução da Heurística Residual Gulosa:

Id Barra	x	Padrão										
1	1	1	0	0	0	0	0	1	0	0	0	0
1	1	0	0	0	0	0	0	0	0	1	0	0
2	1	0	0	1	0	1	0	0	0	0	0	0
2	1	0	0	0	0	0	1	0	1	0	1	0
3	2	2	2	0	0	0	0	2	0	0	0	0
3	1	1	0	0	1	1	0	0	0	0	3	0
3	3	1	0	0	0	0	1	0	0	2	0	0
3	1	0	3	2	0	0	0	1	0	0	0	0
3	1	0	0	0	4	0	0	2	0	1	0	0
4	1	0	0	0	0	0	3	0	0	1	1	0
4	4	0	0	0	0	0	0	0	2	0	1	0
5	1	0	0	0	3	0	0	1	0	0	0	0

Tabela 7.23: *Solução da heurística residual Gulosa.*

Solução da Heurística Residual Nova:

Id Barra	x	Padrão										
1	1	0	0	1	0	0	0	0	0	0	0	0
2	1	0	0	0	0	0	1	0	1	0	1	
3	2	2	2	0	0	0	0	2	0	0	0	
3	1	1	0	0	1	1	0	0	0	0	3	
3	4	1	0	0	0	0	1	0	0	2	0	
3	1	0	3	2	0	0	0	1	0	0	0	
3	1	0	0	0	4	0	0	2	0	1	0	
4	1	0	0	0	0	1	2	1	0	0	1	
4	4	0	0	0	0	0	0	0	2	0	1	
5	1	0	0	0	3	0	0	1	0	0	0	

Tabela 7.24: *Solução da heurística residual Nova.*

Solução da Heurística Residual Nova - versão 2:

Id Barra	x	Padrão										
2	2	1	0	0	0	1	0	0	0	0	0	
2	1	0	0	0	0	0	1	0	1	0	1	
3	2	2	2	0	0	0	0	2	0	0	0	
3	3	1	0	0	0	0	1	0	0	2	0	
3	1	0	3	2	0	0	0	1	0	0	0	
3	2	0	0	0	4	0	0	2	0	1	0	
4	1	0	0	0	0	0	3	0	0	1	1	
4	4	0	0	0	0	0	0	0	2	0	1	
5	1	0	0	1	0	0	0	0	0	0	3	

Tabela 7.25: *Solução da heurística residual Nova - versão 2.*

	Número de Objetos	Número de Padrões	Perda Total	Tempo (s)
Relaxada	15,398	10	27,21	0,17
FFD Pura	28	13	343	0
Gulosa Pura	25	13	477	0,11
FFD Residual	20	13	264	0,44
Gulosa Residual	18	12	79	0,43
Nova	17	10	129	0,72
Nova - versão 2	17	9	43	0,38

Tabela 7.26: *Solução do Exemplo 3.*

7.4 Exemplo 4

Exemplo da Classe 13, com número de tipos de barras: $K = 7$ e número de itens: $m = 10$.

Id Barra	Comprimento	Estoque
1	477	40
2	359	35
3	928	10
4	811	5
5	380	35
6	263	30
7	832	5

Tabela 7.27: *Dados do Exemplo 4 - Barras.*

Item	Comprimento	Demanda
1	108	4
2	40	8
3	52	7
4	9	8
5	91	7
6	103	5
7	35	6
8	47	9
9	73	8
10	85	7

Tabela 7.28: *Dados do Exemplo 4 - Itens.*

Solução do Método Simplex com Geração de Colunas:

Id Barra	$\tilde{x}$	Padrão										
3	0,581	4	8	0	0	1	0	0	0	0	1	
3	0,221	3	0	6	1	0	0	6	0	1	0	
3	1,013	1	0	2	0	6	0	0	0	0	2	
3	0,36	0	6	2	0	0	0	0	0	8	0	
3	0,057	0	6	0	8	6	0	2	0	0	0	
3	0,128	0	1	0	8	0	0	2	5	7	0	
3	0,717	0	1	0	0	0	0	6	9	0	3	
3	0,207	0	0	7	6	0	0	0	0	0	6	
3	1	0	0	0	4	0	5	0	0	4	1	
7	0,212	0	0	7	5	0	0	0	9	0	0	

Tabela 7.29: *Solução relaxada.*

Solução da Heurística FFD Pura:

Id Barra	x	Padrão										
1	1	0	1	0	0	2	0	0	0	0	0	3
1	1	0	0	1	0	0	0	0	9	0	0	0
2	1	3	0	0	0	0	0	1	0	0	0	0
2	1	0	5	0	2	0	0	4	0	0	0	0
3	1	0	0	1	1	5	4	0	0	0	0	0
5	1	0	1	0	0	0	0	0	0	0	0	4
6	1	1	0	1	0	0	1	0	0	0	0	0
6	1	0	0	0	5	0	0	1	0	0	0	0
7	1	0	1	4	0	0	0	0	0	8	0	0

Tabela 7.30: *Solução da heurística FFD pura.*

Solução da Heurística Gulosa Pura:

Id Barra	x	Padrão										
1	1	4	0	0	5	0	0	0	0	0	0	0
1	1	0	8	2	2	0	0	1	0	0	0	0
1	1	0	0	5	1	1	0	2	1	0	0	0
1	1	0	0	0	0	1	0	2	0	2	2	0
1	1	0	0	0	0	1	0	1	1	3	1	0
1	1	0	0	0	0	0	1	0	3	2	1	0
3	1	0	0	0	0	4	3	0	0	0	0	3
6	1	0	0	0	0	0	0	0	4	1	0	0
6	1	0	0	0	0	0	1	0	0	0	0	0

Tabela 7.31: *Solução da heurística Gulosa pura.*

Solução da Heurística Residual FFD:

Id Barra	x	Padrão											
1	1	0	4	0	0	0	0	1	6	0	0		
2	1	3	0	0	0	0	0	1	0	0	0		
2	1	0	0	4	1	0	0	0	3	0	0		
3	1	1	0	2	0	6	0	0	0	0	2		
3	1	0	0	0	4	0	5	0	0	4	1		
4	1	0	0	1	0	1	0	1	0	4	4		
6	1	0	4	0	3	0	0	2	0	0	0		
6	1	0	0	0	0	0	0	1	0	0	0		

