SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito: 24.04.2000

Assinatura: _____

O problema da mochila compartimentada^{*}

Fabiano do Prado Marques

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
Abril/2000

^{*} Projeto financiado pela Fapesp.

Sumário

Motivação	1
Organização do Texto.....	4
Capítulo 1. Introdução aos Problemas de Cortes	6
1.1. Considerações Iniciais.....	6
1.2. Classificação dos Problemas de Cortes.....	8
1.2.1. Dimensionalidade.....	9
1.2.1.1 Problema Unidimensional.....	9
1.2.1.2 Problema Bidimensional	10
1.2.1.3 Problema Tridimensional.....	10
1.2.1.4 Problemas 1.5-dimensional e 2.5-dimensional	11
1.2.1.5 Problema Multidimensional.....	11
1.3. Restrições sob os Planos de Corte	12
1.4. Objetivos	12
Capítulo 2. Problemas da Mochila.....	13
2.1. Considerações Iniciais.....	13
2.2. Problema da Mochila Inteiro.....	14
2.3. Problema da Mochila 0-1	14
2.4. Problema da Mochila Restrito.....	15

2.5. Problema da Soma de Subconjuntos	16
2.6. Problema do Troco	16
2.7. Problema da Mochila Quadrático.....	17
2.8. Problema de Múltiplas Mochilas 0-1	18
2.9. Problema de Designação Generalizada.....	19
2.10. Problema <i>Bin-Packing</i>	19

Capítulo 3. O Problema da Mochila Compartimentada 21

3.1. Definição do Problema.....	21
3.2. Agrupamento dos Itens	22
3.3. O Problema de Corte em Bobinas de Aço	25
3.3.1. Considerações Iniciais.....	25
3.3.2. O Processo de Corte	26
3.3.3. A Cortadeira	28
3.3.4. Laminação, Encruador, Recortadeira e Recozimento	29
3.3.5. Custos dos Processos Envolvidos no Problema.....	31
3.4. Abordagens para o Problema de Corte em Bobinas de Aço	32
3.4.1. Uma Abordagem por Programação Linear Inteira [Pereira,1993].....	32
3.4.2. Um Problema de Corte em Duas Fases [Carvalho, 1991]	32
3.4.3. Um Problema de Corte Unidimensional Sujeito a Restrições de Agrupamento [Hoto, 1996]	33

Capítulo 4. Modelagem do Problema da Mochila Compartimentada 36

4.1. Uma Modelagem Matemática	36
4.2. Limitantes para o Problema da Mochila Compartimentada.....	38
4.2.1. Modelo Agregado.....	38
4.2.2. Relaxação de Integralidade	42

Capítulo 5. Métodos de Resolução e Implementações 45

5.1. Heurística de Decomposição.....	45
5.2. Heurística do Melhor Compartimento	47

5.3. Heurística dos "z" Melhores Compartimentos	49
5.4. Programas Desenvolvidos	51
5.4.1. Objetivos	51
5.4.2. Estruturação do Programa	51
5.4.3. Estruturas de Armazenamento dos Dados dos Programas	53
5.4.4. Fluxo das Informações nos Programas	55
5.4.5. Atualização dos Dados	58
5.4.5.1. Atualização das Demandas	58
5.4.6. Cálculo da Função Objetivo e Aproveitamento da Mochila	58
Capítulo 6. Resultados Computacionais	60
6.1. Considerações Iniciais	60
6.2. Dados da Mochila	61
6.3. Tabelas de Resultados	61
6.4. Análise dos Planos de Corte	96
Capítulo 7. Conclusões e Propostas Futuras.....	100
Referências Bibliográficas	102

Lista de Figuras

Figura 1: Plano de corte unidimensional gerado em um objeto.	7
Figura 2: Plano de corte bidimensional em uma placa retangular.	7
Figura 3: Problema tridimensional.	11
Figura 4: Representação alternativa para o agrupamento dos itens em uma Mochila Compartimentada.	24
Figura 5: O processo de corte em bobinas de aço.	27
Figura 6: Corte em bobinas de aço com refugos embutidos.	27
Figura 7: Perfil do processo de cortagem em bobinas de aço.	29
Figura 8: Perfil do processo de laminação.	30
Figura 9: Fluxo da produção de tubos de aço.	31
Figura 10: Ilustração da heurística de Hoto.	34
Figura 11: Relacionamento entre as <i>units</i> e os programas principais.	53
Figura 12: <i>DFD</i> de <i>nível 1</i> para os programas desenvolvidos.	56
Figura 13: <i>DFD</i> para a Heurística de Decomposição.	56
Figura 14: <i>DFD</i> para a Heurística do Melhor Compartimento.	57
Figura 15: <i>DFD</i> para a Heurística dos "z" Melhores Compartimentos.	57

Lista de Tabelas

Tabela 1: Classificação geométrica dos problemas de cortes.	12
Tabela 2: Exemplo de agrupamento de itens para uma mochila compartimentada.	23
Tabela 3: Resultados computacionais obtidos a partir da Heurística de Decomposição para o exemplo 1.	63
Tabela 4: Resultados computacionais obtidos a partir da Heurística do Melhor Compartimento para o exemplo 1.	65
Tabela 5: Resultados computacionais obtidos a partir da Heurística dos "z" Melhores Compartimentos ($z = 2$) para o exemplo 1.	67
Tabela 6: Resultados computacionais obtidos a partir da Heurística dos "z" Melhores Compartimentos ($z = 3$) para o exemplo 1.	69
Tabela 7: Resultados computacionais obtidos a partir da Heurística de Decomposição para o exemplo 2.	72
Tabela 8: Resultados computacionais obtidos a partir da Heurística do Melhor Compartimento para o exemplo 2.	75
Tabela 9: Resultados computacionais obtidos a partir da Heurística dos "z" Melhores Compartimentos ($z = 2$) para o exemplo 2.	78
Tabela 10: Resultados computacionais obtidos a partir da Heurística dos "z" Melhores Compartimentos ($z = 3$) para o exemplo 2.	81
Tabela 11: Resultados computacionais obtidos a partir da Heurística de Decomposição para o exemplo 3.	85
Tabela 12: Resultados computacionais obtidos a partir da Heurística do Melhor Compartimento para o exemplo 3.	88

Tabela 13: Resultados computacionais obtidos a partir da Heurística dos "z" Melhores	
Compartimentos ($z = 2$) para o exemplo 3.	91
Tabela 14: Resultados computacionais obtidos a partir da Heurística dos "z" Melhores	
Compartimentos ($z = 3$) para o exemplo 3.	94

Lista de Símbolos

Símbolos	Descrição
Σ	- Somatório
$\exists$	- Quantificador existencial
$\in$	- Relação de pertinência
$\subset$	- Relação de inclusão
$\cup$	- Reunião de Conjuntos
$\cap$	- Intersecção de conjuntos
$\neq$	- Relação de diferença
$=$	- Relação de igualdade
$\geq$	- Maior ou igual a ...
$\leq$	- Menor ou igual a ...
$>$	- Maior que ...
$<$	- Menor que ...
$\leftarrow$	- Atribuição
$[]$	- Delimitadores de intervalo
$\{ \}$	- Delimitador de conjunto
ϕ	- Conjunto vazio
$+, -, .$	- Operações aritméticas habituais
$\%$	- Porcentagem

Resumo

Nesse trabalho, estudamos um problema de otimização combinatorial conhecido por *Problema da Mochila Compartimentada*, que é uma extensão do clássico *Problema da Mochila*. O problema consiste em determinar as capacidades adequadas de vários compartimentos que podem vir a ser alocados em uma mochila e como esses compartimentos devem ser carregados, respeitando as restrições de capacidades dos compartimentos e da mochila. Busca-se maximizar o valor de utilidade total. O problema é muito pouco estudado na literatura, apesar de surgir naturalmente em aplicações práticas. Nesse estudo, propomos uma modelagem matemática não linear para o problema e verificamos algumas heurísticas para sua resolução.

Abstract

In this work, we studied a combinatorial optimization problem called the Clustered Knapsack Problem, that is an extension of the standard Knapsack Problem. The problem is to determine the right capacities of several clusters which can be allocated in a knapsack and how these clusters should be placed so as to respect the constraints on the capacities of the clusters and the knapsack. The objective is to maximize a total utility value. The problem has seldom been studied in the literature, even though it appears naturally in practical applications. In this study, we propose a non-linear model for the problem and we insert some heuristics for its resolution.

Motivação

A otimização vem ganhando muito respaldo nas últimas décadas, sendo alvo de muitas pesquisas devido à sua importância operacional e, sobretudo, econômica.

Problemas de otimização se caracterizam por dois aspectos: o possível e o melhor. O possível indica a existência de um conjunto de restrições a serem consideradas, revelando assim que nem toda solução é aceitável, sugerindo um conjunto de *soluções factíveis*, isto é, passíveis de execução. O melhor indica a existência de um critério para qualificar as várias soluções factíveis, permitindo compará-las, revelando a melhor dentre todas, chamada *solução ótima*.

Dentre os problemas de otimização vale ressaltar o problema de corte que consiste em cortar uma peça grande em um conjunto de itens de tamanhos menores, sendo conhecido também na literatura como *Problema da Mochila* (principalmente o problema unidimensional, onde apenas uma dimensão no processo de cortagem é relevante) por uma alusão ao problema de um alpinista que deve carregar sua mochila, selecionando dentre vários itens disponíveis, aqueles que lhe oferecem um máximo conforto. Para isto, o alpinista deve obedecer a um conjunto de restrições a serem impostas como, por exemplo, o peso suportado pelo próprio alpinista.

Um problema de corte envolvendo uma demanda de itens, onde mais de uma peça será cortada, é conhecido por problema de corte de estoque. Problemas de cortes de estoque são essenciais ao planejamento da produção em algumas indústrias, tais como indústrias de papel, vidro, metalúrgica, plástica, têxtil, etc, onde melhorias no processo de produção podem representar inúmeras vantagens econômicas e operacionais. Nestas indústrias, a

redução dos custos de produção e a melhoria da eficiência estão, freqüentemente, associados à utilização de estratégias adequadas de cortes.

Os trabalhos pioneiros de Gilmore e Gomory, [Gilmore e Gomory, 1961, 1963, 1965], despertaram grande interesse na comunidade de pesquisadores e práticos da otimização combinatória por problemas de cortes e empacotamento o que pode ser notado pelos diversos trabalhos que têm sido desenvolvidos desde então.

A importância econômica e operacional de tais problemas, bem como a dificuldade de resolução, têm motivado os pesquisadores para o desenvolvimento de inúmeros trabalhos, como pode ser verificado em vários livros e artigos de revisão, tais como: [Brown, 1971], [Sweeney e Paternoster, 1992], [Dowsland e Dowsland, 1992], [Dyckhoff e Finke, 1992], [Morabito, 1992], [Morabito e Arenales, 1995] e muitos outros.

Por um longo tempo, técnicas na tentativa de solucionar problemas de cortes vêm sendo desenvolvidas. Estas, apoiam-se em algumas abordagens especializadas como: enumeração implícita, programação dinâmica, relaxação lagrangiana, heurísticas, busca em grafos E/OU e outras.

Os problemas de cortes e empacotamento, conforme [Garey e Johnson, 1979], pertencem a uma classe de problemas denominada *NP-completos*. A grosso modo, podemos dizer que são problemas improváveis de serem resolvidos num tempo limitado por uma função polinomial. É interessante observar que existem resultados teóricos afirmando que, se um problema desta classe puder ser resolvido em um tempo polinomial, então todos os problemas da classe terão solução em tempo polinomial, mas parece pouco provável tal obtenção. Este tem sido o motivo da pesquisa em métodos heurísticos, onde soluções “boas” são produzidas.

Neste projeto enfocamos uma variação do clássico Problema da Mochila, chamado de *Problema da Mochila Compartimentada*, pois a mochila pode ser dividida em compartimentos que, por sua vez, agrupam itens com características semelhantes tais como alimentos, produtos de limpeza, roupas, etc (cada compartimento pode ser visto como um Problema da Mochila clássico). Tal problema tem sido ainda pouco estudado na literatura e podemos citar [Ferreira *et al.*, 1990], [Carvalho, 1991], [Pereira, 1993], [Carvalho e Rodrigues, 1994], [Hoto, 1996], [Hoto e Arenales, 1996]. Na prática, este problema tem grande importância econômica e é difícil de ser resolvido. Tal problema surge nas linhas de

produção de algumas indústrias como na produção de placas de circuitos impressos (caso bidimensional) e, também, nas indústrias de tubos de aço, onde as bobinas do estoque são cortadas em *bobinas intermediárias* para então serem recortadas em fitas que darão origem aos tubos, seguindo uma série de restrições físicas e técnicas. Como ilustração da importância deste problema, as sobras de material numa indústria de tubos de aço visitada chegam a 10%, acarretando um prejuízo acima dos padrões.

Organização do Texto

O texto está dividido em 7 capítulos e possui a seguinte estrutura:

No Capítulo 1 mostramos uma visão geral sobre *problemas de cortes* e uma classificação geral.

No Capítulo 2 um resumo dos vários tipos de Problemas da Mochila é apresentado. Esse resumo consta de uma breve descrição dos problemas, bem como de suas modelagens segundo os trabalhos de [Martello e Toth, 1990] e [Lin, 1998]. Esta descrição visa mostrar que o problema estudado neste projeto não se enquadra em nenhum dos tipos apresentados.

No Capítulo 3 definimos o *Problema da Mochila Compartimentada* de uma maneira genérica. A seguir, apresentamos uma aplicação do problema detectado na indústria metalúrgica para o caso de cortes em bobinas de aço sujeitas à laminação. Por fim, apresentamos alguns estudos encontrados na literatura.

No Capítulo 4 apresentamos uma modelagem matemática para o *Problema da Mochila Compartimentada* e fazemos uma breve descrição sobre um possível limitante para o problema.

No Capítulo 5 descrevemos métodos de resolução. Apresentamos três heurísticas propostas para a resolução do *Problema de Múltiplas Mochilas Compartimentadas* idênticas e descrevemos os seus respectivos algoritmos. Além disso, descrevemos o processo de implementação dessas heurísticas, mostrando algumas características básicas dos programas desenvolvidos.

No Capítulo 6 são apresentados os resultados computacionais obtidos na execução dos programas que representam as heurísticas propostas.

Por fim, no Capítulo 7, apresentamos as conclusões obtidas e possíveis pesquisas futuras.

Capítulo 1. Introdução aos Problemas de Corte

Neste capítulo daremos uma visão geral sobre *problemas de cortes*, que, devido a algumas particularidades específicas, introduzem uma certa dificuldade na obtenção de métodos gerais.

1.1 Considerações Iniciais

Cortar unidades maiores em unidades menores e empacotar unidades menores dentro de unidades maiores são problemas que podem ser vistos como dois lados de uma mesma moeda. Tal correspondência resulta da dualidade entre esses problemas[Morabito, 1992].

O *Problema de Empacotamento (Packing Problem)* consiste, genericamente, em empacotar unidades menores dentro de uma unidade maior, de forma que certos objetivos sejam otimizados. Um exemplo deste problema consiste em arranjar o maior volume possível de caixas dentro de um contêiner.

Por outro lado, o *Problema de Corte (Cutting Problem)*, de forma genérica, consiste em cortar uma unidade grande (*objeto*), que esteja disponível, para a produção de um conjunto de unidades pequenas (*itens*) que estão sendo requisitadas. As formas e medidas do objeto e dos itens, bem como suas quantidades demandadas, são especificadas. Tais problemas são encontrados na literatura com maior frequência que os Problemas de Empacotamento.

Possíveis objetos a serem cortados devem obedecer algumas restrições técnicas, ou mesmo de mercado, que exigem tratos especiais ou uma série de exigências a serem

consideradas, quando na sua obtenção. Esses objetos podem ser produzidos pela própria indústria ou serem adquiridos junto a fornecedores.

Conforme os itens solicitados, podemos combiná-los dentro de um objeto de inúmeras maneiras a fim de atender a um conjunto de restrições. A estas combinações denominamos *planos de corte*. O número de planos de corte é, na prática, muito elevado, exigindo que técnicas apuradas sejam utilizadas para sua obtenção. Dentre essas técnicas podemos citar: enumeração implícita, programação dinâmica, relaxação lagrangiana, heurísticas e busca em grafos. Na figura 1 podemos visualizar um exemplo de plano de corte gerado em um objeto unidimensional.

Figura 1: Plano de corte unidimensional gerado em um objeto.

Vale salientar que, dificilmente obtemos um plano de corte que utilize todo o objeto. Temos, então, uma sobra ilustrada na figura 1.

Um plano de corte pode, também, ser visto para o caso onde o corte deve ser executado em placas retangulares com espessura constante, produzindo itens menores, igualmente retangulares. Um exemplo típico de problema deste tipo é o corte em placas de madeira na indústria de móveis. A figura 2 mostra uma ilustração de um plano de corte em uma placa retangular.

Figura 2: Plano de corte bidimensional em uma placa retangular.

Existem, também, especificações do processo de cortagem (limitação do número de facas, cortes guilhotinados, estagiados, restritos, compartimentados, etc) que fazem com que métodos gerais para essa classe de problemas sejam praticamente impossíveis. Por exemplo, um método para o problema de corte de bobinas na indústria de papel não é aplicado ao corte de bobinas de aço; da mesma forma, o corte de retângulos na indústria de móveis difere do problema de corte de retângulos na indústria de vidros ou de circuitos impressos, muito embora todos estejam relacionados à classe de problemas de cortes.

Quando estamos considerando que uma certa quantidade de itens deva ser produzida e que, para isto, uma única operação de cortagem de um plano de corte não será suficiente, teremos um problema em que a solução resulta no menor número possível de objetos do estoque a serem utilizados. Este problema é conhecido na literatura como *Problema de Corte de Estoque*.

Como dissemos, os problemas de cortes de estoque são essenciais para o planejamento da produção em algumas indústrias, tais como indústrias de papel, vidro, metalúrgica, plástica, têxtil, etc. Nessas indústrias, as reduções dos custos de produção e melhoria da eficiência estão, freqüentemente, associadas à utilização de estratégias adequadas no processo de cortagem.

1.2 Classificação dos Problemas de Cortes

Devido à grande variedade de problemas de cortes ocorridos na prática, [Dyckhoff e Finke, 1992] apresentaram um extenso estudo reunindo diversos problemas da área.

Segundo [Dyckhoff e Finke, 1992], a determinação do tipo de um problema de corte se dá baseado em aspectos básicos respeitando a sua estrutura lógica: dimensionalidade, tipo de seleção dos objetos/itens, características dos objetos/itens (tipos de medida, aparência, sortimento de objetos e itens e disponibilidade), conjunto de restrições associadas aos planos de corte, objetivos, status da informação, variabilidade dos dados e métodos de solução. Cada problema de corte teria, então, uma quádrupla associada a ele: dimensionalidade / seleção / sortimento objetos / sortimento itens. Na prática, porém, observamos que essa classificação ainda deixa um pouco a desejar visto que alguns problemas não se enquadram facilmente nestes critérios.

