

Dissertação

Raimundo Ricardo Matos da Cunha

1998

Universidade Federal de Santa Catarina
Curso de Pós-graduação em Engenharia Mecânica

UM ALGORITMO DE MINIMIZAÇÃO DE SOBRAS
EM CORTE UNIDIMENSIONAL

Dissertação Submetida à Universidade Federal de Santa Catarina
para Obtenção do Grau de Mestre em Engenharia Mecânica

Raimundo Ricardo Matos da Cunha

Florianópolis, Novembro de 1998

Um Algoritmo de Minimização de Sobras
em Corte Unidimensional

Raimundo Ricardo Matos da Cunha

Esta Dissertação foi Julgada Adequada para a Obtenção do

Título de

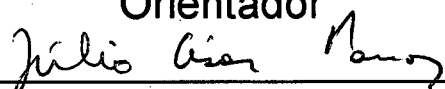
Mestre em Engenharia

Especialidade Engenharia Mecânica, Área de Concentração
Projeto de Sistemas Mecânicos com Ênfase em Sistemas de
CAE/CAD/CAM, e Aprovada em sua Forma Final pelo Curso de
Pós-graduação em Engenharia Mecânica



Altamir Dias, D. Sc.


Orientador



Júlio César Passos, Dr.

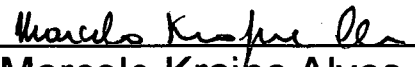
Coordenador da Pós-graduação

Banca Examinadora



João Carlos Espíndola Ferreira, Ph.D.

Presidente



Marcelo Krajinovic Alves, Ph.D.

Membro



Sérgio Fernando Mayerle, Dr.

Membro

A verdade dos fatos nunca está aos olhos de tudo e de todos, pois ela habita o espírito de cada um de nós. E só a sentiremos se formos capazes, e estivermos interessados em buscá-la.

Tilapius, o Sinistra

Que o homem seja capaz de considerar-se a si mesmo é o grande prodígio da natureza, pois ele não consegue conceber o que é o seu corpo e muito menos o que é o seu espírito, e finalmente, como que o corpo pode se unir ao espírito. Este é o cume de sua dificuldade e no entanto é a essência do seu ser.

Blaise Pascal

Buscai em primeiro lugar o Reino de Deus e a sua justiça e todas estas coisas vos serão dadas por acréscimo. Não vos preocupeis, pois, com o dia de amanhã: o dia de amanhã terá as suas preocupações próprias. A cada dia basta o seu cuidado.

Evangelho segundo Mateus, cap. 6, vv. 33 e 34

É costume de um tolo, quando erra, queixar-se dos outros. É costume de um sábio queixar-se de si mesmo.

Sócrates, 470 - 399 a.C.

DEDICATÓRIA

Aos meus pais, Luiz Ricardo e Maria de Lourdes,
pelos ensinamentos de vida a mim repassados.

À minha avó, Maria de Lourdes, pelo carinho.

Aos meus irmãos: Germano, Giovani, Luiz e Clodoaldo.

AGRADECIMENTOS

A Deus, pela minha saúde e pela luz que me guiou na execução do trabalho.

Ao CNPq, pela oportunidade e pelo apoio financeiro para realização do trabalho.

À Universidade Federal de Santa Catarina e ao Curso de Pós-graduação em Engenharia Mecânica, pela oportunidade e pelas dependências e instalações físicas para realização do trabalho.

Ao Prof. Altamir Dias, pela confiança depositada e pela orientação que se fez presente nos momentos decisivos do trabalho.

Aos meus amigos Clodoaldo de Oliveira Carvalho Filho (*Lentoaldo*), Francisco Frederico dos Santos Matos (*Shark Lenda*) e João Carlos Arantes Costa Júnior (*Irmão Sura*), que estiveram juntos comigo, do começo ao fim desta empreitada, e são testemunhas e personagens dos momentos difíceis e felizes aqui em Florianópolis. Ao amigo Cesar Augusto do Canto Vinadé (*Tigrão*), que depois se juntou à República do Ceará, tornando-se um companheiro fiel e sempre prestativo.

Ao Rodolfo Moriconi, mais conhecido como *Gão*, e à sua família, pela hospitalidade e pela ajuda durante os primeiros dias aqui em Florianópolis.

Ao amigo Leonardo Bastos de Toledo, o *Carioca*, que foi companheiro de trabalho e com quem pude compartilhar a experiência vivida durante todo esse tempo.

À minha querida Daniela Regina Royer, pelo conforto e o carinho sempre acolhedor.

À indústria Douat Metalmecânica, quero agradecer a visita técnica que fiz à empresa e as informações que pude colher sobre o tema do trabalho. Em especial, agradeço aos funcionários Pedro Pedrosa e Adolar pela atenção e paciência.

A todos os colegas do Mestrado pelos momentos de descontração, e em especial: à turma do Pará, João Neto, Paulo Roberto e Cláudio Blanco; ao Michael de Jesus Fernandes e família, pelas festas; ao Marcos Luciano, o *Gauchita*, pelas horas de estudo e pelas cuias de chimarrão nas baias; e ao Valdir Mendes Cardoso, pelo companheirismo na reta final do trabalho.

Às amigas conterrâneas Rafaela Frota Reinaldo e Ana Vlândia Cabral Sobral.

Ao Prof. Edison da Rosa, agradeço por me ceder um computador quando o que eu usava quebrou e precisei dar continuidade na escrita da minha dissertação.

Estendo esses agradecimentos também a todos os colegas do GRANTE.

SUMÁRIO

LISTA DE FIGURAS.....	X
LISTA DE TABELAS.....	XI
RESUMO.....	XII
ABSTRACT.....	XIII
CAPÍTULO 1 INTRODUÇÃO.....	1
1.1 Objetivos do Trabalho.....	2
1.2 Importância do Estudo.....	3
1.3 Um Estudo de Caso.....	4
CAPÍTULO 2 PROBLEMA DO CORTE INDUSTRIAL.....	6
2.1 Descrição.....	7
2.2 Classificação e Tipologia.....	8
CAPÍTULO 3 REVISÃO BIBLIOGRÁFICA.....	25
3.1 História dos Processos de Corte e Empacotamento.....	25
CAPÍTULO 4 FUNDAMENTOS TEÓRICOS.....	31
4.1 Formulação Geral.....	33
4.2 Problema de Knapsack.....	36
4.3 Algoritmo de Enumeração.....	39
4.4 Técnica da Geração de Colunas.....	43
CAPÍTULO 5 MÉTODOS E TÉCNICAS.....	47
5.1 Formulação Unidimensional.....	47
5.2 Algoritmo de Otimização Unidimensional Proposto.....	50
CAPÍTULO 6 RESULTADOS.....	54
6.1 Caso Prático.....	54
6.1.1 Descrição do Problema.....	54
6.1.2 Definições Gerais.....	55
6.1.3 Dados do Problema.....	56
6.1.4 Solução e Listagem de um Pedido de Corte.....	58

6.2 Alteração nos Valores das Variáveis do Problema.....	62
6.2.1 Variação na Quantidade de Peças	62
6.2.2 Variação no Comprimento da Matéria-prima	64
6.3 Caso Teórico	65
CAPÍTULO 7 CONSIDERAÇÕES FINAIS.....	71
7.1 Conclusões	71
7.2 Sugestões para Trabalhos Futuros.....	74
7.2.1 Problema com Diferentes Comprimentos de Matéria-prima	74
7.2.2 Encaixe de Retângulos em Corte Tipo Guilhotina	75
REFERÊNCIAS BIBLIOGRÁFICAS.....	77
APÊNDICES	82
Apêndice A. Tópicos de Programação e Implementação Computacional... 82	
Apêndice B. Fluxograma da Entrada de Dados	85
Apêndice C. Fluxogramas dos Algoritmos de Solução Inicial	86
C.1 Algoritmo Initial-Solution.....	86
C.2 Algoritmo FFD-Solution.....	87
Apêndice D. Fluxograma do Algoritmo de Branch-and-Bound.....	89
Apêndice E. Fluxograma do Algoritmo Principal – XCut1D.....	90

Lista de Figuras

Figura 1-1: Exemplo do percentual de desperdício de matéria-prima	4
Figura 2-1: Exemplo do corte de vergalhões.....	9
Figura 2-2: Exemplo de padrões de corte entre os itens da lista de pedidos.....	10
Figura 2-3: Exemplo de empacotamento em contêineres.....	11
Figura 2-4: Exemplo de um padrão de empacotamento	11
Figura 2-5: Exemplo de um problema C&P abstrato.....	13
Figura 2-6: Exemplo de um padrão de ordenamento de tarefas.....	13
Figura 2-7: Exemplo de bidimensionalidade	15
Figura 2-8: Diferenciação entre figuras considerando a orientação.....	16
Figura 2-9: Exemplo de cortes	19
Figura 4-1: Métodos de programação inteira	32
Figura 4-2: Representação em árvore	40
Figura 5-1: Seqüência dos passos 2 a 7 do algoritmo <i>XCut1D</i>	52
Figura 5-2: Seqüência dos passos 8 a 12 do algoritmo <i>XCut1D</i>	53
Figura 6-1: Estrutura proposta para a máquina-embaladeira.....	55
Figura 6-2: Variação na quantidade cortada de peças de aço Tipo 2	63
Figura 6-3: Variação no comprimento da matéria-prima de aço Tipo 1	64
Figura 6-4: Variação no comprimento da matéria-prima de aço Tipo 2	64
Figura 6-5: Variação no comprimento da matéria-prima — 168,00 a 260,00 pol.....	67
Figura 6-6: Variação no comprimento da matéria-prima — 130,00 a 168,00 pol.....	68
Figura 6-7: Variação no comprimento da matéria-prima — 260,00 a 330,00 pol.....	68
Figura 7-1: Desempenho computacional do algoritmo <i>XCut1D</i>	73
Figura 7-2: Corte tipo guilhotina.....	75

Lista de Tabelas

Tabela 1-1: Cotação do preço de bobinas de aço inoxidável (SET/98)	5
Tabela 2-1: Simbologia dos principais tipos de problemas de C&P	24
Tabela 3-1: Pesquisas sobre problemas de C&P.....	26
Tabela 4-1: Crescimento exponencial do tempo de processamento.....	39
Tabela 6-1: Lista de peças para construção de 1 estrutura completa da máquina	57
Tabela 6-2: Lista de peças de aço Tipo 1 para o pedido das 15 estruturas.....	58
Tabela 6-3: Lista de peças de aço Tipo 2 para o pedido das 15 estruturas.....	59
Tabela 6-4: Padrões de corte para as peças de aço Tipo 1 – Algoritmo <i>XCut1D</i>	59
Tabela 6-5: Padrões de corte para as peças de aço Tipo 2 – Algoritmo <i>XCut1D</i>	59
Tabela 6-6: Solução de Degreave e Bemelmans para peças de aço Tipo 1	61
Tabela 6-7: Solução de Degreave e Bemelmans para peças de aço Tipo 2	61
Tabela 6-8: Parcelamento do pedido em lotes.....	63
Tabela 6-9: Lista das peças do pedido	66
Tabela 6-10: Exemplo de economia de matéria-prima.....	67
Tabela 6-11: Padrões de corte — Parte inteira da solução do <i>XCut1D</i>	69
Tabela 6-12: Padrões de corte — Parte real da solução do <i>XCut1D</i>	70

RESUMO

Este trabalho visa implementar um algoritmo computacional para resolver o problema de minimização de rejeitos em cortes de peças unidimensionais. Este problema é freqüentemente encontrado em processos industriais, onde o início da transformação e manufatura da matéria-prima começa com o processo de corte.

Este problema pode ser classificado como uma aplicação de programação linear inteira. Os padrões de corte otimizados são obtidos pela técnica utilizada por Gilmore e Gomory, em 1961, e que recebeu o nome de Técnica da Geração de Colunas. Ela combina todas as peças do pedido e enumera todos os padrões de corte viáveis para a solução do Problema do Corte Industrial. A obtenção dos padrões de corte exige a solução de um conjunto de problemas de Knapsack. O resultado é a seleção de padrões que melhor aproveitam o comprimento nominal útil da matéria-prima dentre aqueles que foram gerados. Estes padrões de corte são conhecidos como padrões maximais. O algoritmo atende a um pedido de peças de comprimentos e quantidades diversas e estipuladas. Além disso, ele informa a disposição espacial e o ordenamento das peças encaixadas na matéria-prima.

Neste trabalho, são apresentadas as formulações matemáticas inerentes ao problema. Adicionalmente, foram introduzidas heurísticas próprias para acelerar a convergência da solução. Os algoritmos resultantes da formulação estudada foram implementados num compilador C++ para sistema operacional Windows 95/98.

ABSTRACT

This research presents a set of techniques to solve the one-dimensional cutting stock problem. Such kind of problem is often found in various industrial plants, where the cutting of a large stock in small parts is a part production of the system.

It is classified as an application of integer linear programming. The optimized cutting patterns are obtained using a technique proposed by Gilmore and Gomory, in 1961, named the Delayed Column-Generation Technique. The parts are sorted and the enumeration of all possible cutting patterns are assigned. An auxiliary set of Knapsack Problem have to be solved, associated with each cutting pattern. This cutting pattern is known as maximal pattern cutting. An order of pieces with different lengths and in specified quantities is solved. Besides, the result is to get the space layout of cutting and the sequencing of the pieces fit in the raw material.

The inherent mathematical formulations for this type of problem were reviewed. Moreover, some heuristics procedures were introduced to accelerate the solution. The routines were implemented in C++ compiler and Windows 95/98 operational system.

Capítulo 1

INTRODUÇÃO

O corte de matérias-primas é um processo normalmente empregado na fabricação dos mais variados produtos encontrados à venda em prateleiras de supermercados, armazéns e lojas de departamentos. Este processo de corte não se limita apenas a uma operação de natureza física e mecânica, tendo também que ser analisado sob o ponto de vista econômico de mercado. E isso fica mais evidente nos casos em que os produtos estão inseridos numa linha de produção seriada de média e/ou grande escala, haja vista o volume de material que é processado. A constatação destes fatos tem mostrado que o assunto assume enorme importância econômica dentro do processo produtivo e por isso ele tem sido levado em conta em muitas pesquisas já publicadas.

Concentrando-se principalmente sob o ponto de vista econômico, este trabalho aborda a questão do desperdício de matéria-prima oriunda de processos de corte realizados durante a fabricação de produtos. Uma parcela importante dos custos advindos da matéria-prima gasta são incorporados diretamente ao preço final de produtos industrializados.

O estudo realizado neste trabalho sobre o Problema do Corte Industrial comenta sobre os tópicos descritos brevemente nas linhas abaixo.

O texto a seguir discorre sobre os objetivos deste trabalho, sobre a importância e aplicação dos problemas de encaixe de peças e fornece alguns dados sobre um estudo de caso realizado numa indústria do setor metalmeccânico. Já o capítulo 2, descreve e classifica as diferentes facetas deste tipo de problema.

No capítulo 3 são feitas as citações dos principais trabalhos que contribuíram para o desenvolvimento das pesquisas realizadas neste tipo de problema.

O capítulo 4 apresenta uma formulação geral do Problema do Corte Industrial e a teoria matemática de programação linear inteira relacionada, quais sejam: Problemas de Knapsack, Algoritmo de Branch-and-Bound, e a Técnica da Geração de Colunas.

Os métodos e técnicas adotados neste trabalho são descritos no capítulo 5. Lá, também são descritas as heurísticas empregadas para obtenção de uma solução aproximada do problema unidimensional. No capítulo 6, apresentam-se os testes usados para validar a presente implementação e analisam-se os resultados obtidos com a aplicação do algoritmo de otimização proposto.

No capítulo 7, fazem-se as considerações finais a respeito do trabalho, concluindo sobre os casos comentados no capítulo 6 de resultados. E também são sugeridos alguns trabalhos relacionados com o que foi resolvido aqui.

E finalmente, os apêndices trazem informação sobre questões levantadas durante a implementação computacional do código do algoritmo e também ilustram os fluxogramas das rotinas implementadas.

1.1 OBJETIVOS DO TRABALHO

Este trabalho procura dar uma contribuição ao estudo da minimização das sobras em corte unidimensional, no qual o resultado deve estar vinculado a propor soluções para a melhora do processo produtivo em termos de redução dos custos de produção. É intrínseco a esse objetivo, rever as formulações matemáticas deste tipo de problema, fazer inferências quanto à eficiência computacional e programá-las de acordo com algoritmos já conhecidos na solução de problemas de programação matemática. Além disso, estes algoritmos resultaram num conjunto de rotinas computacionais programadas para um compilador C++. Em resumo ao que até aqui foi exposto, a implementação deste trabalho basear-se-á nas seguintes proposições:

- (1) Utilizar um conjunto de estratégias para obtenção de soluções para problema de corte unidimensional;
- (2) Propor uma estrutura de interface computacional amigável que facilite a entrada de dados, obtenção da solução e dos resultados para este tipo de problema;
- (3) Propor um algoritmo de otimização das sobras de matérias-primas.

O resultado alcançado deve ser uma ferramenta computacional que auxilie na solução desse tipo de problema, pois no dia-a-dia de muitas empresas, ainda se recorre às soluções obtidas sem o auxílio de um método de otimização.

1.2 IMPORTÂNCIA DO ESTUDO

O estudo sobre o Problema do Corte Industrial está amparado por argumentos consensuais dado a sua importância dentro do processo produtivo, conforme já foi citado anteriormente. Muitos trabalhos têm sido publicados e alguns destes, julgados mais importantes para a realização desta pesquisa, serão listados no capítulo 3.

Atualmente, a derrubada das fronteiras alfandegárias e a conseqüente abertura de novos mercados comerciais, tem obrigado às indústrias a se modernizarem com o propósito de otimizarem os seus custos de produção, e dessa forma, oferecerem produtos com preços compatíveis com os exigidos pelo mercado consumidor internacional. Diante desse quadro, estudos sobre a otimização do corte com o objetivo de melhorar o aproveitamento da área útil ou do comprimento útil da matéria-prima têm aplicação e importância incontestáveis dentro de muitos setores industriais.

Sweeney e Paternoster identificaram mais de 500 artigos que tratam com problemas e aplicações relacionados a *C&P* [22]. As principais razões mencionadas, na maior parte desses artigos, eram que:

- Os problemas de *C&P* ocorrem num grande campo diversificado de indústrias;
- Existe um grande incentivo econômico na busca de soluções mais eficientes e abrangentes;
- É fácil comparar soluções alternativas e identificar benefícios potenciais usando novas soluções propostas.

Embora o Problema do Corte Industrial tenha sido pesquisado em diversos trabalhos [1][11][17][23][31][39], somente alguns deles resultaram num aplicativo que pudesse ser facilmente disseminado para uso industrial. Isso é importante, afinal de contas, o usuário final desse tipo de aplicativo encontra-se dentro das linhas de produção dos mais diversos setores da indústria secundária e terciária, a saber:

- | | |
|---------------------------------|-----------------------------------|
| ▪ Indústria da Construção Civil | ▪ Indústria Siderúrgica |
| ▪ Indústria de Armazenamento | ▪ Indústria Moveleira |
| ▪ Indústria de Papel e Gráfica | ▪ Indústria Mecânica, em geral |
| ▪ Indústria Têxtil e de Moda | ▪ Indústria de Couro e Calçadista |
| ▪ Indústria de Transporte | ▪ Indústria Naval |

Daí a importância desse trabalho.

1.3 UM ESTUDO DE CASO

Em geral, os níveis de desperdício de matéria-prima no processo industrial, giram em torno de 2% a 4% da matéria-prima utilizada. Isso pôde ser constatado a partir de dados coletados em visita técnica realizada numa indústria do setor metalmeccânico instalada na cidade de Joinville. Essa indústria fabrica produtos para cozinha (*basicamente pias inoxidáveis*) e banheiro (*ralos e mictórios*) a partir de chapas de aço inoxidável. As chapas são adquiridas diretamente de siderúrgicas nacionais e internacionais, e vêm enroladas em forma de bobinas.