Tabela 7.32: *Solução da heurística residual FFD.*

Solução da Heurística Residual Gulosa:

Id Barra	x	Padrão											
1	1	3	2	0	0	0	0	0	0	1	0		
1	1	0	6	1	3	0	0	0	0	1	1		
1	1	0	0	4	1	0	0	5	0	0	1		
2	1	0	0	0	0	0	0	0	4	0	2		
3	1	1	0	2	0	6	0	0	0	0	2		
3	1	0	0	0	4	0	5	0	0	4	1		
5	1	0	0	0	0	1	0	0	3	2	0		
6	1	0	0	0	0	0	0	1	2	0	0		

Tabela 7.33: *Solução da heurística residual Gulosa.*

Solução da Heurística Residual Nova:

Id Barra	x	Padrão										
3	1	2	6	5	4	1	0	0	0	0	0	1
3	1	1	0	2	0	6	0	0	0	0	0	2
3	1	0	1	0	0	0	0	6	9	0	0	3
3	1	0	0	0	4	0	5	0	0	4	0	1
5	1	1	1	0	0	0	0	0	0	3	0	0
6	1	0	0	0	0	0	0	0	0	1	0	0

Tabela 7.34: *Solução da heurística residual Nova.*

Solução da Heurística Residual Nova - versão 2:

Id Barra	x	Padrão										
3	1	2	6	5	4	1	0	0	0	0	0	1
3	1	1	0	2	0	6	0	0	0	0	0	2
3	1	0	0	0	4	0	5	0	0	4	0	1
3	1	0	1	0	0	0	0	6	9	0	0	3
6	1	1	0	0	0	0	0	0	0	1	0	0
6	1	0	1	0	0	0	0	0	0	3	0	0

Tabela 7.35: *Solução da heurística residual Nova - versão 2.*

	Número de Objetos	Número de Padrões	Perda Total	Tempo (s)
Relaxada	4,496	10	0	0,18
FFD Pura	9	9	186	0
Gulosa Pura	9	9	164	0,06
FFD Residual	8	8	236	0,49
Gulosa Residual	8	8	137	0,49
Nova	6	6	203	0,44
Nova - versão 2	6	6	86	0,54

Tabela 7.36: *Solução do Exemplo 4.*

7.5 Exemplo 5

Apresentamos, agora, um exemplo gerado aleatoriamente, que não pertence àquelas classes do Capítulo 6. Este é um exemplo com uma variedade maior de tamanhos de demanda: $\mathbf{d} \in [50, 500]$. Neste exemplo, o número de tipos de barras é $K = 7$ e o número de itens é $m = 10$.

Id Barra	Comprimento	Estoque
1	942	1485
2	825	900
3	394	3745
4	276	3160
5	845	1005
6	728	420
7	297	3265

Tabela 7.37: *Dados do Exemplo 5 - Barras.*

Item	Comprimento	Demanda
1	401	181
2	329	341
3	64	215
4	342	299
5	270	113
6	283	392
7	211	59
8	488	134
9	223	182
10	151	238

Tabela 7.38: *Dados do Exemplo 5 - Itens.*

Solução do Método Simplex com Geração de Colunas:

Id Barra	$\tilde{x}$	Padrão										
1	61	2	0	2	0	0	0	0	0	0	0	0
1	59	1	1	0	0	0	0	1	0	0	0	0
1	84,5	0	2	0	0	0	1	0	0	0	0	0
1	113	0	1	0	1	1	0	0	0	0	0	0
1	23,25	0	0	4	2	0	0	0	0	0	0	0
1	153,75	0	0	0	0	0	2	0	0	1	1	1
1	14,125	0	0	0	0	0	0	0	1	2	0	0
1	21,269	0	0	0	0	0	0	0	1	0	3	3
5	20,446	0	0	0	2	0	0	0	0	0	1	1
5	98,607	0	0	0	1	0	0	0	1	0	0	0

Tabela 7.39: *Solução relaxada.*

Solução da Heurística FFD Pura:

Id Barra	x	Padrão									
1	90	2	0	2	0	0	0	0	0	0	0
1	1	1	0	0	1	0	0	0	0	0	1
2	88	0	2	0	0	0	0	0	0	0	1
2	134	0	1	0	0	0	0	0	1	0	0
3	31	0	1	0	0	0	0	0	0	0	0
4	113	0	0	0	0	1	0	0	0	0	0
4	3	0	0	0	0	0	0	1	0	0	0
4	147	0	0	0	0	0	0	0	0	1	0
5	149	0	0	0	2	0	0	0	0	0	1
5	14	0	0	0	0	0	0	4	0	0	0
7	35	0	0	1	0	0	0	0	0	1	0
7	392	0	0	0	0	0	1	0	0	0	0

Tabela 7.40: *Solução heurística FFD pura.*

Solução da Heurística Residual FFD:

Id Barra	x	Padrão									
1	61	2	0	2	0	0	0	0	0	0	0
1	59	1	1	0	0	0	0	1	0	0	0
1	84	0	2	0	0	0	1	0	0	0	0
1	113	0	1	0	1	1	0	0	0	0	0
1	23	0	0	4	2	0	0	0	0	0	0
1	154	0	0	0	0	0	2	0	0	1	1
1	14	0	0	0	0	0	0	0	1	2	0
1	21	0	0	0	0	0	0	0	1	0	3
2	1	0	1	0	0	0	0	0	1	0	0
4	1	0	0	1	0	0	0	0	0	0	0
5	21	0	0	0	2	0	0	0	0	0	1
5	98	0	0	0	1	0	0	0	1	0	0

Tabela 7.41: *Solução heurística residual FFD.*

Solução da Heurística Gulosa Pura:

