
Geração de colunas para problemas de
corte em duas fases

Aline Aparecida de Souza Leão

Livros Grátis

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito: 20 de janeiro de 2009

Assinatura: _____

Geração de colunas para problemas de corte em duas fases¹

Aline Aparecida de Souza Leão

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 - Ciências de Computação e Matemática Computacional.

USP - São Carlos
Janeiro/2009

¹Este trabalho foi financiado pela FAPESP

Dedico este trabalho à Devanira (minha mãe)
e à Carla (minha irmã).

Agradecimentos

Agradeço primeiramente a Deus por ter me abençoado em todas as etapas necessárias deste trabalho e em todos os momentos da minha vida.

A minha mãe, a quem eu amo muito, pelo carinho, paciência e pela preocupação.

A minha irmã Carla pelo apoio e paciência.

Ao meu pai pela torcida.

Aos professores Arenales e Maristela pela orientação, amizade e paciência.

A minha avó Maria, ao tio Hélio e ao tio Osvaldo pelo carinho, preocupação e pelas orações.

A minha madrinha Cida pelo carinho, pelas orações e pela confiança que sempre depositou em mim.

À professora Silvely pela amizade e por ter me incentivado na pesquisa.

A todos os meus amigos que sempre estiveram do meu lado.

Aos colegas da sala de estudo dos alunos de pós-graduação, pelos momentos de estudo e pelas muitas risadas.

Ao pessoal do laboratório pela convivência nos momentos de estudo e de descontração.

A todos que contribuíram direta ou indiretamente para este trabalho.

Resumo

O Problema da Mochila Compartimentada é uma extensão do Problema da Mochila, em que os itens solicitados são divididos em classes, de modo que a mochila deve ser subdividida em compartimentos, os quais têm capacidades limitadas e são carregados com itens da mesma classe. Além disso, a construção de um compartimento tem um custo fixo e ocasiona uma perda no espaço da mochila. O objetivo consiste em maximizar a soma dos valores dos itens, descontado o custo fixo de inclusão de compartimentos. Neste trabalho, são abordados dois métodos de solução. A primeira abordagem é uma heurística, que consiste na combinação de duas heurísticas da literatura. A segunda abordagem é o método Geração de Colunas, que além de fornecer um novo limitante superior para o Problema da Mochila Compartimentada, ao final do método o problema mestre foi resolvido com as variáveis definidas como inteiras, obtendo uma solução factível. Em ambos os métodos, o modelo não-linear é decomposto em dois modelos lineares, no qual, um gera compartimentos e o outro os seleciona. Os resultados obtidos com as duas abordagens foram comparados com um limitante superior e se mostraram bastante satisfatórios.

Palavras-chave: Problemas da Mochila, Problemas de Corte, Geração de Colunas, Heurísticas.

Abstract

The Compartmentalized Knapsack Problem is an extension of the classical Knapsack Problem, where the ordered items are partitioned into classes, in such way that the knapsack must be divided into compartments, each one having limited capacity. In addition, the building of a compartment has a fixed cost and involves a loss of the overall capacity. The objective is to maximize the sum of the items utility value, minus the fixed costs of the compartments. This dissertation presents two solving methods. The first approach is a heuristic method, which is a combination of two heuristics from the literature. The second approach is a Column Generation method, that apart from it gives a new upper bound to the Compartmentalized Knapsack Problem, in the end of the method the master problem was solved with the variables defined as integer, that supplies a feasible solution. In both methods, the mathematical non linear model is decomposed into two linear models, one generates the compartments, and the other selects them to compose the knapsack. The results obtained with these two approaches were compared with an upper bound and they showed very efficient.

Keywords: Knapsack Problems, Cutting Problems, Column Generation, Heuristics.

Sumário

Resumo	v
Abstract	vii
1 Introdução	1
1.1 Propostas do trabalho	2
1.2 Organização do texto	3
2 Revisão da literatura	5
2.1 Revisão para Problemas de Corte em Duas Fases	5
2.2 Revisão para o Problema da Mochila Compartimentada	7
3 Problema da Mochila Compartimentada	11
3.1 Definição do problema	11
3.2 Modelo	12
3.3 Aplicações em Problemas de Corte em Múltiplas Fases	14
3.3.1 Corte de bobinas de aço	15
3.3.2 Descrição do processo industrial	17
4 Uma nova heurística para o Problema da Mochila Compartimentada	21
4.1 Duas heurísticas da literatura	21
4.1.1 Heurística dos “Z” Melhores Compartimentos	21
4.1.2 Heurística do Melhor Compartimento para “W” Capacidades	23
4.2 Heurística Híbrida	23
4.3 Limitante superior para o Problema da Mochila Compartimentada	25
4.4 Gerador de dados	26
4.5 Experimentos computacionais	27

5	Geração de Colunas para o Problema da Mochila Compartimentada	35
5.1	Geração de Colunas	35
5.2	Geração de Colunas para o Problema da Mochila Compartimentada	38
5.3	Experimentos computacionais	41
6	Métodos de enumeração implícita para Problemas da Mochila e extensões	45
6.1	Método de enumeração implícita para o Problema Clássico da Mochila	45
6.2	Método de enumeração implícita para o Problema da Mochila Restrita	49
6.3	Algoritmo Z melhores soluções para o Problema da Mochila Restrita	50
6.4	Heurística para o problema (4.15) - (4.19)	52
6.5	Verificação do <i>backtracking</i> longo	55
7	Conclusões e trabalhos futuros	57

Lista de Figuras

3.1	Padrão compartimentado	14
3.2	Esquema geral de corte em várias fases.	15
3.3	Corte em duas fases em bobinas de aço.	16
3.4	Perfil do processo de corte em bobinas de aço.	18
3.5	Laminação: redução da espessura das bobinas intermediárias.	18
3.6	Obtenção de tubos de aço.	19
4.1	Gráfico do gap médio: à esquerda os resultados com o Problema (4.15) - (4.19) resolvido heurísticamente e à direita exatamente.	32
4.2	Gráfico do gap médio: à esquerda os resultados com o Problema (4.15) - (4.19) resolvido heurísticamente e à direita exatamente.	32
5.1	Gráfico da função objetivo.	42
6.1	Representação da árvore de enumeração implícita	47

Lista de Tabelas

3.1	Dados dos itens.	13
4.1	Arranjo dos problemas	27
4.2	Valores médios do <i>gap</i> (%) para a heurística Melhor Compartimento para W Capacidades.	28
4.3	Valores médios do <i>gap</i> (%) considerando o Passo 1.5	29
4.4	Valores médios do <i>gap</i> (%) considerando o Passo 1.5'	29
4.5	Valores médios do <i>gap</i> (%) considerando o Passo 1.5	30
4.6	Valores médios do <i>gap</i> (%) considerando o Passo 1.5'	30
4.7	Valores médios do <i>gap</i> (%) considerando o Passo 1.5	31
4.8	Valores médios do <i>gap</i> (%) considerando o Passo 1.5'.	31
4.9	Tempo computacional médio (s) considerando o Passo 1.5.	33
4.10	Tempo computacional médio (s) considerando o Passo 1.5'.	33
4.11	Número de soluções ótimas obtidas considerando o Passo 1.5	33
4.12	Número de soluções ótimas obtidas considerando o Passo 1.5'.	34
5.1	Valores médios da função objetivo	41
5.2	Tempo computacional médio	41
5.3	Geração de Colunas: solução inteira	43

Introdução

Problemas de Corte e Empacotamento são bastante importantes para problemas econômicos. A grande maioria dos estudos realizados tem por objetivo melhorar o planejamento e programação da produção industrial, envolvendo análises peculiares a cada tipo de indústria, como por exemplo, indústrias de papel, móveis, aço, plástico, têxtil, entre outras. Estes problemas começaram a ser estudados na década de 30, porém somente a partir dos trabalhos de Gilmore & Gomory (1961, 1963, 1965) que as pesquisas direcionadas a este assunto tiveram grande notoriedade.

Basicamente, os Problemas de Corte consistem em produzir um conjunto de itens de tamanhos e formas definidas a partir do corte de objetos grandes disponíveis. Analogamente, os objetos grandes podem ser vistos como espaços a serem preenchidos por itens menores, de modo que tais problemas são referenciados por Problemas de Corte e Empacotamento. Problemas de Corte e Empacotamento são Problemas de Otimização Combinatória que podem ter como objetivo reduzir a perda, maximizar o lucro, diminuir custos de produção etc. Muitos trabalhos fazem referências a Problemas de Corte de Estoque quando existe um conjunto de objetos disponíveis em estoque a serem cortados para a produção de itens demandados em quantidades específicas.

Existem diversas características que podem ser usadas para classificar os Problemas de Corte e Empacotamento, conforme Dyckhoff & Finke (1992). O primeiro critério para uma classificação é a dimensão do problema, que resumidamente pode ser unidimensional, bidimensional etc., quando uma, duas ou mais dimensões são relevantes no processo de corte, respectivamente. Ainda temos problemas 1.5 dimensionais que são problemas bidimensionais, porém

uma das duas dimensões é variável, 2.5 dimensionais que são problemas tridimensionais com uma dimensão variável e duas fixas.

A literatura é rica em métodos para resolver Problemas de Corte, devido a grande dificuldade de resolução. Um modelo matemático de otimização linear inteiro, com um número muito grande de variáveis, é bastante utilizado para representar Problemas de Corte de Estoque, desde os trabalhos de Gilmore & Gomory (1961), que propuseram a técnica Geração de Colunas para resolver a relaxação linear desse modelo.

Este trabalho abordará Problemas de Corte Unidimensional, porém o processo de corte é realizado em duas fases, um caso pouco explorado na literatura. Entretanto, além dos problemas necessitarem de duas fases de corte, eles possuem características bastante interessantes do ponto de vista técnico e matemático. Para tratar estas particularidades será abordado o Problema da Mochila Compartimentada.

O Problema da Mochila Compartimentada foi modelado em Marques (2000) e Hoto (2001), para gerar padrões para Problemas de Corte em Duas Fases, isto é, os objetos em estoque são cortados em objetos intermediários, que em seguida são cortados nos itens encomendados. As características envolvidas nos problemas, que podem utilizar o Problema da Mochila Compartimentada, é que os itens demandados são agrupados em diferentes classes e cada objeto intermediário é cortado em itens de uma única classe. Estes problemas são difíceis de resolver e a principal dificuldade reside no fato da não linearidade do Problema da Mochila Compartimentada.

1.1 Propostas do trabalho

Na literatura, alguns métodos heurísticos foram propostos por Marques & Arenales (2007) orientado para o corte de bobinas de aço sujeitas à laminação. Os autores propuseram um bom limitante superior para este problema, que foi utilizado para analisar a eficiência das heurísticas. A partir das heurísticas propostas por estes autores, neste trabalho é proposta uma nova heurística, que combina os métodos de solução de Marques & Arenales (2007).

Além disso, apresentamos uma reformulação do Problema da Mochila Compartimentada, pela definição prévia das possíveis soluções dos objetos intermediário (compartimentos), obtendo-se assim um problema mestre equivalente com muitas variáveis. A relaxação linear do problema mestre é resolvida pela técnica de Geração de Colunas. O problema mestre também foi resolvido com as variáveis inteiras, depois de obtido uma solução ótima para o problema mestre relaxado.

1.2 Organização do texto

No Capítulo 2 são apresentados todos os artigos encontrados na literatura sobre Problemas de Corte em Duas Fases e uma revisão dos trabalhos que tratam o Problema da Mochila Compartimentada.

No Capítulo 3 são apresentadas a definição, a modelagem matemática e uma aplicação para o Problema da Mochila Compartimentada.

No Capítulo 4 é proposta uma heurística híbrida e analisados os resultados computacionais.

No Capítulo 5 o Problema da Mochila Compartimentada é reformulado e proposto um método Geração de Colunas e apresentados os resultados computacionais.

No Capítulo 6 são apresentados os métodos de solução utilizados para resolver os Problemas da Mochila que surgem na heurística e no método Geração de Colunas, baseados na enumeração implícita.

No Capítulo 7 são apresentados as conclusões e os trabalhos futuros que poderão ser desenvolvidos para o Problema da Mochila Compartimentada.

Revisão da literatura

2.1 Revisão para Problemas de Corte em Duas Fases

Nesta seção é apresentada uma revisão bibliográfica sobre Problemas de Corte em Duas Fases, em que a compartimentação é tratada de maneira implícita ou desnecessária, relatando todos os trabalhos encontrados na literatura sobre o tema.

Um dos primeiros trabalhos direcionados ao Problema de Corte em Duas Fases é o artigo de Haessler (1971), que aborda o processo de corte de uma indústria de papel. O corte de bobinas de papel é realizado em duas fases devido ao número limitado de facas e para minimizar o tempo de preparação envolvido na mudança de facas do cortador. Na resolução do problema, o autor utilizou uma heurística para gerar os padrões de corte, junto com a heurística de repetição exaustiva (Hinxman, 1980). Com esse procedimento heurístico, o autor relata que a maioria das soluções obtidas foi melhor que as realizadas manualmente pela indústria estudada.

Haessler (1979) apresenta um estudo para uma indústria de filmes plásticos, que consiste em cortar bobinas de filmes plásticos em bobinas intermediárias e, então, recortá-las. O autor relata que o problema em questão não pode ser classificado como um Problema de Corte de Estoque Bidimensional Multiestágio, proposto por Gilmore & Gomory (1965), porque os cortes do primeiro e segundo estágios eram paralelos uns aos outros (problema unidimensional, já que apenas uma dimensão é relevante para o corte). Para resolvê-lo foram utilizados procedimentos heurísticos baseados numa modificação do processo Geração de Colunas, proposto por Gilmore & Gomory (1961).

Um dos primeiros trabalhos desenvolvidos para Problemas de Corte em Duas Fases provenientes de indústrias de ferro e aço, cujo material está sujeito à redução de espessura (laminação), foi apresentado em Ferreira et al. (1990). Para resolvê-lo, os autores utilizaram uma heurística, em que a partir de um número inicial de itens pré-fixados eram planejadas uma quantidade determinada de bobinas intermediárias. Em seguida, eram gerados os padrões de corte utilizando a heurística *First-Fit-Decreasing*, então verificava se eles eram viáveis e aceitáveis, e os melhores eram selecionados para compor a solução. A desvantagem deste procedimento é que os padrões gerados não eram necessariamente factíveis, nestes casos a heurística era executada até encontrar padrões factíveis, a cada execução o número de bobinas intermediárias e itens iniciais eram incrementados.

Em Pereira (1993), o problema de corte de bobinas de aço sujeitas à laminação foi modelado inicialmente como um Problema de Programação Linear Inteiro Misto. Este modelo, no entanto, era difícil de resolver, então o autor propôs outros modelos simplificados. Alguns testes foram realizados até obter o modelo simplificado considerado mais adequado, o qual foi resolvido utilizando um algoritmo *branch-and-bound*. De acordo com o autor, os resultados finais obtidos foram considerados satisfatórios.

Em Carvalho & Rodrigues (1994, 1995) é desenvolvida a técnica Geração de Colunas para o problema de corte de bobinas de aço sujeitas à laminação. A principal característica do método proposto é que as restrições do problema mestre estão expressas em termos dos itens, enquanto no subproblema, dado por um Problema da Mochila, as variáveis são expressas em termos das bobinas intermediárias. Para contornar essa situação, os autores consideraram que cada bobina intermediária é composta por apenas um tipo de item, ou seja, padrão de corte homogêneo. Este processo, conseqüentemente, ocasiona excesso de itens em relação à demanda. Para reduzir este excesso, as bobinas intermediárias chamadas dominadas e as que não contribuíam com custos relativos positivos eram removidas.

Johnson et al. (1997) comparam a eficiência dos processos de corte em uma única fase com o corte em duas fases para a indústria de papel. Apesar do processo poder ser realizado em uma fase, os autores utilizaram o corte em duas fases para comparar os dois procedimentos. Eles adotaram um procedimento heurístico, cuja desvantagem é que não era possível melhorar as bobinas intermediárias obtidas, caso fosse encontrado uma solução ruim. Apesar da necessidade de procedimentos melhores, ficou demonstrado que o corte em duas fases foi bastante eficiente em relação ao corte em uma fase, diminuindo muito a perda.

Dando continuidade à proposta de Johnson et al. (1997) sobre a formação de bobinas intermediárias melhores, Zak (2002a,b) apresenta um estudo realizado para o Problema de Corte em Duas Fases proveniente de uma indústria de papel. Neste trabalho, a necessidade de dois estágios de corte é devido ao número limitado de facas. Para resolver o problema, o autor propõe uma técnica geração de colunas e linhas, em que o tamanho das bobinas intermediárias são as linhas e os padrões de corte são as colunas. Para gerar linhas e colunas o autor considerou três

problemas: dois Problemas Clássicos da Mochila e um não-linear. Os dois Problemas Clássicos da Mochila fazem a geração de colunas, enquanto que, o problema não-linear faz a geração simultânea de linhas e colunas. Durante o processo de resolução foram impostas algumas limitações, que teoricamente restringem a escolha das bobinas intermediárias.

Correia et al. (2004) apresentam um sistema computacional desenvolvido para uma indústria de papel, que devido ao número limitado de facas o corte é realizado em duas fases, no qual, somente as bobinas que têm o mesmo diâmetro interno e externo e o mesmo comprimento podem ser combinadas no mesmo objeto. O procedimento de solução proposto possui três estágios. No primeiro estágio são geradas as bobinas intermediárias e depois os padrões de corte, então estes são enumerados baseado em uma ordenação lexicográfica. No segundo estágio, os padrões aceitos são utilizados como colunas no Problema de Programação Linear e resolvido pelo método simplex. Os autores utilizam dois modelos de programação linear distintos e comparam a eficiência entre eles. No terceiro estágio, as soluções passam por uma heurística de arredondamento, que segundo os autores são necessárias melhorias.

2.2 Revisão para o Problema da Mochila Compartimentada

Nesta seção é feita uma revisão dos trabalhos de Hoto (1996, 2001); Hoto et al. (1999) e Marques (2000, 2004), que abordam o Problema da Mochila Compartimentada.