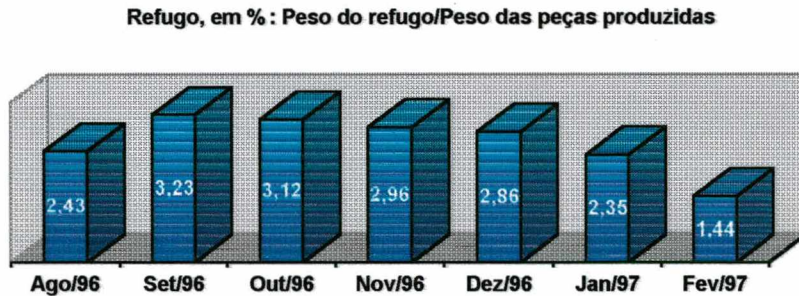


Figura 1-1: Exemplo do percentual de desperdício de matéria-prima

Uma grande quantidade de matéria-prima é cortada durante o processo de fabricação das peças, resultando também em sobras ou refugos da matéria-prima. O gráfico da Figura 1-1 ilustra o percentual de refugo com base no peso de aço inoxidável utilizado para produção mensal das peças. Eles consideravam como aceitáveis, e essa era a meta, níveis de refugo de até 2% do material empregado nos produtos. Para se ter uma idéia da quantidade de matéria-prima desperdiçada, somente no mês de fevereiro do ano de 97 quando a meta inicial foi alcançada, foram refugados quase 3 t de sobras de aço inoxidável. Segundo uma consulta feita no mercado atacadista de São Paulo durante o mês de setembro de 1998, e considerando que a maior parte da matéria-prima cortada eram bobinas de aço inoxidável 304 laminado a frio de espessura 0,50 mm; esse volume de material corresponde a R\$14.100,00 (*catorze mil e cem reais*) mensais. A Tabela 1-1 fornece a cotação do preço de bobinas de diferentes tipos de aço inoxidável para o mês de setembro de 1998.

Tabela 1-1: Cotação do preço de bobinas de aço inoxidável (SET/98)

TIPO DE AÇO INOXIDÁVEL		PREÇO, EM R\$/kg		
		304	316	430
Laminado a quente	3,00 a 8,00 mm	3,30	4,40	*****
Laminado a frio	2,00 mm	3,90	5,15	3,15
	0,50 mm	4,70	*****	4,10

Fonte: Losango Aço Inoxidável Ltda.

Capítulo 2

PROBLEMA DO CORTE INDUSTRIAL

O estudo do corte e do encaixe de figuras abrange uma grande variedade de problemas teóricos e, principalmente, práticos. Estes problemas estão inseridos dentro do contexto de atuação da Engenharia e de muitas outras áreas do conhecimento humano, tais como: Administração, Economia, Matemática, Pesquisa Operacional, Teleinformática e outras.

As informações processadas nesses problemas carregam características próprias interligadas por vínculos de natureza geométrica ou combinatória, e muitas vezes por ambos os tipos e em alguns casos, por nenhum deles. De forma sucinta, o objetivo principal visado nesse tipo de problema pode ser visto como a procura e obtenção de um arranjo geometricamente eficiente de figuras, de áreas menores, inseridas numa área maior. Dentro dessa visão, o estudo do Problema do Corte Industrial, conhecido na literatura inglesa como *Cutting Stock Problem*, interpreta um arranjo geometricamente eficiente como aquele que ao final do processo de corte gera a menor quantidade de sobras de matéria-prima, ou maior rendimento de uso da matéria-prima original.

Na literatura científica, os problemas denominados de *Cutting and Packing*¹, que quer dizer *Corte e Empacotamento*, são denominações de problemas que objetivam otimizar um determinado arranjo de figuras. Fica mais claro, a partir da leitura de trabalhos científicos que versam sobre questões de *C&P*, o porquê da diversidade de nomes em problemas que ostentam a mesma estrutura lógica de dados. A lista seguinte endossa este fato:

- **Problema de Perdas por Corte** (*Trim Loss Problem*);
- **Empacotamento de Caixas, Tiras ou Faixas** (*Bin Packing*);
- **Problema da Mochila** (*Knapsack Problem*);
- **Carregamento de Veículos** (*Vehicle Loading*);

¹ Deste ponto em diante, a citação ao termo em inglês *Cutting and Packing* será substituída pela abreviação *C&P*.

- **Carregamento de Paletes e Contêineres** (*Container and Pallet Loading*);
- **Problemas de Leiaute** (*Layout Problems*);
- **Problemas de Encaixe e Particionamento** (*Nesting and Partitioning Problems*);
- **Problema de Cálculo de Capital** (*Budgeting Capital Problem*);
- **Balanceamento de Linhas de Montagem** (*Assembly Line Balancing*);
- **Problemas de Alocação de Memória** (*Memory Allocation Problems*);
- **Problemas de Ordenamento de Multitarefa** (*Multiprocessor Scheduling Problems*).

No cerne de todos esses tópicos, pode-se vislumbrar uma mesma estrutura lógica básica de dados e informações. As particularidades inerentes a cada um dos tópicos advêm de restrições acrescentadas às formulações das técnicas e algoritmos computacionais de solução do problema específico.

As seções seguintes do texto descrevem os problemas de *C&P*, enfatizando mais o Problema do Corte Industrial, que é o escopo desse trabalho. Também são apresentados detalhes e pontos de vista diferentes sobre os problemas de *C&P*, ajudando-nos a melhor identificá-los e classificá-los.

2.1 DESCRIÇÃO

As sobras de matéria-prima durante todo o ciclo de extração, beneficiamento e processamento são naturalmente inevitáveis. Contudo, nas indústrias de produtos planos, como por exemplo, papel, tecido, couro, plástico e laminado, esse problema atinge níveis preocupantes; o que tem levado ao estudo de soluções para reduzir a quantidade de material desperdiçado no processo de corte.

A matéria-prima, normalmente, é fornecida em rolos ou chapas de diversas larguras. Na maioria dos casos, essa matéria-prima tem que ser recortada em diversas peças de larguras menores e valores distintos [23]. O arranjo dos cortes é, na maioria das vezes, realizado através de métodos práticos e corriqueiros que não empregam um método de otimização. Esses procedimentos geram quantidades inaceitáveis de sobras de matéria-prima e isso repercute diretamente no custo final da produção. A redução deste desperdício pode ser colocado como um problema de otimização e solucionado a partir do uso de técnicas de programação matemática linear [11].

A solução desse tipo de problema consiste em determinar qual deve ser o melhor arranjo, no qual as peças retangulares menores deverão ser dispostas dentro de uma área retangular maior; e também pela determinação do seqüenciamento do corte, de modo que a minimização das sobras de matéria-prima seja atendida [11].

2.2 CLASSIFICAÇÃO E TIPOLOGIA

A classificação e a tipologia dos problemas de *C&P* discutidos aqui, baseiam-se no trabalho de Dyckhoff [14], o qual objetivou unificar a terminologia empregada na literatura e com isso, definir e entender melhor as variantes desse tipo de problema.

A estrutura lógica básica de problemas de *C&P* pode ser colocada nos seguintes termos:

(a) Existem dois grupos de dados básicos cujos elementos definem corpos geométricos de formas precisas em espaços uni ou multidimensionais de números reais. Os dois grupos são:

— Estoque de objetos;

“Objetos” é entendido aqui como uma referência ao material que serve de matéria-prima para o processo de *C&P*.

— Lista de pedidos de itens.

O termo “itens” refere-se às peças ou produtos que são gerados no processo de *C&P* e que pertencem à lista de pedidos.

É oportuno definir dois conceitos importantes para terminologia dos problemas de *C&P*: *padrão de corte* e *sobras*. Entende-se por *padrão de corte* a todas as configurações possíveis que existem de se cortar um determinado comprimento nominal da matéria-prima em comprimentos menores que pertencem à lista de pedidos. Em alguns trabalhos, adota-se também o termo “*atividade de corte*”. Pedacos residuais, isto é, a ocorrência de padrões de corte com figuras que não se referem a itens constantes na lista de pedidos, são denominados de *sobras*.

(b) Os processos de corte ou de empacotamento se caracterizam por padrões gerados a partir de combinações geométricas entre itens inseridos dentro de objetos.

Essa estrutura lógica básica de problemas de *C&P* fica óbvia quando se analisa alguns exemplos clássicos.

Um caso bastante comum no contexto da Engenharia Civil diz respeito ao corte de vergalhões de aço para estrutura metálica de um prédio. Como ilustra a Figura 2-1.

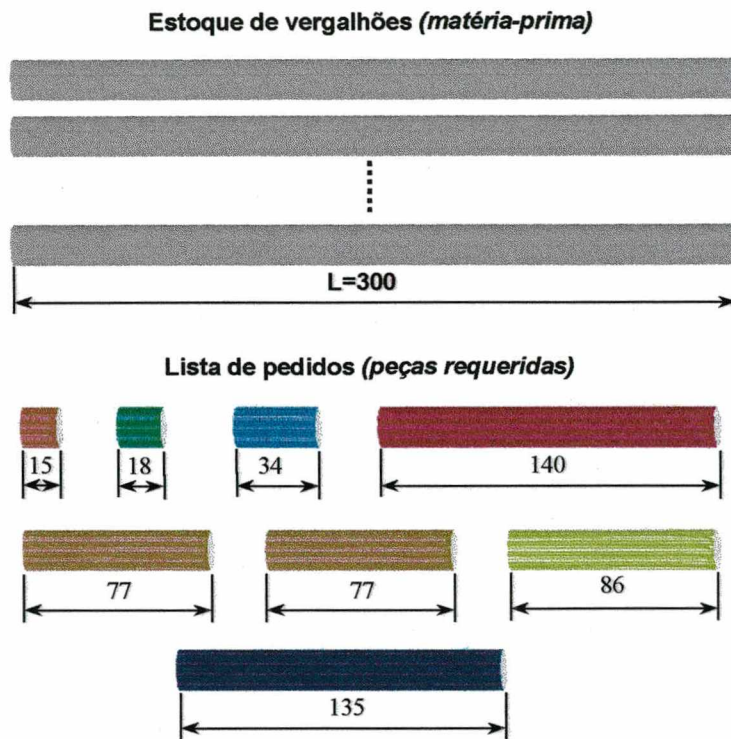


Figura 2-1: Exemplo do corte de vergalhões

Usando como base a terminologia definida acima, pode-se identificar dois grupos de dados principais. De um lado há um estoque de vergalhões de comprimento L , os quais são as matérias-primas do processo de corte, usados para produzir vergalhões de tamanhos menores. A quantidade de vergalhões, em estoque, disponíveis para o corte é considerada ilimitada. Do outro lado há uma lista de pedidos constituída de vergalhões com comprimentos variando entre l_1, l_2, \dots, l_m , a ser produzida para atender uma demanda. Cada comprimento l_i deve ser menor ou igual a L . Estes dois grupos constituem os dados básicos do Problema do Corte Industrial. A partir desses dois grupos de dados, e acrescentando-se certos objetivos e restrições específicos de cada situação, são feitas diversas combinações para gerar padrões de

corte entre os vergalhões da lista de pedidos e os vergalhões do estoque. A Figura 2-2 mostra um exemplo de padrões de corte para atender à lista de pedidos da Figura 2-1.

Vale salientar que este exemplo objetiva apenas mostrar a terminologia definida para este tipo de problema, sem se preocupar em já aplicar algum método de otimização para redução das sobras de material. Uma parte dos itens da lista foram dispostos num primeiro padrão de corte constituído por $I = [140, 86, 34, 18, 15]$, resultando numa sobra de 7 u.c.². O segundo padrão de corte atende ao restante da lista com $I = [135, 77, 77]$ e restando de sobra um comprimento de 11 u.c. da matéria-prima $L = 300$ u.c..

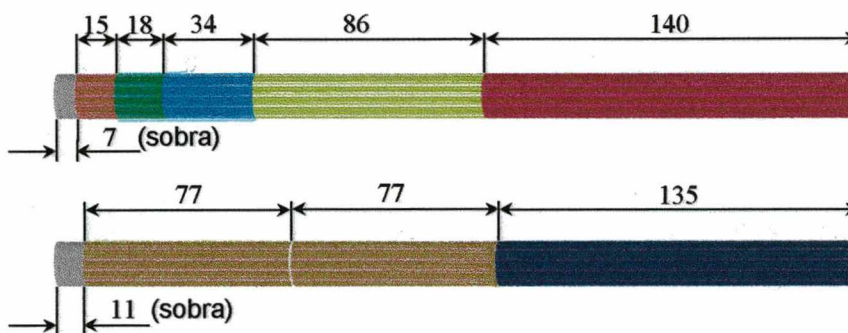


Figura 2-2: Exemplo de padrões de corte entre os itens da lista de pedidos

Um outro tipo de exemplo de problema *C&P* é o problema de empacotamento. A Figura 2-3 ilustra a composição de dados para um problema de empacotamento em contêineres. Neste tipo de problema o objetivo é reduzir os espaços vazios dentro dos contêineres empilhando os itens da lista de pedidos, os quais se constituem de caixas em forma de paralelepípedo. O contêiner possui comprimento L , altura H e largura W definidos. Os itens possuem volumes diversos com uma largura constante w em todos.

A lista de pedidos acima apenas mostra a forma das caixas e suas dimensões principais, não fazendo referência à quantidade requerida de cada uma delas. Para identificação das caixas utiliza-se um código formado por dois algarismos, cujo primeiro algarismo informa a altura h da caixa e o segundo informa o seu comprimento l . Baseando-se neste código, a Figura 2-4 mostra um padrão de empacotamento possível para uma lista de pedido, o qual é constituído por 2 caixas 23, 2 caixas 34, 1 caixa 35, 2 caixas 42 e 1 caixa 45.

² u.c. é abreviatura de unidade de comprimento.

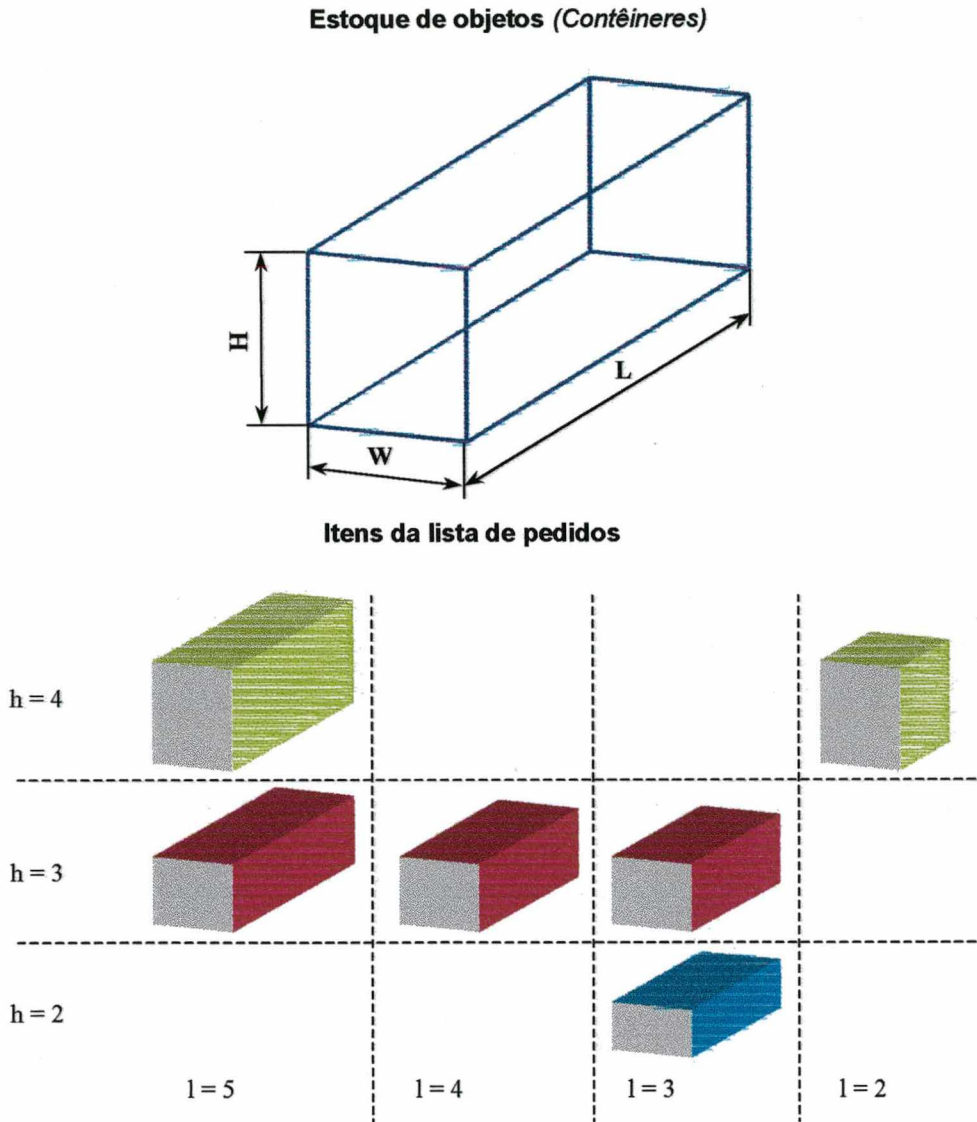


Figura 2-3: Exemplo de empacotamento em contêineres

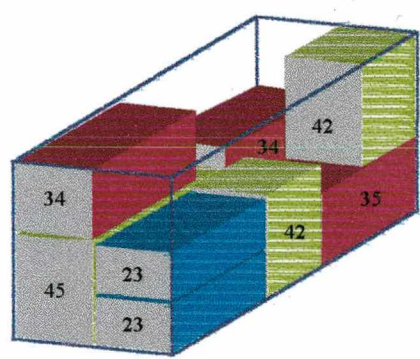


Figura 2-4: Exemplo de um padrão de empacotamento

Com os exemplos acima pode-se perceber um estreito relacionamento entre problemas de corte e problemas de empacotamento, resultante da existência de uma natureza dual entre **material—espaço**. O empacotamento de itens pode ser visto como uma ação de cortar espaços vazios de objetos; transformando alguns desses espaços em itens da lista de pedidos e alguns outros em sobras de material. O corte, por sua vez, pode ser visto como uma ação de empacotar espaços ocupados por itens dentro do espaço ocupado por objetos. Já no caso do Problema do Corte Industrial, os objetos são vistos como materiais sólidos e resistentes a serem cortados em itens, os quais são as peças constantes da lista de pedidos. Levando-se em conta que o objetivo principal é a minimização das sobras de corte, pode-se concluir que o Problema de Perda por Corte tem a mesma natureza que o Problema do Corte Industrial.

Afora certos objetivos e/ou restrições próprios e característicos de cada uma das variantes do problema de *C&P*, o principal aspecto constitui-se da geração de padrões a partir de combinações realizadas entre o grupo dos itens de uma lista de pedidos e o estoque de objetos disponíveis para o processo de corte ou empacotamento.

Num contexto mais restrito, os problemas de *C&P* se preocupam com objetos e itens definidos por uma, duas ou três dimensões do espaço geométrico euclidiano. Todavia, os problemas de *C&P* também podem ser considerados de forma mais generalizada e abstrata com vínculos de natureza adimensional. Pode-se citar alguns trabalhos nessa linha:

- Carregamento de veículos, com enfoque no peso da carga;
- Balanceamento de linhas de montagem e ordenamento de multitarefas, com enfoque no tempo de operação;
- Cálculo de capital de giro, com enfoque em custos financeiros e;
- Alocação de memória, com enfoque no armazenamento de dados.