Em nossos estudos, estaremos preocupados em classificar os problemas de cortes quanto a sua dimensionalidade. Isto será feito na seção seguinte.

1.2.1 Dimensionalidade

Em geral, os problemas de cortes podem ser classificados de acordo com o aspecto geométrico das peças a serem tratadas. Nesta classificação podemos ter problemas de cortes unidimensionais, com apenas uma dimensão relevante ao corte, problemas bidimensionais, com duas dimensões relevantes, tridimensionais, e até multidimensionais.

1.2.1.1 Problema Unidimensional

Um problema é dito unidimensional quando apenas uma dimensão é relevante no processo de cortagem. Como casos típicos de problemas de cortes unidimensionais podemos citar o corte de materiais como papel, tecido, plástico e aço para serem utilizados nos mais diversos setores.

Suponha que um objeto (barra, bobina, etc.) deva ser cortado ao longo de seu comprimento para a produção de itens de comprimentos especificados. Cada item tem um valor associado chamado “*valor de utilidade*”. Itens cujos comprimentos não foram especificados são considerados perdas e têm valores de utilidade nulos. Surge então um problema de otimização combinatória:

Maximizar VALOR DE UTILIDADE TOTAL

Este problema de corte, embora simplificado, surge como um importante subproblema na resolução de problemas de cortes mais gerais. A figura 1, mostrada anteriormente, exhibe um plano de corte para um problema unidimensional.

É fácil observar, ainda, que o problema anterior pode ser resolvido por uma sequência de decisões que consiste em incluir um item por vez, dentre os tipos de itens disponíveis. Esse problema pode ser modelado como um problema de otimização linear inteiro. Uma observação importante baseia-se no fato de que a geração de um simples plano

de corte, ou seja, alocar itens em um único objeto, pode ser formulado em certos casos como o clássico Problema da Mochila unidimensional. No capítulo seguinte faremos um resumo dos vários tipos de Problemas da Mochila que podem ser encontrados na literatura. Vale salientar que, em nossos estudos do *Problema da Mochila Compartimentada*, estaremos assumindo uma abordagem unidimensional.

1.2.1.2 Problema Bidimensional

No problema bidimensional, duas dimensões (comprimento e largura) são relevantes na obtenção da solução, enquanto a espessura é constante. As dificuldades aumentam consideravelmente para se gerar arranjos sem que ocorra sobreposição de itens nos planos de corte. A figura 2, mostrada anteriormente, exibe um plano de corte bidimensional em uma placa retangular.

Entre os problemas bidimensionais podemos citar alguns já bastante estudados, como o corte de placas de madeira na indústria de móveis, chapas de aço, placas de vidro e o Problema de Alocação de Tarefas. Estudos de problemas bidimensionais podem ser vistos em [Gilmore e Gomory, 1965], [Yanasse *et al.*, 1990], [Morabito e Arenales, 1995] e [Gramani, 1997], entre muitos outros.

1.2.1.3 Problema Tridimensional

No problema tridimensional, três dimensões são relevantes para a obtenção da solução. Basicamente, trata-se de arranjar itens espaciais, sem sobrepor-los, dentro de objetos maiores. Podemos citar como exemplos de problemas tridimensionais o Problema de Carregamento de Contêineres, cortes em indústrias de espuma, entre outros [Morabito, 1992]. A figura 3 exibe uma representação de problemas de cortes em três dimensões.

Figura 3: Problema tridimensional.

1.2.1.4 Problemas 1.5-dimensional e 2.5-dimensional

Ainda sob o aspecto geométrico, é possível encontrar problemas do tipo 1.5-dimensional, que são essencialmente bidimensionais, porém uma das duas dimensões consideradas é variável. Este caso tem aplicação no corte de peças de vestuário. Outros são problemas do tipo 2.5-dimensional, onde uma das três dimensões é variável. Uma aplicação é o problema de se efetuar o carregamento de unidades dentro de caixas abertas, ou seja, as bases estão definidas, mas a altura deverá ser definida. Problemas em que uma das dimensões é variável são, genericamente, denominados $n\frac{1}{2}$ -dimensional.

1.2.1.5 Problema Multidimensional

Além dos problemas já expostos, o problema multidimensional também pode aparecer. Uma ocorrência desse tipo de problema pode aparecer associada ao Problema de Alocação de Tarefas [Morabito, 1992].

A tabela 1 resume a classificação dos problemas de cortes segundo seus aspectos geométricos:

Tabela 1: Classificação geométrica dos problemas de cortes.

Problemas	Dimensões Relevantes
Unidimensionais	1
Bidimensionais	2
Tridimensionais	3
n -dimensionais	$n > 3$
$n\frac{1}{2}$ -dimensionais	n fixas e 1 variável

1.3 Restrições sob os Planos de Corte

Basicamente, as restrições geométricas sob os planos de corte estão associadas às distâncias entre os itens, limitação quanto à combinação entre os tipos de itens, limitação do número de itens em um plano de corte e limitação no número de tipos de itens em um plano de corte.

1.4 Objetivos

Os objetivos são de grande importância, pois definem a meta a ser atingida, como por exemplo, minimização da perda absoluta ou relativa de material, minimização dos custos de produção, custos de material, custos de armazenamento ou custos devidos às trocas de planos, maximização de lucros, entre outros.

No capítulo seguinte, daremos uma breve descrição dos vários tipos de Problemas da Mochila que podem ser encontrados na literatura.

Capítulo 2. Problemas da Mochila

2.1 Considerações Iniciais

Neste capítulo faremos uma revisão dos vários tipos de Problemas da Mochila estudados na literatura. O problema a ser focado nesse projeto, embora possa ser catalogado como um Problema da Mochila, não apareceu nos trabalhos de [Martello e Toth, 1990] e [Lin, 1998], apesar de sua importância, como veremos no decorrer do trabalho.

Os Problemas da Mochila caracterizam uma classe de problemas da programação linear inteira e são classificados na literatura, segundo a sua complexidade de resolução, como problemas *NP-hard*.

Genericamente, o Problema da Mochila é definido do seguinte modo: Suponha que um alpinista deva carregar sua mochila selecionando itens, dentre vários disponíveis, e considerando a capacidade da mochila. A cada item é atribuído um valor de utilidade e o alpinista deve selecioná-los buscando maximizar o valor de utilidade total.

Descrevemos a seguir, alguns dos problemas que podem ser catalogados como Problema da Mochila, segundo os trabalhos de [Martello e Toth, 1990] e [Lin, 1998] e apresentamos a modelagem matemática correspondente.

Para o problema do alpinista enunciado anteriormente, considere os seguintes dados:

- m : número de tipos de itens;
- v_i : valor de utilidade do item tipo i , $i=1, \dots, m$;
- l_i : peso do item tipo i , $i=1, \dots, m$;
- L : capacidade da mochila.

2.2 Problema da Mochila Inteiro

O problema modelado a seguir é chamado na literatura de *Problema da Mochila Inteiro* ou simplesmente *Problema da Mochila*. Não há limitação nas quantidades de itens selecionados. Associando-o ao problema de cortes em barras, temos a situação onde uma barra deve ser cortada ao longo de seu comprimento, sem que uma quantidade máxima de itens seja especificada (problema de corte unidimensional irrestrito e a restrição básica pode ser chamada de *restrição física*). Temos, então:

Variável de decisão:

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

Modelagem Matemática:

$$\text{Maximizar } \Phi = \sum_{i=1}^m v_i x_i \quad (1.1)$$

$$\text{Sujeito a: } \sum_{i=1}^m l_i x_i \leq L \quad (1.2)$$

$$x_i \geq 0 \text{ e inteiro, } i = 1, \dots, m. \quad (1.3)$$

Vale observar que esse modelo não representa o Problema da Mochila Compartimentada, que é objeto do nosso estudo, a ser definido no capítulo 2.

2.3 Problema da Mochila 0-1

O Problema da Mochila 0-1 é, talvez, o mais importante Problema da Mochila e um dos mais estudados problemas de programação discreta. A razão para tal interesse está, basicamente, ligada a três fatores: a) pode ser visto como o mais simples problema de programação linear inteira; b) aparece como um subproblema em muitos outros problemas mais complexos; c) pode representar uma gama muito grande de situações práticas.

No Problema da Mochila 0-1 temos a situação em que um único exemplar de cada item pode ser selecionado. Neste caso, as variáveis de decisão são:

$$x_i = \begin{cases} 1, & \text{se o item } i \text{ for selecionado;} \\ 0, & \text{caso contrário.} \end{cases} \quad i = 1, \dots, m,$$

Modelagem Matemática:

$$\text{Maximizar } \Phi = \sum_{i=1}^m v_i x_i \quad (2.1)$$

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

$$x_i = 0 \text{ ou } 1, i = 1, \dots, m. \quad (2.3)$$

Durante as últimas décadas, o Problema da Mochila 0-1 tem sido estudado por diferentes abordagens, tais como a Programação Dinâmica e a Enumeração Implícita.

2.4 Problema da Mochila Restrito

Alguns problemas podem apresentar condições adicionais, como por exemplo, a limitação da quantidade de itens a serem selecionados. Neste caso, o problema passa a ser chamado de *Problema da Mochila Restrito ou Limitado*. Para o Problema da Mochila Restrito, além dos dados necessários para os modelos anteriores, considere ainda:

- d_i : quantidade máxima de itens tipo i que pode ser selecionada, $i=1, \dots, m$;

Modelagem Matemática:

$$\text{Maximizar } \Phi = \sum_{i=1}^m v_i x_i \quad (3.1)$$

$$\text{Sujeito a: } \sum_{i=1}^m l_i x_i \leq L \quad (3.2)$$

$$0 \leq x_i \leq d_i \text{ e inteiro, } i = 1, \dots, m. \quad (3.3)$$

Alguns problemas de cortes são formulados como um Problema da Mochila Restrito onde a produção excedente de itens é considerada perda.

O Problema da Mochila Restrito pode ser visto como uma generalização do Problema da Mochila 0-1, onde $d_i = 1$ para $i = 1, \dots, m$.

2.5 Problema da Soma de Subconjuntos

O Problema da Soma de Subconjuntos consiste em selecionar um subconjunto de itens cuja soma total dos pesos dos itens escolhidos se aproxime ao máximo de L , sem excedê-lo.

Modelagem Matemática:

$$\text{Maximizar } \Phi = \sum_{i=1}^m l_i x_i \quad (4.1)$$

$$\text{Sujeito a: } \sum_{i=1}^m l_i x_i \leq L \quad (4.2)$$

$$x_i = 0 \text{ ou } 1, i = 1, \dots, m. \quad (4.3)$$

O Problema da Soma de Subconjuntos é um caso particular do Problema da Mochila 0-1, onde $v_i = l_i$ para $i = 1, \dots, m$.

Tal problema é conhecido, também, como Problema da Mochila de Valor Independente e surge em situações onde a quantidade desejada deve ser alcançada, tal que a diferença entre os valores da capacidade da mochila (L) e a soma total dos pesos dos itens selecionados seja minimizado, sem que L seja ultrapassado.

2.6 Problema do Troco

O Problema do Troco geralmente aparece na literatura como um problema de minimização. O problema consiste em selecionar um número x_i ($i = 1, \dots, m$) de itens de cada tipo i tal que o peso total seja L e o número total de itens selecionados seja mínimo.

Modelagem Matemática:

$$\text{Minimizar } \Phi = \sum_{i=1}^m x_i \quad (5.1)$$

$$\text{Sujeito a: } \sum_{i=1}^m l_i x_i = L \quad (5.2)$$

$$x_i \geq 0 \text{ e inteiro, } i = 1, \dots, m. \quad (5.3)$$

O problema é chamado Problema do Troco, pois pode ser interpretado como o problema de um caixa (supermercado, banco, etc) que deve devolver uma determinada quantia, L , usando, para isto, um número mínimo de moedas de valores específicos l_i ($i = 1, \dots, m$). Nesse caso, para cada valor, um número ilimitado de moedas está disponível. É importante notar que a condição de igualdade imposta pode fazer com que inexista uma solução para o problema.

2.7 Problema da Mochila Quadrático

O Problema da Mochila Quadrático é um dos mais estudados problemas na área de Problemas da Mochila Não Lineares. Um Problema da Mochila Não Linear, em geral, é o Problema da Mochila com função objetivo não linear ou que envolve restrições não lineares. Um Problema da Mochila Quadrático geral pode ser formulado matematicamente da seguinte forma:

Modelagem Matemática:

$$\text{Maximizar } \Phi = f(x) \quad (6.1)$$

$$\text{Sujeito a: } \sum_{i=1}^m l_i x_i \leq L \quad (6.2)$$

$$x_i \geq 0 \text{ e inteiros, } i = 1, \dots, m. \quad (6.3)$$

Onde $f(x)$ representa uma função quadrática de x da forma $xQx^T + cx^T$ com $x = (x_1, x_2, \dots, x_m)$, $c = (c_1, c_2, \dots, c_m)$ e $Q = (q_1, q_2, \dots, q_m)$ para algum i com $q_i \neq 0$, $i = 1, \dots, m$.

Até aqui, os Problemas da Mochila consistiram em carregar apenas uma mochila. Os próximos problemas, também classificados por [Martello e Toth, 1990] e [Lin, 1998] como Problemas da Mochila, consistem em carregar várias mochilas.

2.8 Problema de Múltiplas Mochilas 0-1

O Problema de Múltiplas Mochilas 0-1 consiste em carregar um conjunto de n mochilas ($n \leq m$) cujas capacidades são dadas por: $L_j, j = 1, \dots, n$. Os demais dados do problema são os mesmos do problema para uma única mochila.

O problema consiste em selecionar subconjuntos disjuntos de itens tal que o valor de utilidade dos itens selecionados seja máximo, e cada subconjunto possa ser associado a uma mochila diferente, cuja capacidade não seja menor que o peso total dos itens no subconjunto.

Variável de decisão:

$$x_{ij} = \begin{cases} 1, & \text{se o item } i \text{ está associado à mochila } j; \\ 0, & \text{caso contrário.} \end{cases}$$

Modelagem Matemática:

$$\text{Maximizar } \Phi = \sum_{j=1}^n \sum_{i=1}^m v_i x_{ij} \quad (7.1)$$

$$\text{Sujeito a: } \sum_{i=1}^m l_i x_{ij} \leq L_j, \quad j = 1, \dots, n \quad (7.2)$$

$$\sum_{j=1}^n x_{ij} \leq 1, \quad i = 1, \dots, m. \quad (7.3)$$

$$x_{ij} = 0 \text{ ou } 1, \quad i = 1, \dots, m \text{ e } j = 1, \dots, n \quad (7.4)$$

Quando $n = 1$, o Problema de Múltiplas Mochilas 0-1 se reduz a um Problema da Mochila 0-1 simples, pois a restrição (7.3) torna-se redundante.

2.9 Problema de Designação Generalizada

O Problema de Designação Generalizada pode ser descrito usando a terminologia aplicada para os Problemas da Mochila. O problema consiste em associar cada item a

exatamente uma mochila, visando maximizar o ganho total, sem associar a nenhuma mochila um peso total que ultrapasse sua capacidade. Considere os seguintes dados:

- v_{ij} : ganho fornecido pela associação do item i à mochila j , $i=1,...,m$ e $j=1,...,n$;
- l_{ij} : peso do item i se associado à mochila j , $i=1,...,m$ e $j=1,...,n$;

Modelagem Matemática:

$$\text{Maximizar } \Phi = \sum_{j=1}^n \sum_{i=1}^m v_{ij} x_{ij} \quad (8.1)$$

$$\text{Sujeito a: } \sum_{i=1}^m l_{ij} x_{ij} \leq L_j \quad j = 1, \dots, n \quad (8.2)$$

$$\sum_{j=1}^n x_{ij} = 1, \quad i = 1, \dots, m. \quad (8.3)$$

$$x_{ij} = 0 \text{ ou } 1, \quad i = 1, \dots, m \text{ e } j = 1, \dots, n \quad (8.4)$$

O problema é, freqüentemente, descrito na literatura como o problema de designação ótima de m tarefas a n processadores, dados o ganho v_{ij} e a quantidade de recursos l_{ij} correspondente à associação da tarefa i ao processador j e a quantidade total de recursos L_j suportados por cada processador j .

2.10 Problema *Bin-Packing*

O Problema *Bin-Packing* pode ser descrito usando a terminologia de Problema da Mochila, onde:

- l_i : peso do item i , $i=1,...,m$;
- L : capacidade de cada caixa.

Associa-se cada item a uma caixa, tal que o peso total dos itens em cada caixa não exceda L e o número de caixas usadas seja mínimo.

Variáveis de decisão:

$$y_j = \begin{cases} 1, & \text{se a caixa } j \text{ é usada;} \\ 0, & \text{caso contrário.} \end{cases}$$
$$x_{ij} = \begin{cases} 1, & \text{se o item } i \text{ está associado à caixa } j; \\ 0, & \text{caso contrário.} \end{cases}$$

Modelagem Matemática:

$$\text{Minimizar } \Phi = \sum_{j=1}^n y_j \quad (9.1)$$

$$\text{Sujeito a: } \sum_{i=1}^m l_i x_{ij} \leq L_j \cdot y_j \quad j = 1, \dots, n \quad (9.2)$$

$$\sum_{j=1}^n x_{ij} = 1, \quad i = 1, \dots, m. \quad (9.3)$$

$$y_j = 0 \text{ ou } 1, x_{ij} = 0 \text{ ou } 1, i = 1, \dots, m \text{ e } j = 1, \dots, n \quad (9.4)$$

No capítulo seguinte, definiremos o *Problema da Mochila Compartimentada* observando a diferença deste em relação aos problemas encontrados na literatura e apresentados anteriormente. Nota-se, claramente, que se trata de um problema ainda não explorado na literatura e, como surge de aplicações práticas importantes, reforça a validade de um estudo mais aprofundado no tema.

Capítulo 3. O Problema da Mochila Compartimentada

Neste capítulo definimos o *Problema da Mochila Compartimentada* e apresentamos uma aplicação no caso de cortes em bobinas de aço sujeitas à laminação detectada na indústria metalúrgica. Além disso, descrevemos, resumidamente, alguns estudos realizados sobre o problema de corte em bobinas de aço.