Id Barra	x	Padrão									
1	59	1	1	0	0	0	0	1	0	0	0
1	21	1	0	0	0	2	0	0	0	0	0
1	105	0	2	0	0	0	1	0	0	0	0
1	71	0	1	3	0	1	0	0	0	0	1
1	22	0	0	0	2	0	0	0	0	1	0
1	1	0	0	0	0	0	2	0	0	1	1
1	79	0	0	0	0	0	0	0	1	2	0
1	55	0	0	0	0	0	0	0	1	0	3
2	50	2	0	0	0	0	0	0	0	0	0
3	1	0	0	0	1	0	0	0	0	0	0
5	1	0	1	0	0	0	1	0	0	1	0
5	1	0	0	2	0	0	2	0	0	0	1
6	1	1	0	0	0	0	0	0	0	0	0
6	127	0	0	0	2	0	0	0	0	0	0
7	282	0	0	0	0	0	1	0	0	0	0

Tabela 7.42: *Solução heurística Gulosa pura.*

Solução da Heurística Residual Gulosa:

Id Barra	x	Padrão									
1	61	2	0	2	0	0	0	0	0	0	0
1	59	1	1	0	0	0	0	1	0	0	0
1	84	0	2	0	0	0	1	0	0	0	0
1	113	0	1	0	1	1	0	0	0	0	0
1	23	0	0	4	2	0	0	0	0	0	0
1	154	0	0	0	0	0	2	0	0	1	1
1	14	0	0	0	0	0	0	0	1	2	0
1	21	0	0	0	0	0	0	0	1	0	3
3	1	0	1	1	0	0	0	0	0	0	0
5	21	0	0	0	2	0	0	0	0	0	1
5	98	0	0	0	1	0	0	0	1	0	0
6	1	0	0	0	0	0	0	0	1	0	0

Tabela 7.43: *Solução heurística residual Gulosa.*

Solução da Heurística Residual Nova:

Id Barra	x	Padrão									
1	62	2	0	2	0	0	0	0	0	0	0
1	57	1	1	0	0	0	0	1	0	0	0
1	1	0	2	1	0	0	0	1	0	0	0
1	84	0	2	0	0	0	1	0	0	0	0
1	113	0	1	0	1	1	0	0	0	0	0
1	1	0	1	0	1	0	0	1	0	0	0
1	22	0	0	4	2	0	0	0	0	0	0
1	2	0	0	0	2	0	0	0	0	1	0
1	154	0	0	0	0	0	2	0	0	1	1
1	13	0	0	0	0	0	0	0	1	2	0
1	22	0	0	0	0	0	0	0	1	0	3
2	1	0	0	2	2	0	0	0	0	0	0
5	18	0	0	0	2	0	0	0	0	0	1
5	99	0	0	0	1	0	0	0	1	0	0

Tabela 7.44: *Solução heurística residual Nova.*

Solução da Heurística Residual Nova - versão 2:

Id Barra	x	Padrão									
1	61	2	0	2	0	0	0	0	0	0	0
1	58	1	1	0	0	0	0	1	0	0	0
1	85	0	2	0	0	0	1	0	0	0	0
1	113	0	1	0	1	1	0	0	0	0	0
1	23	0	0	4	2	0	0	0	0	0	0
1	153	0	0	0	0	0	2	0	0	1	1
1	1	0	0	0	0	0	1	1	0	2	0
1	13	0	0	0	0	0	0	0	1	2	0
1	22	0	0	0	0	0	0	0	1	0	3
1	1	0	0	0	2	0	0	0	0	1	0
2	1	1	0	1	1	0	0	0	0	0	0
5	19	0	0	0	2	0	0	0	0	0	1
5	99	0	0	0	1	0	0	0	1	0	0

Tabela 7.45: *Solução heurística residual Nova - versão 2.*

	Número de Objetos	Número de Padrões	Perda Total	Tempo (s)
Relaxada	648,946	10	3160,34	0,11
FFD Pura	1197	12	21629	0
Gulosa Pura	876	15	12719	1,43
FFD Residual	650	12	3375	0,21
Gulosa Residual	650	12	3396	0,22
Nova	649	14	3293	0,28
Nova - versão 2	649	13	3196	0,27

Tabela 7.46: *Solução do Exemplo 5.*

Capítulo 8

O Problema de Redução de Padrões de Corte

Qualquer solução para o problema de corte de estoque consiste em um conjunto de padrões de corte e o número de vezes que cada um desses padrões deve ser utilizado (*freqüência*). Vários objetivos são sugeridos na literatura para esse problema. Em particular, nem sempre é apropriado considerar somente a minimização de perdas como único objetivo. Pileggi [20] estuda o problema com dois objetivos conflitantes a serem minimizados: perda versus número máximo de pilhas abertas. Cada tipo de item cortado abre uma pilha que será fechada somente quando o último item daquele tipo for cortado. Para este objetivo, a seqüência dos padrões de corte é relevante.

O problema de corte de estoque comumente apresenta diversas soluções ótimas alternativas, isto é, soluções com mesma perda, porém obtida a partir de padrões de corte alternativos. O número de padrões de corte pode afetar consideravelmente o tempo do processo de corte. Este é um caso onde o equipamento de corte tem de ser preparado a cada novo padrão de corte; um procedimento que consome um determinado tempo durante o qual o processo de corte tem de ser interrompido. Em outras palavras, o tempo de preparação de máquina (*setup*) não considerado na construção dos padrões de corte pode ser relevante na programação da produção. Isto sugere que um objetivo deveria ser considerado, de modo a reduzir o número de padrões de corte.

Agora, apresentamos uma heurística para resolver o problema de redução de padrões

em problemas de corte de estoque proposta por Limeira e Yanasse [12]. O método baseia-se na observação empírica de que para se ter uma redução no número de padrões, deve-se impor que dois (ou mais) itens precisam ter suas demandas completadas com o uso de um padrão de corte. Outro requisito do método é que a eficiência do padrão gerado seja maior que uma eficiência determinada na solução inicial inteira do problema.

8.1 Uma Heurística para Redução dos Padrões de Corte

A heurística proposta por *Limeira e Yanasse* [12], consiste em outro tópico do projeto que visa a redução do número de padrões de corte distintos. A heurística, basicamente, consiste em, a partir do problema original P , definir um problema P_1 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 pela divisão de $d_1, d_2, \dots, d_m$ por F_1 e tomando-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 haja excesso de produção de um item.

Para o problema P_1 , um padrão de corte é aceito se completar simultaneamente as demandas de dois ou mais itens e, também, se possuir uma perda inferior a um determinado α (nível de aspiração). Se estas duas condições forem satisfeitas, tal padrão de corte fará parte da solução global do problema P com frequência F_1 .