A Figura 2-5 mostra bem a semelhança entre a estrutura lógica básica de problemas de *C&P* com um contexto definido fisicamente e outro definido de forma mais abstrata exemplificado por um problema de ordenamento de multitarefas. Neste tipo de problema, normalmente se tem um número definido de estações ou bancadas de trabalho disponíveis para execução das tarefas dentro de um período de tempo definido. No exemplo, dispõe-se de 9 bancadas de trabalho *S* com períodos de tempo *T* limitados. As tarefas $[t_1, t_2, \dots, t_{10}, t_{11}]$ a serem executadas constituem os itens da lista

de pedidos. Haja vista que, em alguns casos, as tarefas possuem uma seqüência de execução definida, tal restrição particular do problema deve ser considerada. É o caso das tarefas t_2, t_3, t_4 e t_5 , cuja execução depende da tarefa t_1 ser concluída. Com exceção dessa restrição, o *Problema de Ordenamento de Multitarefas* possui a mesma formulação do *Problema de Empacotamento de Caixas*.

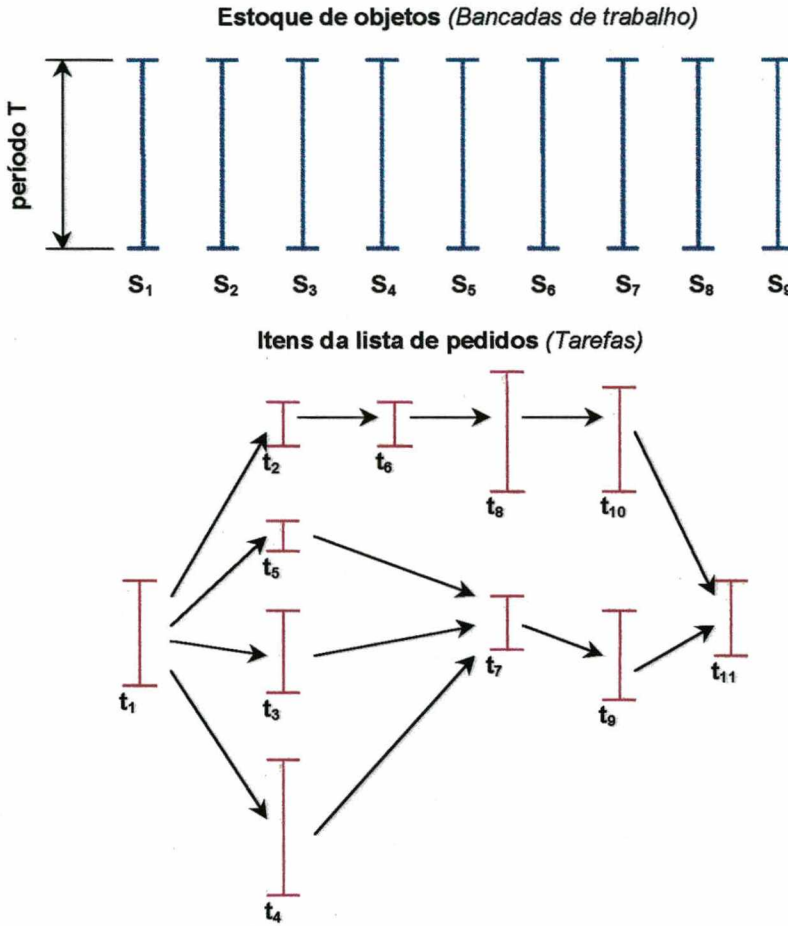


Figura 2-5: Exemplo de um problema C&P abstrato

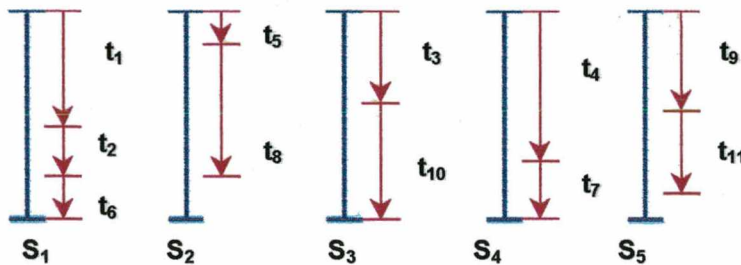


Figura 2-6: Exemplo de um padrão de ordenamento de tarefas

Atendendo a todas as restrições de limitação de tempo e de ordenamento especificadas para o problema, a Figura 2-6 mostra os padrões de alocação das bancadas de trabalho disponíveis, com o respectivo ordenamento das tarefas.

A estrutura sistemática elaborada por Dyckhoff [14] estabeleceu uma classificação dos problemas de *C&P* com base em características que se referiam aos seguintes tópicos: estoque de matéria-prima (*objetos*), lista de pedidos de itens (*peças ou produtos a serem obtidos*), combinações geométricas entre figuras dos padrões; e considerações entre *peças obtidas versus padrões geométricos* e entre *padrões geométricos versus matéria-prima* em estoque. Todos esses tópicos, por sua vez, são também referenciados quanto às características geométricas e/ou combinatórias. Tais características analisadas por Dyckhoff foram: **dimensionalidade**, **sistema de medida**, **forma geométrica**, **sortimento**, **disponibilidade**, **restrições**, **objetivos**, **valor da informação e variabilidade**. Na seqüência, são feitos alguns comentários sobre cada uma dessas características.

DIMENSIONALIDADE

A dimensionalidade é a primeira característica a ser considerada nos problemas de *C&P* devido à importância que a geometria exerce na maioria desses problemas.

A **dimensionalidade** é definida como o número mínimo de dimensões necessárias para descrever a geometria dos padrões. Por essa característica, pode-se classificar os problemas de *C&P* como: **unidimensional** (Ver Figura 2-1), **bidimensional**, **tridimensional** e **multidimensionais**. Problemas de empacotamento tridimensionais (Ver Figura 2-3) que consideram uma variável de tempo, são denominados de problemas multidimensionais. Na Figura 2-7, ilustra-se um exemplo de um encaixe de retângulos com comprimento e largura definidos no espaço bidimensional.

Embora possa parecer, nem sempre é claro e direto classificar problemas de *C&P* com respeito a sua dimensionalidade. Isso porque não são somente aspectos geométricos definem a questão. Considere como exemplo o caso do carregamento de paletes ou contêineres, que normalmente é considerado como bidimensional. Nestes problemas, o empilhamento é feito primeiro em pilhas verticais e, em seguida, as pilhas são dispostas horizontalmente na base do palete ou contêiner. Assim, ao invés de um problema de dimensionalidade "3", tem-se um problema de dimensionalidade "2+1".

Analogamente, problemas de corte de placas de vidro são considerados "1+1"-dimensional visto que as placas só podem ser cortadas por cortes que se prolongam por toda extensão da peça³.

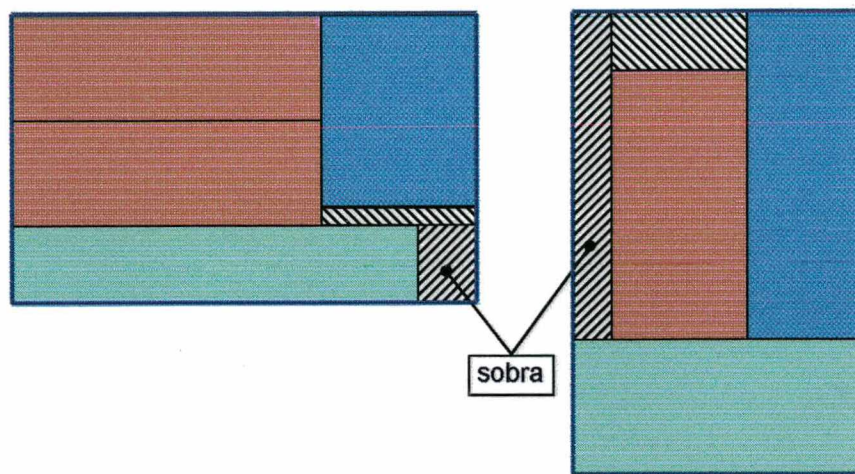


Figura 2-7: Exemplo de bidimensionalidade

SISTEMA DE MEDIDA

Outra característica importante é a maneira de medir a quantidade dos objetos e dos itens. Gilmore, em 1979, definiu os dois sistemas de medida possíveis:

— Sistema de medida discreta ou inteira, i. e., através de números naturais;

Este sistema referencia o número de objetos ou itens que apresentam uma determinada forma geométrica.

— Sistema de medida contínua, i.e., através de números reais.

Já neste caso, o sistema referencia a dimensão contínua da grandeza principal com respeito à dimensão mais relevante de objetos ou itens. Por exemplo, no caso de peças unidimensionais, referencia-se o seu comprimento. Ao invés do comprimento, poder-se-ia ter também grandezas de massa (*peso de uma peça*) ou diâmetro (*em rolos*).

Na literatura especializada, um tipo especial de problema de *C&P* combinado ficou conhecido como $1\frac{1}{2}$ -dimensional. Este caso trata com problemas unidimensionais

³ Este é um tipo especial de corte conhecido como corte guilhotina.

que possuem duas dimensões, sendo que uma delas é definida e constante e a outra é variável e contínua. Um exemplo típico desse caso, refere-se ao problema de corte de placas de vidro produzidas por processo contínuo e cuja largura de corte é constante.

FORMA GEOMÉTRICA

Outra característica principal de problemas de *C&P* diretamente relacionado a dimensionalidade é a forma das figuras dos objetos e dos itens. A **figura** de um objeto ou um item é definida como sendo a representação geométrica no espaço das suas dimensões principais. Objetos ou artigos com mesma figura possuem a mesma representação geométrica, com exceção de algumas translações dentro do espaço de dimensões. Ignorando tais translações, uma figura pode ser determinada univocamente pela sua **forma**, pelo seu **tamanho** e pela sua **orientação**.

Figuras de mesma forma podem diferir no tamanho ou na orientação dentro do espaço dimensional. Para problemas multidimensionais é importante saber se a forma das figuras é regular ou se é irregular. Uma forma é dita regular se puder ser descrita através de um número reduzido de parâmetros. Exemplos de formas bidimensionais regulares são círculos e quadrados. A maioria dos problemas considerados na literatura se preocupam com formas regulares, especialmente formas retangulares. Formas irregulares, não-convexas e assimétricas, são problemas típicos de indústrias que trabalham com corte de metal, tecido, couro, ou madeira. Peças retangulares com diferentes razões de largura e comprimento constituem formas regulares distintas. Todavia figuras que diferem no tamanho tornar-se-ão idênticas se forem realizadas translações e mudanças na escala igualmente em todas as dimensões. Um aspecto importante quanto à facilidade ou dificuldade de se resolver um problema específico está freqüentemente associado ao tamanho relativo dos objetos e itens.

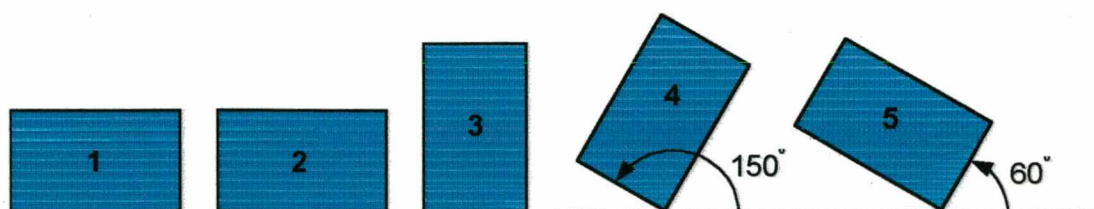


Figura 2-8: Diferenciação entre figuras considerando a orientação

Figuras com forma e tamanho iguais, podem diferir somente com respeito à orientação. Neste caso, elas são ditas **congruentes**. Pode-se distinguir três situações relativas a orientação. A ilustração da Figura 2-8 exemplifica as três situações.

1. Se qualquer orientação é aceita, então objetos e itens com figuras congruentes não podem ser diferenciados. Neste caso, todos os retângulos são idênticos.
2. Se somente rotações de 90° são permitidas, são considerados idênticos somente os objetos e itens que podem ser feitos idênticos por rotações de 90°. Assim, podemos tornar idênticos os retângulos **1** e **2** com o **3**; e também o retângulo **4** com o **5**.
3. Se a orientação é fixada, então objetos e itens com figuras congruentes são diferenciados, com exceção daqueles que podem ser feitos idênticos através de translações. Aqui, apenas pode-se tornar idênticos os retângulos **1** e **2**.

Com estas considerações pode-se afirmar que figuras unidimensionais com forma e orientação iguais só diferem no tamanho, ou seja, em seus comprimentos.

SORTIMENTO

Define-se a diversidade de figuras, conhecido tecnicamente por **sortimento**, como a determinação do número e forma das figuras aceitas pelo problema.

Concentrando-se no aspecto do sortimento de figuras, pode-se distinguir problemas de dois tipos: aqueles que possuem formas diferentes, e aqueles onde todos os objetos e itens têm a mesma forma. Em ambos os casos as figuras podem ser diferentes, não importando se isso ocorre em muitas ou só em algumas figuras. No segundo caso, diferenças entre figuras resultam do tamanho ou da orientação das mesmas. Tipos especiais de problemas surgem quando as figuras são congruentes ou até mesmo idênticas. Por exemplo, o carregamento de paletes e o empacotamento de caixas normalmente se referem a problemas cujos objetos possuem a mesma figura como matéria-prima, sendo que os itens têm figuras congruentes no caso do carregamento de paletes e figuras diferentes no caso do empacotamento de caixas.

DISPONIBILIDADE

O sortimento das figuras nada diz sobre a quantidade de objetos e itens considerados no pedido. Isto é dado pela disponibilidade de objetos ou itens das figuras aceitas. A disponibilidade considera:

- Os limites inferiores e superiores para a quantidade de objetos ou itens;

- O seqüenciamento do processo;
- A data quando objeto ou item pode ser, ou tem que ser cortado ou empacotado.

Pode-se distinguir entre um número infinito ou finito de objetos ou itens. No caso de ser finito, pode-se ter muitos ou só alguns objetos ou itens. Nos problemas de empacotamento de caixas e de corte industrial, normalmente está se tratando com um número infinito de objetos que possuem uma mesma figura como base. Além disso, comparando-se os problemas de empacotamento de caixas e corte industrial nota-se que no caso do empacotamento de caixas existe um baixo sortimento de itens e uma grande quantidade de cada um desses itens; enquanto que no caso do corte industrial existe um grande sortimento de itens e uma pequena quantidade de cada um desses itens. Problemas de carregamento de paletes ou problemas de Knapsack são normalmente caracterizados por possuírem só um objeto.

É comum o número de objetos e itens dados ser interpretado como o limite superior do pedido. Entretanto, em muitos casos, também se considera a existência de um limite inferior. Por exemplo, um objeto no carregamento de paletes tem que ser empacotado com certos itens. Reciprocamente, todos os itens de um problema de empacotamento de caixas ou de corte industrial têm que ser empacotados ou cortados, respectivamente. Quando limites inferiores e superiores diferem, pode-se interpretá-los como tolerâncias de folga aceitáveis com respeito ao número dos itens do pedido.

Características adicionais podem ser consideradas quando existe uma certa ordem para os objetos e itens, ou quando o tempo exerce um papel relevante. Pode-se citar alguns problemas onde isso é importante. Nos problemas de balanceamento de linhas de montagem, uma ordem parcial dos itens tem que ser levada em conta. Por exemplo, as placas de aço produzidas por processo de lingotamento têm que seguir uma seqüência de corte sincronizada com a produção e também sem intervalos ou interrupções do processo. Itens a serem empacotados em contêineres podem ter diferentes datas para serem transportados.

RESTRICÇÕES

As restrições do problema de *C&P* se limitam a dois grupos importantes: restrições de geração dos padrões e restrições da forma de designação dos itens com relação aos objetos. Na geração de padrões, de acordo com o problema em particular, define-se as restrições de cunho geométrico e/ou combinatório aplicáveis. Para o

segundo grupo, define-se as restrições quanto ao número de estágios, à frequência e ordem dos padrões, e quanto às associações realizadas entre padrões.

Por sua vez, as restrições de padrão podem ser agrupadas em quatro grupos que podem ser considerados durante a geração dos padrões. A saber:

- A mínima ou máxima distância entre itens ou entre cortes que dividem objetos. Isso pode ser relevante quando cortamos vidros ou carregamos contêineres;
- A orientação dos itens relativa a um ou outro item e/ou em relação aos objetos. Às vezes isso tem que ser levado em conta quando cortamos tecidos estampados ou quando transportamos materiais frágeis em paletes;
- A frequência de itens num padrão em particular. Normalmente considerada com respeito à associação ou quantidade de itens diferentes no pedido, ou à quantidade de itens em níveis totais ou relativos entre si. Isso é importante quando uma quantidade máxima de desperdício é fixada. Isso também é considerado em situações onde limitamos a quantidade máxima de faixas de tamanhos dos itens ou objetos ou limitamos o número de pedidos;
- O tipo e o número de cortes permitidos são parâmetros essenciais na classificação dos tipos de problemas de corte.

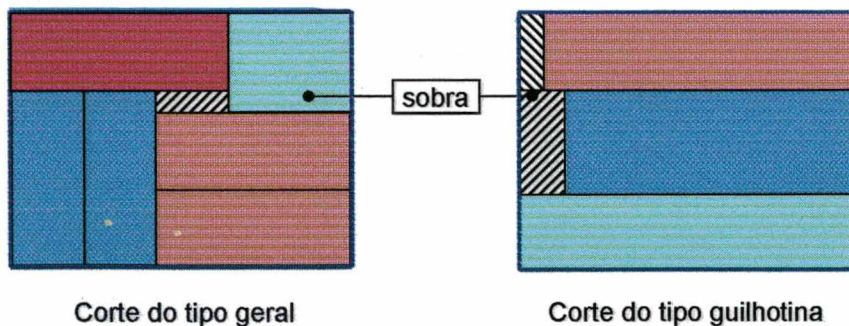


Figura 2-9: Exemplo de cortes

No caso particular dos objetos e itens serem de forma retangular, considerando a posição do retângulo a ser cortado em relação ao retângulo de base, ou também conhecido como retângulo de origem, podemos distinguir padrões de corte ortogonais e não-ortogonais. Se os cortes são realizados paralelos aos lados do retângulo de base, o corte é dito ortogonal. Em caso contrário, temos o corte não-ortogonal que é bastante

aplicado em materiais isotrópicos, cujas propriedades mecânicas se mantêm constantes em todas as direções.

Adicionalmente, pode-se ainda classificar esses cortes como sendo do tipo mais geral ou corte do tipo guilhotina, conforme pode ser visto na Figura 2-9. O corte do tipo guilhotina é caracterizado por se prolongar de um lado ao outro do retângulo de base.

No caso das restrições da forma de designação de itens em relação aos objetos, normalmente dois passos devem ser executados. Primeiramente, os itens são dispostos dentro de um padrão válido, e em seguida são dispostos em objetos. Podem existir restrições quanto:

- **Forma de designação dos itens com relação aos objetos**

A designação aos objetos e aos itens têm que estar bem clara para que se saiba quando a seleção faz menção apenas a um ou a todos que estão dentro de um padrão. O Problema de Knapsack e do empacotamento de caixas são exemplos típicos de problemas onde uma seleção de um dado item tem que ser associada a um padrão, tal que para cada um dos objetos dados um padrão está associado. Numa definição contrária, que é típica para problemas de corte industrial e empacotamento de caixas, todos os itens têm que ser definidos apropriadamente para uma seleção dos objetos.