3.1 Definição do Problema

Um alpinista deve carregar sua mochila com m possíveis itens de sua utilidade. A cada item i , o alpinista atribui um *valor de utilidade* v_i e seu peso l_i . O máximo peso que o alpinista suporta em sua jornada é de L . Até este ponto, o problema coincide com a formulação clássica do Problema da Mochila. Além disso, muitos itens são de *classes* distintas (por exemplo: *classe 1*: alimentos, *classe 2*: utensílios, *classe 3*: roupas, *classe 4*: calçados, etc.) e devem estar em compartimentos separados dentro da mochila (isto é, cada compartimento contém somente itens de uma mesma classe). Os compartimentos da mochila são flexíveis, permitindo que o alpinista carregue maior peso em alimentos do que em roupas, por exemplo. Entretanto, as capacidades dos compartimentos (incógnitas) são limitadas inferior e superiormente, caso estes sejam criados, digamos por: L_{min} e L_{max} (isto é, a soma dos pesos l_i dos itens alocados no compartimento deve ser superior ou igual a L_{min} e inferior ou igual a L_{max}). A cada compartimento é associado um custo c_k , caso este seja preenchido com itens da classe k e, além disso, cada compartimento incluído na mochila produzirá uma perda da capacidade da mesma, digamos, S (isto é, se, por exemplo, 3 compartimentos forem utilizados, então a capacidade disponível na mochila será de $L-3S$, ao

invés de L). É importante ressaltar que vários compartimentos, de capacidades diferentes, podem ser criados para carregar itens de uma mesma classe, ou em outras palavras, itens de uma mesma classe podem ser alocados em mais que um compartimento.

O *Problema da Mochila Compartimentada* consiste em determinar as capacidades adequadas de cada compartimento e como estes devem ser carregados de modo que o valor de utilidade total (soma dos valores de utilidade de todos os itens selecionados) seja máximo, descontando-se os custos dos compartimentos, os quais dependem das classes com que foram preenchidos (c_k).

Como poderemos observar em seções seguintes, tal problema surge no corte de bobinas de aço, onde os itens são *fitas* para a produção de tubos soldados, os compartimentos são as *bobinas intermediárias* e a mochila a ser preenchida é uma *bobina mestre*. Apesar da importância prática, esse problema foi pouco estudado na literatura.

3.2 Agrupamento dos Itens

Com a definição descrita na seção anterior, podemos imaginar que a construção de um plano de corte para o *Problema da Mochila Compartimentada* não se torna algo muito evidente.

Designamos por classe, um subconjunto de itens que podem ser armazenados dentro de um mesmo compartimento. Por exemplo: se desejarmos encher uma mochila com camisetas, calças, sapatos, remédios, biscoitos, canivete, abridor de garrafa, entre outros, podemos considerar que o agrupamento desses itens seja de forma que remédios e alimentos não fiquem em um mesmo compartimento, ou seja, seriam de classes distintas. Já, por outro lado, camisetas e calças podem ser agrupadas em um mesmo compartimento, pertencendo à classe de roupas. Existem, também, aqueles itens que podem ser colocados livremente, isto é, sem a necessidade de compartimentá-los ("*itens livres*"), como o item canivete, por exemplo.

Suponhamos que uma mochila deva ser preenchida com os itens da tabela 2, obedecendo as seguintes relações:

Tabela 2: Exemplo de agrupamento de itens para uma mochila compartimentada.

Tipo de Item	Descrição	Agrupamento	Classe
1	Camisetas	Roupas	A
2	Calças	Roupas	A
3	Bermudas	Roupas	A
4	Meias	Roupas	A
5	Cuecas	Roupas	A
6	Sapatos	Calçados	B
7	Chinelos	Calçados	B
8	Pomada	Remédios	C
9	Antitérmico	Remédios	C
10	Antiinflamatório	Remédios	C
11	Analgésico	Remédios	C
12	Biscoitos	Alimentos	D
13	Enlatados	Alimentos	D
14	Refrigerantes	Alimentos	D
15	Xaropes	Separado	E
16	Abridor de Latas	Livre	F
17	Canivete	Livre	F

Note que, pela tabela anterior, existem seis agrupamentos de itens a serem considerados (Roupas, Calçados, Remédios, Alimentos, Xaropes e Livres). A classe F, representada pelos itens Abridor de Latas e Canivete, corresponde à classe de itens livres, definida anteriormente. Nessa classe, os itens não necessitam de um compartimento para serem alocados. Sendo assim, eles estão livres da necessidade de cumprir as restrições de capacidades impostas aos compartimentos. Apesar disso, eles não poderão estar associados a qualquer outro agrupamento, ou seja, não poderão pertencer a nenhum compartimento estabelecido para as outras classes de itens. Note que, se tivermos uma situação onde somente itens livres são considerados, teremos um clássico Problema da Mochila Unidimensional.

[Hoto, 2000] estudou o problema de múltiplas mochilas compartimentadas, onde certos itens deveriam ser alocados a mochilas específicas, enquanto outros poderiam

pertencer a diferentes mochilas. Ele representou os itens para o agrupamento em bobinas de aço através de um grafo. Faremos o mesmo para o exemplo da tabela anterior. Seja $G(V,E)$ um grafo onde cada vértice V representa um tipo de item e cada aresta E é tal que, se um tipo de item pode ser agrupado com outro, existe uma aresta entre os vértices correspondentes. Veja na figura 4 o grafo relativo ao agrupamento dos itens expostos na tabela anterior.

Figura 4: Representação alternativa para o agrupamento dos itens em uma Mochila Compartimentada.

Considerando $M = \{1, \dots, 17\}$ como o conjunto de índices dos tipos de itens, temos, pelo grafo G , que os tipos de itens estão agrupados em 6 subgrafos $M_1 = \{1, 2, 3, 4, 5\}$, $M_2 = \{6, 7\}$, $M_3 = \{8, 9, 10, 11\}$, $M_4 = \{12, 13, 14\}$, $M_5 = \{15\}$ e $M_6 = \{16, 17\}$. Podemos notar que cada um dos subgrafos é uma clique ou um vértice isolado do grafo G . No estudo de [Hoto, 2000], poder-se ia ter, por exemplo, que os itens 4, 5, 6 e 7 pertenceriam, alternativamente, à uma mesma mochila, seguindo diferentes cliques: $\{ \{1, 2, 3\}, \{4, 5\}, \{6, 7\} \}$ ou $\{ \{1, 2, 3, 4\}, \{5\}, \{6, 7\} \}$. O problema a ser focado neste trabalho pressupõe que as cliques (classes) já tenham sido determinadas.

No capítulo seguinte, estaremos apresentando uma modelagem matemática para o *Problema da Mochila Compartimentada*. Esse modelo representa o preenchimento de uma única mochila compartimentada.

Quando estivermos tratando do problema onde uma demanda de cada item i para $i = 1, \dots, m$ deva ser considerada, e onde será necessário mais que um plano de corte para que estas demandas sejam cumpridas, estaremos então, tratando do *Problema de Múltiplas Mochilas Compartimentadas* idênticas, caracterizando um problema de corte de estoque em mochilas compartimentadas.

No capítulo 5 estaremos apresentando três heurísticas para a resolução do *Problema de Múltiplas Mochilas Compartimentadas* onde as mochilas serão idênticas. Porém, estaremos gerando vários planos de cortes para que as demandas sejam cumpridas.

3.3 O Problema de Corte em Bobinas de Aço

Nessa seção descrevemos o corte em bobinas de aço sujeitas à laminação, procurando enfocar os principais aspectos envolvidos no processo. Além disso, fazemos uma descrição dos custos envolvidos no processo de produção.

Este problema foi observado em uma empresa que produz tubos de aço para diversas aplicações. A linha de produção consiste em produzir fitas, a partir de bobinas de aço em estoque. Essas fitas, por sua vez, serão utilizadas na confecção dos tubos, que terão finalidades específicas.

3.3.1 Considerações Iniciais

Antes de prosseguirmos vamos considerar as seguintes definições que são encontradas em [Ferreira *et al.*, 1990], [Pereira, 1993] e [Hoto, 1996].

Definições:

- *Bobinas mestres* são os *objetos* a serem cortados ou as mochilas a serem preenchidas. Tais bobinas são identificadas pelos seus pesos (entre 12000 e 13500 Kg), larguras (variando de 1100 a 1200 mm), espessura do aço e pelo teor de carbono do aço segundo os critérios do *SAE* (*Society of Automotive Engineers*). Os mais utilizados são: SAE 1008, SAE 1010, SAE 1012, SAE 1017 e SAE 1021.

- *Bobinas intermediárias* são as bobinas obtidas durante a primeira etapa de corte ou os compartimentos que agruparão as classes de itens. As bobinas intermediárias herdam algumas de suas características das bobinas mestres, como a espessura e o teor de carbono do aço. Porém, por sua vez, as bobinas intermediárias terão seus próprios pesos e larguras.
- *Fitas* são os *itens* obtidos durante a segunda etapa de corte, a partir das bobinas intermediárias. As fitas possuem características bem definidas como a largura (de acordo com o diâmetro dos tubos a serem produzidos), a espessura e o tipo de aço.

Observe que uma bobina com o tipo de aço SAE 1008 somente poderá produzir fitas onde uma das características seja o aço SAE 1008. No que se segue, vamos supor que os conjuntos de fitas selecionados tenham as características necessárias para serem cortados a partir de uma mesma bobina mestre.

3.3.2 O Processo de Corte

Uma peculiaridade do problema ao se efetuar cortes em bobinas de aço sujeitas a um processo técnico de laminação, reside no fato de que as bobinas mestres não suportam a laminação devido a sua largura. Em outras palavras, as bobinas mestres devem ser cortadas em bobinas intermediárias cujas larguras aceitem a laminação, e posteriormente recortadas em “fitas” [Ferreira *et al.*, 1990]. Além disso, vale enfatizar que o processo de laminação é executado visando ajustar a espessura das bobinas intermediárias de acordo com as fitas demandadas (na prática, cerca de 40% da produção necessita de laminação).

Sendo assim, temos duas etapas de corte bem caracterizadas, definindo um compartimento de fitas (bobinas intermediárias deverão produzir fitas de mesma espessura) na construção de um plano de corte.

A figura 5 dá uma idéia mais clara do processo de corte das bobinas.

Figura 5: O processo de corte em bobinas de aço.

Denomina-se “*Refugo*” todo tipo de *perda inevitável* de material durante o processo de obtenção das fitas, e “*Sobra*” todas as outras categorias de perdas de material. Temos dois refugos intrínsecos: um durante a primeira etapa de corte e outro durante a segunda etapa de corte [Ferreira *et al.*, 1990]. A figura 6 ilustra melhor este fato:

Figura 6: Corte em bobinas de aço com refugos embutidos.

Na prática, por uma restrição no número de facas, o número de bobinas intermediárias na primeira etapa é limitado.

A dificuldade no processo de cortagem decorre da existência de demandas baixas, de maneira que a utilização de uma bobina mestre com peso consideravelmente alto poderá proporcionar excesso de produção, mesmo que apenas uma fita de um determinado tipo apareça no plano de corte. Para evitar este excesso, [Hoto, 2000] considera uma formulação $1\frac{1}{2}$ -dimensional, introduzindo uma variável no comprimento da bobina mestre. Embora essa abordagem pareça interessante, há uma resistência de ser implementada na prática, portanto, não vamos considerá-la neste trabalho.

3.3.3 A Cortadeira

A grosso modo, uma bobina mestre é desenrolada de maneira uniforme e o processo de corte para obtenção das bobinas intermediárias é feito longitudinalmente por “discos de corte” (não há cortes transversais e por isso podemos entender que o problema é unidimensional). Nessa etapa, existem perdas intrínsecas (que podem ser visualizadas na figura anterior como refugos) nas laterais da bobina, para eliminar as irregularidades das bordas, variando entre 6 a 8 mm. Posteriormente, as bobinas intermediárias são enroladas e passarão ou não para o processo de laminação [Pereira, 1993], uma vez que, se a espessura das fitas de alguma bobina intermediária for idêntica a da bobina mestre, ela não necessitará ser laminada.

O tempo de preparação da máquina é cerca de 40 minutos e o tempo médio para o corte de uma bobina gira em torno de 15 minutos. A figura 7 ilustra o processo de corte.

Figura 7: Perfil do processo de cortagem em bobinas de aço.

3.3.4 Laminação, Encruador, Recortadeira e Recozimento

O processo de laminação tem por objetivo diminuir (quando necessário) a espessura do material de uma bobina intermediária. Esse processo se dá à temperatura ambiente (laminação a frio) por meio de cilindros de laminação que efetuam pressão sobre a lâmina de aço, como veremos na figura 8. Cabe notar que, eventualmente, uma mesma bobina intermediária poderá sofrer mais de uma laminação. Isto é necessário para que se possa obter a espessura desejada do material (a redução é cerca de 15 a 20% da espessura em cada passo, e de 50 a 60% no total). Como todas as outras, a máquina que executará a laminação (o laminador) possui restrições do tipo: Não é possível laminar uma bobina intermediária com menos de 154 mm e mais de 456 mm de largura (isto limita os compartimentos, conforme definição do *Problema da Mochila Compartimentada*, seção 3.1).

Após a última laminação, o material da bobina adquiriu algumas imperfeições superficiais que são corrigidas num aparelho denominado encruador. Este aparelho nada mais é que um laminador, mas ao contrário do convencional, apenas corrige, através de uma laminação suave, as imperfeições da superfície do material da bobina.

A figura 8 ilustra o processo de laminação em perfil.

Figura 8: Perfil do processo de laminação.

Após passar pelo encruador a bobina intermediária é submetida a recortadeira, uma máquina especial, semelhante à cortadeira porém preparada para que se obtenha as fitas. Nesta máquina, os cortes são limitados num total de 8 e, como na cortadeira, existe uma perda intrínseca nas laterais das bobinas variando de 3 a 5 mm. O limite de largura imposto pelas máquinas que irão fazer o recorte é de 600mm, e os tempos de preparação e recorte são os mesmos da primeira etapa de corte.

Devemos salientar que, quando submetemos uma mesma bobina intermediária ao processo de laminação, o material adquire alguns fatores físicos indesejáveis. Em face desse inconveniente é necessário, após o recorte, um tratamento térmico ao material denominado recozimento. No recozimento, as fitas são acondicionadas em fornos especiais onde permanecem por um certo período e, em seguida, são lentamente resfriadas até atingirem a temperatura ambiente. O processo todo leva cerca de 24 horas e constitui um gargalo na produção. Daí a importância de se laminar o mínimo possível.

Basicamente, [Hoto, 2000] resume o fluxo da produção pelo esquema representado na figura 9. Entretanto vale salientar a possibilidade de haver pequenas alterações.

Figura 9: Fluxo da produção de tubos de aço.

3.3.5 Custos dos Processos Envolvidos no Problema

Como visto, custos inerentes ao processo e relativos às perdas de material são típicos do problema. Dentre os custos inerentes, podemos destacar: o custo de laminação; custos devido a refugos que, de uma maneira geral, estão intimamente ligados aos tipos de equipamentos disponíveis e a fatores tecnológicos; e custos devido aos acertos de corte (troca de padrões).

O custo de laminação é uma preocupação fundamental, e nesse sentido, tenta-se evitar o máximo possível este tipo de operação, pois laminar como já ressaltamos, é um processo caro. Naturalmente, o processo de laminação nem sempre poderá ser evitado, portanto, nesse caso é interessante buscar uma minimização deste processo [Hoto, 1996].

Em síntese, as diversas peculiaridades da linha de produção aqui citadas apontam para um problema complexo, e que foge do contexto clássico. Além disso, a necessidade do processo de laminação induz planos de corte nada evidentes, visto que nem todas as fitas são compatíveis entre si. Atualmente, a empresa que trabalha com estes cortes em bobinas de aço indica uma perda de material da ordem de 10%, o que corresponde a uma média mensal de 500 toneladas de aço [Hoto, 2000].

Na seção seguinte, faremos uma breve revisão de alguns estudos do problema de corte em bobinas de aço sujeitos à laminação encontrados na literatura.

3.4 Abordagens para o Problema de Corte em Bobinas de Aço

Nas seções seguintes, apresentaremos brevemente, três abordagens encontradas na literatura para o problema de cortes em bobinas de aço sujeitas ao processo técnico de laminação, segundo os trabalhos de [Pereira, 1993], [Carvalho, 1991] e [Hoto, 1996].

3.4.1 Uma Abordagem por Programação Linear Inteira [Pereira,1993]

Em seus estudos sobre o *Problema da Mochila Compartimentada* aplicado ao corte de bobinas de aço, Pereira procurou resolver o problema propondo uma modelagem baseada em programação linear inteira. O modelo proposto foi resolvido, utilizando-se, para isto, o método *branch-and-bound* (ou de enumeração implícita). Com o objetivo de diminuir o processo de busca, uma estimativa do limitante inferior do modelo foi utilizada, a fim de fornecer uma solução inicial. O emprego desse recurso gerou uma imprevisibilidade na qualidade das soluções, contribuindo para seu posterior abandono.

Um outro problema constatado durante a sua resolução foi relativo ao excesso de produção. Esse inconveniente poderia ser contornado com a introdução de um termo na função objetivo do modelo, que penalizasse o excedente da produção.

Com tal estudo, podemos perceber que, dado às especificidades do problema, é justificável a busca de um algoritmo especializado com abordagem heurística para sua resolução.

3.4.2 Um Problema de Corte em Duas Fases [Carvalho, 1991]

[Carvalho, 1991] estimulado por problemas encontrados na indústria de cortes em bobinas de aço, estudou o problema, procurando resolvê-lo de forma a elaborar um sistema de apoio ao planejamento da produção em uma indústria de aço que, até então, era executado manualmente por uma equipe de programadores experientes.

Carvalho propôs uma abordagem heurística baseada em programação linear: a priori, gerava todas as bobinas intermediárias e formulava o problema como um problema de corte de estoque unidimensional, onde os planos de corte eram formados de bobinas intermediárias (um procedimento de geração de colunas era usado para resolver a formulação PL, onde os subproblemas eram Problemas da Mochila, cujos itens correspondiam às bobinas intermediárias).

3.4.3 Um Problema de Corte Unidimensional Sujeito a Restrições de Agrupamento [Hoto, 1996]

Em seus estudos, Hoto procurou enfatizar a busca por uma heurística gulosa. Inicialmente, era realizado um agrupamento das fitas em classes onde cada classe era constituída apenas de fitas de mesma espessura, ou seja, que podiam pertencer a uma mesma bobina intermediária. Essa estratégia pré estabelecia uma relação classe – bobina intermediária, cujo objetivo final era gerar planos de corte onde as espessuras das bobinas intermediárias fossem próximas da espessura da bobina mestre, minimizando assim, o processo de laminação. Dessa maneira, um processo de ordenação das classes tornou-se necessário de forma a priorizar aquelas que minimizassem o processo de laminação.

A estratégia de Hoto baseava-se em, primeiramente, gerar um plano de referência e, a partir deste, os demais planos de corte. Este plano de referência era formado tomando a classe de maior prioridade e alocando as fitas dessa classe segundo uma ordenação pré-estabelecida, respeitando a largura útil da bobina mestre, o comprimento máximo de laminação, o número máximo de fitas que podiam aparecer numa bobina intermediária, bem como o fato de não exceder a demanda da fita (uma seqüência de Problemas da Mochila Restritos deve ser resolvida).