Se nenhum padrão for escolhido, nenhuma redução será realizada. Um novo fator F_2 é definido e o processo é repetido. O processo termina quando o fator $F_k = 1$, para algum k , for alcançado e nenhum outro padrão alcançar os níveis de aspiração desejados. Caso se tenha, no final, um problema residual, ou seja, demandas de alguns itens ainda não foram atendidas, então esse problema residual deve ser resolvido de maneira independente. No nosso caso, utilizamos as heurísticas descritas no Capítulo 5.

Os fatores $F_1 \geq F_2 \geq \dots \geq 1$ são definidos da seguinte maneira. Como os padrões desejados, quando cortados, precisam finalizar a demanda de pelo menos dois itens, as

quantidades de tais padrões precisam ser submúltiplas das demandas de alguns itens. Portanto, os candidatos a fatores são números submúltiplos das demandas de todos os itens. Esses valores são ordenados de forma decrescente, excluindo-se os valores iguais, e são testados um a um.

Um dos níveis de aspiração para os padrões é que o desperdício precisa estar em torno de um determinado α , já que o objetivo é gerar uma solução com desperdício global o mais próximo possível de α . O valor de α é obtido no cálculo do desperdício gerado pela solução inteira do problema de corte, onde a questão do número de diferentes padrões de corte não foi considerada.

8.1.1 Exemplo

Consideremos o seguinte problema de corte de estoque unidimensional. Temos em estoque uma quantidade ilimitada de barras de comprimento $L = 1000$ e desejamos atender a uma demanda de 10 itens, ou seja, $m = 10$. O comprimento de cada item $i, i = 1, \dots, 10$ e a sua respectiva demanda estão na Tabela 8.1.

Item	Comprimento	Demanda
1	173	8
2	194	10
3	97	5
4	118	12
5	22	8
6	163	10
7	184	12
8	67	9
9	88	3
10	133	15

Tabela 8.1: *Dados do exemplo numérico*

Primeiramente, resolvemos o problema pelo método simplex com geração de colunas e obtivemos a solução fracionária:

$\tilde{x}$	Padrão de Corte									
0,817	0	2	0	0	2	0	2	1	0	1
1,38	3	0	0	1	0	1	0	1	0	1
0,255	2	2	0	0	0	0	0	0	0	2
0,185	0	1	0	4	0	0	0	3	0	1
1,459	0	2	0	5	1	0	0	0	0	0
0,32	0	0	0	0	1	6	0	0	0	0
1,083	0	0	3	1	1	0	2	3	0	0
3,35	1	0	0	0	0	2	2	0	0	1
1,5	0	2	0	1	0	0	1	2	2	0
1,752	0	1	1	0	2	0	0	1	0	5

Total de objetos cortados = 12,101

Número de padrões de corte = 10

A seguir, aplicamos uma heurística para obtenção de uma solução inicial inteira. Neste caso, foi utilizada a heurística residual Nova, seção 5.4, mas poderia ter sido usada qualquer outra descrita no Capítulo 5.

x	Padrão de Corte									
1	3	0	0	1	0	1	0	1	0	1
1	2	0	0	0	0	0	3	0	0	0
3	1	0	0	0	0	2	2	0	0	1
1	0	5	0	0	0	0	0	0	0	0
1	0	2	0	5	1	0	0	0	0	0
1	0	2	0	1	0	0	1	2	2	0
1	0	1	1	0	2	0	0	0	0	5
1	0	0	3	1	1	0	2	3	0	0
1	0	0	1	3	4	1	0	3	1	0
1	0	0	0	1	0	0	0	0	0	1
1	0	0	0	0	0	2	0	0	0	5

Total de objetos cortados = 13

Número de padrões de corte = 11

Partindo desta solução inteira, aplicamos a Heurística de Redução de Padrões de Corte descrita em 8.1.

A porcentagem de desperdício nesta solução é de:

$$\alpha = \left(\frac{\text{perda total}}{L \times \text{total objetos cortados}} \right) = \left(\frac{899}{13000} \right) = 0,0691538$$

O nível de aspiração para a eficiência (média) dos padrões é:

$$\text{ef} = \left\lceil (1 - \alpha) \times L \right\rceil = \left\lceil (1 - 0,0691538) \times 1000 \right\rceil = 931 \text{ unidades}$$

1ª iteração:

Os candidatos a fatores são: $F_1 = \{15, 12, 10, 9, 8, 6, 5, 4, 3, 2, 1\}$. Consideremos $F_1 = 15$. Com este fator, teremos:

Item	Comprimento	Demanda
10	133	$\lfloor 15/15 \rfloor = 1$

Este fator completa a demanda de apenas um item, o item 10. Como a heurística de redução de padrões de corte exige que um fator deve completar a demanda de pelo menos dois itens, este fator é descartado e partimos para o próximo, que é $F_1 = 12$. Para este fator, o problema P_1 será composto pelos itens:

Item	Comprimento	Demanda
4	118	$\lfloor 12/12 \rfloor = 1$
7	184	$\lfloor 12/12 \rfloor = 1$
10	133	$\lfloor 15/12 \rfloor = 1$

Para este fator ser aceito, ele deve satisfazer duas condições:

1. A demanda de pelo menos dois itens deve ser completada e, isso acontece, pois este fator completa os itens 4 e 7;
2. A eficiência para os possíveis padrões de corte devem ser superior a 931, o que não acontece (eficiência = 435).

Portanto, o fator $F_1 = 12$ é descartado, ou seja, nenhum padrão é gerado para esta frequência. O próximo fator a ser considerado é $F_1 = 10$. Para este fator, teremos o seguinte problema P_1 :

Item	Comprimento	Demanda
2	194	$\lfloor 10/10 \rfloor = 1$
4	118	$\lfloor 12/10 \rfloor = 1$
6	163	$\lfloor 10/10 \rfloor = 1$
7	184	$\lfloor 12/10 \rfloor = 1$
10	133	$\lfloor 15/10 \rfloor = 1$

Neste caso, a primeira condição é satisfeita, pois este fator completa os itens 2 e 6. Mas, a segunda condição não é satisfeita pois a eficiência (792) é menor que a eficiência média 931. Portanto, descartamos o fator $F_1 = 10$. Nenhum padrão é gerado para esta frequência.