- **Número de estágios**

O número de estágios refere à questão sobre se os itens são cortados ou empacotados em apenas um passo ou gradativamente em vários passos. O processo de corte pode ser realizado num certo número de fases seqüenciais, conhecidas como **estágios**. Disso, pode-se classificar os cortes de acordo com o número de estágios. Cada estágio diz respeito a uma direção de corte permitida. A complexidade dos padrões de corte tipo guilhotina é função do número de estágios e do número de cortes adicionais paralelos que devem ser feitos dentro de cada estágio. Como exemplo, pode-se citar o caso do corte de materiais caros, tais como seda, onde a direção de corte pode ser alterada a todo momento, assumindo um novo padrão de corte. Por causa da fraca estabilidade dos padrões, os cortes do tipo guilhotina são normalmente de pouco interesse em problemas de carregamentos de paletes. A aplicação desse tipo de corte é mais indicada para indústria de corte de planos.

▪ Número, frequência e seqüência dos padrões

A ordem ou seqüência de padrões dentro do processo de definição pode ser uma restrição; e em geral, induzida por uma relação de dependência entre objetos e itens, ou pela própria tecnologia dos processos de *C&P*. Também existem restrições que se referem à frequência de padrões; ou seja, à quantidade permitida de tipos diferentes de padrões ou à quantidade de padrões do mesmo tipo.

OBJETIVOS

Nem sempre é possível decidir claramente se uma característica é geométrica ou combinatória. Algumas incluem ambos os aspectos e outras nenhuma desses. Os objetivos de problemas de *C&P* têm frequentemente aspectos geométricos e combinatórios interligados. Além disso, eles podem ser atribuídos aos objetos, itens, padrões, ou ao processo de designação. Aqui, um objetivo significa um critério a ser otimizado, seja maximizado ou minimizado.

Grosseiramente, pode-se diferenciar os objetivos dependendo se eles se referem: à quantidade de objetos ou de pedaços pequenos e residuais referenciados para os padrões; à geometria dos padrões; à seqüência, à combinação, ou ao número de padrões.

Funções objetivo que apresentam comportamento linear com respeito às quantidades podem estimar o valor de objetos ou itens pelo seu tamanho, ou como é mais comum, pelo seu preço. Objetivos não lineares aparecem no caso onde as sobras relativas são minimizadas, ou se quantidades prefixadas para os padrões têm que ser levadas em conta. Além disso, é típico em muitos problemas de *C&P* a existência de mais de um objetivo a ser considerado.

VALOR DA INFORMAÇÃO E VARIABILIDADE

As características de estado de informação e de variabilidade são pertinentes na identificação dos dados do problema, classificando-os como determinísticos, estocásticos ou incertos; e se eles são fixos ou podem variar em certos intervalos. Por exemplo, rolos de filme de plástico podem possuir defeitos de borda variando dentro de uma faixa. Outro exemplo é a produção contínua de placas de metal onde tamanhos de faixas bem precisos são cortados de lingotes cujos tamanhos não são precisos. A demanda para itens da lista de pedidos normalmente é determinística, podendo ainda

ser de variável desde que certas tolerâncias sejam aceitas pelos clientes. A imprecisão de medições na prática, também pode ser uma razão para variabilidade dos dados.

Em vista da grande variedade de problemas de *C&P* na prática, as características consideradas acima não provêm uma lista completa de todas as possíveis propriedades. Além disso, algumas propriedades podem se sobrepor. Por exemplo, há uma relação forte entre os limites de disponibilidade, o tipo de designação, e esses objetivos que se referem a quantidades de objetos e itens.

Com o objetivo de integrar os vários tipos de problemas de *C&P* numa tipologia abrangente e representativa, além de estabelecer as bases para definição dos mais importantes problemas desse tipo, para os quais as futuras pesquisas deveriam concentrar esforços, Dyckhoff limitou as características principais em quatro grupos e adotou uma convenção simbólica apropriada para identificá-las.

Foi utilizada uma convenção simbólica denotada por uma seqüência $\alpha/\beta/\delta/\gamma$ constituída por números e letras, de acordo como definido abaixo. Por exemplo, *3/B/O/F* denotam todos os problemas de *C&P* tridimensionais onde um objeto tem que ser empacotado num processo de escolha com pouca diversidade de itens. Chama-se a atenção de que um símbolo ausente significa que são possíveis todas as combinações entre as respectivas propriedades.

[1] Dimensionalidade (α)

- (1) unidimensional;
- (2) bidimensional;
- (3) tridimensional;
- (N) n-dimensional, com $n > 3$;

[2] Tipo de designação (β)

(B) Todos os objetos e uma seleção de itens;

Uma escolha de itens tem que ser combinada a padrões, tal que a cada grande objeto é designado um padrão não trivial.

(V) Uma seleção de objetos e todos os itens;

Todos os itens têm que ser combinados a padrões, os quais em seguida são designados a uma escolha apropriada de objetos.

[3] Sortimento de objetos (δ)

(O) Um objeto:
So há um objeto definido. Como exemplo, temos os problemas de carregamento de paletes.

(I) Figuras idênticas;

Todos os objetos possuem a mesma figura. Como exemplo, temos os problemas de empacotamento de caixas.

(D) Figuras diferentes;

Os objetos têm figuras diferentes. O Problema de Corte Industrial se insere aqui.

[4] Sortimento de itens (γ)

(F) Listas de pedido de poucos itens com figuras distintas;

São problemas de poucos itens e, normalmente, com figuras distintas. Como exemplo, temos os problemas de carregamento de veículos com até 10 itens distintos.

(M) Lista de pedidos de muitos itens com muitas figuras distintas;

Existem muitos itens, muitos dos quais com figuras distintas.

(R) Lista de pedidos de muitos itens com algumas figuras distintas;

Existem muitos itens mas poucas figuras distintas. Em geral, milhares de cortes e itens com menos de 50 figuras distintas.

(C) Lista de pedidos com figuras congruentes;

Todos os itens são congruentes. Como exemplo, temos os problemas de carregamento de paletes.

Para constatar a utilidade deste sistema, podemos nos basear em tipos especiais de problemas de *C&P*, tipicamente usados na literatura e mostrados na Tabela 2-1.

A convenção simbólica deixa evidente que os problemas de carregamento de veículos, empacotamento de caixas e do corte industrial pertencem ao mesmo tipo de problemas combinado com respeito às primeiras três características, quais sejam: **1/V/I**. Porém, eles diferem no sortimento dos itens. Já no caso dos problemas de balanceamento de linhas de montagem, ordenamento de multitarefas e alocação de memória; fica óbvio que pertencem ao mesmo tipo dos problemas clássicos de

empacotamento de caixas. Diferenças entre eles se referem a características que não foram discutidas e nem referenciadas dentro da classificação definida por Dyckhoff.

Conseqüentemente para identificar problemas de forma única, tem-se que ampliar a convenção simbólica a fim de que ela englobe mais características, além daquelas sugeridas por Dyckhoff. Todavia, devido a grande variedade de problemas de C&P, uma extensa listagem de características não faz muito sentido, haja vista que o objetivo é utilizar uma classificação que reúna tais problemas numa terminologia mais geral. Somente no caso de um segundo nível de classificação do tipo de problema específico é que se deve ampliar a simbologia sugerida por Dyckhoff, adicionando-se mais características à convenção. Em princípio, as características adicionais devem pertencer ao grupo daquelas que já foram definidas e comentadas acima.

Tabela 2-1: Simbologia dos principais tipos de problemas de C&P

TIPOS DE PROBLEMAS COMBINADOS	REPRESENTAÇÃO SIMBÓLICA
Knapsack	1/B/O/- ⁴
Carregamento de paletes	2/B/O/C
Knapsack multidimensional	-/B/O/-
Empacotamento de caixas – Dual	1/B/O/M
Carregamento de veículos	1/V//F ou 1/V//M
Carregamento de Contêineres	3/V//I- ou 3/B/O/-
Empacotamento de caixas – Clássico	1/V//M
Corte industrial – Clássico	1/V//R
Empacotamento de caixas – 2D	2/V/D/M
Corte industrial – 2D Clássico	2/V//R
Corte industrial ou Perdas por Corte – Geral	1/-/-/, 2/-/-/ ou 3/-/-/
Balanceamento de linhas de montagem	1/V//M
Ordenamento de multitarefas	1/V//M
Alocação de memória	1/V//M
Cálculo de capital para “n” períodos	n/B/O/-

Fonte: Ver artigo de Dyckhoff [14]

⁴ O hífen significa ausência de símbolo.

Capítulo 3

REVISÃO BIBLIOGRÁFICA

Muitas heurísticas, algoritmos e métodos de solução foram desenvolvidos desde a metade dos anos cinquenta. Estes métodos tratam da aplicação de algoritmos de enumeração, tais como algoritmo de Branch-and-Bound; de Programação Dinâmica; de Simulação e de Programação Linear pelo Método Simplex Revisado.

No texto que se segue são considerados os trabalhos que contribuíram para o desenvolvimento dos estudos relacionados aos problemas de *C&P*.

3.1 HISTÓRIA DOS PROCESSOS DE CORTE E EMPACOTAMENTO

Os primeiros artigos matemáticos no campo do *C&P* foram de:

- L. V. Kantorovich, em 1939, escrito em russo, sendo publicado para o inglês somente em 1960 [11]. O economista Kantorovich foi quem primeiro formulou o problema de encaixe unidimensional.
- R. L. Brooks *et al.*, em 1940, realizaram uma análise do encaixe de retângulos em quadrados [13].

Inicialmente, o problema de encaixe de retângulos começou a ser estudado na década de 50. Respectivamente nos anos de 1955, 1956 e 1958; Paul, Walter e Vадja, apresentaram formulações desse tipo de problema [11]. Todavia, o rápido crescimento de artigos tratando com as variantes e extensões do *C&P* aconteceu por volta da década de 60. Nessa década, foram publicados os três artigos de Gilmore e Gomory [17][18][19] que tratam do Problema do Corte Industrial e que tornaram-se clássicos para o estudo deste tipo de problema. Estes artigos descrevem a primeira técnica empregada na prática para problemas de tamanho médio⁵, a clássica Técnica da

⁵ Um problema do tipo 1N/1/R, cuja lista de pedidos possui em torno de 50 itens e cuja quantidade para cada item diferente fica em torno de centenas ou milhares de peças.

Geração de Colunas (*Delayed Column Generation Technique*). A idéia da geração de colunas já tinha sido usada antes mesmo de Gilmore e Gomory, dentro de um contexto diferente, como parte importante do *Algoritmo de Decomposição Dantzig-Wolfe*. A Técnica da Geração de Colunas é um método de programação linear que foi aplicado aos problemas de corte unidimensional e bidimensional nos artigos de Gilmore e Gomory citados acima. Eles também usaram o processo de Programação Dinâmica para otimização de padrões de corte multidimensionais tipo guilhotina. Além disso, estes autores ressaltaram a importância que tem o *Problema de Knapsack* como um subproblema que ocorre em muitas das variantes do Problema do Corte Industrial [13]. Estes artigos fundamentaram muitos outros trabalhos que se seguiram.

A Tabela 3-1 mostra que, na década de 80, foi atingida uma média de um trabalho publicado por ano sobre o assunto *C&P*.

Tabela 3-1: Pesquisas sobre problemas de C&P.

AUTORES	ANO	ASSUNTO	ÁREA
Brown	1971	Empacotamento	Computação
Salkin / de Kluyver	1975	Knapsack	Logística
Golden	1976	Corte	Engenharia Industrial
Hinxman	1980	Redução de sobras	Pesquisa Operacional
Garey / Johnson	1981	Empilhamento de caixas	Otimização
Israni / Sanders	1982	Leiaute	Fabricação
Rayward-Smith / Shing	1983	Empilhamento de caixas	Matemática
Coffman <i>et al.</i>	1984	Empilhamento de caixas	Computação
Dowland	1985	Empacotamento	Pesquisa Operacional
Dyckhoff <i>et al.</i>	1985	Redução de sobras	Administração
Israni / Sanders	1985	Encaixe de peças	Produção
Berkey / Wang	1987	Empilhamento de caixas	Pesquisa Operacional
Dudzinski / Walukiewicz	1987	Knapsack	Pesquisa Operacional
Martello / Toth	1987	Knapsack	Matemática
Rode / Rosenberg	1987	Redução de sobras	Engenharia / Produção
Dyckhoff <i>et al.</i>	1988	Corte	Produção

Fonte: Ver artigo de Dyckhoff [14]

Haessler, em 1968, abordou Problemas de Corte Industrial usando uma formulação semelhante ao *Problema de Knapsack*. Herz, em 1972, usou combinações

inteiras de comprimentos e larguras discretizadas para normalizar padrões de empacotamento que conduziram a uma redução considerável no custo de processamento computacional. Dyson e Gregory, em 1974, trataram de problemas de encaixe para indústria de vidros usando programação linear e regras heurísticas. Em 1976, Tilanus e Gerhardt usaram programação dinâmica para resolver problemas da indústria de aço. E nesse mesmo ano, Adamowicz e Albano publicaram trabalhos para encaixe de figuras retangulares em contornos irregulares usando uma abordagem de programação dinâmica que permitiu selecionar tiras preenchidas por retângulos. Essa mesma idéia foi estendida para encaixe de formas irregulares em trabalhos subsequentes [11]. Christofides e Whitlock, em 1977, utilizaram um algoritmo de enumeração para problemas de corte bidimensional tipo guilhotina com limitação do número de vezes que uma peça entra no padrão de corte.

Stoyan e Gil, em 1976, publicaram um livro descrevendo métodos e algoritmos para o encaixe de figuras planas e; em 1984, Mukhachova publicou um livro que trata sobre corte eficiente em materiais industriais. Ambos os livros foram impressos na antiga URSS, somente em russo. Basicamente, nestes livros foram considerados problemas de encaixe de retângulos e também não-retângulos sobre tiras. Wang, em 1983, discutiu a geração de padrões de corte tipo guilhotina por combinação sucessiva da união de peças [40]. Em 1987, Terno *et al.* comentaram sobre vários problemas de corte e algoritmos para solução prática [13].

Um passo importante para apoiar a pesquisa internacional em *C&P* foi a criação do *Special Interest Group on Cutting and Packing* (SICUP) por Dyckhoff e Wäscher em 1988. O SICUP tem uma publicação própria publicada duas vezes ao ano. Um número considerável de documentos em *C&P* foram lançados pelo SICUP em conferências internacionais.

As pesquisas publicadas até agora tratavam com aspectos particulares de certos tipos especiais de problemas de *C&P*, principalmente com algoritmos e métodos de solução. Porém, não havia nenhuma pesquisa sobre o assunto de *C&P* que sistematicamente integrasse os vários tipos de problemas e idéias. Para alcançar este propósito, Dyckhoff [14], em 1990, desenvolveu uma classificação baseada na investigação de características da estrutura geométrica e lógica dos problemas de *C&P*. Tal tipologia, permitiu uma identificação das propriedades comuns de problemas que, a princípio, pareciam ser não relacionados. Reciprocamente, diferenças entre

problemas aparentemente semelhantes ficaram mais claras através da análise das suas características principais. Esta tipologia tornou-se uma base para unificar o uso de terminologias diferentes na literatura e ajudou a concentrar pesquisas futuras em tipos especiais de problemas. Dois anos depois, Dyckhoff e Finke publicaram um livro contendo cerca de 700 indicações de referências no campo de *C&P*. Trabalhos de Hinxman, 1980; Haessler *et al.*, 1991; Dowsland *et al.*, 1992; e Cheng *et al.*, concentraram-se em pesquisas sobre *C&P*. A rápida e crescente difusão de artigos relacionados a *C&P* incentivou à edições especiais de alguns jornais científicos, conforme citado em [13].

Aplicações práticas e estudos de caso de *C&P* em problemas unidimensionais com restrições adicionais e/ou generalizadas foram consideradas em problemas que exigiam restrições quanto à existência de defeitos no material de corte ou de outras condições de qualidade. Trabalhos de Sarker, 1988; Richter, 1992; e Scheithauer, 1995; concentraram-se nestes pontos. Foronda *et al.*, 1991; e Carnieri *et al.*, 1994; e Degraeve e Bemelmans realizaram estudos de caso [12].

E nos anos de 1992 e 1993, Ghandforoush *et al.* e MacLeod *et al.*, respectivamente, contemplaram novas heurísticas para o problema do corte tipo guilhotina de retângulos com restrição. Métodos para obtenção da solução ótima do problema do corte tipo guilhotina com restrição foram apresentados por Oliveira e Ferreira, em 1990; Viswanathan e Bagchi, em 1993; Christofides e Hadjiconstantinou, em 1995; e também, Daza *et al.*, em 1995 [13].

Oliveira e Ferreira apresentaram uma versão melhorada do *Algoritmo de Wang* para problemas de cortes bidimensionais com restrição no limite superior do número de peças por padrão de corte [27]. Baseado em trabalhos de Wang [40] e de Oliveira e Ferreira [27], Daza *et al.* [10] propuseram uma generalização dos métodos e melhoramentos adicionais nos algoritmos. Agarwal e Shing, em 1992, apresentaram um trabalho no qual só se permitiam cortes do tipo guilhotina com respeito ao retângulo de corte. Um método de programação dinâmica desenvolvido por Scheithauer e Terno, em 1988, considerou áreas inutilizáveis ou defeituosas no material para cortes três estágios do tipo guilhotina em retângulos. Beasley, em 1985, propôs uma formulação do Problema do Corte Industrial via Programação Inteira Binária usando a *Relaxação Lagrangeana* e um método de procura em árvore aplicado a cortes do tipo não-guilhotina [11]. Arenales e Morabito, em 1995, apresentaram um trabalho que tratou de

cortes do tipo não-guilhotina sem restrição na quantidade das peças geradas, mas considerando apenas cortes ortogonais a pelo menos um lado do retângulo de base [1]. Rinnooy, de Witt e Wijmenga, em 1987, discutiram o problema de encaixe de retângulos idênticos dentro de uma faixa e em posições não-ortogonais [13].

O Problema do Corte Industrial foi discutido sobre vários enfoques em diferentes trabalhos. A saber:

- R. W. Haessler, em 1992, fez considerações sobre procedimentos de solução para problemas, no qual não somente as sobras foram levadas em conta;
- Z. Sinuany-Stern e I. Weiner, em 1994, minimizaram as sobras em cortes de material de trabalho requerendo peças de vários comprimentos. Como objetivo secundário eles tentaram organizar os cortes de modo que a quantidade máxima de sobras ficassem acumuladas nas últimas barras.

Segundo [13], em artigos de Scheithauer *et al.* [32][33][35], de Wäscher *et al.* [41], de Nica [26]; foi investigada a diferença entre os valores ótimos inteiros e o correspondente problema de relaxação de programação linear (conhecido como *Modified Integer Round-Up Property - MIRUP*).

Lins [23], em 1988, apresentou um algoritmo para minimização de sobras usando programação dinâmica e regras heurísticas apropriadas para solucionar o problema de programação matemática.

Em trabalhos de Farley e Vasko *et al.*, ambos em 1992, foram apresentados estudos de caso⁶. Mukhachova e Zalgaller, em 1993, utilizaram programação linear e inteira para resolver problemas de planejamento de corte. Em 1994, Chauny e Loulou usaram uma heurística especial associada a um algoritmo de enumeração para arredondar soluções obtidas através de programação linear. Mas no mesmo ano, Vance *et al.* apresentaram um algoritmo para obter soluções inteiras ótimas [13].

Em 1995, Carvalho e Rodrigues publicaram um artigo que descreve um modelo de programação linear para o Problema do Corte Industrial de dois estágios, o qual é resolvido pela Técnica da Geração de Colunas [5]. Em 1996, um novo método propôs que problemas de natureza bidimensional fossem tratados como dois problemas

⁶ A referência [13] não dispunha de mais informações sobre os estudos de caso, além dos títulos dos mesmos:

Limiting the number of each piece in two-dimensional cutting stock patterns – Farley; e, Bethlehem Steel combines cutting stock and set covering to enhance customers services – Vasco et al..

unidimensionais. Tal método, denominado de Método das Faixas (*Stripe Approach*), foi apresentado por Riehme, Scheithauer e Terno [29].