Uma vez obtido este primeiro plano de corte (plano de referência) tomava-se a primeira bobina intermediária. A partir dessa, retirava-se a maior quantidade possível de fitas de menor prioridade, respeitando a largura mínima permissível para laminação. Em seguida tomava-se a largura útil disponível e alocava-se fitas segundo a mesma estratégia utilizada para gerar o plano de referência.

Dessa forma, obtinha-se o segundo plano de corte. Para que fosse obtido o terceiro plano, tomava-se a primeira e segunda bobinas intermediárias do segundo plano e procedia-se com o mesmo raciocínio até que não fosse mais possível alocar novas bobinas intermediárias. A figura 10 ilustra este processo, onde um plano gerado coincide parcialmente com o plano anterior.

Figura 10: Ilustração da heurística de Hoto.

Com esta heurística, Hoto buscava resolver o problema de cortes em bobinas de aço sujeitas ao processo técnico de laminação através da estratégia do uso de um plano de corte à exaustão. Os resultados obtidos não foram satisfatórios.

Além do corte em bobinas de aço sujeitas à laminação, outro problema surge na literatura com características semelhantes ao *Problema da Mochila Compartimentada*. Tal problema surge nas linhas de produção de algumas indústrias como na produção de placas de circuitos impressos (caso bidimensional).

No capítulo seguinte, apresentaremos uma modelagem proposta, neste estudo, para o *Problema da Mochila Compartimentada*, e mostraremos alguns possíveis limitantes para o problema.

Capítulo 4. Modelagem do Problema da Mochila Compartimentada

Neste capítulo, propomos uma modelagem para o *Problema da Mochila Compartimentada* e fazemos uma breve descrição de um possível limitante para o problema.

4.1 Uma Modelagem Matemática

Inicialmente, apresentamos a notação utilizada para o modelo proposto. É importante lembrar que a cada compartimento da mochila, tem-se associado uma única classe k de itens.

Dados do problema:

- $M = \{1, \dots, m\}$: conjunto dos tipos de itens;
- K : número total de classes distintas;
- C_k : subconjunto de M , contendo itens de mesma classe, $k = 1, \dots, K$,
($C_i \cap C_j \neq \emptyset$ com $i \neq j$);
- C_K : subconjunto que representa a classe dos itens livres;
- N_k : número total de possíveis compartimentos para a classe k ;
- c_k : custo de incluir um compartimento com a classe k na mochila ($c_k \geq 0$), $k = 1, \dots, K$;
- S : perda (Kg) decorrente da inclusão de um novo compartimento na mochila;
- L : capacidade (Kg) da mochila;
- L_{min} : capacidade mínima (Kg) de cada compartimento;
- L_{max} : capacidade máxima (Kg) de cada compartimento ($L_{min} < L_{max} < L$);

- l_i : peso (Kg) do item i ($l_i > 0$), $i = 1, \dots, m$;
- v_i : valor de utilidade do item i ($v_i \geq 0$), $i = 1, \dots, m$;
- d_i : limite máximo de itens i na mochila, $i = 1, \dots, m$;

Variáveis:

- α_{ijk} : número de itens do tipo i , da classe k , no compartimento do tipo j , $i=1, \dots, m$, $k=1, \dots, K$ e $j=1, \dots, N_k$;
- β_{jk} : número de repetições do compartimento do tipo j alocados com a classe k , $k = 1, \dots, K$ e $j=1, \dots, N_k$;

Assim, o j -ésimo compartimento com itens da classe k tem:

- A capacidade ocupada dada por:

$$L_{jk} = \sum_{i \in C_k} l_i \alpha_{ijk}, \quad k = 1, \dots, K \text{ e } j = 1, \dots, N_k. \quad (10)$$

- O valor de utilidade dado por:

$$V_{jk} = \sum_{i \in C_k} v_i \alpha_{ijk}, \quad k = 1, \dots, K \text{ e } j = 1, \dots, N_k. \quad (11)$$

Um modelo matemático para um problema de preencher uma única mochila compartimentada pode ser escrito por:

$$\text{Maximizar} \quad \sum_{k=1}^K \sum_{j=1}^{N_k} (V_{jk} - c_k) \beta_{jk} \quad (12.1)$$

$$\text{Sujeito a :} \quad V_{jk} = \sum_{i \in C_k} v_i \alpha_{ijk}, \quad k = 1, \dots, K \text{ e } j = 1, \dots, N_k \quad (12.2)$$

$$L_{jk} = \sum_{i \in C_k} l_i \alpha_{ijk}, \quad k = 1, \dots, K \text{ e } j = 1, \dots, N_k \quad (12.3)$$

$$L_{\min} \leq L_{jk} \leq L_{\max}, \quad k = 1, \dots, (K-1) \text{ e } j = 1, \dots, N_k \quad (12.4)$$

$$\sum_{k=1}^K \sum_{j=1}^{N_k} \alpha_{ijk} \beta_{jk} \leq d_i, \quad i = 1, \dots, m \quad (12.5)$$

$$\sum_{k=1}^K \sum_{j=1}^{N_k} (L_{jk} + S) \beta_{jk} \leq L \quad (12.6)$$

$$\alpha_{ijk} \geq 0, \text{ inteiro} \quad e \quad (12.7)$$

$$\beta_{jk} \geq 0, \text{ inteiro}, \quad (12.8)$$

$$\text{para } i = 1, \dots, m, \quad k = 1, \dots, K \text{ e } j = 1, \dots, N_k.$$

No modelo, temos, a restrição (12.4) associada aos limites físicos dos compartimentos, a restrição (12.5) que limita o número de itens na mochila e, por fim, a restrição (12.6) correspondente aos limites físicos da mochila. O problema (12.*) é um problema de programação não linear inteiro.

É importante notar que esta formulação é mais geral do que o problema de corte em bobinas de aço visto no capítulo anterior.

4.2 Limitantes para o Problema da Mochila Compartimentada

4.2.1 Modelo Agregado

Uma relaxação do *Problema da Mochila Compartimentada* pode ser obtida da seguinte forma:

Sejam

- x_i : número de itens do tipo i na mochila, $i = 1, \dots, m$;
- y_k : número de compartimentos ocupados por itens da classe k , $k = 1, \dots, K$.

Podemos expressar x_i e y_k em termos das variáveis do modelo (12.*).

$$x_i = \sum_{k=1}^K \sum_{j=1}^{N_k} \alpha_{ijk} \beta_{jk} \quad (13)$$

$$y_k = \sum_{j=1}^{N_k} \beta_{jk} \quad (14)$$

Usando (12.2) podemos escrever (12.1) por:

$$\begin{aligned} & \sum_{k=1}^K \sum_{j=1}^{N_k} \sum_{i \in C_k} v_i \alpha_{ijk} \beta_{jk} - \sum_{k=1}^K \sum_{j=1}^{N_k} c_k \beta_{jk} \\ &= \sum_{i=1}^m v_i \sum_{k=1}^K \sum_{j=1}^{N_k} \alpha_{ijk} \beta_{jk} - \sum_{k=1}^K c_k \sum_{j=1}^{N_k} \beta_{jk} \\ &= \sum_{i=1}^m v_i x_i - \sum_{k=1}^K c_k y_k \end{aligned} \quad (15)$$

Substituindo (12.3) em (12.4) e multiplicando por β_{jk} e somando-se em j temos:

$$\sum_{j=1}^{N_k} \beta_{jk} L_{min} \leq \sum_{j=1}^{N_k} \sum_{i \in C_k} l_i \alpha_{ijk} \beta_{jk} \leq \sum_{j=1}^{N_k} \beta_{jk} L_{max}$$

De onde:

$$y_k L_{min} \leq \sum_{i \in C_k} l_i x_i \leq y_k L_{max} \quad (16)$$

$(\sum_{j=1}^{N_k} \alpha_{ijk} \beta_{jk})$ é o número de itens tipo i da classe C_k , na mochila)

Desta forma, uma relaxação do modelo (12. *) é dada por:

$$\text{Maximizar} \quad v_1x_1 + \dots + v_mx_m - (c_1y_1 + \dots + c_Ky_K) \quad (17.1)$$

$$\text{Sujeito a :} \quad \sum_{i=1}^m l_i x_i \leq L - S \left(\sum_{k=1}^K y_k \right) \quad (17.2)$$

$$y_k \cdot L_{\min} \leq \sum_{i \in C_k} l_i x_i \leq y_k \cdot L_{\max}, \quad k = 1, \dots, K \quad (17.3)$$

$$0 \leq x_i \leq d_i, \text{ inteiro}, \quad i = 1, \dots, m \quad e \quad (17.4)$$

$$y_k \geq 0, \text{ inteiro}, \quad k = 1, \dots, K. \quad (17.5)$$

Com estas variáveis, o objetivo é mais facilmente lido: maximizar o ganho com o preenchimento da mochila, descontado o custo de seus compartimentos.

Observamos que uma possível nova restrição pode ser facilmente adicionada ao modelo:

- Restrição no número de compartimentos que uma mochila pode suportar:

$$\sum_{k=1}^K y_k \leq NC_{\max}, \quad (17.6)$$

onde $NC_{\max}$ é o número máximo de compartimentos admissíveis.

Este, apesar de não representar um modelo preciso para o *Problema da Mochila Compartimentada*, nos fornece um limitante superior para tal, pois todas as soluções factíveis para o problema satisfazem (17.*). Por outro lado, o modelo admite também soluções infactíveis, como poderemos verificar no exemplo a seguir.

Exemplo:

Considere o seguinte conjunto de dados:

Dados da mochila:

- $L = 21$;
- $L_{\min} = 5$ e $L_{\max} = 7$;
- $S = 0$.

Imagine que um possível resultado obtido a partir do modelo fosse o preenchimento de toda a mochila (todos os compartimentos) apenas utilizando uma classe ($k=1$) composta pelos seguintes itens:

Dados da classe $C_1 = \{1, 2, 3\}$:

- $l_1 = 4$;
- $l_2 = 3$;
- $l_3 = 6$.

Poderíamos ter, então, como resultado a seguinte situação:

Vetor solução x – número de itens tipo i na solução:

- $x_1 = 0$;
- $x_2 = 1$;
- $x_3 = 3$.

Vetor solução y – número de compartimentos alocados com a classe k na solução:

- $y_1 = 3$;

Esta solução é factível ao problema modelado em (17.*). Sendo assim, teríamos 3 compartimentos com a classe 1, porém um arranjo dos itens nesses compartimentos, que satisfaça a capacidade máxima admitida, não seria possível (observe que um compartimento não comporta uma unidade do item do tipo 2 e uma unidade do item do tipo 3: $l_2 + l_3 = 9 > L_{max}$). A figura 11 ilustra a situação:

Figura 11: Solução infactível para o problema e admitida por (17.*).

4.2.2 Relaxação de Integralidade

Apesar de (17.*) ser um limitante superior para o modelo (12.*), é ainda de difícil resolução. Uma relaxação de (17.*) nos fornece um outro limitante superior para (12.*).

Relaxando-se a condição de integralidade da variável y_k , faz-se:

$$y_k = \frac{\sum_{i \in C_k} l_i x_i}{L_{max}}, \quad k = 1, \dots, K \quad (18)$$

Substituindo (18) no modelo, temos:

$$\begin{aligned}
 \text{Maximizar} \quad & v_1 x_1 + \dots + v_m x_m - \left(c_1 \frac{\sum_{i \in C_1} l_i x_i}{L_{\max}} + \dots + c_K \frac{\sum_{i \in C_K} l_i x_i}{L_{\max}} \right) \\
 \text{Sujeito a :} \quad & \sum_{i=1}^m l_i x_i + S \left(\frac{\sum_{i \in C_1} l_i x_i}{L_{\max}} + \dots + \frac{\sum_{i \in C_K} l_i x_i}{L_{\max}} \right) \leq L \\
 & 0 \leq x_i \leq d_i, \text{ inteiro, } i = 1, \dots, m.
 \end{aligned}$$

Reescrevendo, temos:

$$\begin{aligned}
 \text{Maximizar} \quad & x_1 \left(v_1 - \frac{c_{k(1)} l_1}{L_{\max}} \right) + \dots + x_m \left(v_m - \frac{c_{k(m)} l_m}{L_{\max}} \right) \\
 \text{Sujeito a :} \quad & x_1 \left(l_1 + \frac{S l_1}{L_{\max}} \right) + \dots + x_m \left(l_m + \frac{S l_m}{L_{\max}} \right) \leq L \\
 & 0 \leq x_i \leq d_i, \text{ inteiro, } i = 1, \dots, m,
 \end{aligned}$$

onde $c_{k(i)} = c_k$, se $i \in C_k$.

Fazendo:

$$\begin{aligned}
 \bar{v}_i &= v_i - \frac{c_{k(i)} l_i}{L_{\max}} \quad e \\
 \bar{l}_i &= l_i + \frac{S l_i}{L_{\max}} \quad \text{para } i = 1, \dots, m.
 \end{aligned}$$

Temos, então:

$$\text{Maximizar} \quad \bar{v}_1 x_1 + \dots + \bar{v}_m x_m \quad (19.1)$$

$$\text{Sujeito a :} \quad \bar{l}_1 x_1 + \dots + \bar{l}_m x_m \leq L \quad (19.2)$$

$$0 \leq x_i \leq d_i, \text{ inteiro, } i = 1, \dots, m \quad (19.3)$$

O modelo (19. *) é o clássico Problema da Mochila Restrito.

Portanto, ao relaxarmos a condição de integralidade da variável y_k , obtemos o clássico Problema da Mochila Restrito. O problema (19. *) fornece um limitante superior para o *Problema da Mochila Compartimentada*, que poderá ser de grande utilidade para um método de enumeração implícita, que entretanto, não será objeto deste estudo.

No capítulo seguinte, apresentaremos três heurísticas para a resolução do *Problema da Mochila Compartimentada*. Além disso, descreveremos algumas características importantes envolvidas no processo de implementação destas heurísticas.

Capítulo 5. Métodos de Resolução e Implementações

Neste capítulo, propomos três heurísticas “gulosas” para a resolução do *Problema da Mochila Compartimentada* onde uma demanda dos itens (d_i , para $i = 1, \dots, m$) deve ser considerada, caracterizando assim, um problema de corte de estoque para mochilas compartimentadas. Faremos uma exposição, em alto nível, dos algoritmos de cada uma das heurísticas propostas e mostraremos alguns detalhes sobre as implementações.

5.1 Heurística de Decomposição

Esta heurística consiste de duas fases: Na primeira fase são resolvidos K Problemas da Mochila, de capacidade L_{max} , um para cada classe de itens, resultando nos respectivos melhores compartimentos associados às classes. Na segunda fase, um problema de programação inteira, é resolvido considerando os compartimentos obtidos na primeira fase como itens, juntamente com os itens livres, para o preenchimento da mochila. Baseado no modelo (12.*), buscaremos uma solução com as seguintes características:

- A função objetivo (12.1) pode ser beneficiada com a construção de V_{jk} grandes e, portanto, para cada classe k , construímos o melhor compartimento ($N_k = 1$, portanto, j seria suprimido) maximizando (12.2) restrito a (12.3), (12.4) e (12.7). A restrição (12.5) é decomposta por classe.
- Com $N_k = 1$ (apenas um compartimento por classe) e α_{ik} já determinado: determina-se L_k (tamanho do compartimento k , veja (12.3), com o índice j suprimido). O valor de β_k é determinado de modo a otimizar (12.1), sujeito às demais restrições do problema (12.*) (veja problema (21.*)). O algoritmo é apresentado a seguir.

Algoritmo:

1. Início;
2. Leitura dos dados do problema;
3. $Plano \leftarrow 1$; “Plano guarda o número do plano que está sendo gerado.”
4. Repita
5. Para $k = 1, \dots, K-1$, tal que $[\sum l_i \cdot d_i (i \in C_k) > L_{min}]$ faça:
6. Selecionar o melhor compartimento para a classe k resolvendo o seguinte Problema da Mochila de capacidade L_{max} :

$$\text{Maximizar} \quad V_k = \sum_{i \in C_k} v_i \alpha_{ik} \quad (20.1)$$

$$\text{Sujeito a :} \quad \sum_{i \in C_k} l_i \alpha_{ik} + S \leq L_{max} \quad (20.2)$$

$$0 \leq \alpha_{ik} \leq d_i \text{ e inteiro, para } i = 1, \dots, m. \quad (20.3)$$

sendo $L_k = \sum_{i \in C_k} l_i \alpha_{ik}$: tamanho do compartimento k .

7. Fim do Para 5;
8. Resolver o seguinte Problema da Mochila envolvendo os compartimentos selecionados e os itens livres:

$$\text{Maximizar} \quad \sum_{k=1}^{K'} (V_k - c_k) \beta_k \quad (21.1)$$

$$\text{Sujeito a :} \quad \sum_{k=1}^{K'} L'_k \beta_k \leq L - S \quad (21.2)$$

$$\alpha_{ik} \beta_k \leq d_i, \quad i \in C_k \quad e \quad k = 1, \dots, K' \quad (21.3)$$

$$\beta_k \geq 0, \text{ inteiro para } k = 1, \dots, K'. \quad (21.4)$$

onde $L'_k = (L_k + S)$ se k representa um compartimento de uma classe,

$L'_k = l_i$ se k representa o item livre i e

K' é o total de compartimentos $(K - 1)$ mais o número total de itens livres.

$$9. d_i \leftarrow d_i - \alpha_{ik} \beta_k;$$

$$10. \text{Plano} \leftarrow \text{Plano} + 1;$$

11. Até que $[d_i = 0 \text{ (para } i = 1, \dots, m)]$ ou $[\sum l_i d_i (i \in C_k) < L_{min}]$ – Fim do Repita 4;

12. Fim do Algoritmo.

5.2 Heurística do Melhor Compartimento

Nesta heurística, como na anterior, o melhor compartimento para cada uma das classes de itens é determinado. Feito isto, esses são comparados aos itens livres e escolhe-se, heurísticamente, o melhor entre todos, sendo então, alocado na mochila. Uma vez feito isto, atualiza-se os dados e as demandas dos itens pertencentes ao compartimento escolhido, ou do item livre, e repete-se o processo até que um plano seja gerado. Neste caso, também temos que compartimento ($N_k = 1$, com j sendo suprimido) e buscaremos uma solução com as mesmas características da Heurística de Decomposição, apenas com uma diferença, o valor de β_k será igual a um para os melhores compartimentos ou itens livres de cada iteração do processo. Com isto estamos evitando o problema (21. *) do algoritmo anterior através de uma solução heurística. O algoritmo é apresentado passo a passo a seguir.