Consideremos o próximo fator $F_1 = 9$. Para este fator, teremos o problema P_1 com os seguintes itens:

Item	Comprimento	Demanda
2	194	$\lfloor 10/9 \rfloor = 1$
4	118	$\lfloor 12/9 \rfloor = 1$
6	163	$\lfloor 10/9 \rfloor = 1$
7	184	$\lfloor 12/9 \rfloor = 1$
8	67	$\lfloor 9/9 \rfloor = 1$
10	133	$\lfloor 15/9 \rfloor = 1$

A primeira condição não é satisfeita, pois este fator completa a demanda de apenas um item. Com isso, nenhum padrão é gerado para a frequência $F_1 = 9$.

Partimos para $F_1 = 8$, cujo problema P_1 será composto pelos seguintes itens:

Item	Comprimento	Demanda
1	173	$\lfloor 8/8 \rfloor = 1$
2	194	$\lfloor 10/8 \rfloor = 1$
4	118	$\lfloor 12/8 \rfloor = 1$
5	22	$\lfloor 8/8 \rfloor = 1$
6	163	$\lfloor 10/8 \rfloor = 1$
7	184	$\lfloor 12/8 \rfloor = 1$
8	67	$\lfloor 9/8 \rfloor = 1$
10	133	$\lfloor 15/8 \rfloor = 1$

A demanda de dois itens é completada (itens 1 e 5), portanto, satisfaz a primeira condição. A eficiência é maior que 945. Portanto, o fator $F_1 = 8$ é aceito. O padrão gerado, resolvendo-se o problema da mochila restrito, é $(1 \ 1 \ 0 \ 1 \ 1 \ 1 \ 1 \ 0 \ 0 \ 1)$. Este padrão é cortado $F_1 = 8$ vezes.

A demanda do problema original P é atualizada e temos o seguinte problema a ser resolvido:

Item	Comprimento	Demanda
1	173	0
2	194	2
3	97	5
4	118	4
5	22	0
6	163	2
7	184	4
8	67	9
9	88	3
10	133	7

2a. iteração:

Calcula-se novamente os candidatos a fatores, baseado na demanda:

$$F_2 = \{9, 7, 5, 4, 3, 2, 1\}$$

e repetimos o procedimento até que o fator $F_k = 1$ seja alcançado e não seja possível gerar qualquer outro padrão de corte com os níveis de aspiração desejados.

Se, ao final da execução do procedimento heurístico, ainda houver itens com demandas não atendidas então resolve-se este problema residual por algum método de arredondamento, alguns estão descritos no Capítulo 5.

O resultado final obtido após a execução da heurística e do problema residual é o seguinte:

x	Padrão de Corte									
8	1	1	0	1	1	1	1	0	0	1
3	0	0	1	1	0	0	1	3	1	2
1	0	2	2	1	0	1	0	0	0	1
1	0	0	0	0	0	1	1	0	0	0

Total de objetos cortados = 13

Número de padrões de corte = 4

A porcentagem de desperdício nesta solução é de:

$$\alpha = \left(\frac{899}{13000} \right) = 0,0691538.$$

Conclusões deste exemplo:

- Podemos concluir que o número de padrões de corte foi reduzido de 11 para 4 com a Heurística de Redução de Padrões de Corte, seção 8.1, mantendo o mesmo número de objetos cortados: 13;
- Este exemplo, também foi resolvido partindo da solução inicial inteira gerada pelas heurísticas FFD e Gulosa, que são diferentes da solução inteira apresentada aqui, usando a heurística Nova. Mas, o resultado obtido pela Heurística de Redução de Padrões de Corte foi o mesmo.

Capítulo 9

Resultados Computacionais - Redução de Padrões de Corte

Os testes computacionais aqui apresentados referem-se ao seguinte problema:

$$\begin{aligned} \text{minimizar} \quad & f(\mathbf{x}) = \sum_{j=1}^n x_j \\ \text{sujeito a:} \quad & \mathbf{Ax} = \mathbf{d}, \quad \mathbf{x} \geq \mathbf{0} \quad \text{e inteiro} \end{aligned} \tag{9.1}$$

onde cada coluna da matriz $\mathbf{A} \in \Re^{m \times n}$ é um vetor associado a um padrão de corte. Este modelo trata o problema de corte unidimensional com apenas um tipo de barra em estoque, em quantidade ilimitadas. A função objetivo é minimizar o total de peças a serem cortadas. Resolvemos este problema através da heurística de redução de padrões de corte descrita na seção anterior.

Realizamos alguns testes computacionais para comparar os resultados obtidos por esta heurística em conjunto com as três heurísticas de arredondamento, descritas no Capítulo 5. Para isto, geramos, aleatoriamente, 18 classes com 30 exemplos em cada classe, totalizando 540 exemplos. Relembramos que as implementações foram desenvolvidas em Delphi 5 e todos os testes foram realizados em um computador Pentium III (866MHz) com 256 MB de memória RAM. A seguir, mostraremos em detalhes os critérios usados para gerar os dados e os resultados obtidos.

9.1 O Gerador Aleatório

Definimos o seguinte gerador aleatório, baseada em Foerster e Wäscher [2]:

- *Dimensão da barra em estoque:* $L = 1000$.
- *Número de tipos de itens a serem produzidos:* Foram utilizados diferentes valores de $m = 10, 20$ e 40 .
- *Dimensão dos itens a serem produzidos:* os comprimentos dos itens l_i foram gerados aleatoriamente nos intervalos $[v_1 L, v_2 L]$, onde v_1 e v_2 definem o menor e o maior valor que $l_i, i = 1, \dots, m$ pode ter. Utilizou-se: $v_1 = 0,01$ e $0,2$ e $v_2 = 0,2$ e $0,8$. Combinando estes valores, gerou-se classes com problemas *pequenos* ($v_1 = 0,01$ e $v_2 = 0,2$), *médios* ($v_1 = 0,01$ e $v_2 = 0,8$) e *grandes* ($v_1 = 0,2$ e $v_2 = 0,8$).
- *Demanda:* Foram utilizados dois geradores diferentes para os valores da demanda dos itens; o primeiro com valores próximos a 10, ($d_i \in [7, 12], i = 1, \dots, m$) e o segundo com valores próximos a 100, ($d_i \in [70, 120], i = 1, \dots, m$).