Nesta área de *C&P* também tivemos trabalhos que contemplaram aplicação de programas computacionais para solução deste tipo de problema. Segundo Dyckhoff *et al.* [13], vale citar aqui os trabalhos de Dincbas *et al.* e Lirov, ambos em 1992. Em 1994, Carvalho e Rodrigues apresentaram um sistema computacional baseado num método interativo para solução de problemas de corte industrial de dois estágios. E em 1996, Scheithauer *et al.* apresentaram uma nova heurística para solução de problemas de empacotamento em vários paletes. Esta heurística se baseia num algoritmo de enumeração para empacotar caixas de vários tamanhos sobre um número mínimo de paletes e considera restrições de empacotamento de interesse prático.

Capítulo 4

FUNDAMENTOS TEÓRICOS

Considerada recente comparada a outros ramos da Matemática, a Programação Linear teve início em 1947 com idealização do Método Simplex pelo matemático norte-americano George B. Dantzig, hoje considerado o pai da Programação Linear. Nos anos cinquenta, o método foi utilizado na solução de problemas de planejamento da Força Aérea Americana. Atualmente, as ferramentas de Programação Linear vêm sendo aplicadas na solução de diversos tipos de problemas lineares práticos. E estes problemas são freqüentemente encontrados em ramos variados da indústria; e mais precisamente, no setor de Planejamento e Controle da Produção (PCP).

Essa nova ferramenta aplicada a processos industriais visando a otimização de operações e procedimentos dos meios de produção, permitiu uma valorização da Matemática Aplicada. Antes disso, a visão do que era melhoramento do sistema produtivo estava baseada em inovações tecnológicas realizadas nos equipamentos mecânicos envolvidos com o processo em si, e/ou nos equipamentos eletrônicos.

A programação linear, na parte de programação inteira e combinatória, tem fornecido soluções de problemas complexos de otimização para a indústria moderna. Da mesma forma que o Cálculo Diferencial, desenvolvido a partir do século 17, foi importante para o estudo da Mecânica, a Programação Linear tem a mesma importância para o estudo de problemas de otimização e gerenciamento racional dos recursos produtivos no século atual. O Problema do Corte Industrial está inserido neste contexto.

A Programação Linear consiste em resolver um problema de otimização (*maximização ou minimização*) de uma função objetivo da forma

$$z = \sum_{j=1}^n c_j \cdot x_j$$

sujeita a $\sum_{j=1}^m a_{ij} \cdot x_j = b_i$,

(4.1)

sujeita a restrições lineares específicas do tipo de problema a ser resolvido, onde:

x_j , representa as variáveis do problema, ou variáveis de projeto;

c_j , representa o custo da variável de projeto x_j .

Algumas definições são importantes na classificação dos tipos de problemas de Programação Linear. Quando todas as variáveis são restritas a somente valores inteiros, esses tipos de problema são denominados de *problemas de programação inteira*; como no caso do Problema do Corte Industrial. Existem também aqueles em que somente algumas das variáveis assumem valores inteiros e as demais assumem valores reais contínuos, e neste caso são denominados de *problemas de programação inteira mista*. Quando as variáveis assumem valores discretos, sejam inteiros ou reais, eles são classificados como *problemas de programação discreta*. E no caso particular, em que todas as variáveis assumem valores 0 ou 1, são classificados como *problemas de programação binária*.

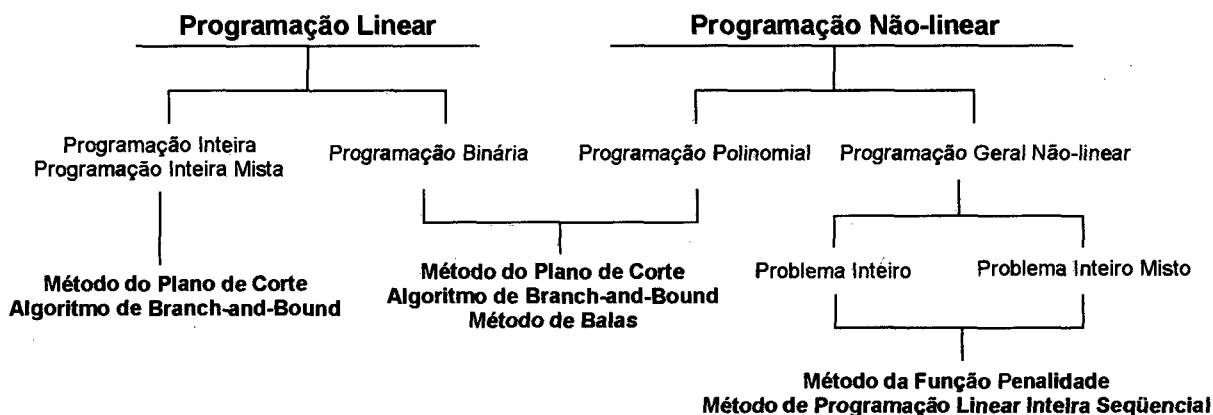


Figura 4-1: Métodos de programação inteira

O método geral para a solução de problemas lineares é obtido pelo Método Simplex. No entanto, este método é restrito a situações em que o número de restrições é finito. Isto é, o problema é dito bem posto. Variações dos métodos de solução de acordo com o tipo de problema são extensivamente tratados nos textos de Programação Linear. Dentre os muitos métodos tratados na literatura [6][28][31], estão os que são considerados mais importantes para solução de problemas de programação inteira, conforme mostra a Figura 4-1.

O *Método do Plano de Corte* e o *Algoritmo de Branch-and-Bound* são os métodos mais usados na solução de problemas de programação inteira e inteira mista.

Os problemas de programação binária também podem ser solucionados pelos métodos anteriormente citados, mas Balas desenvolveu um método enumerativo mais eficiente, o qual leva o seu nome e é conhecido amplamente como Algoritmo de Balas [28][31].

Conforme mostrado nas seções a seguir, nesta pesquisa são ressaltados apenas os métodos que foram usados na solução do Problema do Corte Industrial. O Algoritmo de Branch-and-Bound será utilizado neste trabalho para resolver o Problema de Knapsack que aparece como um subproblema do Problema do Corte Industrial. O Algoritmo de Branch-and-Bound é um algoritmo de enumeração das possíveis soluções, o qual acelera o processo de busca das soluções viáveis. Os métodos de enumeração serão discutidos na seção 4.3.

4.1 FORMULAÇÃO GERAL

No Problema do Corte Industrial é assumido que há um estoque de matérias-primas de comprimentos L_j , com $j = 1, 2, \dots, k$ de um mesmo material, e para os quais se quer cortar diversos comprimentos menores. Assume-se também que a quantidade de peças em estoque para cada matéria-prima de comprimento L_j é ilimitada. Define-se como pedido de corte, o atendimento de uma dada quantidade de peças N_i , cujos comprimentos a serem cortados são l_i , com $i = 1, 2, \dots, m$. Se, para algum valor de j e de todos os valores de i , tem-se satisfeita a inequação $L_j \geq l_i$, então o pedido pode ser preenchido e atendido. Um valor de custo é atribuído a cada um dos k 's comprimentos de matéria-prima armazenada. Sendo assim, o custo total para atender um pedido completo é dado pela soma individual dos custos de cada comprimento cortado.

Define-se como atividade de corte ou padrão de corte, a uma determinada maneira de se cortar uma peça qualquer do estoque de matérias-primas. Por exemplo, o corte de uma peça de 17 u.c. em três peças menores, uma de 5 u.c. e duas de 4 u.c., é uma atividade de corte. Atribuindo-se uma variável x_j , para cada das possíveis atividades que cortam os comprimentos l_i do pedido a partir das matérias-primas armazenadas de comprimentos L_1, L_2, \dots, L_k , este Problema de Corte Industrial pode ser colocado como um problema de programação linear inteira. O valor representado

pela variável x_j indica o número de vezes que a atividade é executada. As variáveis x_1, x_2, \dots, x_n atribuídas às atividades devem satisfazer m desigualdades do tipo abaixo,

$$\begin{aligned} a_{i1} \cdot x_1 + a_{i2} \cdot x_2 + \dots + a_{ij} \cdot x_j + \dots + a_{in} \cdot x_n &\geq N_i \\ \text{com } i &= 1, 2, \dots, m \text{ e } j = 1, 2, \dots, n \end{aligned} \quad (4.2)$$

para um pedido de N_i peças de comprimento l_i . O termo a_{ij} é o número de peças de comprimento l_i gerada pela j -ésima atividade. Estas equações são chamadas de restrições de corte.

A função custo para este problema é dada por:

$$z(\mathbf{x}) = \sum_{j=1}^n c_j \cdot x_j = c_1 x_1 + c_2 x_2 + \dots + c_n x_n, \quad (4.3)$$

onde:

c_j , representa o custo do comprimento usado na j -ésima atividade.

A transformação das restrições de corte (4.2) para uma equação de igualdade é dada pela introdução de variáveis de folga ou de relaxamento x_{n+i} , como mostra a teoria geral para problemas de otimização.

Assim, o Problema do Corte Industrial pode ser genericamente colocado como minimizar a Equação (4.3), achando os valores inteiros para x_1, x_2, \dots, x_{n+m} , de forma que satisfaça m equações dadas por:

$$a_{i1} \cdot x_1 + a_{i2} \cdot x_2 + \dots + a_{in} \cdot x_n - x_{n+i} = N_i, \quad i = 1, 2, \dots, m; \quad (4.4)$$

$$\text{sujeito a } x_j \in Z_+, \quad j = 1, 2, \dots, n+m. \quad (4.5)$$

Existem dois fatores que tornam esta formulação do Problema de Corte Industrial impraticável do ponto de vista computacional. O primeiro é o tamanho de n , o qual pode ser enorme, quando a quantidade k de comprimentos armazenados e a quantidade m de comprimentos a serem cortados, são razoavelmente grandes. O segundo inconveniente advém da própria natureza do problema em restringir os valores das variáveis para inteiros.

Considerando o segundo fator, pode-se comentar o seguinte: se a restrição (4.5) fosse retirada, o problema torna-se em obter uma solução real. Dada uma solução real,

existem vários modos de determinar a solução inteira aproximada. Por exemplo, ao arredondar o valor da variável para o inteiro mais próximo superior, implica em aumentar o custo; e ao truncar para o valor inferior, significa aumentar o volume de rejeitos, cuja solução pode ser melhorada com uso de métodos heurísticos a posteriori.

Quando usado para obter uma solução da Equação (4.4), sujeita a (4.5) e que minimize (4.3); o Método Simplex fornece a partir de qualquer solução básica inicial, uma outra solução, para a qual o valor de (4.3) é menor que para a solução dada [17].

Se uma solução básica de (4.4) e (4.5) é dada, o algoritmo Simplex testa cada uma das outras variáveis até que seja encontrada uma outra que substitua uma das variáveis básicas correntes. Assume-se que as variáveis numa dada solução básica sejam x_1, x_2, \dots, x_m . Seja P_j o vetor-coluna $(a_{1j}, a_{2j}, \dots, a_{mj})$ e c_j o custo em (4.3) associado com a variável x_j . Assim, se x_j é uma variável de relaxamento, o valor correspondente de c_j é 0 e ao vetor P_j é atribuído uma coordenada de valor -1 .

Define-se $P = (a_1, a_2, \dots, a_m)$ uma nova indeterminada atividade que corta um comprimento L de custo c . Seja A uma matriz com colunas P_1, P_2, \dots, P_m . Visto que P_1, P_2, \dots, P_m forma uma base para a matriz A então, segundo a álgebra linear, existe um vetor-coluna u que satisfaz a seguinte equação

$$A \cdot u = P. \quad (4.6)$$

E a nova atividade P pode ser usada para um melhoramento sobre a solução dada, se e somente se for satisfeita a seguinte equação:

$$C \cdot u^T > c, \quad (4.7)$$

onde:

C , representa um vetor-linha cujas componentes são c_1, c_2, \dots, c_m .

Desta forma, se o vetor-linha dado por $C \cdot A^{-1}$ tem componentes b_1, b_2, \dots, b_m então, conclui-se a partir das equações (4.6) e (4.7), que aí existe uma atividade cortando L e que pode ser rentável se somente se, existe um conjunto de inteiros não-negativos a_1, a_2, \dots, a_m satisfazendo as seguintes equações

$$\sum_{i=1}^m I_i \cdot a_i \leq L \quad (4.8)$$

$$\text{e } \sum_{i=1}^m b_i \cdot a_i > c. \quad (4.9)$$

Um método para determinar se existem inteiros positivos a_i satisfazendo (4.8) e (4.9), seria determinar um conjunto de inteiros não-negativos satisfazendo (4.8) para os quais a Equação (4.9) é um máximo. Se tais inteiros não satisfazem (4.9), então nenhum outro conjunto de valores o fará.

Sendo assim, o problema de escolher uma nova variável no Método Simplex para o Problema do Corte Industrial pode ser expresso como o problema de achar a solução para até k problemas auxiliares (*um para cada comprimento* L_1, L_2, \dots, L_k). Sendo que cada um desses problemas, é um problema de programação linear inteira.

O problema de maximização da Equação (4.9) sujeita a (4.8) é uma generalização do Problema de Knapsack. Desta forma, demonstra-se que uma maneira de resolver o Problema do Corte Industrial significa resolver um conjunto de problemas de Knapsack.

4.2 PROBLEMA DE KNAPSACK

O Problema de Knapsack, conhecido como Problema da Mochila, foi batizado e formulado por Dantzig em 1957. Este problema foi assim denominado, em alusão ao fato de querer resolver o seguinte problema: uma mochila de tamanho b deve ser carregada com n itens de pesos e tamanhos, c_i e a_i , respectivamente. O objetivo é carregar a mochila com o máximo de peso que o seu tamanho comporte. Logo, isso sugere que um item i cujo peso por unidade tem um valor maior, deve ser usado mais vezes. E o número de vezes que um item é colocado no padrão é representado pela variável x_i .

Matematicamente, ele pode ser definido de maneira geral de acordo com a formulação abaixo:

$$\text{maximizar } z = \sum_{i=1}^n c_i \cdot x_i, \quad (4.10)$$

onde:

z , é uma função linear representada pelo produto dos coeficientes c_i , conhecidos como pesos, sobre a correspondente variável x_i .

A Equação (4.10) é denominada de função custo e está sujeita a uma restrição dada pela seguinte desigualdade:

$$\sum_{i=1}^n a_i \cdot x_i \leq b. \quad (4.11)$$

É importante delimitar o intervalo de valores da variável x_i .

$$\underline{u}_i \leq x_i \leq \bar{u}_i. \quad (4.12)$$

onde:

\underline{u}_i , representa o extremo inferior da variável x_i ;

\bar{u}_i , representa o extremo superior da variável x_i .

Ao assumir x_i como sendo uma solução no domínio dos inteiros, tem-se:

$$x_i \in \mathbb{Z}_+, i = 1, 2, \dots, n; \quad (4.13)$$

Sem perda de generalidade, pode-se assumir

$$\begin{aligned} b > 0 \text{ e } c_i > 0, i = 1, 2, \dots, n; \\ 0 < a_i \leq b; \\ 0 \leq \underline{u}_i < \bar{u}_i < +\infty. \end{aligned} \quad (4.14)$$

Num caso particular da formulação geral, fazendo a substituição das restrições (4.12) e (4.13) por $x_i \in \{0, 1\}$, $i = 1, 2, \dots, n$; obtém-se a formulação para o Problema de Knapsack Binário.

No caso em que a restrição de integralidade (4.13) é eliminada e o intervalo de valores da variável x_i é redefinido, obtém-se a seguinte formulação para o Problema de Knapsack:

$$\text{maximizar } z = \sum_{i=1}^n c_i \cdot x_i, \quad (4.15)$$

$$\text{sujeito a } \sum_{i=1}^n a_i \cdot x_i \leq b, \quad (4.16)$$

$$\underline{u}_i = 0 \text{ e } \bar{u}_i = +\infty, \quad (4.17)$$

$$x_i \geq 0, i = 1, 2, \dots, n. \quad (4.18)$$

De acordo com as equações (4.17) e (4.18), percebe-se que a diferença fundamental dessa nova formulação está nos valores assumidos pela variável x_i , os quais agora estão no domínio dos números reais não-negativos.

Uma solução para o Problema de Knapsack pode ser obtida da seguinte forma: a partir de (4.16), adiciona-se uma variável de folga x_{n+1} à formulação do problema de forma a obter a igualdade abaixo.

$$\sum_{i=1}^n a_i \cdot x_i + x_{n+1} = b, \text{ com } x_{n+1} \geq 0 \quad (4.19)$$

Considerando a razão de eficiência da variável (c_i / a_i) para uma variável de índice $i = k$, tal que $\frac{c_k}{a_k} = \text{máximo} \left\{ \frac{c_i}{a_i}, i = 1, 2, \dots, n \right\}$, pode-se obter o valor de x_k a partir de (4.19), o qual sendo substituído em (4.15), obtém-se o seguinte valor para a função objetivo:

$$z = b \cdot \frac{c_k}{a_k} + \sum_{\substack{i=1 \\ i \neq k}}^n \left\{ \left[\frac{(c_i a_k - c_k a_i)}{a_k} \right] \cdot x_i \right\} - \frac{c_k}{a_k} \cdot x_{n+1}. \quad (4.20)$$

Haja vista que para todo e qualquer i , tem-se $\frac{c_i}{a_i} \leq \frac{c_k}{a_k} \Leftrightarrow c_i a_k \leq c_k a_i$, e sendo $a_k > 0$, portanto $\frac{(c_i a_k - c_k a_i)}{a_k} \leq 0$. Assim, conclui-se que $z \leq b \cdot \frac{c_k}{a_k}$.

Para maximizar a função objetivo, tem-se que $z_{\text{máx}} = b \cdot \frac{c_k}{a_k}$, e isso ocorre quando $x_k = \frac{b}{a_k}$ e $x_i = 0 \forall i = 1, 2, \dots, k-1, k+1, \dots, n, n+1$. Portanto, $z^* = b \cdot \frac{c_k}{a_k}$, é o valor ótimo da função objetivo para o Problema de Knapsack.

A implementação de um algoritmo para solução computacional do Problema de Knapsack formulado acima, utiliza um algoritmo de enumeração, o qual será comentado e descrito no texto a seguir.

4.3 ALGORITMO DE ENUMERAÇÃO

De acordo com a literatura corrente [4][31], a enumeração completa de todas as soluções viáveis⁷ de um problema de otimização, conduzirá à solução ótima num número finito de passos. Infelizmente, mostra-se que, mesmo para problemas triviais, a enumeração completa das soluções causa um tempo de processamento computacional impraticável.

Considere um problema cujo número de soluções viáveis seja $(n!)$. Admita que cada solução possa ser avaliada num intervalo de tempo de 1 nanosegundo⁸. De acordo com a Tabela 4-1, se n for acrescentado em apenas uma unidade, o tempo de processamento computacional cresce rapidamente. Este comportamento numérico é conhecido como explosão exponencial (*Exponential Blow-up*) do tempo de processamento computacional total.

Tabela 4-1: Crescimento exponencial do tempo de processamento

n	TEMPO DE PROCESSAMENTO, EM ANOS
20	≈ 80
21	≈ 1620
22	≈ 35640

Fonte: Cap. 1 da referência [4]

A conclusão é que muito embora este procedimento seja direto e natural, a enumeração de todas as soluções não é recomendável do ponto de vista prático.