Hoto (1996) apresentou uma heurística para resolver o problema de corte de bobinas de aço. Apesar do Problema da Mochila Compartimentada não ter sido definido, o Problema da Mochila Restrita é utilizado para gerar compartimentos e uma heurística é utilizada para determinar um padrão de corte compartimentado. Para determinar um padrão, é construído um compartimento para a classe de itens com alta eficiência e incluído na mochila, em seguida, são construídos outros compartimentos para o espaço restante com outras classes. A partir do primeiro padrão de corte são construídos outros, e por fim, com os padrões obtidos, o problema de corte de bobinas de aço é resolvido. Em Hoto et al. (1999) foi proposto um modelo linear inteiro para o Problema da Mochila Compartimentada, que é um caso particular do problema abordado em Hoto (2001) e Marques (2000, 2004), no qual para cada classe de itens é formado no máximo um compartimento para compor a mochila. Para resolvê-lo foi proposto um provável *branch-and-bound*, que não se mostrou eficaz.

Marques (2000) propõe duas heurísticas gulosas para o Problema da Mochila Compartimentada, que são: Heurística do Melhor Compartimento e a dos “Z” Melhores Compartimentos. Em Marques (2004) é apresentada uma terceira heurística: a Heurística do Melhor Compartimento para “W” Capacidades, que dependendo do valor de W fornece resultados melhores que as anteriores.

A Heurística dos “Z” Melhores Compartimentos determina os “Z” melhores compartimentos para cada classe de itens. Para isso são determinadas as “Z” melhores soluções do Problema da Mochila Restrita, cujas restrições são a capacidade máxima do compartimento e o limite máximo dos itens na mochila. O Problema de Programação Linear Inteiro a ser resolvido na segunda etapa determina quais serão os compartimentos e itens livres (itens que não precisam ser compartimentados) que formarão a mochila. Um caso particular é para $Z = 1$, neste caso o problema a ser resolvido na segunda etapa é um Problema da Mochila Restrita.

A Heurística do Melhor Compartimento determina o melhor compartimento para cada classe de itens. Em seguida, determina o valor utilidade por unidade dos compartimentos e itens livres, sendo atribuído à mochila o compartimento (ou item livre) de maior valor, para então, a demanda dos itens e a capacidade da mochila serem atualizadas. O processo é repetido até que não seja possível incluir mais nenhum compartimento na mochila. Nessa heurística é necessário apenas resolver Problemas da Mochila Restrita.

A Heurística do Melhor Compartimento para “W” capacidades determina para cada classe o melhor compartimento para “W” possíveis capacidades diferentes dos compartimentos. No segundo passo, é resolvido o problema que seleciona os compartimentos e itens livres que irão preencher a mochila. Mais detalhes sobre as heurísticas “Z” Melhores Compartimentos e Melhor Compartimento para “W” Capacidades são dadas no capítulo 4.

Hoto (2001) considera duas abordagens para o Problema da Mochila Compartimentada, o problema na sua forma restrita e irrestrita. A forma irrestrita considera o número de itens ilimitado e para resolvê-lo, o autor utiliza um procedimento exato e outro heurístico.

Na resolução do Problema da Mochila Compartimentada Irrestrita por meio do procedimento exato, o autor considera duas etapas para a determinação dos compartimentos. Na primeira etapa são determinados os valores utilidades de todos os compartimentos, resolvendo-se vários Problemas da Mochila, com capacidades variando entre a mínima e máxima. Na segunda etapa é resolvido um Problema Clássico da Mochila, para determinar quantas vezes cada compartimento é incluído na mochila. Note que a ausência de limitantes sobre o número de itens permite incluir qualquer número de compartimentos (já preenchidos na primeira etapa).

A resolução do Problema da Mochila Compartimentada Irrestrita do modo heurístico tem por objetivo evitar a resolução dos vários Problemas da Mochila da primeira etapa do procedimento exato. Para isso, foi utilizado o cálculo do limitante de Martello & Toth (1990) para resolver os Problemas da Mochila, determinando assim uma aproximação superior para a utilidade dos compartimentos. Foi considerado o mesmo custo fixo para os compartimentos formados por itens da mesma classe. Após a resolução dos Problemas da Mochila na primeira etapa, resolve-se apenas o Problema da Mochila para os compartimentos selecionados.

Para o Problema da Mochila Compartimentada Restrita, o autor considera que um compartimento pode ser incluído no máximo uma vez, obtendo assim um outro Problema da Mochila

Compartimentada Restrita e, resolvido de modo análogo à heurística de decomposição descrita anteriormente.

Hoto (2001) apresenta uma abordagem para resolver o problema de corte de bobinas de aço utilizando o método simplex com Geração de Colunas, no qual a condição de integralidade do Problema de Corte em Duas Fases é relaxada e para obter soluções factíveis são utilizadas heurísticas de arredondamento. Os padrões compartimentados são obtidos por meio do Problema da Mochila Compartimentada. Para a demanda inicial, o autor utiliza o algoritmo exato ou a heurística proposta para o Problema da Mochila Compartimentada Irrestrita, pois supõe alta demanda. Para os problemas residuais, a demanda é baixa e é utilizado a heurística de decomposição descrita para o Problema da Mochila Compartimentada Restrita.

Problema da Mochila Compartimentada

3.1 Definição do problema

O Problema da Mochila Compartimentada é modelado como um Problema de Otimização Não-Linear Inteiro e é uma extensão do Problema Clássico da Mochila, o qual tem apenas um compartimento. A mochila compartimentada é dividida em compartimentos, no qual cada um deve ser preenchido por itens de uma dada classe.

Para a definição do problema, considere que existem n itens a serem alocados a uma mochila, cada item possui um peso l_i , um valor utilidade v_i e um limitante superior σ_i (isto é, o número máximo de itens do tipo i permitido na mochila). Esses itens estão particionados em K classes (itens de características em comum pertencem a uma mesma classe). O espaço da mochila é organizado em compartimentos, de modo que apenas itens de mesma classe podem ser colocados no mesmo compartimento. Existem também, itens que não precisam ser compartimentados, formando uma classe de “itens livres”. A mochila possui uma capacidade limitada L e as capacidades dos compartimentos a serem incluídos na mochila são flexíveis, porém, são limitadas inferiormente (L_{min}^k) e superiormente (L_{max}^k). Os compartimentos possuem um custo fixo c_k e uma perda fixa S_k por serem incluídos, que variam de acordo com as classes, e também, um valor utilidade, dado pela soma do valor utilidade dos itens que os compõem. Além disso, pode existir a formação de mais de um compartimento com itens de uma determinada classe. Deste modo, o Problema da Mochila Compartimentada consiste em definir os compartimentos

que comporão a mochila, visando a maximização do valor utilidade subtraídos os custos dos compartimentos.

Para ilustrar este problema, considere que um viajante deva preencher uma mochila com os itens necessários para sua viagem, que são divididos em classes: roupas, alimentos, remédios e calçados. Cada grupo deve ser guardado na mochila em compartimentos separados, que são limitados por uma capacidade relacionada ao peso. Pode existir a formação de mais de um compartimento para a mesma classe, como por exemplo, ao compartimento de maior necessidade. Além disso, o número de itens de um determinado tipo é limitado. Sob estas condições, é preciso preencher a mochila de forma que a sua capacidade não seja ultrapassada e o seu valor utilidade (soma do valor utilidade dos itens alocados nos compartimentos) seja maximizado.

3.2 Modelo

Nesta seção será apresentado o modelo matemático para o Problema da Mochila Compartimentada, que foi proposto em Marques (2004). Para a apresentação do modelo considere os seguintes dados e variáveis.

Dados:

- L : capacidade da mochila;
- $N = \{1, \dots, n\}$: define o conjunto de tipos de itens;
- l_i : peso do item tipo i , $i = 1, \dots, n$;
- v_i : valor utilidade do item tipo i , $i = 1, \dots, n$;
- σ_i : número máximo de itens tipo i permitido na mochila, $i = 1, \dots, n$;
- K : número total de classes;
- A_k : conjunto dos itens pertencentes à classe k , $k = 1, \dots, K$, $N = A_1 \cup A_2 \cup \dots \cup A_K$ e $A_i \cap A_j = \emptyset$ para $i \neq j$;
- c_k : custo de incluir um compartimento para itens da classe k , $k = 1, \dots, K$;
- L_{mim}^k : capacidade mínima do compartimento para itens da classe k , $k = 1, \dots, K$;
- L_{max}^k : capacidade máxima do compartimento para itens da classe k , $k = 1, \dots, K$;
- S_k : perda na capacidade da mochila devido à inclusão de um compartimento para itens da classe k , $k = 1, \dots, K$;
- N_k : número total de possíveis compartimentos da classe k , $k = 1, \dots, K$.

Variáveis:

- a_{isk} : número de itens do tipo i da classe k no compartimento s , $i = 1, \dots, n$, $k = 1, \dots, K$ e $s = 1, \dots, N_k$;
- β_{sk} : número de vezes que o compartimento s para itens da classe k é repetido na mochila, $k = 1, \dots, K$ e $s = 1, \dots, N_k$.

Para evitar situações triviais, supomos que para $k = 1, \dots, K$:

- $\sum_{i \in A_k} \sigma_i l_i > L_{max}^k$;
- $l_i < L_{max}^k, i \in A_k$;
- $L_{min}^k < L_{max}^k \leq L$.

O modelo matemático apresentado em Marques (2004) é dado por:

$$\text{maximize} \quad \sum_{k=1}^K \sum_{s=1}^{N_k} \left(\sum_{i \in A_k} v_i a_{isk} - c_k \right) \beta_{sk} \quad (3.1)$$

$$\text{sujeito a :} \quad \sum_{k=1}^K \sum_{s=1}^{N_k} \left(\sum_{i \in A_k} l_i a_{isk} + S_k \right) \beta_{sk} \leq L \quad (3.2)$$

$$\sum_{s=1}^{N_k} a_{isk} \beta_{sk} \leq \sigma_i, i \in A_k, k = 1, \dots, K \quad (3.3)$$

$$L_{min}^k \leq \sum_{i \in A_k} l_i a_{isk} + S_k \leq L_{max}^k, k = 1, \dots, K, s = 1, \dots, N_k \quad (3.4)$$

$$a_{isk} \geq 0, \text{ inteiro}, i = 1, \dots, n, k = 1, \dots, K, s = 1, \dots, N_k \quad (3.5)$$

$$\beta_{sk} \geq 0, \text{ inteiro}, k = 1, \dots, K, s = 1, \dots, N_k. \quad (3.6)$$

Neste modelo, a função objetivo maximiza o valor utilidade menos o custo de incluir o compartimento. A restrição (3.2) refere-se à restrição de capacidade da mochila, considerando os itens que compõem a mochila mais a perda ocasionada pela inclusão de um compartimento. As restrições (3.3) limitam o total de itens a compor a mochila e as restrições (3.4) limitam a capacidade dos compartimentos.

Exemplo: Considere uma mochila com tamanho igual a 28mm e os itens que podem compor a mochila são dados na Tabela 3.1.

Tabela 3.1: Dados dos itens.

	Classe 1			Classe 2			Classe 3		
item	1	2	3	4	5	6	7	8	9
utilidade	7	8	4	3	5	7	6	2	4
largura	3	4	6	5	2	4	5	3	4

As classes são: $A_1 = \{1, 2, 3\}$, $A_2 = \{4, 5, 6\}$ e $A_3 = \{7, 8, 9\}$; os tamanhos mínimo (L_{min}) e máximo (L_{max}), respectivamente, para os compartimentos da classe 1 são dados por

5 e 9, para a classe 2 são 5 e 10, e para a classe 3 são 3 e 10. A inclusão de compartimentos com itens da classe 2 na mochila ocasionam perda de $S_2 = 4mm$ ($2mm$ em cada lateral), para a classe 1 e 3 não há perdas, $S_1 = 0$ e $S_3 = 0$, respectivamente. A Figura 3.1 representa um possível padrão compartimentado.

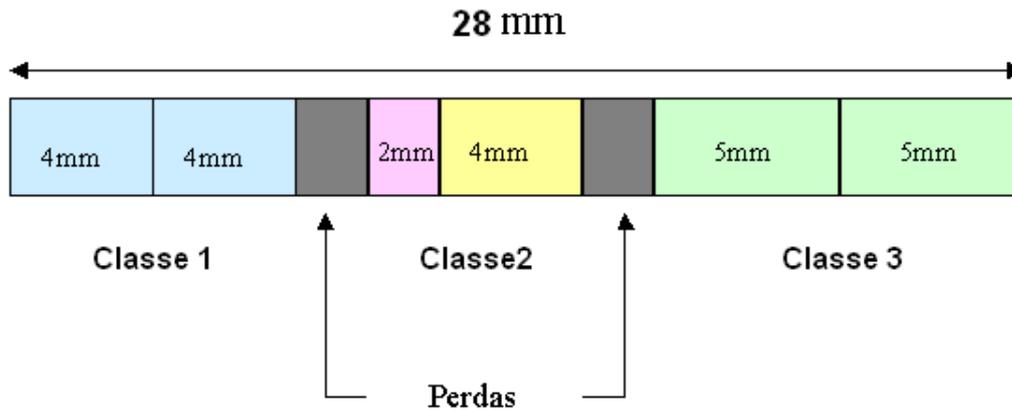


Figura 3.1: Padrão compartimentado

No padrão acima foram incluídos: dois itens do tipo 2 da classe 1; um item do tipo 5 e um do tipo 6 da classe 2; dois itens do tipo 7 da classe 3. Entre as classes 1 e 2, e as classes 2 e 3 encontram-se as perdas por ter incluído um compartimento formado por itens da classe 2.

O Problema da Mochila Compartimentada é de grande importância para Problemas de Corte em Múltiplas Fases. Na literatura, este problema foi aplicado no corte de bobinas de aço sujeitas à laminação, descrito a seguir.

3.3 Aplicações em Problemas de Corte em Múltiplas Fases

A classe de Problemas de Corte Unidimensional se torna mais generalizada quando um objeto em estoque é cortado em objetos intermediários, possuindo duas ou mais fases de corte para a obtenção de itens finais, denominado Problema de Corte em Múltiplas Fases. Diversas indústrias utilizam o corte em múltiplas fases, como indústrias de papel, filme e aço. A necessidade de mais de uma fase pode surgir devido às restrições dos processos de produção e as características de itens encomendados, como por exemplo, a limitação do número de facas, revestimentos do material (papel e lâminas de aço revestidos), laminação (bobinas de aço) entre outros tipos de processamento. A Figura 3.2 representa o caso geral de um processo de corte em múltiplas fases, em que um objeto é cortado sucessivamente até obter os itens demandados, os objetos obtidos nas fases intermediárias de corte são denominados objetos intermediários.

Na literatura, a maioria dos trabalhos sobre cortes unidimensionais trata problemas de corte em uma única fase. Para Problemas de Corte em Múltiplas Fases existem poucos estudos, no

qual a maioria das formulações e testes realizados são para o corte em duas fases, o que não impede extensões.

A maior dificuldade para a resolução dos Problemas de Corte em Múltiplas Fases é que não conhecemos qual será a melhor formação dos objetos intermediários para a construção de um bom padrão de corte. Além disso, essa dificuldade pode ser ainda maior quando as características dos itens encomendados exigem agrupá-los em diversas classes, no qual apenas os itens pertencentes à mesma classe devem estar presentes num mesmo objeto intermediário, como acontece em algumas indústrias metalúrgicas, destacando-se as características da mochila compartimentada.

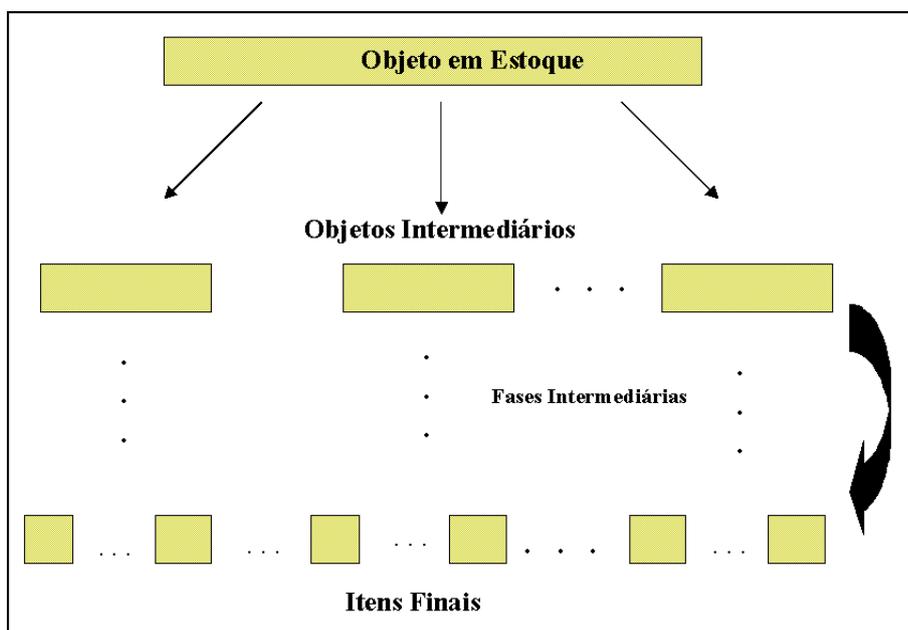


Figura 3.2: Esquema geral de corte em várias fases.

O corte de bobinas de aço é um caso bastante explorado na literatura e surge, por exemplo, em indústrias metalúrgicas que fabricam tubos de aço que serão utilizados em estruturas de treliça, com aplicações na construção civil, na fabricação de maquinário industrial, automóveis, aviões, bicicletas, entre outros. Este problema é descrito a seguir.