Foram criadas 18 classes de problemas combinando (L, m, d) e gerados 30 exemplos para cada classe.

9.2 Resultados

A seguir, apresentamos uma tabela com o número médio (dos 30 exemplos) de objetos cortados na solução original, ou seja, sem aplicar a heurística de redução de padrões de corte e na solução reduzida obtida pela heurística de redução de padrões de corte.

				Solução Original			Solução Reduzida		
	m	l_i	d	FFD	Gulosa	$Nova$	FFD	Gulosa	$Nova$
C1	10	P	10	<i>10,97</i>	10,93	10,93	<i>10,97</i>	10,93	10,93
C2	10	P	100	106,97	106,97	106,97	<i>107,37</i>	<i>107,37</i>	107,20
C3	20	P	10	21,03	21,00	21,00	<i>21,07</i>	21,03	21,00
C4	20	P	100	207,00	206,97	206,97	<i>207,30</i>	207,27	207,03
C5	40	P	10	41,70	41,67	<i>41,77</i>	41,70	41,70	41,70
C6	40	P	100	423,97	423,97	424,00	<i>424,70</i>	424,50	424,03
C7	10	M	10	44,87	45,00	45,13	45,83	45,83	<i>46,07</i>
C8	10	M	100	452,87	452,87	453,53	480,93	<i>481,73</i>	480,53
C9	20	M	10	83,07	83,37	83,33	89,53	89,57	<i>89,77</i>
C10	20	M	100	842,33	842,67	<i>843,00</i>	927,93	927,73	925,93
C11	40	M	10	161,77	162,43	161,87	178,67	178,30	<i>179,13</i>
C12	40	M	100	1649,77	1650,73	1650,10	<i>1846,50</i>	1839,87	1843,73
C13	10	G	10	55,20	55,20	55,30	56,47	<i>56,57</i>	56,53
C14	10	G	100	581,87	581,87	582,13	608,97	<i>609,00</i>	<i>609,00</i>
C15	20	G	10	106,60	106,60	106,70	109,87	109,90	<i>109,97</i>
C16	20	G	100	1097,43	1097,47	1097,83	1150,07	<i>1150,87</i>	1150,27
C17	40	G	10	206,47	206,53	206,53	221,30	221,37	<i>221,57</i>
C18	40	G	100	2097,27	2097,30	2097,47	<i>2228,53</i>	<i>2228,53</i>	2226,97

Tabela 9.1: Número médio de objetos cortados.

A seguir, apresentamos uma tabela com o número médio de padrões de corte na solução original, ou seja, sem aplicar a heurística de redução de padrões de corte e na solução reduzida obtida pela heurística de redução de padrões de corte.

				Solução Original			Solução Reduzida		
	m	l_i	d	FFD	Gulosa	$Nova$	FFD	Gulosa	$Nova$
C1	10	P	10	9,33	9,37	8,30	3,57	3,60	3,60
C2	10	P	100	14,07	14,03	12,90	10,80	10,83	10,13
C3	20	P	10	18,50	18,47	16,90	5,17	5,67	5,53
C4	20	P	100	27,43	27,40	25,30	15,20	15,57	13,90
C5	40	P	10	37,23	37,20	34,80	7,53	8,63	7,50
C6	40	P	100	54,80	54,77	51,50	40,50	38,23	30,00
C7	10	M	10	10,43	10,63	9,60	8,10	8,10	7,90
C8	10	M	100	11,93	11,90	11,93	12,43	12,50	12,07
C9	20	M	10	20,70	21,70	18,40	15,37	15,23	14,93
C10	20	M	100	24,27	25,37	22,77	21,80	21,93	21,87
C11	40	M	10	41,40	43,80	35,67	28,50	28,30	27,87
C12	40	M	100	48,20	51,10	43,77	39,13	39,30	39,17
C13	10	G	10	9,80	9,80	9,60	9,40	9,43	9,37
C14	10	G	100	10,77	10,77	11,27	11,43	11,43	11,43
C15	20	G	10	19,63	19,63	19,17	16,33	16,33	16,20
C16	20	G	100	21,83	22,00	22,30	22,60	22,70	22,53
C17	40	G	10	39,83	39,97	36,23	32,20	32,20	32,07
C18	40	G	100	44,37	44,50	42,73	42,13	42,13	42,13

Tabela 9.2: Número médio de padrões de corte.

9.3 Análise dos Resultados

Figura 9.1: Número médio de padrões de corte obtidos inicialmente, sem a aplicação da Heurística de Redução.

Inicialmente, a Heurística *Nova* encontra soluções para o problema de corte de estoque unidimensional com menos padrões de corte que as Heurísticas Residuais *FFD* e *Gulosa*, veja Figura 9.1. Com a aplicação da Heurística de Redução de Padrões de Corte, as soluções encontradas, na maioria das classes, apresentaram-se muito próximas para as três heurísticas de arredondamento, o que sugere que a Heurística de redução de padrões de corte é pouco sensível à solução inteira inicial, isto pode ser visto na Figura 9.2.

Fazemos agora uma análise da redução obtida pela heurística, analisando o gráfico comparativo do número de padrões de corte antes e depois da aplicação da heurística de redução. Os dados mostrados no gráfico da Figura 9.3 se referem às soluções obtidas pela heurística *Nova*.

Podemos notar, pelo gráfico da Figura 9.3, que a grande redução do número de padrões de corte ocorre apenas para as primeiras classes, que são aquelas que contêm itens pequenos, mais fáceis de serem combinados. Para as demais classes notamos uma pequena redução no número de padrões de corte.

Figura 9.2: Número médio de padrões de corte obtidos pela Heurística de Redução, usando as três heurísticas para obter a solução inteira.

Figura 9.3: Número médio de padrões de corte antes (com a heurística Nova) e depois da aplicação da Heurística de Redução.

Capítulo 10

Alguns Exemplos Resolvidos

10.1 Exemplo 1

Exemplo da Classe 7, com número de itens: $m = 10$ e o comprimento da barra disponível em estoque: $L = 1000$.