Estes algoritmos têm como função principal de enumerar, avaliar e comparar todas as soluções viáveis de um problema. Baseando-se em critérios de integralidade, de não-negatividade e das restrições próprias de cada problema, pode-se avaliar de forma implícita que muitas das soluções viáveis não conduzem a uma melhora da solução, podendo ser descartadas do processo de enumeração. Assim, a eficiência e

⁷ É toda e qualquer solução possível para o problema, não necessariamente a solução ótima. No Problema de Knapsack, uma

solução viável deve satisfazer a inequação $b - \sum_{i=1}^n a_i \cdot x_i < a_n$. E neste caso, a solução é conhecida como solução sensível.

⁸ 1 ns equivale a 10^{-9} s.

diferenças entre os algoritmos de enumeração em geral, resumem-se aos critérios usados na enumeração implícita das soluções viáveis.

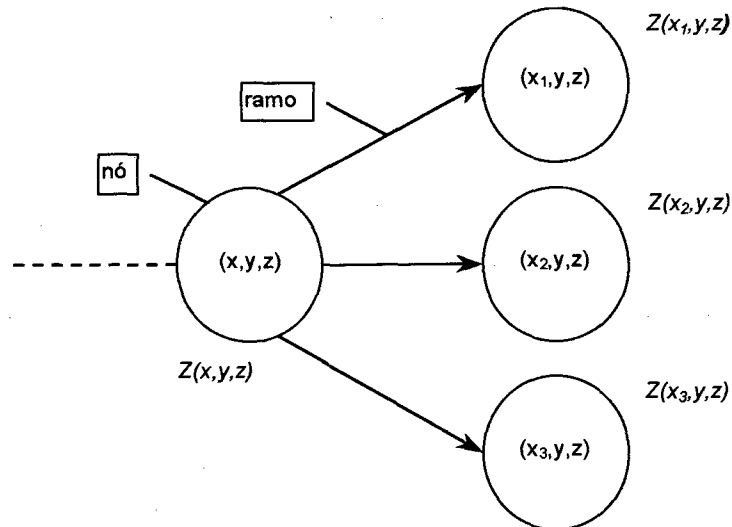


Figura 4-2: Representação em árvore

Uma maneira bastante usual de representar os algoritmos de enumeração é através da representação gráfica em forma de árvore, pois isso possibilita uma visualização gráfica de fácil interpretação.

A Figura 4-2 mostra parte de uma representação em árvore, enfatizando os elementos básicos nó e ramo. Um nó⁹ representa um ponto, tal qual (x, y, z) , de onde são gerados vários ramos. A geração de um novo ramo é dado com alteração do valor de uma variável, mantendo-se constantes as demais, conforme se pode verificar com o valor de x que assume os valores x_1 , x_2 e x_3 , respectivamente, em cada um dos novos ramos gerados.

Como é correntemente aceito na literatura, uma maneira de resolver o Problema de Knapsack é usar o algoritmo de enumeração denominado Algoritmo de Branch-and-Bound. Segundo Rao [28], este algoritmo é muito eficiente na solução de problemas de programação linear e não linear inteira mista (*mixed-integer linear and nonlinear programming problems*). A seguir, é descrita a maneira de como usar este algoritmo na resolução dos problemas de Knapsack resultantes da formulação adotada para o Problema de Corte Industrial.

⁹ Também chamado de nodo em algumas literaturas. No caso do primeiro nó, ele é a raiz.

Assume-se uma representação em árvore, na qual o primeiro nó é a raiz e os demais são as folhas. Então, partindo-se da raiz, é iniciado o processo de enumeração das variáveis x_i do Problema de Knapsack. Inicialmente, são usados os maiores valores possíveis que uma variável x_i qualquer pode assumir. A explicação para isso é bastante simples e está fundamentada na ordenação decrescente da razão (c_i / a_i) , denominada *eficiência* da variável x_i .

$$\frac{c_1}{a_1} \geq \frac{c_2}{a_2} \geq \dots \geq \frac{c_i}{a_i} \geq \dots \geq \frac{c_n}{a_n} \quad (4.21)$$

Com base na eficiência da variável x_i , a representação em árvore dada pela enumeração de uma solução (x_1, x_2, \dots, x_n) , segue o seguinte processo: vai-se usando o maior valor de x_1 , e uma vez fixado, usa-se o valor de x_2 , e assim por diante com as demais variáveis, até x_n .

Formalmente, as variáveis x_i são enumeradas pela seguinte recursão¹⁰:

$$x_j = \left\lfloor \left(b - \sum_{i=1}^{j-1} a_i \cdot x_i \right) / a_j \right\rfloor, \quad j = 2, 3, \dots, n. \quad (4.22)$$

Para $j = 1$, em particular, tem-se que: $x_1 = \lfloor b / a_1 \rfloor$.

Sabendo-se como determinar os valores das variáveis, deve-se agora descrever o processo de enumeração das soluções viáveis. A diretiva básica que rege o funcionamento do Algoritmo de Branch-and-Bound é retornar à raiz após todos os ramos serem explorados, ou seja; todos os possíveis valores são testados. A cada novo nó, deve-se seguir a diretiva de explorar todos os possíveis ramos. Nos casos de empate nos valores da função objetivo, prevalece o valor correspondente ao ramo mais alto na estrutura da árvore. Embora não fique evidente e claro os procedimentos acima, numa forma algébrica eles se resumem a: considere uma solução viável (x_1, x_2, \dots, x_n) que acabou de ser avaliada; então um valor k assume o valor $n-1$ e entra num ciclo que reduz k em uma unidade até que um valor $x_k > 0$ seja encontrado; e neste caso, o valor correspondente x_k é substituído por $x_k - 1$; e em seguida, recorre-se à Equação (4.22) para obter os valores de $x_{k+1}, x_{k+2}, \dots, x_n$.

¹⁰ O símbolo $\lfloor x \rfloor$ significa o maior inteiro menor que x .

Para acelerar o processo de enumeração, adota-se um critério de enumeração implícita das soluções viáveis, o qual visa evitar a avaliação de ramos que são obviamente piores que os já avaliados, ou seja, não conduzem a uma solução melhor que a atual. Esse critério é mostrado a seguir.

Considere que durante a exploração dos ramos da representação em árvore até uma determinada iteração, o maior valor obtido para a função objetivo $z(x)$ foi tal que

$$z^* = \sum_{i=1}^n c_i \cdot x_i^* = M.$$

Retornando-se à raiz ao encontrar o maior k tal que, $k \leq n-1$ e $x_k > 0$; deve-se proceder como descrito antes, fazendo

$$\begin{aligned} \hat{x}_i &= x_i, \text{ para } i = 1, 2, \dots, k-1; \\ \hat{x}_k &= x_k - 1, \text{ para } i = k. \end{aligned}$$

Em seguida, recorre-se à Equação (4.22) para obter os diversos valores de $\hat{x}_{k+1}, \hat{x}_{k+2}, \dots, \hat{x}_n$ que conduzem a uma solução viável. Mas antes de investigar esses valores, deve-se verificar se eles são melhores que a solução atual $(x_1^*, x_2^*, \dots, x_n^*)$.

Nota-se que, a partir de (4.21), cada uma das variáveis $x_{k+1}, x_{k+2}, \dots, x_n$ tem uma eficiência de, no máximo, (c_{k+1} / a_{k+1}) . Assim:

$$\sum_{i=k+1}^n c_i \cdot \hat{x}_i \leq \frac{c_{k+1}}{a_{k+1}} \cdot \sum_{i=k+1}^n a_i \cdot \hat{x}_i.$$

Implica que, a partir de (4.16), tem-se

$$\sum_{i=1}^n c_i \cdot \hat{x}_i \leq \sum_{i=1}^k c_i \cdot \hat{x}_i + \frac{c_{k+1}}{a_{k+1}} \cdot \left(b - \sum_{i=1}^k a_i \cdot \hat{x}_i \right). \quad (4.23)$$

Numa estrutura em árvore, a inequação (4.24) faz esse teste. Esse teste de enumeração implícita foi adotado no algoritmo proposto neste trabalho.

$$\sum_{i=1}^k c_i \cdot \hat{x}_i + \frac{c_{k+1}}{a_{k+1}} \cdot \left(b - \sum_{i=1}^k a_i \cdot \hat{x}_i \right) \leq M. \quad (4.24)$$

Se a inequação (4.24) for verdadeira, então nenhuma das soluções $(\hat{x}_1, \hat{x}_2, \dots, \hat{x}_n)$ têm chance de ser melhor que $(x_1^*, x_2^*, \dots, x_n^*)$; logo, elas podem ser

ignoradas no processo de enumeração. Em caso contrário, a solução ótima atual $(x_1^*, x_2^*, \dots, x_n^*)$ deve ser substituída por $(\hat{x}_1, \hat{x}_2, \dots, \hat{x}_n)$.

No caso particular de todos os coeficientes c_i serem inteiros positivos, então quem faz o teste é a inequação fraca (4.25).

$$\sum_{i=1}^k c_i \cdot \hat{x}_i + \frac{c_{k+1}}{a_{k+1}} \cdot \left(b - \sum_{i=1}^k a_i \cdot \hat{x}_i \right) < M + 1. \quad (4.25)$$

Em muitos dos algoritmos de enumeração, tipo o de Branch-and-Bound, é freqüente acontecer que uma solução ótima do problema original seja obtida nos estágios iniciais. Uma boa heurística é terminar a procura num estágio intermediário, obtendo uma solução aproximada da ótima.

4.4 TÉCNICA DA GERAÇÃO DE COLUNAS

Em 1960, Gilmore e Gomory [17] propuseram a Técnica da Geração de Colunas. A principal função dessa técnica é gerar novas colunas, as quais correspondem aos novos padrões de corte. Além disso, os novos padrões de corte gerados são melhores que os já existentes e os mesmos são avaliados implicitamente. Isso evita uma enumeração completa das soluções viáveis.

Adotando-se esta técnica, a formulação do Problema do Corte Industrial fica dividida em: problema principal, o qual otimiza a escolha a ser feita; e no subproblema, que gera os novos padrões de corte. Estes dois problemas devem estar inter-relacionados de forma que a resposta (*novo padrão de corte*) do subproblema seja a entrada do problema principal, e o problema principal deve fornecer informações ao subproblema de como gerar um novo e melhorado padrão de corte. Em suma, deve haver uma constante interação entre o problema principal e o subproblema.

De acordo com Gilmore e Gomory [17], os passos da rotina computacional para a solução das equações (4.4) e (4.5), para a qual a Equação (4.3) é um mínimo, são os seguintes:

1. Determinar m atividades e os seus respectivos custos;

Para cada i , escolhe-se um comprimento $L_j > l_i$ e define-se a i -ésima atividade que corta $a_{ii} = \lfloor L_j / l_i \rfloor$ peças de comprimento l_i . O custo correspondente da atividade é dado pelo custo c_j do comprimento L_j usado para cortar as peças de comprimento l_i .

2. Montar e armazenar a matriz \mathbf{B} com os a_{ii} 's e os c_j definidos no passo 1;

A matriz \mathbf{B} assume a seguinte forma:

$$\mathbf{B} = \begin{array}{c} \left\| \begin{array}{cccccc} 1 & -c_1 & -c_2 & \cdots & -c_i & \cdots & -c_m \\ \hline 0 & a_{11} & 0 & \cdots & 0 & \cdots & 0 \\ 0 & 0 & a_{22} & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \cdots & \vdots \\ 0 & 0 & 0 & \cdots & a_{ii} & \cdots & 0 \\ \vdots & \vdots & \vdots & \cdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & \cdots & a_{mm} \end{array} \right\| \end{array}$$

3. Montar e armazenar a matriz \mathbf{S} e o vetor-coluna \mathbf{N}' ;

A matriz \mathbf{S} , de dimensão $(m+1) \times m$, corresponde às variáveis de folga e é formada por vetores-colunas $\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_m$. Sendo que cada vetor-coluna tem zeros em todas as posições, exceto para o elemento da $(i+1)$ -ésima linha, o qual possui valor -1 . O vetor-coluna \mathbf{N}' é formado pelas respectivas quantidades das peças pedidas, com exceção do primeiro elemento que é zero. Assim, a matriz \mathbf{S} e o vetor-coluna \mathbf{N}' assumem a forma:

$$\mathbf{S} = \begin{array}{c} \left\| \begin{array}{cccc} 0 & 0 & \cdots & 0 \\ \hline -1 & 0 & \cdots & 0 \\ 0 & -1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & -1 \end{array} \right\| \end{array} \text{ e } \mathbf{N}' = \begin{array}{c} \left| \begin{array}{c} 0 \\ N_1 \\ N_2 \\ \vdots \\ N_i \\ \vdots \\ N_m \end{array} \right| \end{array}$$

4. Se a i -ésima variável de folga não está aparecendo na solução atual, ela só deverá trazer um melhoramento para a solução se e somente se os $(i+1)$ elementos da primeira linha da matriz inversa de \mathbf{B} são negativos. A matriz inversa de \mathbf{B} assume a seguinte forma:

$$B^{-1} = \left(\begin{array}{cccccc} 1 & \frac{c_1}{a_{11}} & \frac{c_2}{a_{22}} & \dots & \frac{c_j}{a_{ji}} & \dots & \frac{c_m}{a_{mm}} \\ \hline 0 & \frac{1}{a_{11}} & 0 & \dots & 0 & \dots & 0 \\ 0 & 0 & \frac{1}{a_{22}} & \dots & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \dots & \vdots \\ 0 & 0 & 0 & \dots & \frac{1}{a_{ji}} & \dots & 0 \\ \vdots & \vdots & \vdots & \dots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 0 & \dots & \frac{1}{a_{mm}} \end{array} \right)$$

5. Se nenhuma variável de folga conduz a uma melhora da solução atual, deve-se então verificar se existe uma nova atividade que melhore a solução atual;

Para tanto deve-se determinar se existe algum comprimento L de custo c , para o qual as equações (4.8) e (4.9) têm solução; sendo que os termos b_i são os últimos m elementos da primeira linha da matriz B^{-1} atual. Se as citadas equações não possuem solução para nenhum valor de L_1, L_2, \dots, L_k com respectivos custos c_1, c_2, \dots, c_k , então a solução atual é o mínimo da Equação (4.3). Neste caso, a solução atual é dada pelo vetor-coluna N , obtido do produto da matriz B^{-1} pelo vetor-coluna N' . O primeiro elemento de N é o custo da solução atual e as demais m linhas são os valores das variáveis correspondentes às colunas da matriz B^{-1} . Se entretanto, uma nova atividade melhorar a solução atual, então ela formará o vetor-coluna P de elementos $-c, a_1, a_2, \dots, a_m$.

6. No caso de uma variável, seja ela de folga ou referente a uma nova atividade P , melhorar a solução atual, deve-se então proceder da seguinte forma:

Considerando o vetor-coluna P como sendo o vetor correspondente da variável. Para determinar os novos valores de B^{-1} e N , faz-se o seguinte: calcula-se o produto de B^{-1} por P e chama os elementos deste resultado de $y_1, y_2, \dots, y_m, y_{m+1}$. Os elementos do vetor-coluna N atual são chamados de $x_1, x_2, \dots, x_m, x_{m+1}$. Em seguida, determina-se a menor razão x_i/y_i , tal que $i \geq 2, x_i \geq 0$ e $y_i > 0$. O valor do índice i correspondente à razão mínima é chamado de k .

Se o valor da razão mínima x_i/y_i não é zero, então o k -ésimo elemento de $B^{-1} \cdot P$ deverá ser o pivô da eliminação de Gauss processada simultaneamente em $B^{-1}, B^{-1} \cdot P$ e

N . O novo valor de N é resultado dessa eliminação de Gauss. Já a matriz B^{-1} , após a eliminação de Gauss, deve trocar a coluna de índice correspondente ao k -ésimo elemento, pelo valor do vetor-coluna P correspondente à nova atividade. Obtém-se então, o novo valor da matriz B^{-1} .

Agora, se o valor da razão mínima x_i/y_i é zero, então deve-se adotar o seguinte procedimento: Uma nova coluna N^1 de elementos positivos $x_1^1, x_2^1, \dots, x_m^1, x_{m+1}^1$, e independentes de N , é adicionada à matriz G , a qual é formada pela junção de B^{-1} , $B^{-1} \cdot P$ e N . Daí, repete-se o mesmo procedimento feito anteriormente: Escolhe-se o termo $y_i > 0$ para o qual $x_i = 0$, de modo tal que $x_i^1 > 0$ e x_i^1/y_i é mínimo. Proceda-se à eliminação de Gauss como antes, só que agora, simultaneamente sobre a matriz G acrescida da nova coluna N^1 . E assim, a coluna é mantida em G até que exista um i tal que a razão x_i/y_i seja positiva e finita. Senão, se for o caso de nenhum x_i/y_i ou x_i^1/y_i ser positivo e finito, então adiciona-se uma nova coluna N^2 à matriz G e repete todo o processo. E como as novas colunas N^i acrescidas à matriz G são independentes antes e depois da eliminação de Gauss, então esse processo vai se repetir não mais que m vezes. Assim, a cada nova coluna adicionada à matriz G tem-se um novo problema de programação linear a ser resolvido e que também deve ser checado sobre a possibilidade de possuir uma razão x_i/y_i infinita.

A dificuldade associada à aplicação desta Técnica da Geração de Colunas para a solução do Problemas do Corte Industrial deve-se ao grande número de padrões de corte ou atividades gerados. Sem considerar ainda a dificuldade que é implementar o código computacional desta rotina. Por isso, neste trabalho foram adotadas as modificações implementadas por Gilmore e Gomory [18], as quais tornaram muito mais rápido e eficiente o método de resolução dos problemas de Knapsack associados ao Problema do Corte Industrial.

No capítulo seguinte, faz-se uma descrição e explanação da formulação e do algoritmo empregados para a solução do problema de minimização das sobras de matéria-prima em cortes unidimensionais.

Capítulo 5

MÉTODOS E TÉCNICAS

5.1 FORMULAÇÃO UNIDIMENSIONAL

O Problema do Corte Industrial Unidimensional Geral é definido da seguinte forma:

“Matérias-primas unidimensionais de um dado comprimento L devem ser cortadas em pedaços menores de comprimentos l_i , com i variando no intervalo $i = 1, 2, \dots, m$, e atendendo à demanda N_i correspondente. O objetivo é minimizar a quantidade total de matéria-prima utilizada para atender à demanda solicitada.”

Para solução do Problema do Corte Industrial, pode-se adotar uma formulação baseada em programação linear inteira dada por:

$$\text{minimizar } \sum_{j=1}^n x_j, \quad (5.1)$$

$$\text{sujeito a } \sum_{j=1}^n a_{ij} \cdot x_j \geq N_i, \quad (5.2)$$

$$\text{e } \sum_{i=1}^m a_{ij} \cdot l_i \leq L, \quad (5.3)$$

$$a_{ij} \text{ e } x_j \in \mathbb{Z}_+, \text{ com } i = 1, 2, \dots, m \text{ e } j = 1, 2, \dots, n. \quad (5.4)$$

onde:

n , representa o número de padrões de corte gerados;

m , representa o número de comprimentos distintos na lista de pedidos de peças;

a_{ij} , representa o número de vezes que a peça de comprimento l_i entra no padrão de corte j ;

x_j , representa o número de vezes que o padrão de corte j é usado ou, equivalentemente, a quantidade de matéria-prima usada pelo padrão j ;

N_i , representa a quantidade solicitada da peça de comprimento l_i .