3.3.1 Corte de bobinas de aço

O corte de bobinas de aço para a produção de fitas de aço de diferentes características, como espessura e teor de carbono, é classificado como um Problema de Corte de Estoque Unidimensional. A partir dessas fitas são obtidos tubos de aço, em alguns casos as fitas não têm a mesma espessura das bobinas em estoque, sendo necessário o procedimento de laminação e para isso elas são cortadas em bobinas intermediárias, como mostra a Figura 3.3.

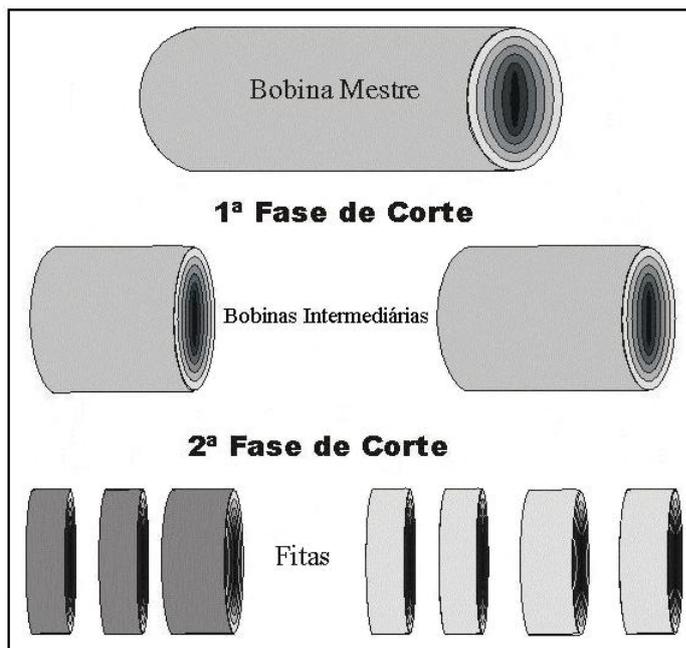


Figura 3.3: Corte em duas fases em bobinas de aço.

Na literatura, para distinguir as particularidades relativas a cada fase do processo de corte, é proposta a classificação dos objetos obtidos em cada fase, em que:

- os objetos a serem cortados ou objetos em estoque são denominados bobinas mestre (Hoto, 1996, 2001), (Marques, 2000, 2004) e (Correia et al., 2004) ou matéria-prima (Ferreira et al., 1990) e (Carvalho & Rodrigues, 1994, 1995);
- os objetos obtidos após a primeira fase de corte são denominados bobinas intermediárias (Hoto, 1996, 2001), (Marques, 2000, 2004), (Carvalho & Rodrigues, 1994, 1995) e (Zak, 2002b,a); ou bobinas primárias (Ferreira et al., 1990); ou ainda bobinas auxiliares (Correia et al., 2004);
- os produtos finais ou itens são os objetos obtidos após a segunda fase de corte.

Além disso, há a perda causada pelos cortes realizados nas laterais das bobinas mestre e intermediárias para eliminar irregularidades, e as sobras que são os materiais que podem ser aproveitados em ocasiões futuras.

Em geral, as características das bobinas de aço envolvem o peso, a largura, a espessura e o teor de carbono. A partir das bobinas em estoque se obtêm as bobinas intermediárias que, se necessário, são laminadas para a redução de espessura, conforme especificado para a produção de tubos.

Assim, as fitas são separadas em grupos de acordo com a sua espessura e teor de carbono, definindo as classes, e os itens que pertencem a classes diferentes não podem ser cortados

de uma mesma bobina intermediária. Para definir um padrão de corte deve ser levado em conta como as bobinas intermediárias são combinadas na bobina mestre e como os itens são combinados nas bobinas intermediárias, o que define um padrão de corte compartimentado. Neste problema, as bobinas intermediárias são denominadas compartimentos.

A necessidade do corte em duas fases é porque existe um número limitado de facas para a obtenção dos itens finais, e também, a largura das bobinas a serem laminadas é limitada inferiormente e superiormente, que podem ser bastante inferior à largura da bobina mestre. Existem situações que são necessários cortes em mais de duas fases, como por exemplo, casos que além da laminação envolvem o revestimento da superfície do aço e o recorte.

Existem várias sociedades que fornecem especificações sobre a qualidade e as características de aço, entre elas, SAE (*Society of Automotive Engineers*), ABNT (*Associação Brasileira de Normas Técnicas*), API (*American Petroleum Institute*), ASTM (*American Society for Testing and Materials*), DIN (*Deustaches Institute for Normuns*) e AISI (*American Institute of Steel and Iron*). Tais especificações variam de acordo com o uso do aço.

3.3.2 Descrição do processo industrial

Os processos de uma indústria analisada por Hoto (2001) e Marques (2004) são descritos nesta seção.

As bobinas mestre são retiradas de um salão de estoque e uma ponte rolante é utilizada para manipulá-las, então cada bobina é desenrolada e uma máquina chamada cortadeira a corta longitudinalmente por cisalhamento, o corte é feito por meio de discos. Em seguida, as tiras obtidas são enroladas novamente, obtendo assim as bobinas intermediárias (veja Figura 3.4). Existe um número limitado de cortes que podem ser feitos nas bobinas mestre, além disso, é realizado cortes nas laterais para retirar as irregularidades das bordas. A máquina que efetua o corte exige um tempo de preparação.

Após a obtenção das bobinas intermediárias, se necessário, ocorre o processo de laminação que é realizado em temperatura ambiente, chamado laminação a frio. Esse processo é realizado por dois cilindros que pressionam a lâmina, no qual as bobinas em uma extremidade são desenroladas, passam pelos cilindros e na outra extremidade enroladas novamente (veja Figura 3.5). A laminação aumenta a dureza do aço e o comprimento da bobina.

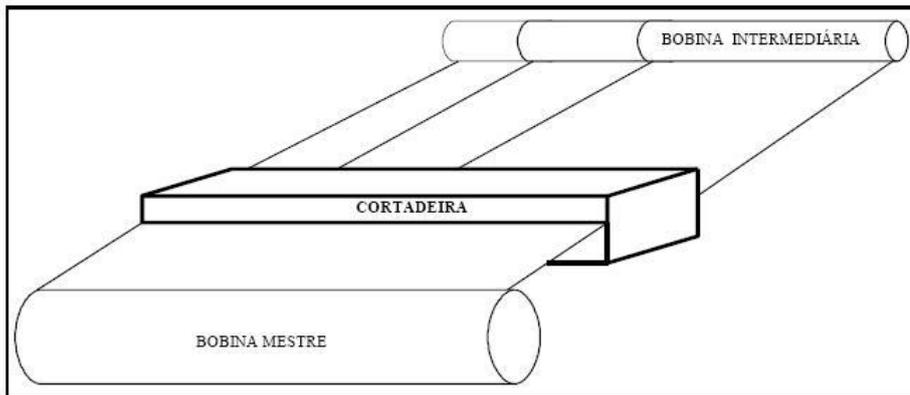


Figura 3.4: Perfil do processo de corte em bobinas de aço.
Fonte: Marques (2004)

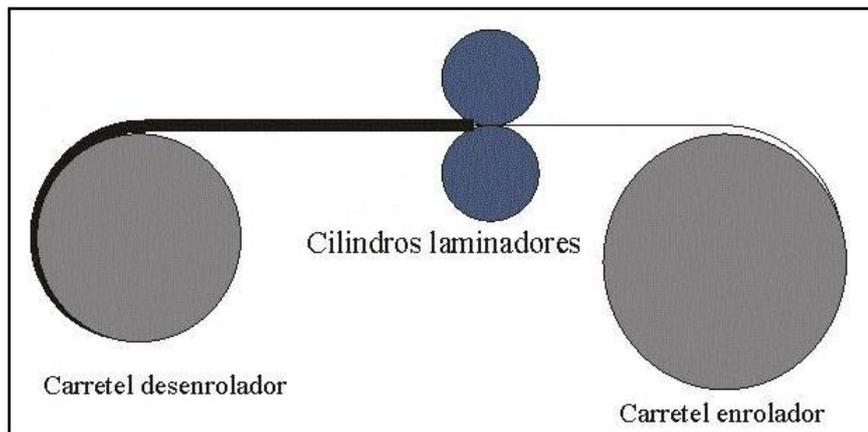


Figura 3.5: Laminação: redução da espessura das bobinas intermediárias.
Adaptado de: Pereira (1993)

Após este processo, é realizado o segundo corte, conhecido como recorte, sendo esse aplicado até obter as fitas. Durante esta etapa são feitos cortes nas laterais das bobinas intermediárias para eliminar as irregularidades. Análogo à máquina que realiza o corte, existe um número limitado de cortes e tempo de preparação da máquina.

Para a obtenção das fitas existe ainda o processo cuja finalidade é corrigir as imperfeições causadas na superfície do aço após o processo de laminação, que consiste numa laminação mais delicada realizada pelo encruador. Por fim, as fitas são acondicionadas em fornos especiais, onde permanecem por um certo período e depois são lentamente resfriadas até atingir a temperatura ambiente, processo denominado recozimento (MARQUES, 2004).

Para obter os tubos, as fitas são cortadas de acordo com o comprimento desejado para os tubos e soldadas como mostra a Figura 3.6, conhecidos como tubos com costura.

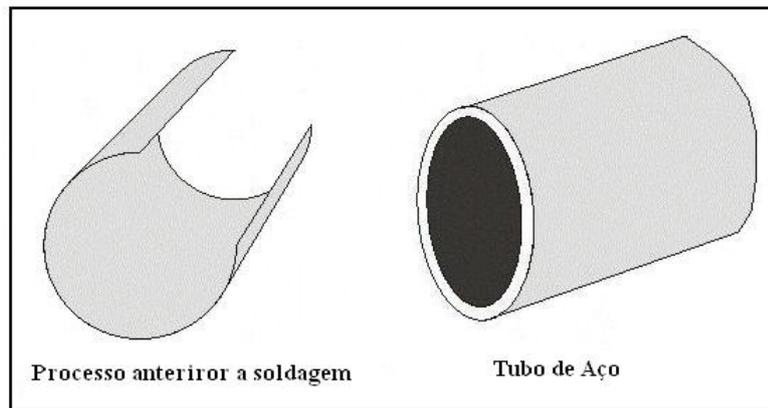


Figura 3.6: Obtenção de tubos de aço.

Há vários custos envolvidos na produção de tubos de aço, sendo relevantes a perda de material, a laminação e a preparação de máquinas. A maioria dos trabalhos sobre o planejamento de produção de indústrias metalúrgicas tem como objetivo reduzir as perdas, estando presente na função objetivo proposta em Pereira (1993), Carvalho & Rodrigues (1994, 1995) e Hoto (1996, 2001). A perda de material pode ser bastante grande, conforme relatado em Hoto (2001) e Marques (2004), que estudaram uma empresa onde a perda de material estava entre 10% e 15%, que equivale a 500 e 750 toneladas de aço perdidas por mês. A formulação matemática deste problema pode ser encontrada em Hoto (2001) e Hoto et al. (2007).

Para o problema de corte de bobinas de aço, o Problema da Mochila Compartimentada determina um padrão de corte (colunas a serem incluídas no problema mestre durante o processo de Geração de Colunas), porém, em alguns casos, outras restrições podem ser incluídas ao modelo (3.1) - (3.6), como o limite de itens em um compartimento e o número de compartimentos em uma mochila, que é determinado pela limitação do número de facas e podem ser representadas por:

$$\sum_{i \in A_k} a_{isk} \leq f$$

$$\sum_{s=1}^{N_k} \sum_{k=1}^K \beta_{sk} \leq F,$$

em que, f representa o número máximo de itens permissível em um compartimento e F representa o número máximo de compartimentos em uma mochila.

Uma nova heurística para o Problema da Mochila Compartimentada

Neste capítulo é apresentada uma nova heurística para o Problema da Mochila Compartimentada, que consiste na combinação de duas heurísticas da literatura: a Heurística dos “Z” Melhores Compartimentos e a Heurística do Melhor Compartimento para “W” Capacidades; propostas em Marques & Arenales (2007). Nestas heurísticas, o modelo não-linear (3.1) - (3.6) é decomposto, obtendo-se dois modelos lineares, no primeiro são gerados os compartimentos e, no segundo os compartimentos e itens livres são selecionados para formar a mochila compartimentada.

4.1 Duas heurísticas da literatura

4.1.1 Heurística dos “Z” Melhores Compartimentos

A heurística dos “Z” Melhores Compartimentos consiste em determinar os Z melhores compartimentos para cada classe de itens, ou seja, as Z melhores soluções para o Problema da Mochila Restrita (4.1) - (4.3). Em seguida, é resolvido o problema (4.4) - (4.8), determinando quais serão os compartimentos e itens livres que formarão a mochila. Os itens livres pertencem a classe K , o custo de incluir um compartimento desta classe é zero e os limitantes para o tamanho do compartimento são dados por $[L_{min}^k, L_{max}^k] = [0, L]$, assim não é necessário resolver o Problema da Mochila Restrita para gerar compartimentos para essa classe, o algoritmo é dado como segue:

Algoritmo: “Z” Melhores Compartimentos

1. Para $k = 1, \dots, K - 1$ determine as Z melhores soluções para o seguinte problema:

$$\text{maximize} \quad \sum_{i \in A_k} v_i a_{ik} \quad (4.1)$$

$$\text{sujeito a:} \quad \sum_{i \in A_k} l_i a_{ik} + S_k \leq L_{max}^k \quad (4.2)$$

$$0 \leq a_{ik} \leq \sigma_i, \text{ inteiro, } i \in A_k, \quad (4.3)$$

obtendo $\hat{a}_{izk} = (a_{ik})_{i \in A_k}$, $V_{zk} = \sum_{i \in A_k} v_i \hat{a}_{izk}$ e $L_{zk} = \sum_{i \in A_k} l_i \hat{a}_{izk} + S_k$, em que z corresponde a z -ésima solução, tal que V_{zk} é o z -ésimo melhor valor da função objetivo, isto é, $V_{1k} \geq V_{2k} \geq \dots \geq V_{Zk}$.

Observação: Se $L_{zk} < L_{min}^k$ a solução é descartada.

2. Resolva o seguinte Problema Linear Inteiro considerando os $((K - 1) * Z)$ compartimentos obtidos no Passo 1 e os itens livres que pertencem ao conjunto A_K . Considere γ_k como a quantidade de itens livres do tipo k alocados a mochila com $k \in A_K$ e, β_{zk} o número de vezes que o z -ésimo compartimento formado por itens da classe k é incluído na mochila.

$$\text{maximize} \quad \sum_{k=1}^{K-1} \sum_{z=1}^Z (V_{zk} - c_k) \beta_{zk} + \sum_{k \in A_K} v_k \gamma_k \quad (4.4)$$

$$\text{sujeito a:} \quad \sum_{k=1}^{K-1} \sum_{z=1}^Z L_{zk} \beta_{zk} + \sum_{k \in A_K} l_k \gamma_k \leq L \quad (4.5)$$

$$\sum_{z=1}^Z \hat{a}_{izk} \beta_{zk} \leq \sigma_i, \forall i \in A_k, k = 1, \dots, K - 1 \quad (4.6)$$

$$0 \leq \gamma_k \leq \sigma_k, \text{ inteiro, } k \in A_K \quad (4.7)$$

$$\beta_{zk} \geq 0, \text{ inteiro, } k = 1, \dots, K - 1, z = 1, \dots, Z. \quad (4.8)$$

Para determinar as Z melhores soluções para o Problema da Mochila Restrita foi utilizado um método de enumeração implícita baseado no método proposto por Gilmore & Gomory (1963), que determina a melhor solução para o Problema Clássico da Mochila. Neste algoritmo, conforme a árvore é percorrida, as Z melhores soluções obtidas até o momento são armazenadas, que são atualizadas à medida que soluções melhores são encontradas.

4.1.2 Heurística do Melhor Compartimento para “W” Capacidades

A heurística do Melhor Compartimento para “W” Capacidades determina o melhor compartimento considerando W possíveis capacidades diferentes para cada classe de itens. No segundo passo, resolve-se o problema (4.4) - (4.8), porém Z é trocado por W , cuja solução fornece os compartimentos e itens livres para preencher a mochila, o algoritmo é dado a seguir.

Algoritmo: Melhor Compartimento para “W” Capacidades

1. Determine o melhor compartimento para cada uma das W capacidades para a classe k . Para $k = 1, \dots, K - 1$, faça:

1.1 Seja $L_{comp} = L_{max}^k$ e $w = 1$,

1.2 Enquanto $w \leq W$ e $L_{comp} \geq L_{min}^k$ faça:

1.2.1 Resolva:

$$V_{wk} = \text{maximize} \sum_{i \in A_k} v_i a_{iwk} \quad (4.9)$$

$$\text{sujeito a:} \quad \sum_{i \in A_k} l_i a_{iwk} + S_k \leq L_{comp} \quad (4.10)$$

$$0 \leq a_{iwk} \leq \sigma_i, \text{ inteiro, } i \in A_k, \quad (4.11)$$

1.2.2 Atualizações: $L_{wk} = \sum_{i \in A_k} l_i \hat{a}_{iwk} + S_k$, se $L_{wk} < L_{min}^k$ a solução é descartada,

$$L_{comp} = L_{wk} - 1,$$

$$w = w + 1.$$

2. Resolva o Problema Linear Inteiro (4.4) - (4.8) considerando os $((K - 1) * W)$ compartimentos obtidos no Passo 1 e os itens livres, entretanto, considere β_{wk} com $w = 1, \dots, W$, ao invés de β_{zk} e, \hat{a}_{iwk} ao invés de \hat{a}_{izk} .

4.2 Heurística Híbrida

Baseado nas heurísticas de Marques & Arenales (2007) apresentadas na seção anterior, nesta seção é proposta uma nova heurística, que consiste na combinação destas. A Heurística Híbrida

determina um conjunto dos melhores compartimentos com diferentes capacidades para cada classe de itens.