Item	Comprimento	Demanda
1	650	12
2	219	8
3	102	12
4	671	10
5	554	9
6	123	8
7	575	11
8	458	9
9	27	9
10	478	9

Tabela 10.1: *Dados do Exemplo 1.*

Solução do Método Simplex com Geração de Colunas:

$\tilde{x}$	Padrão										
12	1	0	0	0	0	0	0	0	0	0	0
3,604	0	2	0	0	1	0	0	0	0	0	0
4	0	0	3	0	1	1	0	0	0	0	0
10	0	0	0	1	0	0	0	0	0	0	0
0,604	0	0	0	0	1	3	0	0	2	0	0
0,729	0	0	0	0	0	3	1	0	2	0	0
10,27	0	0	0	0	0	0	1	0	0	0	0
4,5	0	0	0	0	0	0	0	2	0	0	0
0,791	0	1	0	0	1	0	0	0	8	0	0
4,5	0	0	0	0	0	0	0	0	0	0	2

Tabela 10.2: *Solução relaxada.*

Solução da Heurística Residual Nova:

x	Padrão										
12	1	0	0	0	0	0	0	0	0	0	0
4	0	2	0	0	1	0	0	0	0	0	0
4	0	0	3	0	1	1	0	0	0	0	0
10	0	0	0	1	0	0	0	0	0	0	0
1	0	0	0	0	1	3	0	0	2	0	0
11	0	0	0	0	0	0	1	0	0	0	0
4	0	0	0	0	0	0	0	2	0	0	0
1	0	0	0	0	0	0	0	1	2	1	0
4	0	0	0	0	0	0	0	0	0	2	0
1	0	0	0	0	0	1	0	0	5	0	0

Tabela 10.3: *Solução da heurística residual Nova.*

Solução da Heurística de Redução de Padrões de Corte com a heurística Residual Nova:

x	Padrão									
12	1	0	1	0	0	0	0	0	0	0
8	0	1	0	0	0	1	1	0	0	0
10	0	0	0	1	0	0	0	0	0	0
9	0	0	0	0	1	0	0	0	0	0
3	0	0	0	0	0	0	1	0	0	0
9	0	0	0	0	0	0	0	1	1	1

Tabela 10.4: *Solução da heurística de redução de padrões de corte com a heurística residual Nova.*

	Número de Objetos	Número de Padrões	Tempo (s)
Relaxada	50,998	10	0
Nova	52	10	0,02
Nova Reduzida	51	6	0,05

Tabela 10.5: *Solução do Exemplo 1.*

10.2 Exemplo 2

Exemplo da Classe 14, com número de itens: $m = 10$ e comprimento da barra disponível em estoque: $L = 1000$.

Item	Comprimento	Demanda
1	658	117
2	227	90
3	679	117
4	562	94
5	583	95
6	466	106
7	486	101
8	369	84
9	390	119
10	273	101

Tabela 10.6: *Dados do Exemplo 2.*

Solução do Método Simplex com Geração de Colunas:

$\tilde{x}$	Padrão									
16	1	1	0	0	0	0	0	0	0	0
74	0	1	1	0	0	0	0	0	0	0
43	0	0	1	0	0	0	0	0	0	0
94	0	0	0	1	0	0	0	0	1	0
11	0	0	0	0	1	0	0	0	1	0
53	0	0	0	0	0	2	0	0	0	0
50,5	0	0	0	0	0	0	2	0	0	0
84	0	0	0	0	1	0	0	1	0	0
7	0	0	0	0	0	0	0	0	2	0
101	1	0	0	0	0	0	0	0	0	1

Tabela 10.7: *Solução relaxada.*

Solução da Heurística Residual Nova:

x	Padrão									
15	1	1	0	0	0	0	0	0	0	0
101	1	0	0	0	0	0	0	0	0	1
75	0	1	1	0	0	0	0	0	0	0
42	0	0	1	0	0	0	0	0	0	0
94	0	0	0	1	0	0	0	0	1	0
84	0	0	0	0	1	0	0	1	0	0
11	0	0	0	0	1	0	0	0	1	0
53	0	0	0	0	0	2	0	0	0	0
50	0	0	0	0	0	0	2	0	0	0
7	0	0	0	0	0	0	0	0	2	0
1	1	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0

Tabela 10.8: *Solução da heurística residual Nova.*

Solução da Heurística de Redução de Padrões de Corte com a heurística Residual Nova:

x	Padrão									
117	1	0	0	0	0	0	0	0	0	0
3	0	2	0	0	0	1	0	0	0	0
84	0	1	0	0	0	0	0	1	1	0
35	0	0	0	0	1	0	0	0	1	0
101	0	0	1	0	0	0	0	0	0	1
101	0	0	0	0	0	1	1	0	0	0
94	0	0	0	1	0	0	0	0	0	0
16	0	0	1	0	0	0	0	0	0	0
60	0	0	0	0	1	0	0	0	0	0
1	0	0	0	0	0	2	0	0	0	0

Tabela 10.9: *Solução da heurística de redução de padrões de corte com a heurística residual Nova.*

	Número de Objetos	Número de Padrões	Tempo (s)
Relaxada	533,5	10	0
Nova	534	12	0
Nova Reduzida	612	10	0,11

Tabela 10.10: *Solução do Exemplo 2.*

Capítulo 11

Conclusões e Propostas

11.1 Conclusões

- A abordagem por otimização linear com a aplicação de heurísticas para a resolução de problemas de corte de estoque inteiro mostrou-se muito eficiente. As heurísticas residuais apresentaram excelentes resultados até mesmo para problemas com baixa demanda. O folclore da área de problemas de corte e empacotamento de que a abordagem por otimização linear (geração de colunas) é inadequado para problemas com baixa demanda é infundado. Não foi explorada a classe de problemas *bin-packing* onde a demanda é 1 para todos os itens. Estes experimentos foram com demanda baixa, variando de um a dez;
- a única heurística de redução de padrões de corte analisada mostrou-se bastante eficiente apenas para uma certa classe de problemas, ou seja, problemas cujos itens são de comprimentos pequenos, fáceis de combinar. Provavelmente, a redução do número de padrões de corte fica prejudicada para as demais classes pois os itens de comprimentos médio e grande exigem que mais padrões sejam utilizados, não sendo, então, possível resolver o problema com um número menor de padrões de corte.