Algoritmo:

1. Início;

2. Leitura dos dados do problema;

3. $\text{Plano} \leftarrow 1$; “Plano guarda o número do plano que está sendo gerado.”

4. Repita

5. $lutil \leftarrow L - S$; “ $lutil$ guarda a capacidade disponível no plano atual.”

6. Repita “Comando de repetição que verifica a criação do plano.”

7. Para $k = 1, \dots, K-1$, tal que $[\sum l_i.d_i (i \in C_k) > Lmin]$ faça:

8. Se $(lutil < Lmax)$ e $(lutil \geq Lmin)$ então

9. Selecionar o melhor compartimento para a classe k resolvendo o seguinte Problema da Mochila de capacidade $lutil$:

$$\text{Maximizar } V_k = \sum_{i \in C_k} v_i \alpha_{ik} \quad (22.1)$$

$$\text{Sujeito a : } \sum_{i \in C_k} l_i \alpha_{ik} + S \leq lutil \quad (22.2)$$

$$0 \leq \alpha_{ik} \leq d_i \text{ e inteiro, para } i = 1, \dots, m. \quad (22.3)$$

10. Senão (do comando condicional Se 8)

11. Selecionar o melhor compartimento para a classe k resolvendo o seguinte Problema da Mochila de capacidade $Lmax$:

$$\text{Maximizar } V_k = \sum_{i \in C_k} v_i \alpha_{ik} \quad (23.1)$$

$$\text{Sujeito a : } \sum_{i \in C_k} l_i \alpha_{ik} + S \leq Lmax \quad (23.2)$$

$$0 \leq \alpha_{ik} \leq d_i \text{ e inteiro, para } i = 1, \dots, m. \quad (23.3)$$

12. Fim do Para 7;

13. Se (houver um compartimento ou item livre com $L'_k \leq lutil$) então

14. Encontrar o primeiro compartimento ou item livre com demanda pendente e com $L'_k \leq lutil$;

15. $lutil \leftarrow lutil - L'_k$; “Alocar o compartimento ou item livre na mochila.”

16. $d_i \leftarrow d_i - \alpha_{ik}$; “ $\alpha_{ik} = 1$ se o item for um item livre”

17. Até que $[d_i = 0 \text{ (para } i = 1, \dots, m)]$ ou $[L'_k > lutil, \text{ para } k = 1, \dots, K']$ – Fim do Repita 6;

18. $Plano \leftarrow Plano + 1$;

19. Até que $[d_i = 0 \text{ (para } i = 1, \dots, m)]$ ou $[\sum l_i.d_i (i \in C_k) < Lmin]$ – Fim do Repita 4;

20. Fim do Algoritmo.

5.3 Heurística dos “z” Melhores Compartimentos

Esta heurística consiste basicamente na resolução de um Problema da Mochila para cada classe, resultando nos respectivos “z” melhores compartimentos associados a cada classe. Note que agora estamos armazenando os “z” melhores compartimentos gerados para cada classe. Estes serão, através da resolução de um problema de programação inteira (via método de enumeração implícita), alocados na mochila juntamente com os itens que não necessitam de compartimentos (*itens livres*). Sendo assim, estamos aumentando consideravelmente o número de possíveis combinações a serem consideradas para o preenchimento da mochila. O algoritmo é apresentado a seguir.

Algoritmo:

1. Início;
2. Leitura dos dados do problema;
3. $Plano \leftarrow 1$; “Plano guarda o número do plano que está sendo gerado.”
4. Repita
5. Para $k = 1, \dots, K-1$, tal que $[\sum l_i d_i (i \in C_k) > L_{min}]$ faça:
6. Selecionar os “z” melhores compartimentos para a classe k resolvendo o seguinte Problema da Mochila de capacidade L_{max} :

$$\text{Maximizar } V_{jk} = \sum_{i \in C_k} v_i \alpha_{ijk} \quad (24.1)$$

$$\text{Sujeito a : } \sum_{i \in C_k} l_i \alpha_{ijk} + S \leq L \max \quad (24.2)$$

$$0 \leq \alpha_{ijk} \leq d_i \text{ e inteiro, para } i = 1, \dots, m \text{ e } j = 1, \dots, z. \quad (24.3)$$

Sendo $V_{1k} \geq V_{2k} \geq \dots \geq V_{zk}$ os "z" melhores valores de (24.1) com suas respectivas soluções α_{ijk} associadas.

7. Fim do Para 5;

8. Resolver o seguinte problema de programação inteira envolvendo os $((K-1)*z)$ compartimentos selecionados e os itens livres:

$$\text{Maximizar } \sum_{k=1}^{K'} \sum_{j=1}^{z'} (V_{jk} - c_k) \beta_{jk} \quad (25.1)$$

$$\text{Sujeito a : } \sum_{k=1}^{K'} \sum_{j=1}^z L'_{jk} \beta_{jk} \leq L - S \quad (25.2)$$

$$\sum_{j=1}^{z'} \alpha_{ijk} \beta_{jk} \leq d_i, \quad i \in C_k \text{ e } k = 1, \dots, K'. \quad (25.3)$$

$$\beta_{jk} \geq 0 \text{ e inteiro, para } k = 1, \dots, K' \text{ e } j = 1, \dots, z. \quad (25.4)$$

onde $L'_{jk} = (L_{jk} + S)$ se k representa um compartimento de uma classe,

$L'_{jk} = l_i$ se k representa um item livre i ,

K' é o total de compartimentos (K) mais o número total de itens livres e

$z' = z$ se k representa um compartimento de uma classe e

$z' = 1$ caso contrário.

$$9. d_i \leftarrow d_i - \alpha_{ijk} \beta_{jk};$$

$$10. \text{Plano} \leftarrow \text{Plano} + 1;$$

11. Até que $[d_i = 0 \text{ (para } i = 1, \dots, m)]$ ou $[\sum l_i d_i (i \in C_k) < L_{min}]$ – Fim do Repita 4;

12. Fim do Algoritmo.

5.4 Programas Desenvolvidos

Nesta seção, apresentamos uma descrição das características dos programas desenvolvidos para cada um dos algoritmos expostos anteriormente.

5.4.1 Objetivos

O problema em estudo é bastante complexo e de difícil resolução. Uma mostra disso é a dificuldade de quantificação dos custos das classes e dos valores associados aos itens envolvidos no problema. Definirlos de forma razoável é, no mínimo, uma tarefa extremamente trabalhosa.

Os programas desenvolvidos têm por objetivo a geração de planos de corte que representem as soluções do *Problema de Múltiplas Mochilas Compartimentadas* idênticas segundo as heurísticas propostas. A geração destes planos está diretamente ligada aos dados dos itens envolvidos e ocorre até que uma demanda, previamente determinada para um conjunto de itens, seja cumprida ou aproximada, de forma que as restrições físicas impostas não possam ser cumpridas. A avaliação dos planos é feita com base na utilização das mochilas; no cumprimento das demandas dos itens envolvidos; assim como no tempo gasto na execução dos programas.

5.4.2 Estruturação dos Programas

As implementações dos programas foram desenvolvidas em linguagem de programação Pascal. Trata-se de uma linguagem bastante simples e de fácil utilização. Para isto, utilizou-se um microcomputador com a seguinte configuração:

- Pentium III 550 MHz;
- 128 Mbytes de memória Ram;

As subrotinas utilizadas nos programas estão divididas em seis "*units*". Cada *unit* é composta por subrotinas com funções semelhantes dentro do objetivo principal dos programas, sendo que uma delas abriga a declaração das variáveis globais comuns às implementações das três heurísticas. São as *units*:

- **Variáveis:** Declaração das variáveis globais;
- **Telib:** Recursos de interface utilizados;

- **Ordena:** Algoritmos de ordenação utilizados;
- **Mochila:** Algoritmos de preenchimento de compartimentos e geração dos planos;
- **Controle:** Subrotinas de atualizações e verificações dos dados do problema;
- **Inout:** Recursos de inicialização, leitura, escrita e armazenagem dos dados.

A *unit Variáveis* é composta pela definição das estruturas usadas para armazenagem dos dados do problema, assim como pelas declarações das variáveis globais utilizadas pelos programas principais das três heurísticas propostas. Numa seção seguinte, mostraremos as estruturas utilizadas.

Dentre as *units* mostradas, a *Telib* é a responsável pela interface apresentada pelo programa. Nela estão subrotinas que criam janelas, molduras e menus na tela, além de controle do tamanho do cursor.

A *unit Ordena* apresenta várias subrotinas de ordenação em vetores de variáveis inteiras e reais. O método de ordenação utilizado foi o *Bubble Sort*. Este é um método simples, baseado em trocas entre os elementos e é bastante utilizado para ordenação em vetores com um número pequeno de variáveis, mostrando resultados razoáveis nestes casos. Este método foi escolhido devido ao fato de estarmos trabalhando com problemas que consideram um número reduzido de itens. Esta situação foi apoiada, principalmente, no fato de termos visitado recentemente uma indústria metalúrgica que apresenta o problema, e constatado que o número de itens envolvidos é realmente reduzido.

A mais importante *unit* dentre todas é a *Mochila*. Nesta, estão presentes todas as subrotinas que envolverão o preenchimento das capacidades dos compartimentos e mochilas em questão. A implementação destas subrotinas envolve o método de enumeração implícita proposto em [Gilmore e Gomory, 1963].

Outra *unit* que merece destaque é a *Controle*. Esta, apresenta uma série de subrotinas que executam as tarefas de cálculos e atualizações nos valores dos dados dos itens envolvidos no problema, dentre eles as demandas. Outras subrotinas presentes nesta *unit* fazem as verificações relativas ao controle em cumprir ou não as demandas dos itens de cada classe, assim como dos itens livres.

Por fim, a *unit Inout* apresenta todas as subrotinas de leitura (via teclado ou arquivo) e escrita na tela dos dados do problema, bem como a armazenagem do conjunto de planos resultantes em um arquivo de saída.

Além destas *units*, temos ainda os três programas principais, cada um representando uma das heurísticas propostas. A figura 11 mostra o relacionamento entre as *units* e os programas principais de uma maneira geral:

Figura 11: Relacionamento entre as *units* e os programas principais.

5.4.3 Estruturas de Armazenamento dos Dados do Problema

Os programas desenvolvidos têm acesso a um arquivo de dados de entrada relativos às características dos itens, das casses e da mochila a ser preenchida. Com base neste conjunto de dados, os programas têm capacidade para determinar todos os dados necessários à execução dos programas, como, por exemplo, os possíveis compartimentos gerados para cada uma das classes.

Para a execução dos programas, foram necessários três tipos de registros (estruturas) diferentes. Numa primeira etapa os dados dos itens são carregados, utilizando-se para isto, o registro “*classe*”. Uma vez carregado esses dados, numa segunda etapa da execução, são gerados os compartimentos para cada classe envolvida. Para o armazenamento dos dados

destes compartimentos, bem como dos dados dos itens livres que estarão concorrendo com estes compartimentos por uma alocação na mochila, foi utilizado um outro registro, o “*compartimento*”. Por fim, temos uma terceira etapa da execução (preenchimento da mochila) onde obtemos os planos finais. Para armazenar estes planos um novo registro foi utilizado, o “*solução final*”.

As tabelas a seguir mostrarão os três registros criados para a execução dos programas com seus respectivos campos de armazenagem dos dados:

Registro *Classe*

<i>Classe</i>	
Número Classe	Número de Itens
Vetor de Itens	
Número	Demanda
Capacidade	Valor

Registro *Compartimento*

<i>Compartimento</i>	
Número Compartimento	
Número Classe	Demanda
Capacidade	Valor
Vetor Ocorrência de Itens	

Registro *Solução Final*

<i>Solução Final</i>
Capacidade Final
Função Objetivo
Vetor Número Compartimento
Vetor Ocorrência Compartimentos

Além desses registros, contamos ainda com algumas variáveis que armazenam os demais dados relevantes à solução do problema. Dentre elas, podemos citar as variáveis que armazenam os dados relativos aos limitantes físicos dos compartimentos, às perdas por compartimentos e à capacidade da mochila.

5.4.4 Fluxo das Informações nos Programas

Os programas desenvolvidos passaram por um processo de análise estruturada e foram projetados segundo uma abordagem orientada ao fluxo de dados. Tal abordagem é útil quando as informações são processadas sequencialmente e, à medida que se movimentam pelo programa, novos dados vão sendo obtidos até que se chegue aos resultados esperados [Pressman, 1995]. Para melhor entendermos o fluxo das informações dentro dos programas desenvolvidos, *Diagramas de Fluxo de Dados (DFD)* foram elaborados. Inicialmente iremos expor um *DFD de nível 1*, visto na figura 12, comum aos três programas, e desenvolvido segundo as heurísticas propostas.

Figura 12: DFD de nível 1 para os programas desenvolvidos.

À medida que os dados fluem pelo programa, novas informações são geradas. Nesse ponto, cada uma das heurísticas propostas difere no que diz respeito ao fluxo das informações. Estas diferenças são pequenas, porém, são importantes para o desenvolvimento do programa até que os resultados sejam obtidos. Para melhor representar esta sequência de passos, o *DFD de nível 1* anterior foi refinado para cada um dos três

programas desenvolvidos até que fossem obtidos detalhes suficientes para um melhor entendimento do fluxo de informação no programa. A seguir, mostraremos os *Diagramas de Fluxo de Dados* obtidos para cada uma das heurísticas propostas.

Figura 13: DFD para a Heurística de Decomposição.

Figura 14: DFD para a Heurística do Melhor Compartimento.

Figura 15: DFD para a Heurística dos “z” Melhores Compartimentos.

Com o desenvolvimento dos diagramas anteriores, uma visualização dos programas se tornou bastante fortalecida facilitando inclusive o desenvolvimento dos mesmos.

5.4.5 Atualização dos Dados

Um ponto importante a ser discutido em relação aos programas desenvolvidos, diz respeito à atualização das demandas dos itens a cada iteração.

5.4.5.1 Atualização das Demandas

À medida que os planos (ou alocação dos compartimentos, no caso da Heurística do Melhor Compartimento) vão sendo gerados, uma atualização nas demandas pendentes dos itens torna-se necessária. No início da execução do programa, cada item tem uma demanda d_i . Com a geração dos planos, esta demanda é atualizada. Temos então:

$$d_i = d_i - dcp_i, \quad (26)$$

onde dcp_i é a demanda do item i cumprida pela geração do último plano de corte (ou compartimento alocado – Heurística do Melhor Compartimento).

5.4.6 Cálculo da Função Objetivo e Aproveitamento da Mochila

De acordo com o modelo proposto, temos uma função objetivo a ser maximizada segundo a seguinte fórmula:

$$\text{Maximizar} \quad \sum_{k=1}^K \sum_{j=1}^{N_k} (V_{jk} - c_k) \beta_{jk} \quad (27)$$

Temos então um valor c_k associado a cada classe que representa um compartimento ou item livre. No caso onde k representa um compartimento associado a uma classe, temos que c_k é o custo de alocação de um compartimento com a classe k , como definido anteriormente. Porém, quando c_k representa um item livre, este custo c_k tem valor zero ($c_k=0$), pois estamos considerando que não existe um custo associado à alocação de um item livre na mochila.

Uma outra característica importante diz respeito ao aproveitamento da mochila. Segundo observações na indústria, se o plano gerado tem uma sobra maior que $Lmin$, esta não é considerada perda, pois pode ser reaproveitada como uma mochila parcial.

Apresentaremos no capítulo seguinte alguns resultados computacionais obtidos a partir das implementações apresentadas neste capítulo.

Capítulo 6. Resultados Computacionais

Neste capítulo, apresentamos alguns resultados obtidos a partir da implementação das três heurísticas propostas no capítulo anterior.

6.1 Considerações Iniciais

Os resultados foram obtidos considerando apenas um tipo de mochila (cujos dados serão expostos posteriormente) para os três exemplos a serem mostrados. Cada exemplo envolve um conjunto de classes e itens diferentes (com seus respectivos dados) e foi executado para cada uma das heurísticas (Decomposição, Melhor Compartimento e “z” Melhores Compartimentos). No caso da heurística dos “z” Melhores Compartimentos padronizamos a execução para $z = 2$ e $z = 3$, ou seja, consideramos os dois melhores compartimentos ($z = 2$) e os três melhores compartimentos ($z = 3$) para cada classe. Por fim, mostramos as estatísticas finais da execução envolvendo o tempo de execução, a perda média por plano de corte, o valor total da função objetivo (soma das funções objetivo de cada plano) e um gráfico mostrando o número de planos gerados segundo um percentual de perda.

É importante frisar que os exemplos a serem mostrados possuem alguns dados baseados em situações reais verificadas em uma indústria metalúrgica que trata do corte de aço, com características semelhantes ao problema estudado.

Os resultados obtidos serão expostos em tabelas. Para um melhor entendimento dessas tabelas consideramos a seguinte notação:

Notação:

- **Itens:** Número de identificação dos itens considerados;
- **Peso:** Peso dos itens a serem alocados na mochila;
- **Valor:** Valor associado ao item;
- **Demanda (d_i):** Número de itens (para cada tipo) que estão sendo requisitados.;
- **Classe Livre:** Conjunto de itens livres;
- **Custo:** Custo da classe considerada.

Notação para a tabela de resultados:

- **Compartimentos:** linha que mostra o número de compartimentos de cada classe alocados na mochila segundo o plano X (da linha seguinte);
- **Plano X :** linha que mostra a quantidade de cada item presente no plano de corte;
- **Utilização:** peso total alocado por plano de corte;
- **FO.:** Valor da função objetivo alcançada para o plano de corte;
- **Perda:** Percentual de perda (em peso) do plano de corte gerado.
- **$d[i]$:** Demanda inicial do item i ;
- **$dfinal[i]$:** Demanda final não alocada ou alocada em excesso (se $dfinal_i < 0$).

6.2 Dados da Mochila

Para todos os exemplos a serem mostrados a seguir, os seguintes dados relativos à mochila e seus compartimentos são considerados:

Dados do Problema:

Capacidade da Mochila	$L = 1200$
Capacidade Mínima do Compartimento	$L_{min} = 154$
Capacidade Máxima do Compartimento	$L_{max} = 456$
Perda pela Alocação de um Compartimento	$S = 12$

6.3 Tabelas de Resultados

Exemplo 1:

O exemplo 1 tem seus dados gerados aleatoriamente, como os demais, porém, para este exemplo, os valores associados aos itens (“*valor*” ou v_i) não obedecem a uma relação linear com os pesos dos itens. Isso será verificado a partir do exemplo 2.