Para a descrição da heurística, considere Z o número de melhores soluções desejadas, W o número de diferentes capacidades para os compartimentos e β_k^{zw} o número de vezes que o compartimento formado por itens da classe k com a w -ésima capacidade da z -ésima melhor solução, é incluído na mochila. A estratégia combinada pode ser descrita por:

Algoritmo: “Z” Melhores Compartimentos/ “W” Capacidades

1. Geração de compartimentos alternativos.

1.1 Para $k = 1, \dots, K - 1$ faça:

1.2 $w = 1$ e $L_w^k = L_{max}^k$.

1.3 Enquanto $w \leq W$ e $L_w^k \geq L_{min}^k$ faça:

1.4 Determine as Z melhores soluções do seguinte Problema da Mochila:

$$\text{maximize} \quad \sum_{i \in A_k} v_i a_{ik}^w \quad (4.12)$$

$$\text{sujeito a:} \quad \sum_{i \in A_k} l_i a_{ik}^w + S_k \leq L_w^k \quad (4.13)$$

$$0 \leq a_{ik}^w \leq \sigma_i, \text{ inteiro } \forall i \in A_k, k = 1, \dots, K - 1, \quad (4.14)$$

$$\text{obtendo } \hat{a}_{ik}^{zw} = (a_{ik}^w)_{i \in A_k}, V_k^{zw} = \sum_{i \in A_k} v_i \hat{a}_{ik}^{zw}, L_k^{zw} = \sum_{i \in A_k} l_i \hat{a}_{ik}^{zw} + S_k,$$

para $z = 1, \dots, Z$.

Observação: Se $L_k^{zw} < L_{min}^k$ então a solução é descartada.

1.5 Redefina: $L_{w+1}^k = \bar{L} - 1$ e $w = w + 1$, sendo $\bar{L} = \min\{L_k^{zw}, z = 1, \dots, Z\}$.

2. Solução para a mochila compartimentada.

$$\text{maximize} \quad \sum_{k=1}^{K-1} \sum_{z=1}^Z \sum_{w=1}^W (V_k^{zw} - c_k) \beta_k^{zw} + \sum_{k \in A_K} v_k \gamma_k \quad (4.15)$$

$$\text{sujeito a:} \quad \sum_{k=1}^{K-1} \sum_{z=1}^Z \sum_{w=1}^W L_k^{zw} \beta_k^{zw} + \sum_{k \in A_K} l_k \gamma_k \leq L \quad (4.16)$$

$$\sum_{z=1}^Z \sum_{w=1}^W \hat{a}_{ik}^{zw} \beta_k^{zw} \leq \sigma_i, \forall i \in A_k, k = 1, \dots, K - 1 \quad (4.17)$$

$$0 \leq \gamma_k \leq \sigma_k, \text{ inteiro, } k \in A_K \quad (4.18)$$

$$\beta_k^{zw} \geq 0, \text{ inteiro, } k = 1, \dots, K - 1, z = 1, \dots, Z. \quad (4.19)$$

O Passo **1.5** da Heurística Híbrida pode ser substituído pelo seguinte passo:

$$\mathbf{1.5}': L_{w+1}^k = L_k^{1w} - 1 \text{ e } w = w + 1,$$

isto é, o próximo compartimento a ser gerado é decrementado uma unidade do tamanho do melhor compartimento obtido, ao invés do menor tamanho obtido. Isso fornece diferentes capacidades para os compartimentos.

4.3 Limitante superior para o Problema da Mochila Compartimentada

Em Marques & Arenales (2007) é apresentado um limitante superior para o Problema da Mochila Compartimentada (3.1) - (3.6), que consiste em um Problema de Programação Linear Inteiro, o qual considera o custo, a perda de incluir um compartimento na mochila, bem como o arranjo dos itens nos compartimentos. Esta relaxação será utilizada para obter o limitante superior para as heurísticas implementadas. Para a formulação matemática, considere a_i o número de itens do tipo i ($i = 1, \dots, n$) e β_k o número de compartimentos preenchidos por itens da classe k ($k = 1, \dots, K$) a serem incluídos na mochila, o modelo é dado por:

$$\text{maximize} \quad \sum_{i=1}^n v_i a_i - \sum_{k=1}^K c_k \beta_k \quad (4.20)$$

$$\text{sujeito a :} \quad \sum_{i=1}^n l_i a_i \leq L - \left(\sum_{k=1}^K S_k \beta_k \right) \quad (4.21)$$

$$L_{min}^k \beta_k \leq \sum_{i \in A_k} l_i a_i + S_k \beta_k \leq L_{max}^k \beta_k, \quad k = 1, \dots, K \quad (4.22)$$

$$0 \leq a_i \leq \sigma_i, \text{ inteiro}, \quad i = 1, \dots, n \quad (4.23)$$

$$\beta_k \geq 0, \text{ inteiro}, \quad k = 1, \dots, K. \quad (4.24)$$

Note que, as restrições (4.22) consideram para cada classe k um compartimento de tamanho entre $L_{min}^k \beta_k$ e $L_{max}^k \beta_k$, o qual não necessariamente pode ser dividido em β_k compartimentos que possam conter o itens selecionados. Em Marques & Arenales (2007) é demonstrado que este problema fornece um limitante superior para o problema (3.1) - (3.6).

4.4 Gerador de dados

As heurísticas foram testadas com exemplos gerados aleatoriamente, como descrito em Marques & Arenales (2007), apenas o tamanho da mochila e a perda por incluir um novo compartimento foram fixados em tamanhos diferentes. A medida utilizada foi o comprimento da mochila, o gerador descrito em Marques & Arenales (2007) é dado a seguir.

Parâmetros fixos:

- $L = 1200$ mm;
- $[L_{min}^k, L_{max}^k] = [154mm, 456mm]$, $k = 1, \dots, K - 1$;
- $S_k = 4$ mm, $k = 1, \dots, K - 1$ e $S_K = 0$ mm.

Parâmetros gerados aleatoriamente:

- $l_i \in [20mm, 444mm]$, $i \in A_k$, $k = 1, \dots, K$;
- $v_i = l_i + \varepsilon$, $\varepsilon \in [0, l_i]$, $i \in A_k$, $k = 1, \dots, K$;
- $c_k \in [1, 100]$, $k = 1, \dots, K - 1$ e $c_K = 0$.

Parâmetros variáveis (foram consideradas duas possibilidades para cada parâmetro):

• Número de classes:

- (i) número pequeno de classes (P): $K \in [3, 10]$;
- (ii) número médio de classes (M): $K \in [11, 20]$.

• Número de itens por classe:

- (i) número pequeno de itens (P): $|A_k| \in [1, 6]$, $k = 1, \dots, K - 1$;
- (ii) número médio de itens (M): $|A_k| \in [7, 15]$, $k = 1, \dots, K - 1$.

• Número de itens livres:

- (i) número pequeno de itens livres (P): $|A_K| \in [1, 7]$;
- (ii) número médio de itens livres (M): $|A_K| \in [8, 20]$.

• Limitantes para o número de itens a serem incluídos na mochila:

- (i) limitante pequeno (P): $\sigma_i \in [1, 3]$, $i \in A_k$, $k = 1, \dots, K$;
- (ii) limitante médio (M): $\sigma_i \in [4, 15]$, $i \in A_k$, $k = 1, \dots, K$.

Assim foram obtidos 16 tipos de problemas distribuídos de acordo com as características abaixo:

- (i) número de classes;
- (ii) número de itens por classe;
- (iii) número de itens livres;
- (iv) limites sobre o número de itens.

A Tabela 4.1 mostra as características de cada tipo de problema (classe) gerado, detalhando cada parâmetro.

Tabela 4.1: Arranjo dos problemas

Classe	número de classes	número de itens	número de itens livres	Demanda
1	P	P	P	P
2	P	P	P	M
3	P	P	M	P
4	P	P	M	M
5	P	M	P	P
6	P	M	P	M
7	P	M	M	P
8	P	M	M	M
9	M	P	P	P
10	M	P	P	M
11	M	P	M	P
12	M	P	M	M
13	M	M	P	P
14	M	M	P	M
15	M	M	M	P
16	M	M	M	M

4.5 Experimentos computacionais

A finalidade dos testes computacionais foi analisar o comportamento da heurística proposta em relação aos parâmetros Z e W . Para cada classe de problemas foram gerados 20 exemplos. Para o cálculo do *gap* foi utilizado como limitante superior o problema (4.20) - (4.24), que foi resolvido pelo *software* ILOG CPLEX 11.0, o qual determinou uma solução ótima rapidamente. O *gap* é dado por:

$$\frac{\bar{f} - f}{f},$$

em que, \bar{f} corresponde o valor ótimo da função objetivo (4.20) e f o valor da função objetivo encontrada pela heurística.

As heurísticas foram implementadas em linguagem de programação C. Os testes computacionais foram realizados em um computador Pentium 4 com processador de 3GHz e 1GB de memória RAM. Para resolver os problemas (4.12) - (4.14) e (4.15) - (4.19) foram implementados dois métodos, que são extensões do algoritmo de enumeração implícita de Gilmore & Gomory (1963). O método proposto para o problema (4.15) - (4.19) é uma heurística e para analisar seu desempenho o problema, também, foi resolvido pelo *software* ILOG CPLEX 11.0. Os algoritmos implementados estão descritos no Capítulo 6.

Vários testes foram realizados variando os parâmetros Z e W da Heurística Híbrida: $Z = 3, 5$ e 7 e $W = 2, 3, 5$ e 7 . As Tabelas 4.3, 4.4, 4.5, 4.6, 4.7 e 4.8 mostram a média dos *gaps* dos exemplos gerados. Na Tabela 4.2 têm-se os resultados obtidos com a heurística Melhor Compartimento para “W” Capacidades com $W = 2, 3, 5$ e 7 (Heurística Híbrida com $Z = 1$). Em Marques & Arenales (2007), a heurística Melhor Compartimento para “W” Capacidades com $W = 5$ teve o melhor desempenho dentre todas as heurísticas analisadas pelos autores.

Tabela 4.2: Valores médios do *gap* (%) para a heurística Melhor Compartimento para W Capacidades.

Z=1	Problema (4.15) - (4.19) heurísticamente				Problema (4.15) - (4.19) exatamente			
classe	W=2	W=3	W=5	W=7	W=2	W=3	W=5	W=7
1	1,675	0,955	0,416	0,290	1,675	0,955	0,416	0,290
2	1,935	1,007	0,731	0,617	1,879	0,951	0,670	0,556
3	0,376	0,228	0,173	0,173	0,290	0,143	0,088	0,088
4	0,538	0,450	0,100	0,100	0,538	0,450	0,100	0,100
5	4,378	3,410	1,296	0,879	4,378	3,410	1,296	0,879
6	3,464	2,509	1,545	0,958	3,464	2,335	1,545	0,958
7	1,231	0,791	0,446	0,308	1,199	0,759	0,414	0,276
8	0,943	0,521	0,283	0,185	0,943	0,521	0,283	0,185
9	1,931	0,973	0,624	0,437	1,931	0,973	0,624	0,437
10	1,375	0,964	0,479	0,431	1,375	0,964	0,479	0,431
11	0,467	0,216	0,074	0,056	0,397	0,216	0,074	0,056
12	0,641	0,408	0,261	0,214	0,504	0,355	0,208	0,162
13	4,565	2,785	0,830	0,542	4,565	2,785	0,755	0,542
14	3,744	2,501	1,171	0,786	3,744	2,501	1,171	0,786
15	1,267	1,004	0,547	0,310	1,267	1,004	0,547	0,310
16	1,113	0,843	0,516	0,346	1,113	0,843	0,516	0,346
Média	1,853	1,223	0,593	0,414	1,829	1,198	0,574	0,400

Tabela 4.3: Valores médios do *gap* (%) considerando o Passo 1.5

Z=3	Problema (4.15) - (4.19) heurísticamente				Problema (4.15) - (4.19) exatamente			
classe	W=2	W=3	W=5	W=7	W=2	W=3	W=5	W=7
1	0,449	0,186	0,042	0,042	0,449	0,186	0,042	0,042
2	0,716	0,539	0,535	0,535	0,655	0,478	0,478	0,478
3	0,173	0,173	0,173	0,173	0,088	0,088	0,088	0,088
4	0,145	0,100	0,100	0,100	0,145	0,100	0,100	0,100
5	1,734	0,908	0,576	0,343	1,734	0,908	0,576	0,343
6	2,045	1,336	0,756	0,440	1,974	1,336	0,756	0,440
7	0,611	0,359	0,079	0,060	0,579	0,327	0,047	0,043
8	0,634	0,290	0,157	0,157	0,634	0,290	0,157	0,157
9	0,604	0,453	0,348	0,348	0,604	0,453	0,348	0,348
10	0,527	0,431	0,336	0,274	0,527	0,431	0,336	0,274
11	0,127	0,056	0,045	0,045	0,127	0,056	0,045	0,045
12	0,337	0,174	0,143	0,095	0,284	0,121	0,090	0,042
13	1,114	0,547	0,442	0,297	1,114	0,547	0,442	0,297
14	1,354	0,937	0,631	0,398	1,354	0,937	0,631	0,398
15	0,888	0,254	0,217	0,148	0,888	0,254	0,217	0,148
16	0,615	0,429	0,226	0,128	0,615	0,429	0,186	0,128
Média	0,755	0,448	0,300	0,224	0,736	0,434	0,284	0,211

Tabela 4.4: Valores médios do *gap* (%) considerando o Passo 1.5'

Z=3	Problema (4.15) - (4.19) heurísticamente				Problema (4.15) - (4.19) exatamente			
classe	W=2	W=3	W=5	W=7	W=2	W=3	W=5	W=7
1	0,885	0,416	0,347	0,042	0,885	0,416	0,347	0,042
2	1,014	0,796	0,613	0,535	0,957	0,735	0,556	0,478
3	0,228	0,173	0,173	0,173	0,143	0,088	0,088	0,088
4	0,377	0,169	0,100	0,100	0,377	0,169	0,100	0,100
5	3,332	1,690	1,014	0,643	3,332	1,690	1,014	0,643
6	2,578	2,005	1,096	0,741	2,345	2,005	1,096	0,741
7	0,820	0,573	0,351	0,166	0,788	0,541	0,319	0,134
8	0,859	0,432	0,266	0,185	0,859	0,432	0,266	0,185
9	0,884	0,761	0,453	0,437	0,884	0,761	0,453	0,437
10	0,791	0,495	0,431	0,381	0,791	0,495	0,431	0,381
11	0,196	0,185	0,056	0,056	0,196	0,185	0,056	0,056
12	0,390	0,344	0,219	0,169	0,337	0,291	0,166	0,116
13	2,247	1,229	0,564	0,500	2,247	1,229	0,564	0,500
14	2,607	1,500	0,815	0,723	2,607	1,500	0,815	0,723
15	1,020	0,815	0,356	0,230	1,020	0,815	0,356	0,230
16	0,909	0,615	0,459	0,291	0,909	0,615	0,459	0,291
Média	1,196	0,762	0,457	0,336	1,167	0,748	0,443	0,322

Tabela 4.5: Valores médios do *gap* (%) considerando o Passo 1.5

Z=5	Problema (4.15) - (4.19) heurísticamente				Problema (4.15) - (4.19) exatamente			
classe	W=2	W=3	W=5	W=7	W=2	W=3	W=5	W=7
1	0,100	0,100	0,042	0,042	0,100	0,100	0,042	0,042
2	0,539	0,535	0,535	0,535	0,478	0,478	0,478	0,478
3	0,173	0,173	0,173	0,173	0,088	0,088	0,088	0,088
4	0,100	0,100	0,078	0,078	0,100	0,100	0,078	0,078
5	0,979	0,692	0,368	0,345	0,979	0,692	0,368	0,345
6	1,548	0,868	0,543	0,230	1,548	0,868	0,543	0,230
7	0,431	0,145	0,060	0,043	0,399	0,113	0,043	0,026
8	0,396	0,209	0,157	0,118	0,396	0,209	0,157	0,118
9	0,453	0,364	0,348	0,348	0,453	0,364	0,348	0,348
10	0,440	0,338	0,274	0,274	0,440	0,338	0,274	0,274
11	0,067	0,045	0,013	0,000	0,067	0,045	0,013	0,000
12	0,219	0,143	0,095	0,095	0,166	0,090	0,042	0,042
13	0,684	0,405	0,263	0,182	0,684	0,405	0,263	0,182
14	1,109	0,731	0,398	0,329	1,109	0,731	0,398	0,329
15	0,343	0,223	0,160	0,048	0,343	0,223	0,160	0,048
16	0,439	0,319	0,117	0,115	0,439	0,275	0,117	0,115
Média	0,501	0,337	0,227	0,185	0,487	0,320	0,213	0,171

Tabela 4.6: Valores médios do *gap* (%) considerando o Passo 1.5'

Z=5	Problema (4.15) - (4.19) heurísticamente				Problema (4.15) - (4.19) exatamente			
classe	W=2	W=3	W=5	W=7	W=2	W=3	W=5	W=7
1	0,416	0,397	0,100	0,042	0,416	0,397	0,100	0,042
2	0,727	0,727	0,535	0,535	0,670	0,670	0,478	0,478
3	0,173	0,173	0,173	0,173	0,088	0,088	0,088	0,088
4	0,289	0,169	0,100	0,100	0,289	0,169	0,100	0,100
5	1,981	1,103	0,862	0,643	1,981	1,103	0,862	0,643
6	2,245	1,720	1,057	0,637	2,245	1,720	1,057	0,637
7	0,636	0,504	0,278	0,098	0,604	0,472	0,246	0,066
8	0,517	0,432	0,185	0,147	0,517	0,432	0,185	0,147
9	0,604	0,453	0,453	0,391	0,604	0,453	0,453	0,391
10	0,479	0,479	0,431	0,381	0,479	0,479	0,431	0,381
11	0,172	0,071	0,056	0,045	0,172	0,071	0,056	0,045
12	0,390	0,241	0,214	0,169	0,337	0,188	0,162	0,116
13	1,786	0,777	0,516	0,457	1,786	0,693	0,516	0,457
14	1,567	1,404	0,786	0,723	1,567	1,404	0,786	0,723
15	1,004	0,514	0,239	0,217	1,004	0,514	0,239	0,217
16	0,729	0,527	0,371	0,197	0,729	0,527	0,371	0,197
Média	0,857	0,606	0,397	0,310	0,843	0,586	0,383	0,295

Tabela 4.7: Valores médios do *gap* (%) considerando o Passo 1.5

Z=7	Problema (4.15) - (4.19) heuristicamente				Problema (4.15) - (4.19) exatamente			
classe	W=2	W=3	W=5	W=7	W=2	W=3	W=5	W=7
1	0,100	0,042	0,042	0,042	0,100	0,042	0,042	0,042
2	0,535	0,535	0,535	0,521	0,478	0,478	0,478	0,465
3	0,173	0,173	0,173	0,173	0,088	0,088	0,088	0,088
4	0,100	0,100	0,078	0,078	0,100	0,100	0,078	0,078
5	0,887	0,564	0,325	0,296	0,887	0,564	0,325	0,296
6	1,136	0,728	0,368	0,209	1,136	0,728	0,368	0,209
7	0,259	0,126	0,043	0,043	0,227	0,094	0,026	0,026
8	0,252	0,181	0,157	0,118	0,252	0,181	0,157	0,118
9	0,364	0,348	0,348	0,348	0,364	0,348	0,348	0,348
10	0,391	0,277	0,274	0,274	0,391	0,277	0,274	0,274
11	0,056	0,045	0,013	0,000	0,056	0,045	0,013	0,000
12	0,174	0,143	0,095	0,095	0,121	0,090	0,042	0,042
13	0,684	0,403	0,257	0,182	0,684	0,403	0,257	0,182
14	0,893	0,580	0,398	0,267	0,893	0,580	0,398	0,267
15	0,245	0,217	0,048	0,044	0,245	0,217	0,048	0,044
16	0,364	0,182	0,117	0,076	0,364	0,182	0,117	0,076
Média	0,413	0,290	0,204	0,173	0,399	0,276	0,191	0,160

Tabela 4.8: Valores médios do *gap* (%) considerando o Passo 1.5'.