11.2 Extensões a serem investigadas

- Estender a heurística de redução de padrões de corte para problemas com limitação de estoque;

- investigar os trabalhos da literatura que tratam o problema de redução de padrões de corte e comparar com os resultados aqui apresentados;
- estender as heurísticas de arredondamento para o problema de corte de estoque bidimensional.

Bibliografia

- [1] ARENALES, M. N., MORABITO, R., (editores), “O Problema de Corte e Empacotamento e Aplicações Industriais ”, Livro-texto de Mini curso, XX CNMAC - Congresso Nacional de Matemática Aplicada e Computacional, Gramado - RS, (1997).
- [2] FOERSTER, H., WÄSCHER, G., “*Pattern reduction in one-dimensional cutting-stock problems*”. International Journal of Production Research, 38: 1657-1676 (2000).
- [3] GAU, T., WÄSCHER, “*CUTGEN: A problem generator for the standard one-dimensional cutting stock problem*”. European Journal of Operational Research, 84: 572-579 (1995).
- [4] GILMORE, P. C., GOMORY, R. E., “*A linear programming approach to the cutting stock problem*”. Operations Research, 9: 848-859 (1961).
- [5] GILMORE, P. C., GOMORY, R. E., “*A linear programming approach to the cutting stock problem - Part II*”. Operations Research, 11: 863-888 (1963).
- [6] GILMORE, P. C., GOMORY, R. E., “*Multi-stage cutting stock problems of two and more dimensions*”. Operations Research, 13: 94-120 (1965).
- [7] HAESSLER, R. W., “*Controlling cutting pattern changes in one-dimensional trim loss problems*”. Operations Research, 23: 483-493 (1975).
- [8] HAESSLER, R. W., “*A note on computational modifications to the Gilmore-Gomory cutting stock algorithm*”. Operations Research, 28: 1001-1005 (1980).
- [9] HINXMAN, A., “*The trim-loss and assortment problems: a survey*”. European Journal of Operational Research, 5: 8-18 (1980).

- [10] HOTO, R. S. V., “*O problema da mochila compartimentada aplicado no corte de bobinas de aço*”. Tese de Doutorado, COPPE - UFRJ, (2001).
- [11] LEFRANÇOIS, P., GASCON, A., “*Solving a one-dimensional cutting-stock problem in a small manufacturing firm: a case study*”. IIE Transactions, 27: 483-496 (1995).
- [12] LIMEIRA, M. S., YANASSE, H. H., “*Uma heurística para o problema de redução de padrões de corte*”. Anais da V Oficina Nacional de Problemas de Corte e Empacotamento, São José dos Campos, SP, pp. 137-145, (2001).
- [13] MARQUES, F. P., “*O problema da mochila compartimentada*”. Dissertação de Mestrado, ICMC - USP, (2000).
- [14] MORABITO, R., “*Corte De Estoque Bidimensional*”. Dissertação de Mestrado, ICMC - USP, (1989).
- [15] MORABITO, R., “*Uma abordagem em grafo-e-ou para o problema de empacotamento: Aplicação ao carregamento de paletes e contêineres*”. Tese de Doutorado, EESC - USP, (1992).
- [16] MORABITO, R., ARENALES, M. N., ARCARO, V. F., “*An AND/OR-graph approach for two-dimensional cutting problems*”. European Journal of Operational Research, 58: 263-271 (1992).
- [17] MORABITO, R., ARENALES, M. N., “*Um exame dos problemas de corte e empacotamento*”. Pesquisa Operacional, V. 12, n° 1, pp. 1-20 (1992).
- [18] MORABITO, R., ARENALES, M. N., ARCARO, V. F., “*Performance of two heuristics for solving large scale two-dimensional guillotine cutting problems*”. INFOR - Canadian Journal of Operational Research and Information Processing, 33: 145-155 (1995).
- [19] MORABITO, R., ARENALES, M. N., ARCARO, V. F., “*Staged and constrained two-dimensional guillotine cutting problems - An AND/OR-graph approach*”. European Journal of Operational Research, 94: 548-560 (1996).

- [20] PILEGGI, G. C. F., “*Abordagens para otimização integrada dos problemas de geração e sequenciamento de padrões de corte*”. Tese de Doutorado, ICMC - USP, (2002).
- [21] PINTO, M. J., “*O problema de corte de estoque inteiro*”. Dissertação de Mestrado, ICMC - USP, (1999).
- [22] POLDI, K. C., “*Heurísticas para o problema de corte de estoque unidimensional inteiro*”. Anais do XXXIV SBPO - Simpósio Brasileiro de Pesquisa Operacional, Rio de Janeiro - RJ, (2002).
- [23] POLDI, K. C., ARENALES, M. N., “*Sobre uma heurística de redução de padrões de corte para o problema de corte de estoque*”. Artigo submetido à revista Tema e apresentado no XXV CNMAC - Congresso Nacional de Matemática Aplicada e Computacional, Nova Friburgo - RJ, (2002).
- [24] SCHEITHAUER, G., TERNO, J., “*The modified Integer Round-up Property of the One-dimensional Cutting Stock Problem*”. European Journal of Operational Research, 84: 562-571 (1995).
- [25] STADTLER, H., “*A one-dimensional cutting stock problem in the Aluminium Industry and its solution*”. European Journal of Operational Research, 44: 209-223 (1990).
- [26] VAHRENKAMP, R., “*Random search in the one-dimensional cutting stock problem*”. European Journal of Operational Research, 95: 191-200 (1996).
- [27] VIANNA, A. C. G., “*Problemas de corte e empacotamento: uma abordagem em grafo E/OU*”. Tese de Doutorado, ICMC - USP, (2000).
- [28] WÄSCHER, G., GAU, T., “*Heuristics for the integer one-dimensional cutting stock problem: a computational study*”. OR Spektrum, 18: 131-144 (1996).
- [29] YANASSE, H. H., LIMEIRA, M. S., “*Pattern reduction in cutting problems*”. trabalho em desenvolvimento, apresentado em INFORMS Conference, San Antonio, (2000).