Introduzindo variáveis de folga x_{n+i} , $i = 1, 2, \dots, m$ na restrição (5.2), obtém-se m equações da forma:

$$a_{i1} x_1 + a_{i2} x_2 + \dots + a_{ij} x_j \dots + a_{in} x_n - x_{n+i} = N_i, \text{ para } i = 1, 2, \dots, m; \quad (5.5)$$

$$x_j \in \mathbb{Z}_+, \text{ para } j = 1, 2, \dots, n \text{ e } x_j \in \mathbb{R}^{11}, \text{ para } j = n+i. \quad (5.6)$$

Eliminando a restrição de integralidade, a Equação (5.5) assume a forma abaixo:

$$a_{i1} x_1 + a_{i2} x_2 + \dots + a_{ij} x_j \dots + a_{in} x_n = N_i, \text{ para } i = 1, 2, \dots, m; \quad (5.7)$$

$$x_j \in \mathbb{R}, \text{ para } j = 1, 2, \dots, n. \quad (5.8)$$

O sistema de equações lineares gerado a partir da Equação (5.7) possui m linhas e n colunas. Cada coluna da matriz¹² de coeficientes \mathbf{A} do sistema constitui um padrão de corte possível utilizando uma matéria-prima de comprimento nominal L , e é dada por $\mathbf{a} = [a_{1j}, a_{2j}, \dots, a_{mj}, \dots, a_{mj}]^T$. Assim, a interpretação para uma coluna j qualquer da matriz \mathbf{A} é a seguinte:

- a_{1j} peças de comprimento l_1 são cortadas no padrão j ;
- a_{2j} peças de comprimento l_2 são cortadas no padrão j ;
- \vdots \vdots \vdots
- a_{mj} peças de comprimento l_m são cortadas no padrão j ;

¹¹ O símbolo \mathbb{R} representa o conjunto dos números reais.

¹² Convenção adotada: as matrizes estão grafadas em maiúsculo e negrito e os vetores estão grafados em minúsculo e negrito.

Resolvendo-se o sistema de equações lineares pela Técnica da Geração de Colunas, obtém-se a matriz \mathbf{A} e os valores de x_j .

Neste ponto da formulação, adotaram-se as seguintes regras heurísticas¹³ para obtenção de uma solução aproximada do problema. E as mesmas estão implementadas no algoritmo de otimização *XCut1D* (Ver fluxogramas nos Apêndices):

Heurística 1: Obtenção de uma solução real com posterior truncamento dos valores.

Faz-se o truncamento dos valores de x_j pertencentes ao domínio dos números reais e que foram obtidos pela Técnica da Geração de Colunas, tal que:

$$x_j^h = \lfloor x_j \rfloor, x_j^h \in Z_+. \quad (5.9)$$

onde:

x_j^h , representa a quantidade de vezes que o padrão j se repete na solução real, a qual é truncada para um valor no domínio dos números inteiros não-negativos.

Essa heurística divide o problema inicial em dois outros, pois a solução real truncada x_j^h , atende a um pedido com N_i^h peças, onde N_i^h é menor ou igual a N_i correspondente do pedido inicial.

Vale salientar que nesta solução, toda e qualquer coluna $\mathbf{a} \in Z_+^m$ da matriz de coeficientes \mathbf{A} , é chamada de padrão de corte maximal pois satisfaz a inequação:

$$0 \leq L - \mathbf{a}^T \cdot \mathbf{l} < \text{mínimo}(l_i), \text{ com } i = 1, 2, \dots, m. \quad (5.10)$$

onde:

\mathbf{a} , representa vetor-coluna cujos componentes dizem o número de vezes que a peça de comprimento i entra no padrão de corte j ;

\mathbf{l} , representa o vetor-linha cujos componentes são os comprimentos das peças a serem cortadas.

Heurística 2: Obtenção de uma solução inteira para completar o pedido inicial.

Geralmente esta complementaridade do pedido inicial possui uma solução trivial, ou seja, é atendida sem a necessidade de uma otimização no encaixe das peças.

Aplica-se então o algoritmo *FFD-Solution* para gerar uma solução inteira de forma a completar o pedido inicial nas quantidades estipuladas.

¹³ Entende-se por regras heurísticas um conjunto de procedimentos sem regras matemáticas a priori, que auxiliam na busca de uma ou mais soluções de um problema [39].

Outras hipóteses podem ser acrescentadas sem perda de generalidade. Para garantir uma solução para o problema, assume-se que a quantidade disponível de uma matéria-prima de comprimento L é ilimitada e, o $máximo(l_i) \leq L$ e $N_i \geq 1 \forall i = 1, 2, \dots, m$. Além disso, assume-se que os comprimentos das peças da lista de pedidos são ordenados em ordem decrescente, ou seja, $l_1 > l_2 > \dots > l_m$.

Embora as regras heurísticas adotadas não garantam a obtenção da solução ótima para o Problema do Corte Industrial, as soluções obtidas são, na maioria dos casos, satisfatórias do ponto de vista prático.

5.2 ALGORITMO DE OTIMIZAÇÃO UNIDIMENSIONAL PROPOSTO

Com base na notação definida nas seções anteriores, em linhas gerais, os passos para a implementação desse algoritmo são os seguintes¹⁴:

(1) Em particular, na primeira iteração tem-se que iniciar a matriz \mathbf{A} e o vetor-coluna \mathbf{x} ,

Em geral, pode-se sempre iniciar a matriz dos coeficientes \mathbf{A} com m padrões de corte tais que, em cada padrão, somente um comprimento de peça é cortado. Em outras palavras, somente um elemento de cada coluna da matriz \mathbf{A} é diferente de zero. Assim, tem-se a matriz inicial $\mathbf{A}^1 = [a_{ij}]_{m \times m}$. O valor do vetor-coluna $\mathbf{x}^1 = [x_j]_{m \times 1}$ é obtido resolvendo-se o sistema de equações lineares $\mathbf{A}^1 \cdot \mathbf{x}^1 = \mathbf{N}$ de m linhas e m colunas.

A maneira adotada acima de se iniciar os valores conduz a uma solução inicial viável para o problema; porém, é desejável que essa estimativa inicial da solução seja a mais próxima possível da solução ótima desejada. Para tanto, podem ser utilizados um dos dois algoritmos para uma boa estimativa da solução inicial. O *Algoritmo Initial-Solution* gera uma estimativa de solução real para o Problema do Corte Industrial. Já o *Algoritmo FFD-Solution* gera uma estimativa de solução inteira a partir de algumas alterações feitas no *Algoritmo Initial-Solution*. A boa estimativa de solução inicial gerada por esses algoritmos permite que o número de iterações necessário para obter a solução do Problema do Corte Industrial seja reduzido. Ambos os algoritmos estão descritos no Apêndice em anexo e mais informações sobre eles pode ser encontrada em Chvátal [6].

¹⁴ O Índice sobrescrito indica o número da iteração atual e, o Índice subscrito indica a componente do vetor.

De acordo com as heurísticas adotadas neste trabalho, inicialmente utilizou-se o algoritmo *Initial-Solution* para gerar uma solução inicial no domínio dos números reais.

Para toda e qualquer iteração j , tem-se que:

(2) Resolver o sistema $\mathbf{y}^j \cdot \mathbf{A}^j = [1 \ 1 \ \dots \ 1]_{1 \times m}$, achando o vetor-linha \mathbf{y}^j ;

A resolução do sistema de equações lineares é feita aplicando-se o *Método de Gauss-Jordan*, o qual pode ser facilmente encontrado em livros de Álgebra ou de Cálculo Numérico. Um exemplo de implementação computacional desse método está em Shammás [36], o qual serviu de suporte para a rotina implementada neste trabalho.

(3) Gerar uma nova coluna, tal que a condição descrita na Equação (5.3) se verifique, e

que também satisfaça a condição $\sum_{i=1}^m a_i \cdot y_i > 1$, onde y_i é cada uma das componentes do vetor-linha \mathbf{y}^j encontrado no passo 2;

Neste ponto, tem-se que resolver o Problema de Knapsack com a aplicação do algoritmo de Branch-and-Bound, o qual está descrito no Apêndice em anexo. Esta é a primeira etapa da Técnica da Geração de Colunas. O Problema de Knapsack a ser resolvido no Problema do Corte Industrial é o seguinte:

$$\text{maximizar } \sum_{i=1}^m a_i \cdot y_i > 1,$$

$$\text{sujeito a } \sum_{i=1}^m a_i \cdot l_i \leq L.$$

(4) Resolver o sistema $\mathbf{A}^j \cdot \mathbf{d}^j = \mathbf{a}^j$, onde \mathbf{a}^j é a nova coluna gerada no passo 3;

Adota-se a rotina do Método de Gauss-Jordan implementada neste trabalho.

(5) Determinar qual coluna da matriz de coeficientes \mathbf{A}^j vai sair para a entrada da nova coluna melhorada e gerada no passo 3;

A escolha é baseada na menor razão obtida entre as componentes correspondentes dos vetores-coluna \mathbf{x}^j e \mathbf{d}^j , ou seja, $t = \text{mínimo}(x_i / d_i)$. Sai a coluna correspondente a componente i do valor t mínimo. Esta é a segunda etapa da Técnica da Geração de Colunas.

(6) Atualizar a matriz A^j e o vetor-coluna x^j ;

O vetor-coluna x^j tem cada uma das suas componentes subtraída pelo valor $t \cdot d_i$, assim, $x_i = x_i - t \cdot d_i$. A exceção fica por conta da componente i correspondente à razão t , a qual assume o próprio valor de t .

(7) Realizar uma nova iteração, retornando ao passo 2 com os valores atualizados da matriz A^j e do vetor x^j , até não mais se verificar as condições do passo 3; pois se isso ocorrer, deve-se realizar o passo 8;

Ver fluxograma da Figura 5-1 para a seqüência de passos 2 até 7.

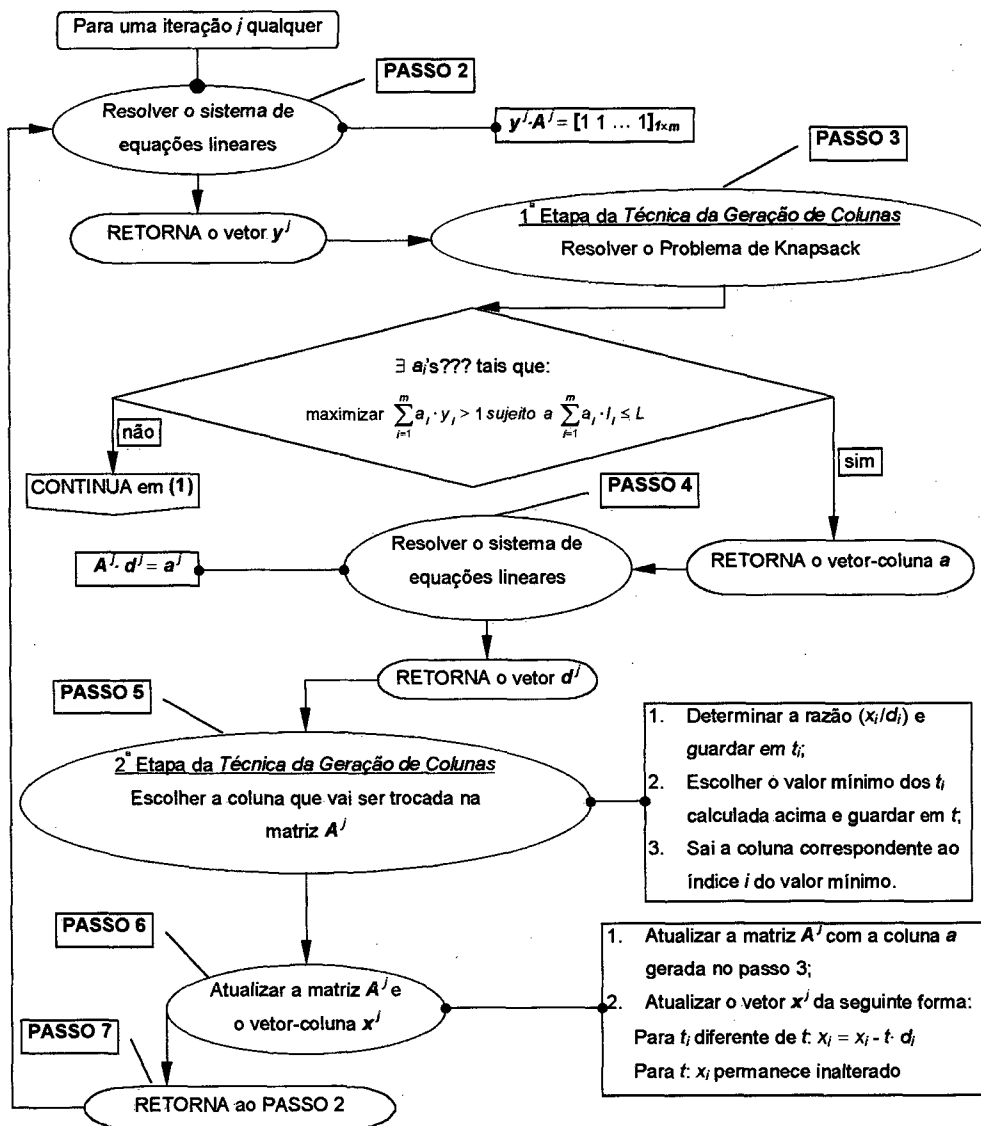


Figura 5-1: Seqüência dos passos 2 a 7 do algoritmo XCut1D

Continuando a seqüência de passos do algoritmo, tem-se que:

- (8) Armazenar e imprimir os valores reais da matriz A^j e do vetor x^j
- (9) Aplicar a Heurística 1: Truncar os valores do vetor-coluna x^j para o domínio dos números reais e atualizar o valor das quantidades de peças do pedido;
- $$\text{Fazer } x_j^h = \lfloor x_j \rfloor \text{ e } N_i^h = \sum_{j=1}^{n=m} a_{ij} \cdot x_j^h. \text{ Então } N_i = N_i - N_i^h$$
- (10) Aplicar a Heurística 2: Aplicar o algoritmo *FFD-Solution* para gerar uma solução inteira que complete a lista do pedido inicial;
- (11) Armazenar e imprimir a solução inteira obtida no passo 10;
- (12) Finalizar o algoritmo.
- Ver fluxograma da Figura 5-2 para a seqüência de passos 8 até 12.

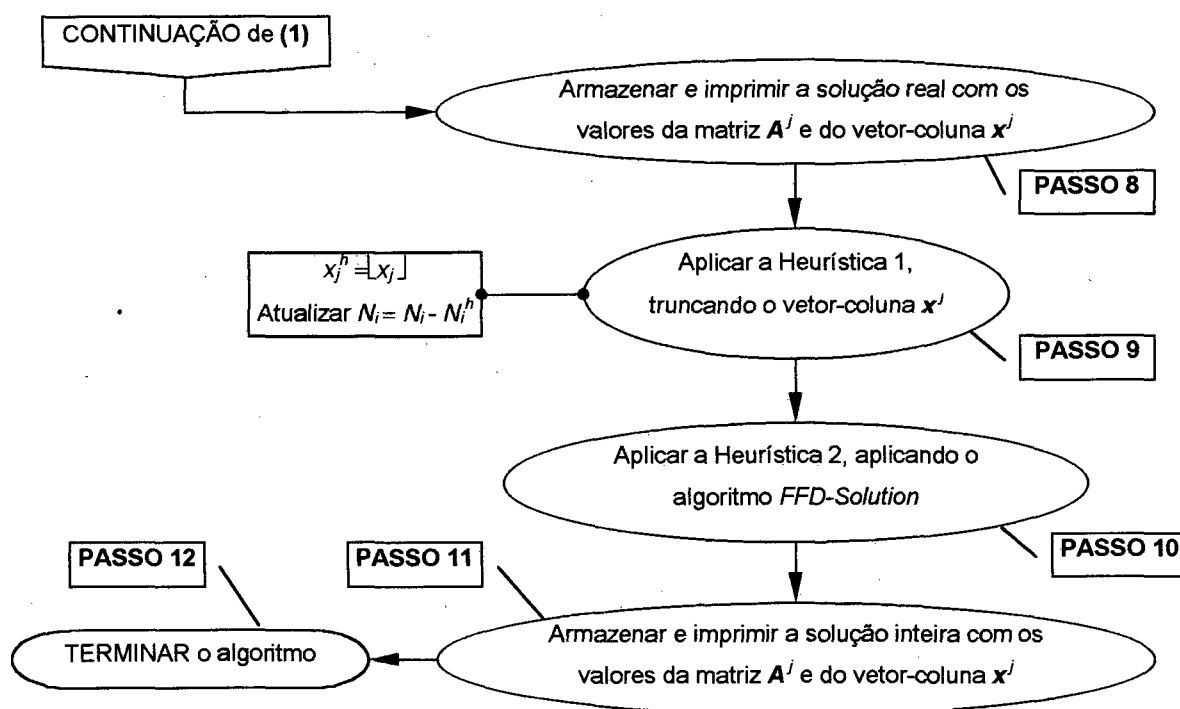


Figura 5-2: Seqüência dos passos 8 a 12 do algoritmo *XCut1D*

Capítulo 6

RESULTADOS

Os exemplos e situações descritos neste capítulo objetivam testar o algoritmo de otimização de corte unidimensional e validar esses resultados com base em algoritmos semelhantes apresentados em outros trabalhos [12][18]. Para tanto, tentou-se reproduzir algumas situações extraídas desses trabalhos. Também, foram feitos alguns comentários sobre questões particulares observadas no algoritmo proposto.

6.1 CASO PRÁTICO

6.1.1 Descrição do Problema

O Problema do Corte Industrial analisado aqui, foi extraído de um artigo de Degreave e Bemelmans [12]. Este artigo descreve um problema real que uma empresa de projetos de engenharia e fabricação de máquinas — a Metal Aarschot — teve que resolver a fim de ganhar uma concorrência para construção de uma máquina.

A concorrência consistia da fabricação de uma máquina de embalagens para acondicionar frios e defumados numa indústria alimentícia. O pedido inicial era constituído de 7 máquinas, sendo que 6 delas seriam para o mercado interno e 1 para o mercado norte-americano. E além desse pedido, posteriormente deveria se seguir um outro de 100 ou mais máquinas, pois se planejava automatizar todo o setor de embalagens.

A Metal Aarschot Engineering (MAE) propôs a máquina ilustrada na Figura 6-1 e recebeu a aprovação do cliente entre todos os outros projetos dos demais fabricantes envolvidos na concorrência. A MAE foi incumbida projetar e fabricar a máquina para atender o pedido inicial do cliente.

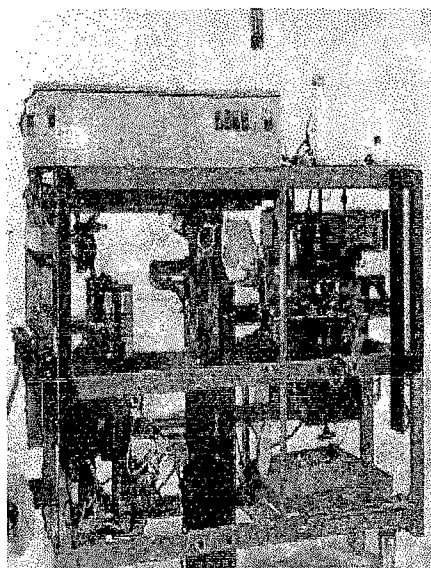


Figura 6-1: Estrutura proposta para a máquina-embaladeira

Conforme vai ser mostrado a seguir, as condicionantes que se seguiram ao projeto e planejamento da fabricação desta máquina exigiram o uso de metodologias de otimização, a fim de melhorar o aproveitamento da matéria-prima.

A situação descrita acima caracteriza e exemplifica bem um problema de otimização no aproveitamento da matéria-prima.

6.1.2 Definições Gerais

Para quantificar e interpretar os valores das respostas fornecidas pelo algoritmo de otimização, foram definidas as seguintes grandezas: sobra do padrão de corte, quantidade de matéria-prima cortada e sobra total de matéria-prima.

A sobra s_j do padrão de corte j é calculada pela Equação (6.1) e fornecida na unidade de comprimento (*u.c.*) correspondente à unidade utilizada para o comprimento da matéria-prima cortada.

$$s_j = L - \sum_{l=1}^m a_{lj} \cdot l_l, \text{ em } u.c., \quad (6.1)$$

onde:

m , é o número de comprimentos distintos de peças a serem cortadas;

L , é o comprimento nominal da matéria-prima;

l_i , é o comprimento da peça i a ser cortada;

a_{ij} , é o número de vezes que a peça i entra no padrão de corte j .