Z=7	Problema (4.15) - (4.19) heuristicamente				Problema (4.15) - (4.19) exatamente			
classe	W=2	W=3	W=5	W=7	W=2	W=3	W=5	W=7
1	0,416	0,100	0,042	0,042	0,416	0,100	0,042	0,042
2	0,727	0,711	0,535	0,535	0,670	0,655	0,478	0,478
3	0,173	0,173	0,173	0,173	0,088	0,088	0,088	0,088
4	0,169	0,169	0,100	0,100	0,169	0,169	0,100	0,100
5	1,402	1,008	0,837	0,614	1,402	1,008	0,837	0,614
6	2,120	1,452	0,888	0,600	2,120	1,452	0,888	0,600
7	0,530	0,504	0,179	0,098	0,498	0,472	0,147	0,066
8	0,517	0,432	0,185	0,118	0,517	0,432	0,185	0,118
9	0,464	0,453	0,391	0,348	0,464	0,453	0,391	0,348
10	0,479	0,440	0,381	0,274	0,479	0,440	0,381	0,274
11	0,145	0,056	0,056	0,045	0,145	0,056	0,056	0,045
12	0,326	0,219	0,169	0,143	0,273	0,166	0,116	0,090
13	1,396	0,777	0,516	0,437	1,396	0,693	0,516	0,437
14	1,469	1,262	0,786	0,649	1,469	1,262	0,786	0,649
15	0,607	0,282	0,230	0,217	0,607	0,282	0,230	0,217
16	0,527	0,516	0,319	0,147	0,527	0,516	0,319	0,147
Média	0,717	0,535	0,362	0,284	0,703	0,515	0,348	0,269

Os resultados das tabelas estão resumidos nos gráficos contidos nas Figuras 4.1 e 4.2.

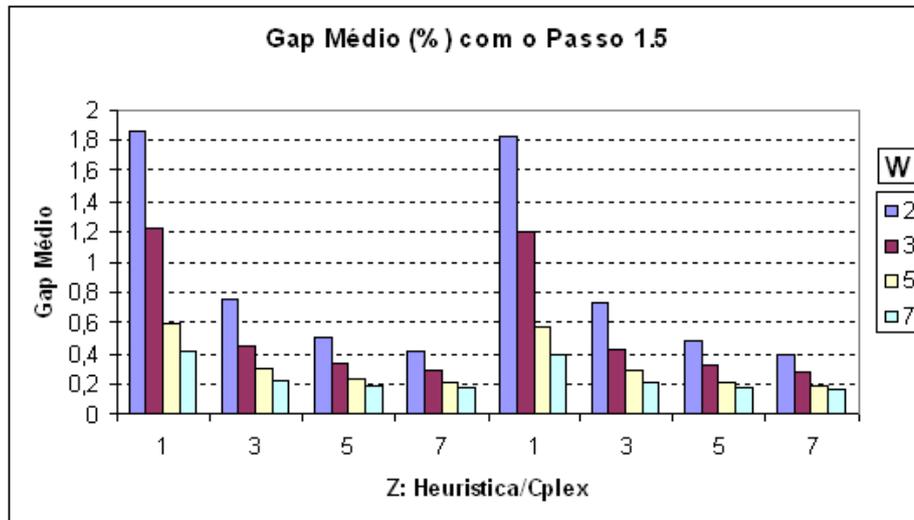


Figura 4.1: Gráfico do gap médio: à esquerda os resultados com o Problema (4.15) - (4.19) resolvido heurísticamente e à direita exatamente.

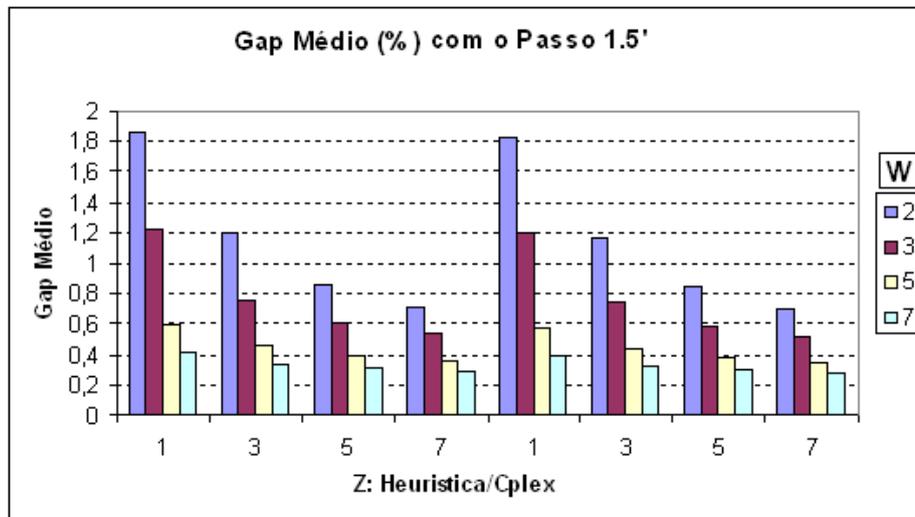


Figura 4.2: Gráfico do gap médio: à esquerda os resultados com o Problema (4.15) - (4.19) resolvido heurísticamente e à direita exatamente.

As Tabelas mostram que, à medida que Z e W aumentam, o valor da função objetivo melhora, além disso, aumentar W influencia mais o desempenho do método que aumentar Z . Ao resolver o problema (4.15) - (4.19) pelo *software* ILOG CPLEX, em algumas classes o *gap* diminui, apesar da heurística ser bastante eficiente. Em geral, o Passo 1.5 foi melhor que o Passo 1.5', pois o Passo 1.5' gera menos compartimentos. As Tabelas 4.9 e 4.10 mostram o tempo médio de resolução dos exemplos, em segundos, considerando os diversos parâmetros.

Tabela 4.9: Tempo computacional médio (s) considerando o Passo 1.5.

	Problema (4.15) - (4.19) heuristicamente				Problema (4.15) - (4.19) exatamente			
	W=2	W=3	W=5	W=7	W=2	W=3	W=5	W=7
Z=1	0,000	0,001	0,001	0,001	0,001	0,002	0,003	0,004
Z=3	0,001	0,001	0,001	0,003	0,004	0,006	0,008	0,010
Z=5	0,001	0,002	0,004	0,004	0,006	0,008	0,012	0,014
Z=7	0,001	0,003	0,005	0,006	0,008	0,010	0,014	0,016

Tabela 4.10: Tempo computacional médio (s) considerando o Passo 1.5'.

Z=3	Problema (4.15) - (4.19) heuristicamente				Problema (4.15) - (4.19) exatamente			
	W=2	W=3	W=5	W=7	W=2	W=3	W=5	W=7
Z=1	0,000	0,001	0,000	0,001	0,001	0,002	0,003	0,004
Z=3	0,001	0,001	0,003	0,005	0,003	0,004	0,007	0,009
Z=5	0,002	0,003	0,007	0,010	0,005	0,006	0,010	0,012
Z=7	0,002	0,004	0,011	0,017	0,006	0,007	0,012	0,017

Esta heurística é bastante rápida, como mostram as Tabelas 4.9 e 4.10, independentemente se o problema (4.15) - (4.19) é resolvido pelo *software* ILOG CPLEX ou pela heurística. Considerando os 320 problemas gerados, as Tabelas 4.11 e 4.12 mostram o número total de soluções ótimas obtidas por parâmetro.

Tabela 4.11: Número de soluções ótimas obtidas considerando o Passo 1.5

	Problema (4.15) - (4.19) heuristicamente				Problema (4.15) - (4.19) exatamente			
	W=2	W=3	W=5	W=7	W=2	W=3	W=5	W=7
Z=1	110	140	168	190	114	144	172	194
Z=3	155	184	219	230	159	188	223	240
Z=5	177	207	235	248	181	211	239	252
Z=7	189	219	242	256	193	223	246	260

Tabela 4.12: Número de soluções ótimas obtidas considerando o Passo 1.5'.

	Problema (4.15) - (4.19) heurísticamente				Problema (4.15) - (4.19) exatamente			
	W=2	W=3	W=5	W=7	W=2	W=3	W=5	W=7
Z=1	110	140	168	190	114	144	172	194
Z=3	140	156	184	202	144	160	188	200
Z=5	154	167	191	210	158	171	195	214
Z=7	159	175	198	220	163	179	202	224

Como se pode observar, o número de soluções ótimas tem um aumento significativo a medida que Z e W aumentam. Comparando o pior e o melhor caso, obtem-se uma melhora de 46,88% do número de soluções ótimas obtidas. Com a Heurística Híbrida foi possível obter até 81,25% de soluções ótimas, que foram dadas pela igualdade da função objetivo do limitante superior e da heurística. Este fato mostra a qualidade do limitante superior, dado pelo problema (4.20) - (4.24).

De maneira geral, a combinação da heurística do Melhor Compartimento para “W” Capacidades com a heurística das “Z” Melhores Soluções mostrou-se bastante eficiente, capaz de encontrar soluções ótimas para a maioria dos exemplos gerados.

Geração de Colunas para o Problema da Mochila Compartimentada

A resolução do Problema da Mochila Compartimentada apresenta algumas dificuldades devido à condição de integralidade de suas variáveis e a não-linearidade presente na função objetivo e em restrições. Se enumerássemos todos os compartimentos seria possível determinar a sua solução ótima e eliminaríamos a não linearidade. Entretanto, o número de compartimentos pode ser muito grande para problemas práticos, o que inviabiliza este procedimento.

Para lidar com estas dificuldades, neste capítulo descrevemos a técnica Geração de Colunas para o Problema da Mochila Compartimentada, em que o problema (3.1) - (3.6) é reformulado em um problema mestre com tantas colunas quantos forem os possíveis compartimentos. O subproblema, que gera colunas para o problema mestre, é um Problema da Mochila Restrita.

O método Geração de Colunas foi inicialmente usado por Manne (1958), Dantzig & Wolfe (1960) e Gilmore & Gomory (1961). Estes últimos autores aplicaram esta técnica para resolver Problemas de Corte de Estoque. A próxima seção descreve o método Geração de Colunas de Gilmore & Gomory (1961), na seção 5.2 é apresentada a Geração de Colunas para o Problema da Mochila Compartimentada.

5.1 Geração de Colunas

Para a apresentação do método, considere que existe um conjunto de objetos de largura L em estoque e um conjunto de pedidos formado por itens de largura l_i e demanda b_i , $i = 1, \dots, m$.

O Problema de Corte de Estoque Unidimensional consiste em cortar os objetos em estoque para produzir os itens demandados e pode ter como objetivo minimizar a perda, maximizar o lucro, minimizar o número de objetos cortados etc. A formulação matemática apresentada em Gilmore & Gomory (1961) é dada por:

$$\text{minimize} \quad \sum_{j \in J} x_j \quad (5.1)$$

$$\text{sujeito a :} \quad \sum_{j \in J} a_{ij} x_j \geq b_i, \quad i = 1, \dots, m \quad (5.2)$$

$$x_j \geq 0, \text{ inteiro}, \quad j \in J, \quad (5.3)$$

em que, J representa o conjunto de padrões de corte, a_{ij} representa o número de itens do tipo i no padrão de corte j e x_j o número de vezes que o padrão de corte j é utilizado. O objetivo do problema é minimizar o número total de objetos a serem cortados. Uma coluna da formulação (5.1) - (5.3) representa um padrão de corte, ou seja, é um vetor $\alpha_j = (a_{1j}, \dots, a_{mj})$ que satisfaz as condições:

$$l_1 a_1 + l_2 a_2 + \dots + l_m a_m \leq L \quad (5.4)$$

$$a_i \geq 0, \text{ inteiro}, \quad i = 1, \dots, m. \quad (5.5)$$

A técnica Geração de Colunas proposta por Gilmore & Gomory (1961) consiste em relaxar a condição de integralidade das variáveis do problema (5.1) - (5.3) e resolver o problema resultante pelo método simplex. O número de variáveis pode chegar à ordem de milhões em problemas práticos, o que inviabiliza a enumeração explícita dos padrões de corte. Para contornar isso, ao invés de analisar todas as colunas para selecionar aquela a entrar na base, a coluna é gerada pela resolução de um Problema Clássico da Mochila, que surge no cálculo do menor custo relativo. Ao final do processo, a solução obtida pode não ser inteira e para obtê-la é utilizado outros métodos, como por exemplo, heurísticas de arredondamento (Wäscher & Gau, 1996; Poldi & Arenales, 2009).

No método simplex, devemos analisar os custos relativos das variáveis não-básicas para verificar a otimalidade da solução básica atual, para isso determinam-se as variáveis duais $\pi = (\pi_1, \dots, \pi_n)$ associadas as restrições do problema mestre e calcula-se o custo relativo, dado por (Regra de Dantzig):

$$c_k - \pi^t \alpha_k = \min\{c_j - \pi^t \alpha_j, \quad j \in J\},$$

ou seja, a variável com menor custo relativo negativo entra na base, caso $c_k - \pi^t \alpha_k \geq 0$, a solução obtida é ótima. Como não se conhece todas as colunas $\alpha_j, j \in J$. Surge assim o sub-problema para encontrar a coluna com o menor custo relativo. A função objetivo corresponde a:

$$\min\{c_j - \pi^t \alpha\} = \min\left\{1 - \sum_{j=1}^m \pi_j a_j\right\} = 1 - \max\left\{\sum_{j=1}^m \pi_j a_j\right\},$$

em que α corresponde a um padrão de corte, ou seja, satisfaz as condições (5.4) e (5.5), portanto, o subproblema é dado por:

$$\text{maximize} \quad \sum_{j=1}^m \pi_j a_j \quad (5.6)$$

$$\text{sujeito a : } l_1 a_1 + l_2 a_2 + \dots + l_m a_m \leq L \quad (5.7)$$

$$a_i \geq 0, \text{ inteiro}, i = 1, \dots, m. \quad (5.8)$$

O método Simplex com Geração de Colunas para o problema (5.1) - (5.3), com a condição de integralidade das variáveis relaxadas, pode ser descrito como segue.

Algoritmo: Método Simplex com Geração de Colunas

Fase 1: Determine uma matriz básica inicial B , formada por padrões de corte simples.

Faça Pare=Falso e iteração=0.

Fase 2: Enquanto Pare=Falso faça.

Determine a solução básica primal factível $x_B = B^{-1}b$.

Determine o vetor multiplicador simplex $\pi^t = c_B^t B^{-1}$.

Resolva o problema (5.6) - (5.8) e obtenha um padrão de corte, representado pela coluna α .

Se $1 - \pi^t \alpha \geq 0$, então: Solução ótima foi encontrada.

Pare=Verdadeiro.

Senão :

Determine as coordenadas básicas da direção simplex:

$$y = B^{-1}\alpha.$$

Determine a coluna l que sai da base:

$$\frac{x_{B_l}}{y_l} = \min\left\{\frac{x_{B_i}}{y_i} \mid y_i > 0, i = 1, \dots, m\right\}.$$

Substitua a l -ésima coluna pela coluna gerada.

Faça: iteração = iteração + 1.

Fim Enquanto.