Dados das Classes de Itens:

Classe 1	Custo = 1.50		Classe 2	Custo = 0.50			Classe 3	Custo = 2.50		
Itens	1	2	3	4	5	6	7	8	9	10
Peso	55	58	57	54	56	50	52	54	55	58
Valor	28.00	30.00	7.00	26.00	30.00	27.00	8.00	29.00	10.00	35.00
Demanda	15	16	12	14	20	13	14	10	12	11

Dados dos Itens Livres:

Classe Livre		Custo = 0.00			
11	12	13	14	15	16
53	56	55	58	60	51
23.00	12.00	21.00	9.00	26.00	33.00
21	23	26	21	17	15

Resultados Obtidos:

I - Heurística de Decomposição

Tabela 3: Resultados computacionais obtidos a partir da Heurística de Decomposição para o exemplo 1.

Classes	1		2				3				Livres								
Itens	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16			
d[i]	15	16	12	14	20	13	14	10	12	11	21	23	26	21	17	15			
Compartimentos	0		0				1				1						Utilização	FO.	Perda
Plano 1	0	0	0	0	0	0	0	5	0	3	0	0	0	0	0	14	1182	709,50	18
Compartimentos	0		1				1				1						Utilização	FO.	Perda
Plano 2	0	0	0	0	7	1	0	5	0	3	2	0	0	0	2	1	1199	615,00	1
Compartimentos	1		1				0				1						Utilização	FO.	Perda
Plano 3	7	1	0	0	7	1	0	0	0	0	3	0	0	0	2	0	1200	582,00	0
Compartimentos	1		1				0				1						Utilização	FO.	Perda
Plano 4	7	1	0	0	6	2	0	0	0	0	3	0	0	0	2	0	1194	579,00	6
Compartimentos	1		1				0				1						Utilização	FO.	Perda
Plano 5	0	7	0	0	0	8	0	0	0	0	1	0	0	0	5	0	1195	577,00	5
Compartimentos	1		1				0				1						Utilização	FO.	Perda
Plano 6	0	7	0	7	0	1	0	0	0	0	5	0	0	0	1	0	1195	558,00	5
Compartimentos	0		1				1				1						Utilização	FO.	Perda
Plano 7	0	0	1	7	0	0	0	0	2	5	5	0	0	0	1	0	1196	522,00	4
Compartimentos	0		0				0				1						Utilização	FO.	Perda
Plano 8	0	0	0	0	0	0	0	0	0	0	2	0	15	0	4	0	1183	465,00	17
Compartimentos	0		0				0				1						Utilização	FO.	Perda
Plano 9	0	0	0	0	0	0	0	0	0	0	0	10	11	0	0	0	1177	351,00	23
Compartimentos	0		0				1				1						Utilização	FO.	Perda
Plano 10	0	0	0	0	0	0	0	0	8	0	0	13	0	0	0	0	1192	233,50	8
Compartimentos	0		0				1				1						Utilização	FO.	Perda
Plano 11	0	0	0	0	0	0	6	0	2	0	0	0	0	13	0	0	1200	182,50	0
Compartimentos	0		1				1				1						Utilização	FO.	Perda
Plano 12	0	0	7	0	0	0	8	0	0	0	0	0	0	6	0	0	1199	164,00	1
Compartimentos	0		1				0				1						Utilização	FO.	Perda
Plano 13	0	0	4	0	0	0	0	0	0	0	0	0	0	2	0	0	368	45,50	Parcial ¹
dfinal[i]	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			

¹ Uma Mochila Parcial foi gerada, não caracterizando perda.

Estatísticas:

Número de Planos	13
Tempo de Execução	0' 1"81
Perda Média por Plano	0,611%
Função Objetivo Total	5584,00

Gráfico de Aproveitamento dos Planos de Corte:

II- Heurística do Melhor Compartimento

Tabela 4: Resultados computacionais obtidos a partir da Heurística do Melhor Compartimento para o exemplo 1.

Classes	1		2				3				Livres								
Itens	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16			
d[i]	15	16	12	14	20	13	14	10	12	11	21	23	26	21	17	15			
Compartimentos	0		0				1				1						Utilização	FO.	Perda
Plano 1	0	0	0	0	0	0	0	0	0	7	0	0	0	0	0	15	1195	735,00	5
Compartimentos	0		1				2				0						Utilização	FO.	Perda
Plano 2	0	0	0	0	7	1	0	8	0	4	0	0	0	0	0	0	1154	598,00	46
Compartimentos	0		3				0				0						Utilização	FO.	Perda
Plano 3	0	0	0	0	13	8	0	0	0	0	0	0	0	0	0	0	1176	603,00	24
Compartimentos	2		1				0				0						Utilização	FO.	Perda
Plano 4	7	6	0	4	0	4	0	0	0	0	0	0	0	0	0	0	1197	581,00	3
Compartimentos	3		0				0				1						Utilização	FO.	Perda
Plano 5	8	10	0	0	0	0	0	0	0	0	2	0	0	0	0	0	1174	561,00	26
Compartimentos	0		1				1				1						Utilização	FO.	Perda
Plano 6	0	0	0	8	0	0	0	2	0	0	11	0	0	0	0	0	1159	513,00	41
Compartimentos	0		0				0				1						Utilização	FO.	Perda
Plano 7	0	0	0	0	0	0	0	0	0	0	8	0	0	0	12	0	1156	496,00	44
Compartimentos	0		1				0										Utilização	FO.	Perda
Plano 8	0	0	0	2	0	0	0	0	0	0	0	0	13	0	5	0	1147	454,00	53
Compartimentos	0		0				0				1						Utilização	FO.	Perda
Plano 9	0	0	0	0	0	0	0	0	0	0	0	8	13	0	0	0	1175	369,00	25
Compartimentos	0		0				1				1						Utilização	FO.	Perda
Plano 10	0	0	0	0	0	0	0	0	6	0	0	15	0	0	0	0	1194	235,00	6
Compartimentos	0		0				1				1						Utilização	FO.	Perda
Plano 11	0	0	0	0	0	0	2	0	6	0	0	0	0	12	0	0	1154	179,00	46
Compartimentos	0		0				2				1						Utilização	FO.	Perda
Plano 12	0	0	0	0	0	0	12	0	0	0	0	0	0	9	0	0	1182	167,00	18
Compartimentos	0		1				0				0						Utilização	FO.	Perda
Plano 13	0	0	12	0	0	0	0	0	0	0	0	0	0	0	0	0	720	82,00	Parcial
dfinal[i]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			

Estatísticas:

Número de Planos	13
Tempo de Execução	0'0"71
Perda Média por Plano	2,340%
Função Objetivo Total	5573,00

Gráfico de Aproveitamento dos Planos de Corte:

III - Heurística dos “z” Melhores Compartimentos ($z = 2$)

Tabela 5: Resultados computacionais obtidos a partir da Heurística dos “z” Melhores Compartimentos ($z=2$) para o exemplo 1.

Classes	1		2				3				Livres								
Itens	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16			
d[i]	15	16	12	14	20	13	14	10	12	11	21	23	26	21	17	15			
Compartimentos	0		0				1				1						Utilização	FO.	Perda
Plano 1	0	0	0	0	0	0	0	0	0	7	0	0	0	0	0	15	1195	737,50	5
Compartimentos	0		1				1				1						Utilização	FO.	Perda
Plano 2	0	0	0	0	6	2	0	5	0	3	3	0	0	0	2	0	1195	602,00	5
Compartimentos	0		2				0				1						Utilização	FO.	Perda
Plano 3	0	0	0	0	13	3	0	0	0	0	2	0	0	0	3	0	1200	594,00	0
Compartimentos	1		1				0				1						Utilização	FO.	Perda
Plano 4	7	1	0	0	1	7	0	0	0	0	0	0	0	0	5	0	1185	573,00	15
Compartimentos	1		1				0				1						Utilização	FO.	Perda
Plano 5	7	1	0	7	0	1	0	0	0	0	1	0	0	0	4	0	1200	560,00	0
Compartimentos	1		0				0				1						Utilização	FO.	Perda
Plano 6	0	14	0	0	0	0	0	0	0	0	3	0	0	0	3	0	1187	564,00	13
Compartimentos	0		1				1				1						Utilização	FO.	Perda
Plano 7	0	0	0	7	0	0	0	5	2	1	6	0	0	0	0	0	1170	517,00	30
Compartimentos	0		0				0				1						Utilização	FO.	Perda
Plano 8	0	0	0	0	0	0	0	0	0	0	6	0	15	0	0	0	1155	453,00	45
Compartimentos	0		0				0				1						Utilização	FO.	Perda
Plano 9	0	0	0	0	0	0	0	0	0	0	0	10	11	0	0	0	1177	351,00	23
Compartimentos	0		0				1				1						Utilização	FO.	Perda
Plano 10	0	0	0	0	0	0	0	0	8	0	0	13	0	0	0	0	1192	233,50	8
Compartimentos	0		0				1				1						Utilização	FO.	Perda
Plano 11	0	0	0	0	0	0	6	0	2	0	0	0	0	13	0	0	1200	182,50	0
Compartimentos	0		1				0				1						Utilização	FO.	Perda
Plano 12	0	0	12	0	0	0	0	0	0	0	0	0	0	8	0	0	1184	155,00	16
Compartimentos	0		0				1				1						Utilização	FO.	Perda
Plano 13	0	0	0	0	0	0	8	0	0	0	0	0	0	0	0	0	440	61,50	Parcial
dfinal[i]	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			

Estatísticas:

Número de Planos	13
Tempo de Execução	0' 01"97
Perda Média por Plano	1,111%
Função Objetivo Total	5584,00

Gráfico de Aproveitamento dos Planos de Corte:

IV - Heurística dos “z” Melhores Compartimentos ($z = 3$)

Tabela 6: Resultados computacionais obtidos a partir da Heurística dos “z” Melhores Compartimentos (z=3) para o exemplo 1.

Classes	1		2				3				Livres								
Itens	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16			
d[i]	15	16	12	14	20	13	14	10	12	11	21	23	26	21	17	15			
Compartimentos	0		0				1				1						Utilização	FO.	Perda
Plano 1	0	0	0	0	0	0	0	0	0	7	0	0	0	0	0	15	1195	737,50	5
Compartimentos	0		1				1				1						Utilização	FO.	Perda
Plano 2	0	0	0	0	6	2	0	5	0	3	3	0	0	0	2	0	1195	602,00	5
Compartimentos	0		2				0				1						Utilização	FO.	Perda
Plano 3	0	0	0	0	13	3	0	0	0	0	2	0	0	0	3	0	1200	594,00	0
Compartimentos	1		1				0				1						Utilização	FO.	Perda
Plano 4	0	7	0	0	1	7	0	0	0	0	2	0	0	0	4	0	1194	577,00	6
Compartimentos	2		0				0				1						Utilização	FO.	Perda
Plano 5	8	7	0	0	0	0	0	0	0	0	6	0	0	0	0	0	1200	569,00	0
Compartimentos	1		0				0				1						Utilização	FO.	Perda
Plano 6	7	1	0	7	0	1	0	0	0	0	1	0	0	0	4	0	1200	560,00	0
Compartimentos	0		1				1				1						Utilização	FO.	Perda
Plano 7	0	0	0	7	0	0	0	5	2	1	2	0	0	0	4	0	1198	529,00	2
Compartimentos	0		0				0				1						Utilização	FO.	Perda
Plano 8	0	0	0	0	0	0	0	0	0	0	5	0	16	0	0	0	1157	451,00	43
Compartimentos	0		0				0				1						Utilização	FO.	Perda
Plano 9	0	0	0	0	0	0	0	0	0	0	0	11	10	0	0	0	1178	342,00	22
Compartimentos	0		0				1				1						Utilização	FO.	Perda
Plano 10	0	0	0	0	0	0	0	0	8	0	0	12	0	1	0	0	1194	230,50	6
Compartimentos	0		0				1				1						Utilização	FO.	Perda
Plano 11	0	0	0	0	0	0	6	0	2	0	0	0	0	13	0	0	1200	182,50	0
Compartimentos	0		1				1				1						Utilização	FO.	Perda
Plano 12	0	0	6	0	0	0	8	0	0	0	0	0	0	7	0	0	1200	166,00	0
Compartimentos	0		1				0				0						Utilização	FO.	Perda
Plano 13	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	366	41,50	Parcial
dfinal[i]	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0			

Estatísticas:

Número de Planos	13
Tempo de Execução	0' 02"14
Perda Média por Plano	0,618%
Função Objetivo Total	5582,00

Gráfico de Aproveitamento dos Planos de Corte:

Exemplo 2:

Nesse exemplo, assim como no exemplo 3, os valores relativos aos pesos (l_i), às demandas (d_i), e aos custos das classes (c_k) foram gerados aleatoriamente dentro de um determinado intervalo. Temos:

$$1 \leq c_k \leq 2, \text{ para } k = 1, \dots, K \quad (28)$$

$$36 \leq l_i \leq 105, \text{ para } i = 1, \dots, m \quad (29)$$

$$1 \leq d_i \leq 50, \text{ para } i = 1, \dots, m. \quad (30)$$

O intervalo para a geração dos pesos (l_i) foi baseado em valores encontrados na prática enquanto os demais foram estabelecidos procurando evitar demandas muito grandes, o que foge a realidade do problema.

Além disso, os valores associados aos itens (v_i) foram gerados obedecendo a seguinte relação:

$$v_i = \alpha_i \cdot l_i, \text{ para } i = 1, \dots, m \quad (31)$$

onde $0.08 \leq \alpha_i \leq 0.12$.

Dados das Classes de Itens:

Classe 1 Custo = 1.42			
Itens	1	2	3
Peso	62	86	78
Valor	6.44	8.96	8.10
Demanda	22	14	21

Classe 2 Custo = 1.89			
4	5	6	7
51	75	85	88
5.35	7.98	8.76	9.06
12	7	9	16

Classe 3 Custo = 1.98			
Itens	8	9	10
Peso	37	55	56
Valor	3.94	5.78	5.91
Demanda	15	10	22

Classe 4 Custo = 1.77			
11	12	13	14
67	68	90	78
6.93	7.01	9.81	8.19
11	9	14	18

Dados dos Itens Livres:

Classe Livre		Custo = 0.00					
15	16	17	18	19	20	21	22
39	76	40	70	63	45	42	51
3.76	7.44	3.78	6.55	6.01	4.22	4.01	5.82
38	41	38	36	45	34	31	46

Resultados Obtidos:

I - Heurística de Decomposição

Tabela 7: Resultados computacionais obtidos a partir da Heurística de Decomposição para o exemplo 2.

Classes	1			2				3			4				Livres												
Itens	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22					
d[i]	22	14	21	12	7	9	16	15	10	22	11	9	14	18	38	41	38	36	45	34	31	46					
Compartimentos	0			0				0			0				1										Utilização	FO.	Perda
Plano 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	23	1185	1168,86	15		
Compartimentos	0			0				0			0				1										Utilização	FO.	Perda
Plano 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	23	1185	1168,86	15		
Compartimentos	0			0				0			2				1										Utilização	FO.	Perda
Plano 3	0	0	0	0	0	0	0	0	0	0	0	0	8	2	2	0	0	0	0	0	5	0	1200	118,89	0		
Compartimentos	0			0				1			1				1										Utilização	FO.	Perda
Plano 4	0	0	0	0	0	0	0	12	0	0	0	0	4	1	0	2	0	0	0	1	2	0	1199	118,08	1		
Compartimentos	1			0				1			0				1										Utilização	FO.	Perda
Plano 5	3	3	0	0	0	0	0	3	3	3	0	0	0	0	0	2	1	0	0	0	2	0	1200	116,37	0		
Compartimentos	2			0				0			0				1										Utilização	FO.	Perda
Plano 6	6	6	0	0	0	0	0	0	0	0	0	0	0	0	0	2	1	0	0	0	2	0	1200	116,24	0		
Compartimentos	1			0				1			0				1										Utilização	FO.	Perda
Plano 7	3	3	0	0	0	0	0	0	4	4	0	0	0	0	0	2	1	0	0	0	2	0	1200	116,24	0		
Compartimentos	1			1				0			0				1										Utilização	FO.	Perda
Plano 8	2	1	3	4	2	0	1	0	0	0	0	0	0	0	0	2	0	0	0	0	3	0	1200	116,16	0		
Compartimentos	1			1				0			0				1										Utilização	FO.	Perda
Plano 9	2	1	3	4	2	0	1	0	0	0	0	0	0	0	0	2	0	0	0	0	3	0	1200	116,16	0		
Compartimentos	0			1				0			0				1										Utilização	FO.	Perda
Plano 10	0	0	0	4	2	0	1	0	0	0	0	0	0	0	0	8	0	0	0	0	3	0	1200	116,08	0		
Compartimentos	2			0				0			0				1										Utilização	FO.	Perda
Plano 11	4	0	8	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	1	0	0	0	1199	116,05	1		
Compartimentos	0			0				2			0				1										Utilização	FO.	Perda
Plano 12	0	0	0	0	0	0	0	0	0	14	0	0	0	0	0	5	0	0	0	0	0	0	1200	115,98	0		

Continua na próxima página.

[illegible]

Estatísticas:

Número de Planos	26
Tempo de Execução	0'17"52
Perda Média por Plano	0,170%
Função Objetivo Total	5090,08

Gráfico de Aproveitamento dos Planos de Corte:

II- Heurística do Melhor Compartimento

Tabela 8: Resultados computacionais obtidos a partir da Heurística do Melhor Compartimento para o exemplo 2.

Classes	1			2				3			4				Livres												
Itens	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22					
d[i]	22	14	21	12	7	9	16	15	10	22	11	9	14	18	38	41	38	36	45	34	31	46					
Compartimentos	0			0				0			0				1										Utilização	FO.	Perda
Plano 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	23	1185	1168,86	15		
Compartimentos	0			0				0			0				1										Utilização	FO.	Perda
Plano 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	23	1185	1168,86	15		
Compartimentos	0			0				0			3				0										Utilização	FO.	Perda
Plano 3	0	0	0	0	0	0	0	0	0	0	0	0	11	2	0	0	0	0	0	0	0	0	1194	113,67	6		
Compartimentos	0			0				2			0				1										Utilização	FO.	Perda
Plano 4	0	0	0	0	0	0	0	15	3	3	0	0	0	0	1	3	0	0	0	0	0	0	1191	112,33	9		
Compartimentos	2			0				0			0				1										Utilização	FO.	Perda
Plano 5	6	6	0	0	0	0	0	0	0	0	0	0	0	0	1	3	0	0	0	0	0	0	1191	112,80	9		
Compartimentos	2			0				0			0				1										Utilização	FO.	Perda
Plano 6	6	6	0	0	0	0	0	0	0	0	0	0	0	0	1	3	0	0	0	0	0	0	1200	116,24	0		
Compartimentos	0			1				1			0				1										Utilização	FO.	Perda
Plano 7	0	0	0	4	2	0	1	0	4	4	0	0	0	0	1	3	0	0	0	0	0	0	1189	111,52	11		
Compartimentos	0			2				0			0				1										Utilização	FO.	Perda
Plano 8	0	0	0	8	4	0	2	0	0	0	0	0	0	0	1	3	0	0	0	0	0	0	1187	111,36	13		
Compartimentos	2			0				0			0				1										Utilização	FO.	Perda
Plano 9	4	2	6	0	0	0	0	0	0	0	0	0	0	0	1	3	0	0	0	0	0	0	1191	112,68	9		
Compartimentos	0			0				0			2				1										Utilização	FO.	Perda
Plano 10	0	0	0	0	0	0	0	0	0	0	0	8	2	2	1	3	0	0	0	0	0	0	1183	111,08	17		
Compartimentos	1			0				0			1				1										Utilização	FO.	Perda
Plano 11	2	0	4	0	0	0	0	0	0	0	3	1	1	1	1	3	0	0	0	0	0	0	1176	110,78	24		
Compartimentos	2			0				0			0				1										Utilização	FO.	Perda
Plano 12	4	0	8	0	0	0	0	0	0	0	0	0	0	0	1	3	0	0	0	0	0	0	1175	110,96	25		
Compartimentos	0			0				0			0				1										Utilização	FO.	Perda
Plano 13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	14	0	0	0	0	0	0	1193	115,44	7		

Continua na próxima página.