5.2 Geração de Colunas para o Problema da Mochila Compartimentada

Para a descrição do método, considere que todos os possíveis compartimentos para a classe k ($k = 1, \dots, K$) é dado por N_k , isto é todas as soluções que satisfazem:

$$L_{min}^k \leq \sum_{i \in A_k} l_i a_i + S_k \leq L_{max}^k$$

$$0 \leq a_i \leq \sigma_i, \text{ inteiro, } i \in A_k.$$

Considere, também, que os itens livres pertencem a uma única classe A_K . Para esta classe o limitante inferior e superior para sua alocação na mochila são dados por $L_{min}^K = 0$ e $L_{max}^K = L$, respectivamente, e o custo de alocar um compartimento é dado por $c_K = 0$. Por conveniência de notação, o conjunto de índice dos itens será definido por: $N = \{n_0 + 1, \dots, n_1, n_1 + 1, \dots, n_2, \dots, n_{K-1} + 1, \dots, n_K\}$, e as classes $A_1 = \{n_0 + 1, \dots, n_1\}$, $A_2 = \{n_1 + 1, \dots, n_2\}$, ..., $A_K = \{n_{K-1} + 1, \dots, n_K\}$, em que $n_0 = 0$ e $n_K = n$. Com isto, podemos omitir o índice k em a_{isk} , pois um tipo i está na classe k se: $n_{k-1} + 1 \leq i \leq n_k$. Assim, o problema (3.1) - (3.6) é equivalentemente escrito por:

Problema Mestre:

$$\text{maximize} \quad \sum_{k=1}^K \sum_{s=1}^{N_k} \left(\sum_{i=n_{k-1}+1}^{n_k} v_i a_{is} - c_k \right) \beta_{sk} \quad (5.9)$$

$$\text{sujeito a :} \quad \sum_{k=1}^K \sum_{s=1}^{N_k} \left(\sum_{i=n_{k-1}+1}^{n_k} l_i a_{is} + S_k \right) \beta_{sk} \leq L \quad (5.10)$$

$$\sum_{s=1}^{N_k} a_{is} \beta_{sk} \leq \sigma_i, \quad k = 1, \dots, K, \quad i = n_{k-1} + 1, \dots, n_k \quad (5.11)$$

$$\beta_{sk} \geq 0, \text{ inteiro, } k = 1, \dots, K, \quad t = 1, \dots, N_k. \quad (5.12)$$

O problema mestre tem um número muito grande de variáveis, porém com uma lei de formação para as colunas relativamente simples, de modo que a técnica de geração de colunas pode ser aplicada. Sejam $L_{sk} = \sum_{i=n_{k-1}+1}^{n_k} l_i a_{is} + S_k$ e $V_{sk} = \sum_{i=n_{k-1}+1}^{n_k} v_i a_{is}$, a estrutura matricial do problema (5.9) - (5.12) é dada por:

$V_{11} - c_1$	\dots	$V_{1N_1} - c_1$	$V_{21} - c_2$	\dots	$V_{2N_2} - c_2$	\dots	$V_{K1} - c_K$	\dots	V_{KN_K}
L_{11}	\dots	L_{1N_1}	L_{21}	\dots	L_{2N_2}	\dots	L_{K1}	\dots	L_{KN_K}
$a_{(n_0+1)1}$	\dots	$a_{(n_0+1)N_1}$	0	\dots	0	\dots	0	\dots	0
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
a_{n_11}	\dots	$a_{n_1N_1}$	0	\dots	0	\dots	0	\dots	\vdots
0	\dots	0	$a_{(n_1+1)1}$	\dots	$a_{(n_1+1)N_2}$	\dots	0	\dots	0
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
0	\dots	0	a_{n_21}	\dots	$a_{n_2N_2}$	\dots	0	\dots	0
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\dots	\vdots	\vdots	\vdots
0	\dots	0	0	\dots	0	\dots	$a_{(n_{K-1}+1)1}$	\dots	$a_{(n_{K-1}+1)N_K}$
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
0	\dots	0	0	\dots	0	\dots	a_{n_K1}	\dots	$a_{n_KN_K}$

Observe que existem K subproblemas, um para cada classe de itens. Para determinar quais colunas entrarão na base, os custos relativos são analisados, para isso consideremos as seguintes observações.

1. O vetor das variáveis duais é dado por:

$$(\lambda, \pi) = (\lambda, \pi^1, \pi^2, \dots, \pi^K) = (\lambda, \pi_{n_0+1}, \dots, \pi_{n_1}, \pi_{n_1+1}, \dots, \pi_{n_2}, \dots, \pi_{n_{K-1}+1}, \dots, \pi_{n_K}).$$

A variável λ está associada a restrição (5.10) e π as restrições (5.11).

2. A j -ésima coluna do modelo (5.9) - (5.12) associada à classe k , omitindo-se o índice j , é dada por: $\alpha_k = (L_k, 0, \dots, 0, a_{n_{k-1}+1}, \dots, a_{n_k}, 0, \dots, 0)$.

3. Como o problema é de maximização, deseja-se determinar o maior custo relativo. Para obter a expressão dos custos relativos considere o seguinte desenvolvimento para cada classe k , $k = 1, \dots, K$ (o índice j é omitido):

$$C_{\bar{k}} - \Pi^t \alpha_{\bar{k}} = \max\{C_k - (\lambda, \pi)^t \alpha_k\}$$

$$C_{\bar{k}} - \Pi^t \alpha_{\bar{k}} = \max\{(V_k - c_k) - \left(\lambda, (\pi^1, \dots, \pi^k, \dots, \pi^K) \right) \begin{pmatrix} L_k \\ 0 \\ \vdots \\ a_{n_{k-1}+1} \\ \vdots \\ a_{n_k} \\ \vdots \\ 0 \end{pmatrix}\}.$$

$$C_{\bar{k}} - \Pi^t \alpha_{\bar{k}} = \max\{(V_k - c_k) - \lambda L_k - (\pi_{n_{k-1}+1}, \dots, \pi_{n_k}) \begin{pmatrix} a_{n_{k-1}+1} \\ \vdots \\ a_{n_k} \end{pmatrix}\}.$$

$$\text{Como } V_k = \sum_{i=n_{k-1}+1}^{n_k} v_i a_i \text{ e } L_k = \sum_{i=n_{k-1}+1}^{n_k} l_i a_i + S_k,$$

substituindo, tem-se:

$$C_k - \Pi^t \alpha_k = \max \left\{ \sum_{i=n_{k-1}+1}^{n_k} v_i a_i - c_k - \lambda \left(\sum_{i=n_{k-1}+1}^{n_k} l_i a_i + S_k \right) - \sum_{i=n_{k-1}+1}^{n_k} \pi_i a_i \right\}.$$

$$C_k - \Pi^t \alpha_k = \max \left\{ \sum_{i=n_{k-1}+1}^{n_k} (v_i - \lambda l_i - \pi_i) a_i - (c_k + \lambda S_k) \right\}.$$

Logo, para $k = 1, \dots, K$, tem-se o seguinte subproblema, dado por um Problema da Mochila Restrita:

$$\text{maximize} \quad \sum_{i=n_{k-1}+1}^{n_k} (v_i - \lambda l_i - \pi_i) a_i \quad (5.13)$$

$$\text{sujeito a :} \quad L_{min}^k \leq \sum_{i=n_{k-1}+1}^{n_k} l_i a_i + S_k \leq L_{max}^k \quad (5.14)$$

$$0 \leq a_i \leq \sigma_i, \text{ inteiro, } i = n_{k-1} + 1, \dots, n_k. \quad (5.15)$$

Em resumo, o subproblema (5.13) - (5.15) fornece os elementos para a construção de uma coluna associada de maior custo relativo para a classe k , $k = 1, \dots, K$. Se $\sum_{i=n_{k-1}+1}^{n_k} (v_i - \lambda l_i - \pi_i) a_i - (c_k + \lambda S_k) \leq 0$, $k = 1, \dots, K$, então a solução atual é ótima para a relaxação linear do problema mestre (5.9) - (5.12).

O algoritmo do Método Simplex com Geração de Colunas para o Problema da Mochila Compartimentada pode ser descrito como o algoritmo da seção 5.1, porém, na Fase 2 deverá ser resolvido o problema (5.13) - (5.15) para determinar a coluna $(L_K, 0, \dots, 0, a_{n_{k-1}+1}, \dots, a_{n_k}, 0, \dots, 0)$ a entrar na base, uma para cada classe. Este método fornece um limitante superior para o Problema da Mochila Compartimentada.

Baseado na heurística de decomposição de Marques (2004), em Hoto et al. (2005) é sugerido uma heurística com geração de colunas. Entretanto, a heurística proposta em Hoto et al. (2005) difere do método geração de colunas proposto neste trabalho em diversos aspectos. A heurística de Hoto et al. (2005) possui dois passos. No primeiro passo, para cada classe k ($k = 1, \dots, K$) são resolvidos vários problemas da mochila bidimensional, um para cada valor inteiro entre L_{min} e L_{max} , ou seja, determina compartimentos de capacidade $L_{comp} \in \{L_{min}, L_{min} + 1, \dots, L_{max}\}$ para cada classe k . Os problemas da mochila bidimensional possuem uma restrição de igualdade para o tamanho do compartimento L_{comp} , a segunda restrição

é devido a restrição do número de facas incluído pelos autores. Os limitantes superiores e inferiores dos compartimentos são iguais para todas as classes. No segundo passo é resolvido um problema linear inteiro, obtendo uma solução factível para o Problema da Mochila Compartimentada. Os autores relatam que no pior caso serão resolvidos $K(L_{max} - L_{min} + 1)$ problemas da mochila, que é o número total de compartimentos que o método considera, e pode ser excessivamente alto. Para evitar isto Hoto et al. (2005) sugerem utilizar cálculos de limitantes superiores para o valor da função objetivo dos problemas da mochila. Cálculo de limitantes superiores, também, foi utilizado em Hoto (2001) nas heurísticas propostas para o Problema da Mochila Compartimentada e para o Problema da Mochila Compartimentada Irrestrito, como citado no capítulo 2 desta dissertação.

5.3 Experimentos computacionais

Nesta seção são apresentados os resultados computacionais obtidos com o método Geração de Colunas para o Problema da Mochila Compartimentada. Para resolver o problema (5.13) - (5.15) foi utilizado um método de enumeração implícita apresentado no capítulo 6. Este método determina uma solução exata e é extremamente rápido para exemplares com dezenas de itens. Para a resolução do problema mestre foi utilizado o *software* ILOG CPLEX 11.0.

As Tabelas 5.1 e 5.2 mostram os resultados obtidos com o método Geração de Colunas e comparam com o limitante superior dado pelo Problema Inteiro (4.20) - (4.24). Na Tabela 5.1 são apresentados os valores médios da função objetivo para cada classe, a Tabela 5.2 mostra o tempo computacional médio em segundos.

Tabela 5.1: Valores médios da função objetivo

Classe	Geração de Colunas	Problema (4.20) - (4.24)	Aumento
1	2195,11	2146,45	2,27%
2	2250,67	2220,55	1,35%
3	2253,61	2221,20	1,46%
4	2297,25	2261,25	1,62%
5	2211,66	2183,40	1,29%
6	2272,33	2244,05	1,27%
7	2282,54	2255,15	1,22%
8	2356,72	2319,45	1,61%
9	2226,22	2191,60	1,57%
10	2270,44	2233,85	1,64%
11	2290,55	2264,40	1,16%
12	2324,75	2299,20	1,11%
13	2281,36	2260,30	0,93%
14	2313,21	2293,80	0,84%
15	2304,22	2282,25	0,96%
16	2340,70	2318,35	0,97%

Tabela 5.2: Tempo computacional médio

Classe	Geração de Colunas	Problema (4.20) - (4.24)
1	0,000	0,009
2	0,000	0,008
3	0,001	0,007
4	0,000	0,006
5	0,002	0,017
6	0,001	0,011
7	0,002	0,015
8	0,001	0,007
9	0,004	0,013
10	0,002	0,010
11	0,006	0,013
12	0,004	0,007
13	0,015	0,025
14	0,015	0,017
15	0,017	0,023
16	0,016	0,016

Os resultados que estão na Tabela 5.1 estão representados no gráfico contido na Figura 5.1.

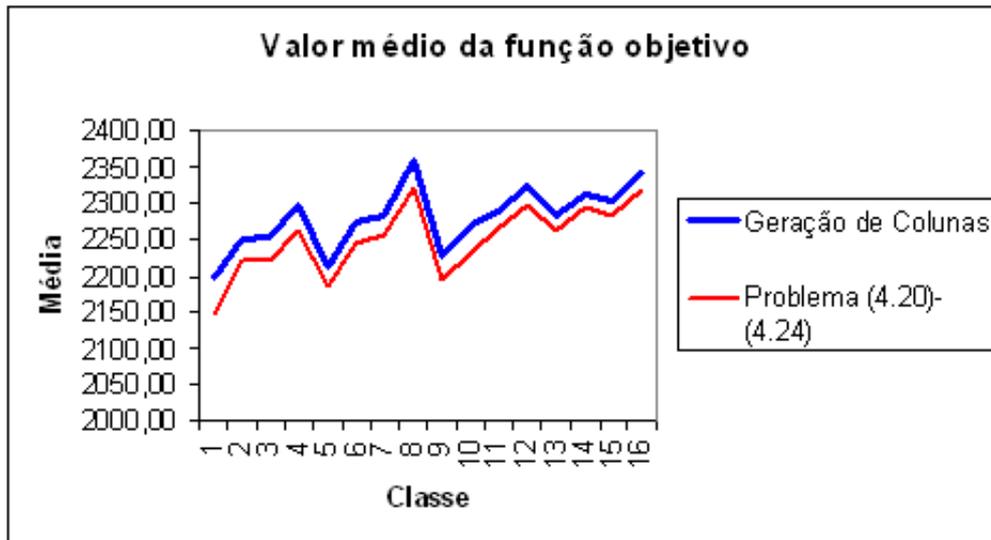


Figura 5.1: Gráfico da função objetivo.

A partir dos testes realizados podemos observar que o limitante superior obtido pela Geração de Colunas é um pouco pior que o limitante superior obtido pelo Problema Linear Inteiro (4.20) - (4.24). Para analisar esta diferença, a última coluna da tabela 5.1 mostra o aumento médio do valor da função objetivo do método Geração de Colunas, ocorrendo um aumento médio geral de 1,3%. Apenas para 4,1% dos problemas, a Geração de Colunas forneceu limitante superior melhor e conforme o número de classes e de itens aumentam ela se torna mais eficiente. Apesar do tempo computacional ter sido extremamente baixo para ambos os métodos, a Geração de Colunas foi um pouco mais rápida. Outra observação relevante é que o número de iterações do método simplex é relativamente baixo, o número médio de iterações foi 3,5.

Com os compartimentos obtidos, no final do método Geração de Colunas, o problema mestre (5.9) - (5.12) foi resolvido fixando as variáveis como inteiras, obtendo assim uma solução factível para o Problema da Mochila Compartimentada. A Tabela 5.3 apresenta os resultados.

Tabela 5.3: Geração de Colunas: solução inteira

Classe	Gap médio(%)
1	2,360
2	3,039
3	0,633
4	1,570
5	4,892
6	8,000
7	1,431
8	1,605
9	1,588
10	1,591
11	1,043
12	0,937
13	2,603
14	6,023
15	1,285
16	0,720
Média	2,458

Com este novo método as soluções obtidas foram piores que as obtidas pela heurística apresentada no capítulo 4, entretanto, ao considerarmos o número de compartimentos gerados, os resultados obtidos foram bastante satisfatórios. De 320 problemas foram obtidas 100 soluções ótimas, dadas pela igualdade da função objetivo deste método com a função objetivo do limitante superior (4.20) - (4.24), e o tempo computacional médio foi de 0,01 segundos.

O método Geração de Colunas, de modo geral, se comportou como o esperado, o principal objetivo deste método não foi obter um limitante superior eficiente, mas para que a partir dele possa ser desenvolvido um método exato para o Problema da Mochila Compartimentada, isto é, um método *branch-and-price*.

Métodos de enumeração implícita para Problemas da Mochila e extensões

Neste capítulo são apresentados os métodos utilizados para resolver os Problemas Lineares Inteiros que surgem na Heurística Híbrida, apresentada no capítulo 4, e no método Geração de Colunas, apresentado no capítulo 5. Foram utilizados três diferentes métodos de solução, todos baseados no algoritmo de enumeração implícita de Gilmore & Gomory (1963), os quais foram propostos em Marques (2004).

6.1 Método de enumeração implícita para o Problema Clássico da Mochila

Nesta seção é apresentado o método proposto por Gilmore & Gomory (1963). Os autores o utilizaram durante o processo de Geração de Colunas para resolver o subproblema, dado pelo Problema da Mochila (5.6) - (5.8). Para a descrição do algoritmo, daremos novamente a formulação matemática do Problema da Mochila:

$$\text{maximize:} \quad g(a) = \sum_{i=1}^m v_i a_i \quad (6.1)$$

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

$$a_i \geq 0, \text{ inteiro}, i = 1, \dots, m. \quad (6.3)$$

No modelo, m representa o número de tipos de itens que devem ser alocados na mochila, v_i e l_i representam o valor utilidade e a largura dos itens, respectivamente, para $i = 1, 2, \dots, m$. A variável a_i representa a quantidade que o item i é incluído na mochila, $i = 1, 2, \dots, m$. Para eliminar soluções triviais, supomos $v_i > 0$, $l_i > 0$ e $l_i \leq L$, $i = 1, \dots, m$.

O método pode ser dividido em 5 passos. O limitante inferior é dado pela melhor solução obtida até o momento (inicialmente nulo) e o limitante superior é definido pela relaxação linear. No Passo 1, os itens são ordenados em ordem não-crescente do valor por unidade de largura $\frac{v_i}{l_i}$, isto é: $\frac{v_1}{l_1} \geq \frac{v_2}{l_2} \geq \dots \geq \frac{v_m}{l_m}$.