[illegible]

Estatísticas:

Número de Planos	27
Tempo de Execução	0'02"25
Perda Média por Plano	1,423%
Função Objetivo Total	5041,04

Gráfico de Aproveitamento dos Planos de Corte:

III - Heurística dos “z” Melhores Compartimentos ($z = 2$)

Tabela 9: Resultados computacionais obtidos a partir da Heurística dos “z” Melhores Compartimentos ($z=2$) para o exemplo 2.

Classes	1			2				3			4				Livres												
Itens	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22					
d[i]	22	14	21	12	7	9	16	15	10	22	11	9	14	18	38	41	38	36	45	34	31	46					
Compartimentos	0			0				0			0				1										Utilização	FO.	Perda
Plano 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	23	1185	1168,86	15		
Compartimentos	0			0				0			0				1										Utilização	FO.	Perda
Plano 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	23	1185	1168,86	15		
Compartimentos	0			0				0			2				1										Utilização	FO.	Perda
Plano 3	0	0	0	0	0	0	0	0	0	0	0	1	8	1	0	3	0	1	0	0	0	0	1200	119,01	0		
Compartimentos	0			0				1			1				1										Utilização	FO.	Perda
Plano 4	0	0	0	0	0	0	0	12	0	0	0	0	4	1	0	2	0	0	0	1	2	0	1199	118,08	1		
Compartimentos	0			1				1			0				1										Utilização	FO.	Perda
Plano 5	0	0	0	1	4	0	1	3	3	3	0	0	0	0	0	2	0	0	0	1	2	0	1200	116,47	0		
Compartimentos	2			0				0			0				1										Utilização	FO.	Perda
Plano 6	6	6	0	0	0	0	0	0	0	0	0	0	0	0	0	2	1	0	0	0	2	0	1200	116,24	0		
Compartimentos	2			0				0			0				1										Utilização	FO.	Perda
Plano 7	6	6	0	0	0	0	0	0	0	0	0	0	0	0	0	2	1	0	0	0	2	0	1200	116,24	0		
Compartimentos	0			1				1			0				1										Utilização	FO.	Perda
Plano 8	0	0	0	4	2	0	1	0	4	4	0	0	0	0	0	2	0	0	0	0	3	0	1200	116,22	0		
Compartimentos	2			0				0			0				1										Utilização	FO.	Perda
Plano 9	4	2	6	0	0	0	0	0	0	0	0	0	0	0	0	2	1	0	0	0	2	0	1200	116,12	0		
Compartimentos	1			0				0			1				1										Utilização	FO.	Perda
Plano 10	3	0	3	0	0	0	0	0	0	0	0	4	1	1	0	4	0	0	0	0	0	0	1200	116,23	0		
Compartimentos	2			0				0			0				1										Utilização	FO.	Perda
Plano 11	4	0	8	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	1	0	0	0	1199	116,05	1		
Compartimentos	0			0				2			0				1										Utilização	FO.	Perda
Plano 12	0	0	0	0	0	0	0	0	0	14	0	0	0	0	0	5	0	0	0	0	0	0	1200	115,98	0		
Compartimentos	0			0				0			0				1										Utilização	FO.	Perda
Plano 13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	14	1	0	0	0	2	0	1200	115,96	0		

Continua na próxima página.

[illegible]

Estatísticas:

Número de Planos	26
Tempo de Execução	0'18"50
Perda Média por Plano	0,176%
Função Objetivo Total	5089,92

Gráfico de Aproveitamento dos Planos de Corte:

IV - Heurística dos “z” Melhores Compartimentos ($z = 3$)

Tabela 10: Resultados computacionais obtidos a partir da Heurística dos “z” Melhores Compartimentos ($z=3$) para o exemplo 2.

Classes	1			2				3			4				Livres												
Itens	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22					
d[i]	22	14	21	12	7	9	16	15	10	22	11	9	14	18	38	41	38	36	45	34	31	46					
Compartimentos	0			0				0			0				1										Utilização	FO.	Perda
Plano 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	23	1185	1168,86	15		
Compartimentos	0			0				0			0				1										Utilização	FO.	Perda
Plano 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	23	1185	1168,86	15		
Compartimentos	0			0				0			2				1										Utilização	FO.	Perda
Plano 3	0	0	0	0	0	0	0	0	0	0	0	0	8	2	0	3	0	1	0	0	0	0	1200	119,01	0		
Compartimentos	0			0				1			1				1										Utilização	FO.	Perda
Plano 4	0	0	0	0	0	0	0	12	0	0	0	0	4	1	0	2	0	0	0	1	2	0	1199	118,08	1		
Compartimentos	0			1				1			0				1										Utilização	FO.	Perda
Plano 5	0	0	0	1	4	0	1	3	3	3	0	0	0	0	0	2	0	0	0	1	2	0	1200	116,47	0		
Compartimentos	2			0				0			0				1										Utilização	FO.	Perda
Plano 6	6	6	0	0	0	0	0	0	0	0	0	0	0	0	0	2	1	0	0	0	2	0	1200	116,24	0		
Compartimentos	1			0				0			0				1										Utilização	FO.	Perda
Plano 7	6	6	0	0	0	0	0	0	0	0	0	0	0	0	0	2	1	0	0	0	2	0	1200	116,24	0		
Compartimentos	0			1				1			0				1										Utilização	FO.	Perda
Plano 8	0	0	0	4	2	0	1	0	4	4	0	0	0	0	0	2	0	0	0	0	3	0	1200	116,22	0		
Compartimentos	2			0				0			0				1										Utilização	FO.	Perda
Plano 9	4	2	6	0	0	0	0	0	0	0	0	0	0	0	0	2	1	0	0	0	2	0	1200	116,12	0		
Compartimentos	1			0				0			1				1										Utilização	FO.	Perda
Plano 10	3	0	3	0	0	0	0	0	0	0	0	4	1	1	0	4	0	0	0	0	0	0	1200	116,23	0		
Compartimentos	1			0				0			1				1										Utilização	FO.	Perda
Plano 11	3	0	3	0	0	0	0	0	0	0	0	4	1	1	0	4	0	0	0	0	0	0	1200	116,23	0		
Compartimentos	0			0				1			0				1										Utilização	FO.	Perda
Plano 12	0	0	0	0	0	0	0	0	0	14	0	0	0	0	0	5	0	0	0	0	0	0	1200	115,98	0		
Compartimentos	0			1				0			0				1										Utilização	FO.	Perda
Plano 13	0	0	0	7	0	1	0	0	0	0	0	0	0	0	0	8	0	0	0	0	3	0	1200	115,87	0		

Continua na próxima página.

[illegible]

Estatísticas:

Número de Planos	26
Tempo de Execução	0'30"09
Perda Média por Plano	0,385%
Função Objetivo Total	5083,15

Gráfico de Aproveitamento dos Planos de Corte:

Exemplo 3:

Dados das Classes de Itens:

Classe 1 Custo = 1.95					
Itens	1	2	3	4	5
Peso	101	80	88	105	36
Valor	10.40	8.64	9.59	11.45	3.96
Demanda	3	1	2	2	1

Classe 2 Custo = 1.78				
6	7	8	9	10
82	82	64	36	62
8.20	8.77	6.59	3.60	6.20
11	13	11	8	8

Classe 3 Custo = 1.21				
Itens	11	12	13	14
Peso	71	87	76	87
Valor	7.10	9.48	7.30	8.00
Demanda	9	15	17	17

Classe 4 Custo = 1.24			
15	16	17	18
101	52	56	82
10.30	4.73	5.21	7.79
17	5	13	9

Classe 5 Custo = 1.93				
Itens	19	20	21	22
Peso	49	42	94	96
Valor	5.34	3.95	9.96	10.37
Demanda	3	9	6	18

Dados dos Itens Livres:

Classe Livre Custo = 0.00				
23	24	25	26	27
44	70	80	79	86
4.12	6.44	7.44	7.53	8.29
25	23	29	30	22

Resultados Obtidos:

I - Heurística de Decomposição

Tabela 11: Resultados computacionais obtidos a partir da Heurística de Decomposição para o exemplo 3.

Classes	1					2					3					4				5				Livres							
Itens	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27				
d[i]	3	1	2	2	1	11	13	11	8	8	9	15	17	17	17	5	13	9	3	9	6	18	25	23	29	30	22				
Compartimentos	0					0					2					0				0				1					Utilização	FO.	Perda
Plano 1	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	1	0	0	2	1	1194	119,85	6	
Compartimentos	0					0					1					0				1				1					Utilização	FO.	Perda
Plano 2	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	3	0	0	3	1	0	0	2	1	1194	118,86	6	
Compartimentos	1					0					0					0				1				1					Utilização	FO.	Perda
Plano 3	1	0	1	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	4	1	0	0	0	3	1199	117,39	1	
Compartimentos	0					0					0					0				2				1					Utilização	FO.	Perda
Plano 4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	8	0	2	0	0	2	1200	116,46	0	
Compartimentos	0					1					0					0				1				1					Utilização	FO.	Perda
Plano 5	0	0	0	0	0	0	3	1	2	1	0	0	0	0	0	0	0	0	0	1	1	3	1	0	0	1	2	1199	115,84	1	
Compartimentos	0					1					0					0				1				1					Utilização	FO.	Perda
Plano 6	0	0	0	0	0	0	3	1	2	1	0	0	0	0	0	0	0	0	0	1	4	0	1	0	0	0	3	1200	115,37	0	
Compartimentos	0					1					0					0				0				1					Utilização	FO.	Perda
Plano 7	0	0	0	0	0	0	3	1	2	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	8	1200	114,96	0	
Compartimentos	1					1					0					0				0				1					Utilização	FO.	Perda
Plano 8	2	1	1	0	0	0	3	1	2	1	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	2	1198	114,66	2	
Compartimentos	0					0					0					0				0				1					Utilização	FO.	Perda
Plano 9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15	0	1197	112,95	3	
Compartimentos	0					1					0					0				0				1					Utilização	FO.	Perda
Plano 10	0	0	0	0	0	0	0	5	0	2	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	7	0	1197	112,76	3	
Compartimentos	0					1					0					0				0				1					Utilização	FO.	Perda
Plano 11	0	0	0	0	0	2	1	2	0	1	0	0	0	0	0	0	0	0	0	0	0	0	8	1	1	3	0	1199	112,20	1	
Compartimentos	0					0					0					2				0				1					Utilização	FO.	Perda
Plano 12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	2	2	0	0	0	0	3	1	1	0	0	1200	111,56	0	
Compartimentos	0					0					0					2				0				1					Utilização	FO.	Perda
Plano 13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	2	2	0	0	0	0	0	4	0	0	0	1198	111,08	2	

Continua na próxima página.

[illegible]

Estatísticas:

Número de Planos	22
Tempo de Execução	0'03"73
Perda Média por Plano	0,553%
Função Objetivo Total	2355,21

Gráfico de Aproveitamento dos Planos de Corte:

II- Heurística do Melhor Compartimento

Tabela 12: Resultados computacionais obtidos a partir da Heurística do Melhor Compartimento para o exemplo 3.

Classes	1					2					3					4				5				Livres							
Itens	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27				
d[i]	3	1	2	2	1	11	13	11	8	8	9	15	17	17	17	5	13	9	3	9	6	18	25	23	29	30	22				
Compartimentos	0					0					3					0				0				0					Utilização	FO.	Perda
Plano 1	0	0	0	0	0	0	0	0	0	0	0	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1179	115,98	21	
Compartimentos	1					0					0					0				1				1					Utilização	FO.	Perda
Plano 2	1	0	1	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	3	0	0	0	0	3	1164	111,09	36	
Compartimentos	0					0					0					0				3				0					Utilização	FO.	Perda
Plano 3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	11	0	0	0	0	0	1188	110,39	12	
Compartimentos	0					1					0					0				1				1					Utilização	FO.	Perda
Plano 4	0	0	0	0	0	0	3	1	2	1	0	0	0	0	0	0	0	0	0	1	0	4	0	0	0	0	3	1164	109,18	36	
Compartimentos	0					2					0					0				0				1					Utilização	FO.	Perda
Plano 5	0	0	0	0	0	0	6	2	4	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	1182	110,35	18	
Compartimentos	0					1					0					0				1				1					Utilização	FO.	Perda
Plano 6	0	0	0	0	0	0	3	1	2	1	0	0	0	0	0	0	0	0	0	1	4	0	1	0	0	0	3	1200	111,66	0	
Compartimentos	1					0					1					0				0				1					Utilização	FO.	Perda
Plano 7	2	1	1	0	0	0	0	0	0	0	5	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	1192	110,85	8	
Compartimentos	0					1					0					0				0				1					Utilização	FO.	Perda
Plano 8	0	0	0	0	0	0	0	5	0	2	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	2	6	1186	110,71	14	
Compartimentos	0					1					0					1				0				1					Utilização	FO.	Perda
Plano 9	0	0	0	0	0	2	1	2	0	1	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	4	0	1192	109,83	8	
Compartimentos	0					0					0					0				0				1					Utilização	FO.	Perda
Plano 10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15	0	1197	112,95	3	
Compartimentos	0					0					0					1				0				1					Utilização	FO.	Perda
Plano 11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	1	1	0	0	0	0	0	0	0	9	0	1176	109,19	24	
Compartimentos	0					0					0					2				0				1					Utilização	FO.	Perda
Plano 12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	2	2	0	0	0	0	6	0	0	0	0	1182	107,56	18	
Compartimentos	0					0					1					1				0				1					Utilização	FO.	Perda
Plano 13	0	0	0	0	0	0	0	0	0	0	4	1	0	0	3	0	1	1	0	0	0	0	8	0	0	0	0	1200	109,84	0	

Continua na próxima página.

[illegible]

Estatísticas:

Número de Planos	22
Tempo de Execução	0'01"42
Perda Média por Plano	1,373%
Função Objetivo Total	2294,96

Gráfico de Aproveitamento dos Planos de Corte:

III - Heurística dos “z” Melhores Compartimentos ($z = 2$)

Tabela 13: Resultados computacionais obtidos a partir da Heurística dos “z” Melhores Compartimentos ($z=2$) para o exemplo 3.

Classes	1					2					3					4				5				Livres							
Itens	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27				
d[i]	3	1	2	2	1	11	13	11	8	8	9	15	17	17	17	5	13	9	3	9	6	18	25	23	29	30	22				
Compartimentos	0					0					2					0				0				1					Utilização	FO.	Perda
Plano 1	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	1	0	0	2	1	1194	119,85	6	
Compartimentos	0					0					1					0				1				1					Utilização	FO.	Perda
Plano 2	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	1	0	0	4	1	0	0	1	2	1199	119,31	1	
Compartimentos	1					0					0					0				1				1					Utilização	FO.	Perda
Plano 3	1	0	1	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	4	1	0	0	1	2	1199	118,02	1	
Compartimentos	0					1					0					0				1				1					Utilização	FO.	Perda
Plano 4	0	0	0	0	0	0	3	3	0	0	0	0	0	0	0	0	0	0	1	0	1	3	1	0	0	1	2	1200	117,01	0	
Compartimentos	0					0					0					0				2				1					Utilização	FO.	Perda
Plano 5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	1	7	0	1	0	2	1	1200	116,38	0	
Compartimentos	0					2					0					0				0				1					Utilização	FO.	Perda
Plano 6	0	0	0	0	0	0	6	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	2	1	1200	116,07	0	
Compartimentos	0					1					0					0				1				1					Utilização	FO.	Perda
Plano 7	0	0	0	0	0	0	3	1	2	1	0	0	0	0	0	0	0	0	0	1	4	0	1	0	0	0	3	1200	115,37	0	
Compartimentos	1					0					0					0				0				1					Utilização	FO.	Perda
Plano 8	2	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	2	7	1198	114,29	2	
Compartimentos	0					1					0					0				0				1					Utilização	FO.	Perda
Plano 9	0	0	0	0	0	2	1	1	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	3	1200	113,43	0	
Compartimentos	0					0					0					0				0				1					Utilização	FO.	Perda
Plano 10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	13	0	1199	112,77	1	
Compartimentos	0					0					0					2				0				1					Utilização	FO.	Perda
Plano 11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	2	2	0	0	0	0	3	1	1	0	0	1200	111,56	0	
Compartimentos	0					0					0					2				0				1					Utilização	FO.	Perda
Plano 12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	2	2	0	0	0	0	3	1	1	0	0	1200	111,56	0	
Compartimentos	0					1					0					1				0				1					Utilização	FO.	Perda
Plano 13	0	0	0	0	0	0	0	0	2	6	0	0	0	0	3	1	0	1	0	0	0	0	3	1	1	0	0	1199	111,04	1	

Continua na próxima página.

[illegible]

Estatísticas:

Número de Planos	22
Tempo de Execução	0'05"56
Perda Média por Plano	0,438%
Função Objetivo Total	2351,70

Gráfico de Aproveitamento dos Planos de Corte:

IV - Heurística dos “z” Melhores Compartimentos ($z = 3$)

Tabela 14: Resultados computacionais obtidos a partir da Heurística dos “z” Melhores Compartimentos ($z=3$) para o exemplo 3.

Classes	1					2					3				4				5				Livres							
Itens	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27			
d[i]	3	1	2	2	1	11	13	11	8	8	9	15	17	17	17	5	13	9	3	9	6	18	25	23	29	30	22			
Compartimentos	0					0					2				0				0				1					Utilização	FO.	Perda
Plano 1	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	1	0	0	2	1	1194	119,85	6
Compartimentos	0					0					1				0				1				1					Utilização	FO.	Perda
Plano 2	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	3	0	0	3	1	0	0	2	1	1194	118,86	6
Compartimentos	1					0					0				0				1				1					Utilização	FO.	Perda
Plano 3	0	0	2	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	4	0	0	0	4	0	1200	117,71	0
Compartimentos	0					0					0				0				2				1					Utilização	FO.	Perda
Plano 4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	8	0	2	0	0	2	1200	116,46	0
Compartimentos	0					1					0				0				1				1					Utilização	FO.	Perda
Plano 5	0	0	0	0	0	0	3	3	0	0	0	0	0	0	0	0	0	0	0	1	1	3	1	0	0	0	3	1200	116,38	0
Compartimentos	0					2					0				0				0				1					Utilização	FO.	Perda
Plano 6	0	0	0	0	0	0	6	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	2	1	1200	116,07	0
Compartimentos	0					1					0				0				1				1					Utilização	FO.	Perda
Plano 7	0	0	0	0	0	0	4	0	3	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	4	1192	115,17	8
Compartimentos	0					0					0				0				0				1					Utilização	FO.	Perda
Plano 8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	3	10	1197	113,73	3
Compartimentos	1					0					0				0				0				1					Utilização	FO.	Perda
Plano 9	3	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0	1197	113,19	3
Compartimentos	0					0					0				1				0				1					Utilização	FO.	Perda
Plano 10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	1	0	1	0	0	0	0	6	0	0	6	0	1199	112,08	1
Compartimentos	0					1					0				1				0				1					Utilização	FO.	Perda
Plano 11	0	0	0	0	0	1	0	2	3	2	0	0	0	0	3	0	1	1	0	0	0	0	3	1	0	1	0	1200	111,79	0
Compartimentos	0					0					0				2				0				1					Utilização	FO.	Perda
Plano 12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	2	2	0	0	0	0	3	1	1	0	0	1200	111,56	0
Compartimentos	0					1					0				1				0				1					Utilização	FO.	Perda
Plano 13	0	0	0	0	0	3	0	0	0	3	0	0	0	0	3	0	1	1	0	0	0	0	5	1	0	0	0	1199	111,12	1

Continua na próxima página.

Estatísticas:

Número de Planos	22
Tempo de Execução	0'08"70
Perda Média por Plano	0,838%
Função Objetivo Total	2352,15

Gráfico de Aproveitamento dos Planos de Corte:

6.4 Análise dos Planos de Corte

Comparando o desempenho das heurísticas propostas através dos resultados obtidos, podemos verificar que para uma demanda razoável, os planos gerados têm um aproveitamento bastante satisfatório, aproximando-se sempre de 100%.

À medida que a quantidade de itens (d_i) vai diminuindo (para os últimos planos), principalmente quando as demandas dos itens livres se aproximam de zero, o índice de aproveitamento dos planos também diminui. Quando as demandas dos itens livres zeram, as heurísticas começam a gerar planos parciais (isto é, com sobra grande, porém não o suficiente para acomodar um compartimento com utilização próxima de L_{max}). Dependendo dos dados (como no exemplo 3), as heurísticas começam a gerar esses planos parciais, pois uma combinação entre os compartimentos não torna possível um aproveitamento satisfatório da mochila.

A capacidade ociosa dos planos parciais não é considerada perda, mas um novo processo de alocação dos itens deve ser considerado para esta capacidade ociosa. Todas as heurísticas apresentaram a geração de planos parciais, basicamente no último plano gerado (que era de se esperar), onde as demandas restantes não eram suficientes para preencherem toda a mochila. Porém, a Heurística de Decomposição apresentou esta geração não somente no último plano. Isso compromete um pouco a geração dos planos, portanto, um processo que evitasse tal acontecimento seria bem vindo. Uma forma de se resolver esse problema seria, logo que um plano parcial fosse obtido, uma nova mochila (novo compartimento) de capacidade igual ao espaço ocioso do plano parcial seria preenchida. Testes foram feitos para comprovar essa proposta e os resultados foram animadores. Uma outra forma de evitar a geração de planos parciais depende dos dados dos itens. Se for possível manter uma demanda de itens livres até que o último plano seja obtido, o problema de geração de planos parciais não ocorreria. Na prática sabe-se que 60% da demanda são de itens livres, o que nos fornece boas perspectivas. Porém, a princípio, não se pode garantir tal situação.

Na heurística dos “z” Melhores Compartimentos surge um problema de otimização linear inteira (problema (25.*)). Na implementação, relaxamos esse problema, substituindo

$$\sum_{k=1}^K \sum_{j=1}^{z'} \alpha_{ijk} \beta_{jk} \leq d_i,$$

por:

$$\sum_{j=1}^{z'} \alpha_{ijk} \beta_{jk} \leq d_i$$

como visto na seção 5.3 (algoritmo para Heurística dos “z” Melhores Compartimentos), resolvendo o Problema da Mochila resultante. Isto, entretanto, faz com que um excesso de

produção ocorra em alguns exemplos (como no exemplo 2, para $z = 2$). Num futuro, serão pesquisados métodos adequados para resolver o problema (25. *).

Outro fator observado (e serve de comparação entre as heurísticas) é relativo às perdas médias por plano de corte gerado. Essas perdas não consideram planos parciais, pois esses poderão ser reaproveitados. O que se pôde notar foi a ocorrência de perdas muito baixas. Das três heurísticas propostas, a Heurística de Decomposição e a Heurística dos “z” Melhores Compartimentos apresentaram perdas médias abaixo de 1%, o que é muito bom se considerarmos que na indústria as perdas giram em torno de 10%. Comparativamente, a Heurística do Melhor Compartimento apresentou um índice de perdas um pouco maior (em torno de 2%). Se formos considerar um grande volume de demanda, essa diferença para as outras heurísticas torna-se representativa podendo, inclusive, acarretar num aumento do número de planos de cortes necessários para suprir a demanda. O exemplo 2 ilustra essa situação. A Heurística do Melhor Compartimento gerou 27 planos de corte (último plano foi parcial) para suprir a demanda inicial enquanto as demais heurísticas o fizeram com 26 planos. O gráfico e a tabela apresentados a seguir permitirão visualizar o número de planos de cortes gerados por cada heurística e, também, fazer uma comparação das perdas geradas pelas heurísticas, considerando os 3 exemplos e todos os planos de cortes gerados.

Heurísticas	Decomposição	Melhor Compartimento	$z=2$ Melhores Compartimentos	$z=3$ Melhores Compartimentos
Número Total De Planos	61	62*	61	61

* Note que a Heurística do Melhor Compartimento gerou um plano de corte a mais que as outras.

Aproveitamento dos Planos Gerados para cada Heurística

Outro fator a ser considerado é o tempo de execução. Os resultados têm mostrado que as heurísticas rodam relativamente bem. Um fator a ser considerado é que estamos trabalhando com um número pequeno de itens, o que corresponde ao problema observado na prática. Curiosamente, a Heurística do Melhor Compartimento mostrou-se a mais rápida enquanto as demais forneciam os resultados em tempos aproximados entre si. A Heurística de Decomposição foi ligeiramente mais rápida que a Heurística dos “z” Melhores Compartimentos, o que é natural, pois a primeira tem $z = 1$ enquanto a segunda, nesse caso, foi executada para valores de $z = 2$ e $z = 3$, sendo mais rápida, como era de se esperar, para $z = 2$.

Com essas observações podemos concluir que os resultados obtidos foram muito bons, embora alguns ajustes que venham a ser feitos possam contribuir para a obtenção de planos de cortes ainda melhores, evitando o excesso de produção e um número elevado de planos de cortes parciais.

Capítulo 7. Conclusões e Propostas Futuras

Esse trabalho abordou o *Problema da Mochila Compartimentada*, um problema de otimização combinatorial que é uma variação do clássico Problema da Mochila. Tal problema baseia-se em preencher uma mochila compartimentada de capacidade L com itens de diversas categorias onde cada compartimento tem capacidade variável (entre L_{min} e L_{max}) a ser definida. Além disso, cada compartimento só pode agrupar itens de uma mesma categoria. Existem, também, itens que podem ser colocados livremente, isto é, sem a necessidade de compartimentos para eles. O problema consiste em determinar as capacidades adequadas dos compartimentos a serem alocados na mochila e como esses compartimentos devem ser carregados respeitando as restrições de capacidades dos compartimentos e da mochila, buscando maximizar o valor de utilidade total dos itens inseridos. Trata-se de um problema pouco estudado na literatura e de importantes aplicações práticas como na produção de tubos de aço na indústria metalúrgica.

Nesse estudo, propusemos uma modelagem matemática não linear para o problema que, num primeiro instante, nos ofereceu bastante dificuldade, mas que por fim, representou o problema de maneira objetiva.

Além disso, tratamos o *Problema de Múltiplas Mochilas Compartimentadas* onde uma demanda para cada item foi considerada (fazendo uma analogia ao problema de corte de estoque) e, a partir disso, algumas heurísticas foram propostas para a resolução desse problema.

Foram propostas três heurísticas gulosas para a resolução do problema (Heurística de Decomposição, Heurística do Melhor Compartimento e Heurística dos “z” Melhores Compartimentos). Essas heurísticas forneceram resultados muito bons, gerando planos de

corte com níveis de aproveitamento acima do esperado e cumprindo as demandas de forma satisfatória. Porém, vale salientar que no futuro, algumas melhorias podem ser feitas nessas heurísticas de forma que resultados ainda melhores possam ser obtidos, evitando, por exemplo, a geração de planos de corte parciais e o excesso de produção.

Dentre essas melhorias podemos citar o tratamento dos planos parciais à medida que estes vão sendo gerados (através da resolução de um Problema da Mochila para as mochilas parciais), o estudo de um método mais adequado de resolução para o problema de programação inteira (25. *) evitando o excesso de produção, adaptações à Heurística dos “z” Melhores Compartimentos de forma que essa forneça os melhores compartimentos para “z” alternativas de capacidade do compartimento, entre outras.

Uma continuação deste estudo envolve, além dessas melhorias, um estudo aprofundado, estendendo o problema para o caso de diversos tipos de mochilas com características diferentes sendo consideradas. Essa abordagem seria bastante interessante, pois estaríamos tratando com mochilas que nos forneceria custos (c_k) diferentes para a alocação das diversas classes de itens, além, é claro, da possibilidade de termos mochilas com capacidades variadas.

Outra proposta seria realizar um estudo aprofundado do método de enumeração implícita (que segue a estratégia de busca primeiramente em profundidade) proposto em [Gilmore e Gomory, 1963] que resolve o Problema da Mochila, visando estendê-lo para o *Problema da Mochila Compartimentada*.

Além disso, um estudo de uma abordagem *branch and bound* para a resolução do *Problema de Múltiplas Mochilas Compartimentadas* poderia trazer resultados bastante satisfatórios.

Por fim, poderíamos estudar uma extensão do estudo do *Problema da Mochila Compartimentada* para casos bidimensionais envolvendo situações encontradas na prática como, por exemplo, o problema que surge nas linhas de produção de placas de circuitos impressos.

Referências Bibliográficas

[Arenales, 1993] Arenales, M.N. (1993) Uma Teoria para o problema de Corte. *Tese de Livre-Docência, ICMSC-USP, São Carlos, São Paulo.*

[Arenales e Morabito, 1997] Arenales, M.N. e Morabito, R. (editores) (1997) O Problema de Corte e Empacotamento e Aplicações Industriais, XX Congresso Nacional de Matemática Aplicada e Computacional, Gramado-RS.

[Brown, 1971] Brown, A. R. (1971), “*Optimum Packing and Depletion*”. Macdonald - London and American Elsevier Inc. - New York.

[Carvalho, 1991] Carvalho, J. M. V. (1991) Um Problema de Corte em Duas Fases, *Tese de Doutorado, Universidade do Minho, Portugal.*

[Carvalho e Rodrigues, 1994] Carvalho, J. M. V. and Rodrigues, A.J.G. (1994) A Computer Based Interactive Approach to a Two-stage Cutting Stock Problem, *INFOR*, 32(4):243-252.

[Chvátal, 1980] Chvátal, V. (1980) *Linear Programming*, W.H.Freeman and Company, New York.

[Crowder *et al.*, 1983] Crowder, H., Johnson, E.L. e Padberg, M.W. (1983). Solving Large-Scale Zero-One Linear Programming Problems. *Operations Research*, 31: 803-834.

[Dietrich e Escudero, 1989a] Dietrich, B.L. e Escudero, L.F. (1989a). More Coefficient Reduction for Knapsack-Like Constraints in 0-1 Programs with Variable Upper Bounds. IBM T.J. Watson Research Center. RC-14389, Yorktown Heights (NY).

[Dietrich e Escudero, 1989b] Dietrich, B.L. e Escudero, L.F. (1989a). New Procedures for Preprocessing 0-1 Models with Knapsack-Like Constraints and Conjunctive and/or Disjunctive Variable Upper Bounds. IBM T.J. Watson Research Center. RC-14572, Yorktown Heights (NY).

[Dowsland e Dowsland, 1992] Dowsland, K.A. and Dowsland, W.B. (1992) Packing Problems, *European Journal of Operational Research*, 56:02-14.

[Dyckhoff e Finke, 1992] Dyckhoff, H. and Finke, U. (1992) *Cutting and Packing in Production and Distribution*. Springer-Verlag Co., Heidelberg.

[Dyckhoff et al., 1985] Dyckhoff, H.; Abel, D. and Gal, T. (1985) Trim Loss and Related Problems, *Omega*, 13:59-72.

[Ferreira et al., 1990] Ferreira, J.S., Neves, M.A. and Castro, P.F. (1990) A Two-phase Roll Cutting Problem, *European Journal of Operational Research*, 44:185-196.

[Garey e Johnson, 1979] Garey, M.R. e Johnson, D.S. (1979) *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman and Co., San Francisco, 1979.

[Gau e Wäscher, 1995] Gau, T. e Wäscher, G. (1995) *CUTGENI: A Problem Generator for the Standard One-Dimensional Cutting Stock Problem*. Eur. J. Opl. Res. 84, 572-579.

[Gilmore e Gomory, 1961] Gilmore, P. e Gomory, R. (1961) A Linear Programming Approach to the Cutting Stock Problem, *Operations Research*, 9:849-859.

[Gilmore e Gomory, 1963] Gilmore, P. and Gomory, R. (1963) A Linear Programming Approach to the Cutting Stock Problem, part II, *Operations Research*, 14:94-120.

[Gilmore e Gomory, 1965] Gilmore, P. and Gomory, R. (1965) MultiStage Cutting Stock Problems in Two and More Dimensions, *Operations Research*, 14:1045-1074.

[Gramani, 1997] Gramani, M. C. N. (1997) Problema de Corte Bidimensional Guilhotinado e Restrito em 2-Estágios, *Dissertação de Mestrado, ICMSC-USP, São Carlos, São Paulo*.

[Haessler, 1974] Haessler, R. W. (1974) Controlling Cutting Pattern Changes in One-dimensional Trim Problems. *Operations Research*, 23(3):483-493.

[Haessler, 1979] Haessler, R. W. (1979) A note on computational modifications to the Gilmore-Gomory cutting stock algorithm. *Operations Research*, 28(4):1001-1005.

[Hoto, 1996] Hoto, R. S. V. (1996) Otimização no Corte de Peças Unidimensionais com Restrições de Agrupamento, *Dissertação de Mestrado, ICMSC-USP, São Carlos, São Paulo*.

[Hoto, 2000] Hoto, R. S. V. (2000) Uma Abordagem 1-1/2-dimensional para um Problema de Corte em Duas Fases com Agrupamento dos Itens, *Monografia de Qualificação para Doutorado, COPPE-UFRJ, Rio de Janeiro, Rio de Janeiro*.

[Hoto e Arenales, 1996] Hoto, R. S. V. e Arenales, M.N. (1996) Um Problema de Corte Unidimensional com Restrições de Agrupamento e Aplicações Industriais, I Oficina Nacional em Problemas de Corte e Empacotamento, São Paulo - SP, 31-36.

[Hoto e Arenales, 1997] Hoto, R. S. V. e Arenales, M.N. (1997) O Problema do Corte em Bobinas de Aço, XX CNMAC, Gramado - RS, 1997, pp 545,546.

[Hoto et al., 1998] Hoto, R. S. V., Maculan N. e Arenales, M.N. (1998) O Problema do Corte em Bobinas de Aço Via Geração de Colunas, XX CNMAC, Gramado - RS, 1997, pp 545,546.

[Lin, 1998] Lin, E.Y.H. (1998). A Bibliographical Survey on Some Well-Known Non-Standard Knapsack Problems. *INFOR*, 4, 36 (1998), 274-317.

[Martello e Toth, 1990] Martello, S. e Toth, P. (1990) *Knapsack Problems Algorithms and Computer Implementations*, John Wiley & Sons, Chichester.

[Morabito, 1992] Morabito, R. (1992) Uma abordagem em grafo E/OU para o problema de empacotamento: Aplicações ao Carregamento de Paletes e Contêineres. *Tese de Doutorado, EESC-USP, São Carlos, São Paulo*.

[Morabito e Arenales, 1995] Morabito, R. e Arenales, M.N. (1995) Performance of Two Heuristics for Solving Large Scale Two-dimensional Guillotine Cutting Problems, *INFOR*, 33(2):145-155.

[Pereira, 1993] Pereira, M.A. (1993) Uma Abordagem Matemática para o Problema de Corte e Laminação de Fitas de Aço, *Dissertação de Mestrado, UNICAMP, Campinas, São Paulo*.

[Pressman, 1995] Pressman, R. S. (1995) *Engenharia de Software*, MAKRON Books do Brasil Editora Ltda, Editora Mc Graw-Hill Ltda, São Paulo, Brasil.

[Soma et al., 1997] Soma, N.; Yanasse, H. e Maculan, N. (1997) O Problema de Corte e Empacotamento e Aplicações Industriais: O Problema da Mochila, XX Congresso Nacional de Matemática Aplicada e Computacional, Gramado-RS, 24-58.

[Silveira, 1999] Silveira, R. J. (1999). O Problema de Corte Bidimensional Guilhotinado e Restrito: Aplicação na Indústria de Circuitos Impressos. *Dissertação de Mestrado, DEP – UFSCar, São Carlos, São Paulo.*

[Sweeney e Paternoster, 1992] Sweeney, P. and Paternoster, E. (1992) Cutting and Packing Problems : A Categorized, Application-Oriented Research Bibliography, *Journal of the Operational Research Society*, 43:691-706.

[Yanasse et al., 1990] Yanasse, H.H., Zinober, A.S.I. and Harris, R.G. (1990) A Heuristic Procedure for a Two-dimensional Guillotine Cutting Stock Problem, *Paper presented at INFOR, Athens, Grece.*

[Yanasse et al., 1997] Yanasse, H.H., Soma, N.Y. e Maculan, N. (1997) An Algorithm for Determining the K-best Solutions of the One-dimensional Knapsack Problem. Submetido para *Discrete Applied Mathematics*.