Em seguida (Passo 2), determina-se uma solução inicial utilizando uma busca em profundidade, no qual uma solução é construída a partir da alocação do item mais valioso. Para representar a solução definiremos um vetor $x = (x_1, x_2, \dots, x_m)$, em que a posição i do vetor armazena a quantidade que o item i foi alocado a mochila. Inicialmente temos $x = (0, 0, \dots, 0)$, então uma unidade do item tipo 1 é alocado na mochila, obtendo a solução $x = (1, 0, \dots, 0)$, em seguida o item 1 é colocado novamente, até que não seja mais possível incluí-lo, o processo é repetido para o item 2 e assim sucessivamente. A cada solução x obtida associamos um nó da árvore de decisão (Figura 6.1) e, a cada nó $x = (x_1, \dots, x_s, 0, \dots, 0)$ temos um Problema da Mochila, em que a decisão de incluir itens do tipo $i = 1, \dots, s$ já foram examinadas:

$$\text{maximize:} \quad g(a) = \sum_{i=s+1}^m v_i a_i + \sum_{i=1}^s v_i x_i \quad (6.4)$$

$$\text{sujeito a:} \quad \sum_{i=s+1}^m l_i a_i \leq L - \sum_{i=1}^s l_i x_i \quad (6.5)$$

$$a_i \geq 0, \text{ inteiro}, i = s + 1, \dots, m. \quad (6.6)$$

Entretanto, no algoritmo os cálculos são realizados diretamente da seguinte forma:

$$\begin{aligned} x_1 &= \lfloor \frac{L}{l_1} \rfloor \\ x_2 &= \lfloor \frac{L-l_1 x_1}{l_2} \rfloor \\ &\vdots \end{aligned}$$

$$x_m = \lfloor \frac{L - (l_1x_1 + l_2x_2 + \dots + l_{m-1}x_{m-1})}{l_m} \rfloor$$

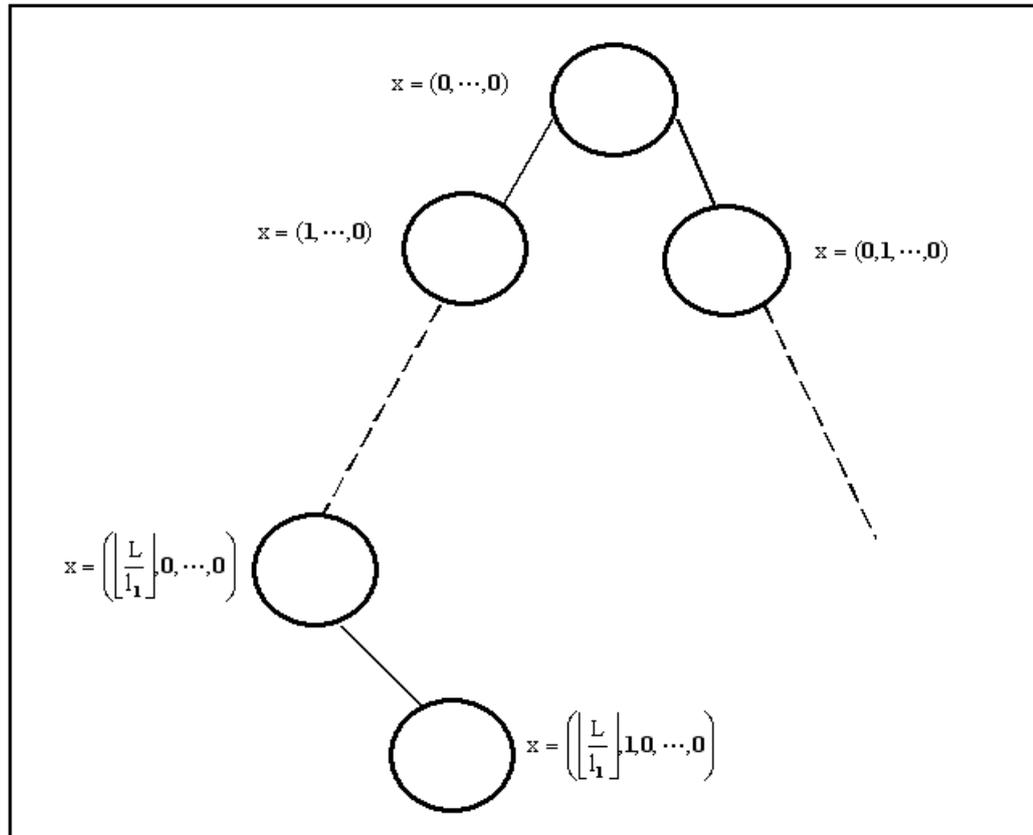


Figura 6.1: Representação da árvore de enumeração implícita

No Passo 3, a solução $x = (x_1, x_2, \dots, x_r, 0, \dots, 0)$ é avaliada, se essa solução for melhor que a melhor solução obtida até o momento, então ela é armazenada, correspondendo ao novo limitante inferior, isto é, $\underline{G} = g(x)$. No Passo 4 retornamos ao nó (solução) imediatamente anterior (*backtracking*) $x = (x_1, \dots, x_r - 1, 0, \dots, 0)$, para verificar a existência de soluções melhores. Uma condição necessária para que exista solução melhor a partir desta solução, pela inclusão de itens $r + 1, \dots, m$, é que $\bar{G} > \underline{G}$, em que \bar{G} é o limitante superior:

$$\bar{G} = v_1x_1 + v_2x_2 + \dots + v_r(x_r - 1) + \frac{v_{r+1}}{l_{r+1}}(L - l_1x_1 - l_2x_2 - \dots - l_{r-1}x_{r-1} - l_r(x_r - 1))$$

No Passo 5 essa condição é testada, se for verdadeira, significa que uma busca em profundidade a partir desta solução pode fornecer uma solução melhor que a melhor solução conhecida, após esta investigação, volta-se ao Passo 3. A busca em profundidade a partir de $x = (x_1, \dots, x_r, 0, \dots, 0)$ é:

$$x_r = x_r - 1 \text{ e}$$

$$L - \sum_{i=1}^{j-1} l_i x_i$$

$$x_j = \lfloor \frac{i=1}{l_j} \rfloor, j = r + 1, \dots, m$$

Caso $\bar{G} \leq \underline{G}$, é feito o retorno longo, isto é, $x_r = 0$, voltando alguns nós na árvore de enumeração, especificamente x_r nós, não sendo necessário investigá-los um a um. Esse retorno longo

não perde a otimalidade e a demonstração é feita na seção 6.5. Após o retorno longo, volta-se ao Passo 4. O algoritmo pára quando toda a árvore foi implicitamente investigada, isto é, quando volta ao nó inicial $x = (0, 0, \dots, 0)$. O algoritmo é descrito a seguir, conforme foi apresentado em Marques (2004):

Algoritmo de enumeração implícita para o Problema Clássico da Mochila

1. Início

1.1 Calcule as razões π_i dadas por: $\pi_i = \frac{v_i}{l_i}, i = 1, \dots, m$

1.2 Ordene os valores π_i : $\pi_1 \geq \pi_2 \geq \dots \geq \pi_m$

1.3 Faça $\underline{G} = 0$ (limitante inferior)

2. Determinação da solução inicial

2.1 Faça:

$$x_1 = \lfloor \frac{L}{l_1} \rfloor$$

$$x_2 = \lfloor \frac{L-l_1x_1}{l_2} \rfloor$$

⋮

$$x_m = \lfloor \frac{L-(l_1x_1+l_2x_2+\dots+l_{m-1}x_{m-1})}{l_m} \rfloor$$

3. Avaliação da solução corrente e armazenamento da mais valiosa

3.1 Faça $g = v_1x_1 + v_2x_2 + \dots + v_mx_m$

3.2 Se $\underline{G} < g$ faça:

3.2.1 $\underline{G} = g$ e $x^* = x$

4. Teste de parada e cálculo do limitante superior

4.1 Em x determine o maior índice r tal que $x_r \neq 0$

4.2 Se $x = (0, \dots, 0)$ então Pare, x^* é a solução ótima, senão:

4.3 Faça $\bar{G} = v_1x_1 + v_2x_2 + \dots + v_r(x_r - 1) + \frac{v_{r+1}}{l_{r+1}}(L - l_1x_1 - l_2x_2 - \dots - l_r(x_r - 1))$

5. Backtracking:

5.1 Se $\bar{G} \leq \underline{G}$ faça $x_r = 0$ e volte ao passo 4

5.2 Senão ($\bar{G} > \underline{G}$) então faça:

$$x_r = x_r - 1$$

$$L - \sum_{i=1}^{j-1} l_i x_i$$

$$x_j = \lfloor \frac{L - \sum_{i=1}^{j-1} l_i x_i}{l_j} \rfloor, j = r + 1, \dots, m \text{ e volte ao Passo 3.}$$

Apesar deste algoritmo não ter sido utilizado para resolver os Problemas da Mochila que surgem nos métodos de solução implementados neste trabalho, ele é a base para o desenvolvimento do método para determinar as melhores soluções para o Problema da Mochila Restrita.

6.2 Método de enumeração implícita para o Problema da Mochila Restrita

Para a descrição do método considere o seguinte Problema da Mochila Restrita:

$$\text{Maximize:} \quad g(a) = \sum_{i=1}^m v_i a_i \quad (6.7)$$

$$\text{Sujeito a:} \quad \sum_{i=1}^m l_i a_i \leq L \quad (6.8)$$

$$0 \leq a_i \leq \sigma_i, \text{ inteiro, } i = 1, \dots, m. \quad (6.9)$$

Este problema pode ser resolvido por meio de uma simples modificação no algoritmo anterior, que consiste em considerar o número máximo de vezes que cada item pode ser incluído na mochila. Para isso basta considerar:

$$x_j = \min\{\sigma_j, \lfloor \frac{L - l_1 x_1 - \dots - l_{j-1} x_{j-1}}{l_j} \rfloor\} \text{ para } j = 1, \dots, m,$$

modificando assim, os Passos 2, 4 e 5 do método anterior. O algoritmo foi dado em Marques (2004) e é apresentado a seguir.

Método de enumeração implícita para o Problema da Mochila Restrita

1. Início

1.1 Calcule as razões π_i dadas por: $\pi_i = \frac{v_i}{l_i}$ para $i = 1, \dots, m$

1.2 Ordene os valores π_i da seguinte forma: $\pi_1 \geq \pi_2 \geq \dots \geq \pi_m$

1.3 Faça $\underline{G} = 0$ (limitante inferior)

2. Determinação da solução inicial

2.1 Faça:

$$x_1 = \min\{\sigma_1, \lfloor \frac{L}{l_1} \rfloor\}$$

$$x_2 = \min\{\sigma_2, \lfloor \frac{L - l_1 x_1}{l_2} \rfloor\}$$

⋮

$$x_m = \min\{\sigma_m, \lfloor \frac{L - l_1 x_1 - \dots - l_{m-1} x_{m-1}}{l_m} \rfloor\}$$

3. Avaliação da solução corrente e armazenamento da mais valiosa

3.1 Faça $g = v_1 x_1 + v_2 x_2 + \dots + v_m x_m$

3.2 Se $\underline{G} < g$ faça:

3.2.1 $\underline{G} = g$ e $x^* = x$

4. Teste de parada e cálculo do limitante superior

4.1 Em x determine o maior índice r tal que $x_r \neq 0$

4.2 Se $x = (0, \dots, 0)$ então Pare, x^* é a solução ótima, senão:

4.3 (Cálculo do limitante superior)

Faça $\bar{L} = L - l_1x_1 - l_2x_2 - \dots - l_r(x_r - 1)$ e

$\bar{G} = v_1x_1 + v_2x_2 + \dots + v_r(x_r - 1)$

4.3.1 Seja $h = r + 1$

4.3.2 Faça $x_{m+1} = 0, l_{m+1} = 1$ e $\sigma_{m+1} = 0$ (variável de folga)

4.3.3 Enquanto ($l_h\sigma_h \leq \bar{L}$ e $h \leq m$) faça:

$\bar{L} = \bar{L} - l_h\sigma_h$ e $\bar{G} = \bar{G} + v_h\sigma_h$

$h = h + 1$

4.4.4 Faça: $\bar{G} = \bar{G} + \frac{\bar{L}}{l_h}v_h$

5. Backtracking

5.1 Se $\bar{G} \leq g$ faça $x_r = 0$ e volte ao passo 4

5.2 Senão ($\bar{G} > g$) faça

$x_r = x_r - 1$

$x_j = \min\{\sigma_j, \lfloor \frac{L - \sum_{i=1}^{j-1} l_i x_i}{l_j} \rfloor\}, j = r + 1, \dots, m$ e volte ao Passo 3.

Este algoritmo foi utilizado para resolver o problema (5.13) - (5.15) do método Geração de Colunas.

6.3 Algoritmo Z melhores soluções para o Problema da Mochila Restrita

Para descrição do algoritmo que determina as Z melhores soluções para o Problema da Mochila Restrita (6.7) - (6.9), considere que as Z melhores soluções sejam armazenadas na matriz X de dimensão $m \times Z$, em que cada coluna z da matriz representa a z -ésima melhor solução obtida. O valor da função objetivo, correspondente a solução armazenada na z -ésima coluna da matriz X , é armazenado na posição z do vetor \underline{G} dado por $\underline{G} = (\underline{G}_1, \underline{G}_2, \dots, \underline{G}_Z)$. O algoritmo é dado a seguir:

Algoritmo Z melhores soluções para o Problema da Mochila Restrita

1. Início

1.1 Calcule as razões π_i dadas por: $\pi_i = \frac{v_i}{l_i}$ para $i = 1, \dots, m$

1.2 Ordene os valores π_i da seguinte forma: $\pi_1 \geq \pi_2 \geq \dots \geq \pi_m$

1.3 Faça $X = 0$ e $\underline{G}_1 = \underline{G}_2 = \dots = \underline{G}_Z = 0$

2. Determinação das Z melhores soluções iniciais

2.1 Faça: $x = (x_1, x_2, \dots, x_m) = (0, 0, \dots, 0)$

$$g = 0$$

$$\bar{L} = L$$

2.2 Para $i = 1, \dots, m$ faça:

Enquanto $(\bar{L} - l_i) \geq 0$ faça:

$$x_i = x_i + 1$$

$$\bar{L} = \bar{L} - l_i$$

$$g = g + v_i$$

Se $g > \underline{G}_Z$, armazene g em \underline{G} de forma que o vetor \underline{G} fique ordenado, isto é, $\underline{G}_1 \geq \underline{G}_2 \geq \dots \geq \underline{G}_Z$ e x em X na coluna correspondente a que g for armazenada.

3. Teste de parada e cálculo do limitante superior

3.1 Em x determine o maior índice r tal que $x_r \neq 0$

3.2 Se $x = (0, \dots, 0)$ então Pare, não existe solução melhor que V_Z , senão:

3.3 Faça $\bar{L} = L - l_1x_1 - l_2x_2 - \dots - l_r(x_r - 1)$ e $\bar{G} = v_1x_1 + v_2x_2 + \dots + v_r(x_r - 1)$

3.4 Cálculo do limite superior:

3.4.1 Seja $h = r + 1$

3.4.2 Faça $x_{m+1} = 0$ e $l_{m+1} = 1$ e $\sigma_{m+1} = 0$

3.4.3 Enquanto $(l_h\sigma_h \leq \bar{L}$ e $h \leq m)$ faça:

$$\bar{L} = \bar{L} - l_h\sigma_h \text{ e } \bar{G} = \bar{G} + v_h\sigma_h$$

$$h = h + 1$$

3.4.4 Faça: $\bar{G} = \bar{G} + \frac{\bar{L}}{l_h}v_h$

4. Backtracking

4.1 Se $\bar{G} \leq \underline{G}_Z$ faça $g = g - v_r x_r$, $x_r = 0$ e volte ao passo 3

4.2 Senão ($\bar{G} > \underline{G}_Z$) então faça $x_r = x_r - 1$; $\bar{L} = \bar{L} - l_r$ e $g = g - v_r$:

4.2.1 Para $i = r + 1, \dots, m$ faça:

Enquanto $(\bar{L} - l_i) \geq 0$ faça:

$$x_i = x_i + 1$$

$$\bar{L} = \bar{L} - l_i$$

$$g = g + v_i$$

Se $g > \underline{G}_Z$, armazene g em V e \underline{G} em X de forma que o vetor \underline{G} fique ordenado e volte ao Passo 3.

6.4 Heurística para o problema (4.15) - (4.19)

Nesta seção é apresentado o algoritmo proposto por Marques (2004) para o problema (4.15) - (4.19), que seleciona os compartimentos e itens livres que comporão a mochila nas heurísticas aqui descritas. Este método é uma heurística baseada no método de enumeração implícita de Gilmore & Gomory (1963). Para facilitar a notação, o problema (4.15)- (4.19) será redefinido, para isso considere $Q = Z + W$, isto é os compartimentos obtidos foram apenas divididos por classe, sem distinguir se ele foi obtido pela z -ésima melhor solução ou w -ésima capacidade, assim a formulação matemática do problema fica da seguinte forma:

$$\text{maximize} \quad \sum_{k=1}^{K-1} \sum_{j=1}^Q (V_{jk} - c_k) \beta_{jk} + \sum_{k \in A_K} v_k \gamma_k \quad (6.10)$$

$$\text{sujeito a:} \quad \sum_{k=1}^{K-1} \sum_{j=1}^Q L_{jk} \beta_{jk} + \sum_{k \in A_K} l_k \gamma_k \leq L \quad (6.11)$$

$$\sum_{j=1}^Q a_{ijk} \beta_{jk} \leq \sigma_i, \forall i \in A_k, k = 1, \dots, K-1 \quad (6.12)$$

$$0 \leq \gamma_k \leq \sigma_k, \text{ inteiro}, k \in A_K \quad (6.13)$$

$$\beta_{jk} \geq 0, \text{ inteiro}, j = 1, \dots, Q, k = 1, \dots, K-1. \quad (6.14)$$

Para a descrição do algoritmo, os compartimentos são considerados como *super itens*. Os itens livres, também, são considerados como compartimentos, totalizando $T = (K-1) * Q + |A_K|$ compartimentos. Como a quantidade de cada item é limitada, então deve-se calcular o número de vezes que cada compartimento pode ser incluído na mochila, para que a demanda dos itens não seja ultrapassada, para isso serão utilizadas variáveis \hat{d}_t , $t = 1, \dots, T$, que são atualizadas à medida que um compartimento é incluído na mochila. Durante o algoritmo serão utilizados três procedimentos de atualização das variáveis \hat{d}_t (A, B e C), que estão descritos no final. O algoritmo para o problema (6.10) - (6.14) é dado a seguir e foi retirado de Marques (2004).

Algoritmo para o problema (6.10) - (6.14)

1. Definição do problema

1.1 Defina os valores $\hat{\pi}_t, \hat{l}_t, \hat{v}_t$ e \hat{d}_t :

$$t = 1$$

Para $k = 1, \dots, K - 1$ faça:

Para $j = 1, \dots, Q$ faça:

$$\hat{\pi}_t = \frac{V_{jk} - c_k}{L_{jk}}, \hat{l}_t = L_{jk}, \hat{v}_t = V_{jk} - c_k;$$

$$\hat{d}_t = \min\{\lfloor \frac{\sigma_i}{a_{ijk}} \rfloor\}, i \in A_k$$

$$t = t + 1;$$

Para todo $k \in A_K$ faça:

$$\hat{\pi}_t = \frac{v_k}{l_k}, \hat{l}_t = l_k, \hat{v}_t = v_k;$$

$$\hat{d}_t = \sigma_k;$$

$$t = t + 1.$$

1.2 Ordene os valores $\hat{\pi}_i$ da seguinte forma: $\hat{\pi}_1 \geq \hat{\pi}_2 \geq \dots \geq \hat{\pi}_T$

2. Determinação da solução inicial

2.1 Faça:

$$x_1 = \min\{\hat{d}_1, \lfloor \frac{L}{\hat{l}_1} \rfloor\}$$

Execute A(t=1)

$$x_2 = \min\{\hat{d}_2, \lfloor \frac{L - \hat{l}_1 x_1}{\hat{l}_2} \rfloor\}$$

Execute A(t=2)

⋮

$$x_T = \min\{\hat{d}_T, \lfloor \frac{L - \hat{l}_1 x_1 - \dots - \hat{l}_{T-1} x_{T-1}}{\hat{l}_T} \rfloor\}$$

Execute A(t=T)

3. Avaliação da solução corrente e armazenamento da mais valiosa

3.1 Faça $g = \hat{v}_1 x_1 + \hat{v}_2 x_2 + \dots + \hat{v}_T x_T$

3.2 Se $\underline{G} < g$ faça:

3.2.1 $\underline{G} = g$ e $x^* = x$

4. Teste de parada e cálculo do limitante superior

4.1 Em x determine o maior índice r tal que $x_r \neq 0$

Execute C

4.2 Se $x = (0, \dots, 0)$ então Pare, x^* é a melhor solução obtida, senão:

4.3 (Cálculo do limitante superior)

Faça $\bar{L} = L - l_1x_1 - l_2x_2 - \dots - l_r(x_r - 1)$ e

$$\bar{G} = \hat{v}_1x_1 + \hat{v}_2x_2 + \dots + \hat{v}_r(x_r - 1)$$

4.3.1 Seja $h = r + 1$

4.3.2 Faça $x_{T+1} = 0$, $l_{T+1} = \bar{L} + 1$ e $d_{T+1} = 0$

4.3.3 Enquanto ($\hat{l}_h\hat{d}_h \leq \bar{L}$ e $h \leq T$) faça:

$$\bar{L} = \bar{L} - \hat{l}_h\hat{d}_h \text{ e } \bar{G} = \bar{G} + \hat{v}_h\hat{d}_h$$

$$h = h + 1$$

4.4.4 Faça: $\bar{G} = \bar{G} + \frac{\bar{L}}{\hat{l}_h}\hat{v}_h$

5. Backtracking

5.1 Se $\bar{G} \leq g$

Execute B

faça $x_r = 0$ e volte ao passo 4

5.2 Senão ($\bar{G} > g$) faça

$$x_r = x_r - 1$$

$$x_{t+1} = \min\left\{\sigma_i, \left\lfloor \frac{L - \sum_{i=1}^j \hat{l}_i x_i}{\hat{l}_{j+1}} \right\rfloor\right\} \text{ e execute A para } t = r, \dots, T - 1.$$

Volte ao Passo 3.

A) Atualizar os limites máximos para os itens envolvidos no cálculo de x_t :

A1. Se t representa um item livre:

$$\text{Então faça: } \hat{d}_t = \hat{d}_t - x_t;$$

Senão: atualizar os limites máximos dos itens envolvidos

no compartimento t da seguinte forma:

Para todo $i \in A_k$, em que k é a classe associada ao

compartimento t , temos:

$$\sigma_i = \sigma_i - x_t * a_{ijk}, j = 1, \dots, Q.$$

A2. Recalcular os limites máximos permitidos para todos os $j = 1, \dots, Q$

compartimentos da classe k associada a t , fazendo:

$$\hat{d}_t = \min\left\{\left\lfloor \frac{\sigma_i}{a_{ijk}} \right\rfloor \text{ para todo } i \in A_k\right\}$$

B) Atualizar os limites máximos para os itens envolvidos na mudança do valor de x_r :

B1. Se r representa um item livre:

Então faça: $\hat{d}_r = \hat{d}_r + (x_r - 1)$ para o item associado a r ;

Senão: atualizar os limites máximos dos itens envolvidos no compartimento r da seguinte forma:

Para todo $i \in A_k$, em que k é a classe associada ao compartimento r , temos:

$$\sigma_i = \sigma_i + (x_r - 1) * a_{ijk}, j = 1, \dots, Q.$$

B2. Recalcular os limites máximos permitidos para todos os $j = 1, \dots, Q$ compartimentos da classe k associada a r , fazendo:

$$\hat{d}_r = \min\{\lfloor \frac{\sigma_i}{a_{ijk}} \rfloor \text{ para todo } i \in A_k\}$$

C) Atualizar os limites máximos para os itens envolvidos na mudança do valor de x_r :

C1. Se r representa um item livre:

Então faça: $\hat{d}_r = \hat{d}_r + 1$ para o item associado a r ;

Senão: atualizar os limites máximos dos itens envolvidos no compartimento r da seguinte forma:

Para todo $i \in A_k$, em que k é a classe associada ao compartimento r , temos:

$$\sigma_i = \sigma_i + a_{ijk}, j = 1, \dots, Q.$$

C2. Recalcular os limites máximos permitidos para todos os $j = 1, \dots, Q$ compartimentos da classe k associada a r , fazendo:

$$\hat{d}_r = \min\{\lfloor \frac{\sigma_i}{a_{ijk}} \rfloor \text{ para todo } i \in A_k\}$$

6.5 Verificação do *backtracking* longo

Nesta seção demonstramos que o *backtracking* longo proposto por Gilmore e Gomory (1963) não perde a solução ótima. Para isso considere o vetor solução $x = (x_1, x_2, \dots, x_r, 0, \dots, 0)$, denotaremos $x^{-s} = (x_1, x_2, \dots, x_r - s, 0, \dots, 0)$ com $x_r \geq s$ e $s \geq 1$. O limitante superior de x^{-s} é dado por:

$$V(x^{-s}) = v_1x_1 + v_2x_2 + \dots + v_r(x_r - s) + \frac{v_{r+1}}{l_{r+1}}(L - l_1x_1 - l_2x_2 - \dots - l_r(x_r - s)).$$

Lema: $V(x^{-s}) \leq V(x^{-1})$.

Prova:

$$V(x^{-s}) = v_1x_1 + v_2x_2 + \dots + v_r(x_r - s) + \frac{v_{r+1}}{l_{r+1}}(L - l_1x_1 - l_2x_2 - \dots - l_r(x_r - s))$$

$$= v_1x_1 + v_2x_2 + \dots + v_r(x_r - s - 1 + 1) + \frac{v_{r+1}}{l_{r+1}}(L - l_1x_1 - l_2x_2 - \dots - l_r(x_r - s - 1 + 1))$$

$$\begin{aligned}
&= v_1x_1 + v_2x_2 + \dots + v_r(x_r - 1) + \frac{v_{r+1}}{l_{r+1}}(L - l_1x_1 - l_2x_2 - \dots - l_r(x_r - 1)) + v_r(1 - s) + \frac{v_{r+1}}{l_{r+1}}l_r(s - 1) \\
&= V(x^{-1}) + v_r(1 - s) + \frac{v_{r+1}}{l_{r+1}}l_r(s - 1)
\end{aligned}$$

Assim basta mostrar que: $v_r(1 - s) + \frac{v_{r+1}}{l_{r+1}}l_r(s - 1) \leq 0$

De fato, dividindo $v_r(1 - s) + \frac{v_{r+1}}{l_{r+1}}l_r(s - 1)$ por l_r tem-se: $\frac{v_r}{l_r}(1 - s) + \frac{v_{r+1}}{l_{r+1}}(s - 1)$

$$= \left(\frac{v_{r+1}}{l_{r+1}} - \frac{v_r}{l_r}\right)(s - 1)$$

Como os itens estão ordenados de acordo com a razão entre o valor utilidade e a largura então: $\frac{v_{r+1}}{l_{r+1}} - \frac{v_r}{l_r} \leq 0$.

Logo, $\left(\frac{v_{r+1}}{l_{r+1}} - \frac{v_r}{l_r}\right)(s - 1) \leq 0$

Portanto: $V(x^{-s}) \leq V(x^{-1})$.

Considerando uma solução x com valor \underline{G} do algoritmo proposto por Gilmore e Gomory (1963) e G_Z o valor da Z -ésima melhor solução armazenada no algoritmo das Z -melhores soluções para o problema da mochila restrita (esses valores são atualizados durante a construção da árvore *branch-and-bound*), temos os seguintes teoremas.

Teorema 1: Se $V(x^{-1}) \leq \underline{G}$ então $V(x^{-s}) \leq \underline{G}$.

Prova: Pelo Lema temos:

$V(x^{-s}) \leq V(x^{-1})$, como por hipótese $V(x^{-1}) \leq \underline{G}$, logo:

$$V(x^{-s}) \leq \underline{G}.$$

Teorema 2: Se $V(x^{-1}) \leq G_Z$ então $V(x^{-s}) \leq G_Z$.

Prova: Análogo ao Teorema 1.

O Teorema 1 mostra que o *backtracking* longo proposto por Gilmore e Gomory (1963) não perde a solução ótima. O Teorema 2 mostra que o algoritmo proposto para as Z melhores soluções não perde as Z melhores soluções, pois o limitante inferior é a Z -ésima melhor solução obtida até o momento.

Conclusões e trabalhos futuros

O objetivo deste trabalho foi desenvolver métodos de solução para o Problema da Mochila Compartimentada. Aqui foram apresentadas duas abordagens de solução, cuja característica em comum é a maneira como o problema foi decomposto.

A primeira abordagem, apresentada no Capítulo 4, é uma heurística baseada em duas heurísticas da literatura, propostas em Marques & Arenales (2007). Nesta heurística, o problema original é decomposto em dois problemas, um que gera os compartimentos e o outro que os seleciona. Diversos testes foram realizados com esta heurística, considerando diferentes parâmetros, os quais mostraram a sua eficiência.

A segunda abordagem, apresentada no Capítulo 5, é a técnica Geração de Colunas, no qual, o problema original foi decomposto em problema mestre e subproblema, sendo que o subproblema gera compartimentos, que são as colunas para o problema mestre. A vantagem deste método é que ele nos fornece um limitante superior e chega a uma solução ótima, para o problema relaxado, com um número de iterações bastante pequeno. Entretanto, o principal objetivo da Geração de Colunas é permitir que futuramente possa ser desenvolvido um método de solução exata para o Problema da Mochila Compartimentada.

Para a execução da heurística e da Geração de Colunas é necessário resolver Problemas da Mochila Restrita e um Problema Linear Inteiro, para isso foram utilizados algoritmos baseados no método *branch-and-bound* de Gilmore & Gomory (1963). No total foram utilizados três procedimentos, dos quais, dois foram propostos em Marques (2004) que são: o método que determina a solução para o Problema da Mochila Restrita e outro que determina a solução para o problema (4.15) - (4.19). O terceiro método é um procedimento que determina as Z melhores soluções para o Problema da Mochila Restrita.

Diversas sugestões para o desenvolvimento de trabalhos futuros podem ser apresentadas. Entre estas se encontram o desenvolvimento de um método exato tipo *branch-and-price* para o Problema da Mochila Compartimentada Unidimensional e aplicar métodos de arredondamento para obter uma solução inteira, a partir da solução obtida pela Geração de Colunas. Esta abordagem pode ser estendida na geração de padrões de corte para o caso bidimensional, visto que surgem naturalmente na prática, pelas mesmas razões que o caso unidimensional, ou seja, corte em vários estágios.

Referências Bibliográficas

- CARVALHO, J. M. V.; RODRIGUES, A. J. G. A computer based interactive approach to a two-stage cutting stock problem. *INFOR: Information Systems and Operation Research*, v. 32, n. 4, p. 243–252, 1994.
- CARVALHO, J. M. V.; RODRIGUES, A. J. G. An lp-basead approach to a two-stage cutting stock problem. *European Journal of Operational Research*, v. 84, n. 3, p. 580–589, 1995.
- CORREIA, M. H.; OLIVEIRA, J. F.; FERREIRA, J. S. Reel and sheet cutting at a paper mill. *Computers and Operations Research*, v. 31, n. 8, p. 1223–1243, 2004.
- DANTZIG, G. B.; WOLFE, P. Decomposition principle for linear programs. *Operations Research*, v. 8, p. 101–111, 1960.
- DYCKHOFF, H.; FINKE, U. *Cutting and packing in production and distribution: Typology and bibliography*. Heidelberg: Springer-Verlag., 1992.
- FERREIRA, J. S.; NEVES, M. A.; E CASTRO, P. F. A two-phase roll cutting problem. *European Journal of Operational Research*, v. 44, n. 2, p. 185–196, 1990.
- GILMORE, P. C.; GOMORY, R. E. A linear programming approach to the cutting-stock problem. *Operations Research*, v. 9, n. 6, p. 849–859, 1961.
- GILMORE, P. C.; GOMORY, R. E. A linear programming approach to the cutting stock problem- part ii. *Operations Research*, v. 11, n. 6, p. 863–888, 1963.
- GILMORE, P. C.; GOMORY, R. E. Multi-stage cutting stock problems of two and more dimensions. *Operations Research*, v. 13, n. 1, p. 94–120, 1965.
- HAESSLER, R. W. A heuristic programming solution to a nonlinear cutting stock-problem. *Management Science*, v. 17, n. 12, p. 793–802, 1971.

- HAESSLER, R. W. Solving the two-stage cutting stock problem. *Omega: The International Journal of Management Science*, v. 7, n. 2, p. 145–151, 1979.
- HINXMAN, A. I. The trim-loss and a survey assortment problems. *European Journal of Operational Research*, v. 5, n. 1, p. 8–18, 1980.
- HOTO, R. *Otimização no corte de peças unidimensionais com restrições de agrupamento*. Dissertação de Mestrado, ICMC/USP, São Carlos - SP, 1996.
- HOTO, R. *O problema da mochila compartimentada aplicado no corte de bobinas de aço*. Tese de Doutorado, COPPE/UFRJ, Rio de Janeiro - RJ, 2001.
- HOTO, R.; ARENALES, M. N.; MACULAN, N. *O problema da mochila compartimentada*. Relatório Técnico, Universidade Estadual de Londrina, 1999.
- HOTO, R.; ARENALES, M. N.; MACULAN, N. The one dimensional compartmentalised knapsack problem: A case study. *European Journal of Operational Research*, v. 183, n. 3, p. 1183–1195, 2007.
- HOTO, R.; SPOLADOR, F.; MARQUES, F. Resolvendo mochilas compartimentadas restritas. In: *Anais do XXXVII Simpósio Brasileiro de Pesquisa Operacional*, 2005.
- JOHNSON, M. P.; RENNICK, C.; ZAK, E. Skiving addition to the cutting stock problem in the paper industry. *Society for Industrial and Applied Mathematics*, v. 39, n. 3, p. 472–483, 1997.
- MANNE, A. S. Programming of economic lot sizes. *Management Science*, v. 4, p. 115–135, 1958.
- MARQUES, F. P. *O problema da mochila compartimentada*. Dissertação de Mestrado, ICMC/USP, São Carlos - SP, 2000.
- MARQUES, F. P. *O problema da mochila compartimentada e aplicações*. Tese de Doutorado, ICMC/USP, São Carlos - SP, 2004.
- MARQUES, F. P.; ARENALES, M. N. The constrained compartmentalised knapsack problem. *Computers and Operations Research*, v. 34, n. 7, p. 2109–2129, 2007.
- MARTELLO, S.; TOTH, P. *Knapsack problems: algorithms and computer implementations*. Chichester: John Wiley & Sons., 1990.
- PEREIRA, M. A. *Uma abordagem matemática para o problema do corte e laminação de fitas de aço*. Dissertação de Mestrado, UNICAMP, Campinas - SP, 1993.
- POLDI, K. C.; ARENALES, M. N. Heuristics for the one-dimensional cutting stock problem with limited multiple stock lengths. *Computers and Operations Research*, (no prelo), 2009.

-
- WÄSCHER, G.; GAU, T. Heuristics for the integer one-dimensional cutting stock problem: a computational study. *OR Spektrum*, v. 18, p. 131–144, 1996.
- ZAK, E. J. Modeling a multistage cutting stock problems. *European Journal of Operational Research*, v. 141, n. 2, p. 313–327, 2002a.
- ZAK, E. J. Row and column generation technique for a multistage cutting stock problem. *Computer and Operations Research*, v. 29, n. 9, p. 1143–1156, 2002b.

Livros Grátis

(<http://www.livrosgratis.com.br>)

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)
[Baixar livros de Literatura de Cordel](#)
[Baixar livros de Literatura Infantil](#)
[Baixar livros de Matemática](#)
[Baixar livros de Medicina](#)
[Baixar livros de Medicina Veterinária](#)
[Baixar livros de Meio Ambiente](#)
[Baixar livros de Meteorologia](#)
[Baixar Monografias e TCC](#)
[Baixar livros Multidisciplinar](#)
[Baixar livros de Música](#)
[Baixar livros de Psicologia](#)
[Baixar livros de Química](#)
[Baixar livros de Saúde Coletiva](#)
[Baixar livros de Serviço Social](#)
[Baixar livros de Sociologia](#)
[Baixar livros de Teologia](#)
[Baixar livros de Trabalho](#)
[Baixar livros de Turismo](#)