A quantidade de matéria-prima cortada, N_{mp} , é dada por:

$$N_{mp} = \sum_{j=1}^n x_j, \quad (6.2)$$

onde:

n , é a quantidade de padrões de corte distintos;

x_j , é a quantidade de vezes que o padrão de corte j se repete.

A sobra total de matéria-prima, S_{mp} , no pedido é dada por:

$$S_{mp} = \sum_{j=1}^n s_j \cdot x_j, \text{ em u.c.} \quad (6.3)$$

A sobra percentual total de matéria-prima tomada sobre o comprimento total de corte é dada por:

$$S_{mp} = 100 \cdot \left(\frac{\sum_{j=1}^n s_j \cdot x_j}{N_{mp} \cdot L} \right) \text{ ou } S_{mp} = 100 \cdot \left[1 - \left(\frac{\sum_{i=1}^m N_i \cdot l_i}{N_{mp} \cdot L} \right) \right], \text{ em } \%, \quad (6.4)$$

onde:

N_i , é a quantidade cortada de peças de comprimento l_i .

6.1.3 Dados do Problema

As peças que compõem a estrutura para construção de 1 máquina estão listadas na Tabela 6-1. Dois tipos diferentes de aço foram utilizados como matéria-prima para o corte das peças. O aço¹⁵ denominado de Tipo 1 se refere às barras de comprimento nominal $L_1 = 6000,00$ mm; enquanto o aço do Tipo 2, refere-se às barras de comprimento nominal $L_2 = 3000,00$ mm.

¹⁵ Degreave e Bemelmans [12], chamavam o aço do Tipo 1 de St.37 60*60*5, e o aço do Tipo 2 de St.37 60*20.

Tabela 6-1: Lista de peças para construção de 1 estrutura completa da máquina

LISTA DE PEÇAS		Comprimento, em mm	Quantidade	
Matéria-prima: Aço	Tipo 1	Peça #11	1910,00	1
		Peça #12	1850,00	1
		Peça #13	1790,00	2
		Peça #14	1110,00	1
		Peça #15	1101,00	3
		Peça #16	550,00	1
		Peça #17	500,00	5
		Peça #18	340,00	1
	Tipo 2	Peça #21	560,00	1
		Peça #22	491,00	1
Peça #23		390,00	2	

Durante a construção das máquinas foi detectado um problema de desperdício de matéria-prima. A MAE propunha em seu projeto que a máquina consistia de uma estrutura de barras de aço, as quais deveriam ser cortadas a partir de barras de aço e daí soldadas. Ou seja, a montagem da estrutura se resumia em operações de corte e soldagem de barras de aço. Após a construção das oito primeiras estruturas, detectou-se que uma quantidade considerável de sobras de barras de aço estavam se perdendo em virtude de dois fatos: primeiro, a estrutura era formada por barras de comprimentos distintos e segundo, devido à forma não planejada de como os cortes da matéria-prima estavam sendo combinados e cortados.

Para a quantidade de estruturas construídas até então, o desperdício de matéria-prima não implicava em perdas financeiras consideráveis. Todavia, em vista de novos pedidos que se seguiriam, o desperdício deveria ser minimizado. Por conta disso, a montagem da máquina em si passou a ser uma questão secundária diante do problema de minimizar o desperdício de matéria-prima. Ou seja, o cliente externo estava satisfeito em suas necessidades. A grande questão agora era satisfazer o cliente interno, otimizando o uso da matéria-prima e reduzindo os custos de fabricação.

Um pedido de corte específico foi colocado para ser resolvido pelo algoritmo de otimização de corte unidimensional proposto neste trabalho de dissertação. Este algoritmo recebeu o nome de **XCut1D**. Comentários adicionais sobre a implementação computacional e sobre a organização dos arquivos podem ser lidos no Apêndice A, em

anexo. Na seção seguinte, os resultados obtidos com o algoritmo *XCut1D* foram comparados com as respostas dadas pelo artigo de Degreave e Bemelmans [12].

6.1.4 Solução e Listagem de um Pedido de Corte

Nesta seção vai ser analisado o problema do corte unidimensional para a construção da estrutura de 15 máquinas. Portanto, será preciso ser cortado um conjunto completo de peças para a montagem de 15 estruturas. A Tabela 6-2 lista as variáveis de entrada do problema para cortar as peças que usarão o aço do Tipo 1. Além disso, a tabela informa o comprimento e o número total de cada peça a ser usada na construção das estruturas. O código empregado na identificação das peças segue a seguinte convenção: Peça #XY, indica que a peça cortada é de material Tipo X e comprimento nominal l_Y . Deve-se ressaltar que em algumas partes do texto, o termo “peça cortada” também quer dizer “comprimento cortado”, visto que existe uma relação intrínseca entre a peça e o seu comprimento.

Assim, o problema é constituído das seguintes variáveis de projeto:

Comprimento nominal da matéria-prima a ser cortada: $L = L_1 = 6000,00$ mm;

Número de comprimentos distintos a serem cortados: $m = 8$;

Quantidade total N, de cada uma das peças a serem cortadas:

Tabela 6-2: Lista de peças de aço Tipo 1 para o pedido das 15 estruturas

LISTA DE PEÇAS	¹⁶ Comprimento l , em mm	Quantidade, N
Peça #11	$l_1 = 1910,00$	$N_1 = 15$
Peça #12	$l_2 = 1850,00$	$N_2 = 15$
Peça #13	$l_3 = 1790,00$	$N_3 = 30$
Peça #14	$l_4 = 1110,00$	$N_4 = 15$
Peça #15	$l_5 = 1101,00$	$N_5 = 45$
Peça #16	$l_6 = 550,00$	$N_6 = 15$
Peça #17	$l_7 = 500,00$	$N_7 = 75$
Peça #18	$l_8 = 340,00$	$N_8 = 15$

De maneira semelhante, a Tabela 6-3 lista as variáveis de entrada do problema para cortar as peças fabricadas com o aço do Tipo 2:

¹⁶ Os comprimentos da lista de pedidos devem ser informados ao algoritmo em ordem decrescente.

Comprimento nominal da matéria-prima a ser cortada: $L = L_2 = 3000,00$ mm;

Número de comprimentos distintos a serem cortados: $m = 3$;

Quantidade total N , de cada uma das peças a serem cortadas:

Tabela 6-3: Lista de peças de aço Tipo 2 para o pedido das 15 estruturas

LISTA DE PEÇAS	Comprimento l_i em mm	Quantidade, N
Peça #21	560,00	15
Peça #22	491,00	15
Peça #23	390,00	30

Os valores acima foram processados pelo algoritmo de otimização *XCut1D*. A solução fornecida pelo algoritmo, listando todos os padrões de corte para o pedido de 15 estruturas, é mostrada na Tabela 6-4 e na Tabela 6-5.

Tabela 6-4: Padrões de corte para as peças de aço Tipo 1 – Algoritmo *XCut1D*

PADRÃO DE CORTE		PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	PC10
QUANTIDADE DO PADRÃO		12	14	5	1	1	1	1	1	1	1
COMPRIMENTO, EM MM	$l_1 = 1910,00$	1				2		1			
	$l_2 = 1850,00$		1					1			
	$l_3 = 1790,00$	2		1				1			
	$l_4 = 1110,00$			1	4	1			5		
	$l_5 = 1101,00$		3							3	
	$l_6 = 550,00$			2	1	1				3	
	$l_7 = 500,00$	1	1	4	2	1	12			2	12
	$l_8 = 340,00$		1					1			
SOBRA, EM mm		10,0	7,0	0,0	10,0	20,0	0,0	110,0	450,0	47,0	0,0
SOBRA, EM %		0,17	0,12	0,0	0,17	0,33	0,0	1,83	7,50	0,78	0,0

Tabela 6-5: Padrões de corte para as peças de aço Tipo 2 – Algoritmo *XCut1D*

PADRÃO DE CORTE		PC1	PC2	PC3	PC4
QUANTIDADE DO PADRÃO		4	4	1	1
COMPRIMENTO, EM MM	$l_1 = 560,00$	2	1	3	
	$l_2 = 491,00$	3		2	1
	$l_3 = 390,00$	1	6		2
SOBRA, EM mm		17,0	100,0	338,0	1729,0
SOBRA, EM %		0,57	3,33	11,27	57,63

Os padrões de corte representam a disposição espacial das peças, e de como 1 barra é cortada. É feita ainda a indicação de quais peças, o número de vezes que ela entra no padrão e a quantidade de vezes que o padrão se repete.

Usando as equações definidas na seção 6.1.2, comprova-se da Tabela 6-4, que 38 barras de aço do Tipo 1 serão cortadas. O percentual de sobras totais é apenas 0,38% sobre o comprimento total de corte. Isso equivale a 855,00 mm de sobras totais ou a 14,25% do comprimento nominal de uma barra de aço do Tipo 1.

Vale ressaltar que os padrões de corte PC1, PC2, PC3, PC4, PC5 e PC6 da Tabela 6-4 foram calculados pela parte real do algoritmo *XCut1D*, enquanto os demais padrões de corte foram calculados pela parte inteira do algoritmo. Devido à heurística adotada na implementação do algoritmo, a solução inteira obtida sempre complementa o pedido com a quantidade de peças ainda faltantes. Como o número de peças para completar o pedido geralmente é pequeno, e conseqüentemente o número de combinações entre as peças é pequeno, isso explica o porquê das sobras parciais de cada padrão de corte aumentarem nesta solução. Tanto é que, se fosse preciso mais uma peça de 340,00 mm, ela poderia ser atendida pelo padrão de corte PC8 e assim, a sobra parcial do padrão diminuiria para 110,00 mm e a sobra total do pedido, para 515,00 mm.

Atendendo o pedido para as peças fabricadas como o aço do Tipo 2, a solução dada pelo algoritmo e listada na Tabela 6-5, determina que sejam cortadas 10 barras de aço do Tipo 2. A sobra percentual é de 8,45% sobre o comprimento total de corte. Isso equivale a 2535,00 mm de sobras totais ou a 84% do comprimento nominal de uma barra de aço do Tipo 2.

Os valores da solução acima mostram que um pedido de corte para 15 estruturas não é um bom pedido quando visto sob o enfoque de minimização das sobras. Para tanto, basta verificar que os padrões de corte gerados pela parte inteira do algoritmo apresentaram uma sobra parcial maior que 10% do comprimento nominal da matéria-prima. O padrão de corte PC3 apresentou uma sobra de 11,27% do comprimento nominal da barra de aço Tipo 2. Por sua vez, o padrão de corte PC4 apresentou uma sobra superior à metade do comprimento nominal da matéria-prima, de valor igual a 57,63%. O que pode ser feito nestes casos, é verificar a possibilidade de se alterar o comprimento nominal da matéria-prima ou tentar escolher um pedido mais econômico variando-se a quantidade de estruturas cortadas num pedido.

Para efeito de validação do algoritmo *XCut1D*, a solução para o pedido das 15 estruturas apresentou valores iguais aos que Degreave e Bemelmans [12] conseguiram obter, conforme pode ser comprovado na Tabela 6-6 e Tabela 6-7.

Tabela 6-6: Solução de Degreave e Bemelmans para peças de aço Tipo 1

PADRÃO DE CORTE		PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	PC10
QUANTIDADE DO PADRÃO		12	3	11	3	4	1	1	1	1	1
COMPRIMENTO, EM mm	$l_1 = 1910,00$			1				2		2	
	$l_2 = 1850,00$	1					1		2		
	$l_3 = 1790,00$		1	2		1			1		
	$l_4 = 1110,00$		1		4						
	$l_5 = 1101,00$	3				2		1			
	$l_6 = 550,00$		2		1		6	1			
	$l_7 = 500,00$	1	4	1	2	4	1	1	1	3	12
	$l_8 = 340,00$	1					1			2	
SOBRA, EM mm		7,0	0,0	10,0	10,0	8,0	10,0	29,0	10,0	0,0	0,0
SOBRA, EM %		0,12	0,0	0,17	0,17	0,13	0,17	0,48	0,17	0,0	0,0

Tabela 6-7: Solução de Degreave e Bemelmans para peças de aço Tipo 2

PADRÃO DE CORTE		PC1	PC2	PC3	PC4	PC5
QUANTIDADE DO PADRÃO		4	2	2	1	1
COMPRIMENTO, EM mm	$l_1 = 560,00$	1	3	2	2	
	$l_2 = 491,00$		1	3	1	6
	$l_3 = 390,00$	6	2	1		
SOBRA, EM mm		100,0	49,0	17,0	1389,0	54,0
SOBRA, EM %		3,33	1,63	0,57	46,30	1,80

As diferenças entre os algoritmos de solução ficaram restritas apenas às configurações dos padrões de corte; isto é, à diferença nas quantidades e nas peças que deveriam pertencer ao padrão. Um fato interessante de comentar é que o algoritmo de Degreave e Bemelmans gerou peças além da quantidade pedida. A solução obteve 16 peças de comprimento 550,00 mm (*Peça #16*) e de comprimento 560,00 mm (*Peça #21*), ao invés de apenas 15. Isso pode ser constatado na Tabela 6-6 e Tabela 6-7.

Certamente, isso pode ocorrer pelo simples fato do uso de regras heurísticas diferentes adotadas em cada um dos algoritmos empregados na solução do problema.

6.2 ALTERAÇÃO NOS VALORES DAS VARIÁVEIS DO PROBLEMA

Em virtude da própria natureza do Problema do Corte Industrial, observou-se uma grande sensibilidade com relação às alterações realizadas nas variáveis principais do problema. E isso está de acordo com o que Dyckhoff [14] colocou em seu trabalho. Dyckhoff listou algumas das características que são responsáveis por essa sensibilidade, além de ter analisado os efeitos provocados por elas nesse tipo de problema.

O algoritmo *XCut1D* serviu de ferramenta para observar, comprovar e avaliar alguns desses efeitos ainda sobre o problema da Metal Aarschot, o qual pode ser classificado como um problema tipo 1/V//R, ou seja, é um problema unidimensional (1), onde padrões de corte otimizados são selecionados (V) e cortados a partir de apenas um tipo de matéria-prima (I), e o número de peças é pequeno (R).

6.2.1 Variação na Quantidade de Peças

Dentre as maneiras possíveis de se otimizar o aproveitamento da matéria-prima durante o corte é através da variação na quantidade das peças. O objetivo é verificar e escolher um pedido que apresente sobras totais menores. Isso foi feito no caso prático acima.

O pedido para as peças do aço Tipo 1 não apresentaram melhoras nos valores de sobra a partir das variações realizadas na quantidade de peças. Isso talvez porque o valor apresentado por esse pedido já era bastante satisfatório, apenas 0,38%. Já no caso do pedido de peças de aço Tipo 2, gráfico da Figura 6-2 resume uma série de casos calculados no algoritmo de otimização *XCut1D*.

Verifica-se que a maior quantidade de sobras totais ocorre para um pedido de 19 estruturas, com 10,80%. O menor valor ocorre quando são cortadas 37 estruturas, onde as sobras totais são de 1,82% somente. Ou seja, tem-se uma redução de quase 9,0% apenas variando a quantidade de estruturas cortadas. Vale ressaltar que o mínimo de sobras foi obtido justamente num caso em que a solução fornecida foi calculada apenas pela parte real do algoritmo, ou seja, a parte inteira não foi calculada neste caso. Pode-se observar também no gráfico da Figura 6-2, uma periodicidade nos valores de sobra total. O valor de 2,35% de sobras se repete quando a quantidade

cortada de estruturas é um múltiplo de 8. O mesmo comportamento se verifica para o valor do mínimo da sobras, repetindo-se a cada múltiplo de 37 estruturas cortadas.

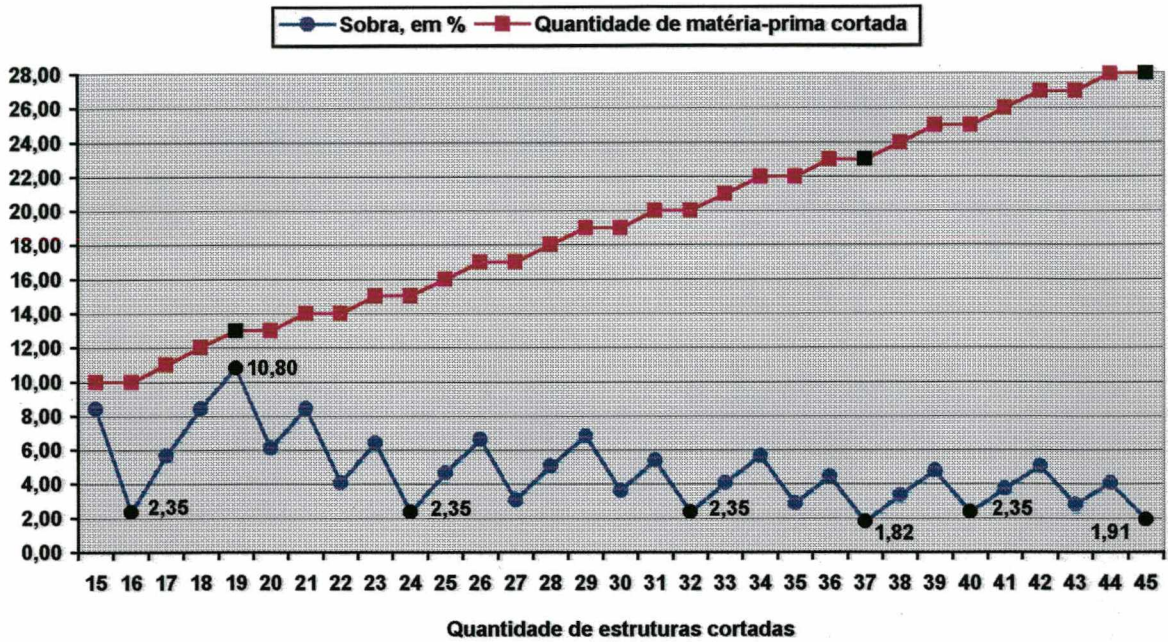


Figura 6-2: Variação na quantidade cortada de peças de aço Tipo 2

É lógico que se não fosse necessário cortar 37 estruturas, mas apenas uma quantidade menor; mesmo assim, o algoritmo ajuda na tarefa de encontrar um valor que melhor se ajusta dentro da quantidade requerida.

Por exemplo, considere o caso de se atender um pedido de 45 estruturas. Se o corte for feito no lote de 45 estruturas de uma vez só, cortam-se 28 barras. Entretanto se o corte for parcelado em dois lotes menores, a maioria das combinações cortam 29 barras. E se aumentarem o número de parcelas, pode ser necessário cortar até 30 barras, ou seja, 2 barras a mais para atender o mesmo pedido de 45 estruturas, conforme se pode verificar na Tabela 6-8.

Tabela 6-8: Parcelamento do pedido em lotes

Tamanho dos lotes	Quantidade de barras cortadas
45	28
35, 10	29
30, 15	29
15, 15, 15	30

6.2.2 Variação no Comprimento da Matéria-prima

Conforme foi sugerido no caso prático estudado acima, outra maneira de otimizar o uso da matéria-prima, é variando-se o comprimento nominal da matéria-prima utilizada para cortar o pedido.

Isso também foi feito no pedido das 15 estruturas. A Metal Aarschot previa que no futuro, as matérias-primas de ambos os tipos seriam fornecidas em comprimentos que variariam de 1,00 m num intervalo de 6,00 m a 15,00 m. Considerando isso, os casos foram simulados no algoritmo *XCut1D*. O objetivo foi verificar e escolher um comprimento que otimizasse o aproveitamento da matéria-prima. Além de verificar a influência da variação do comprimento da matéria-prima.

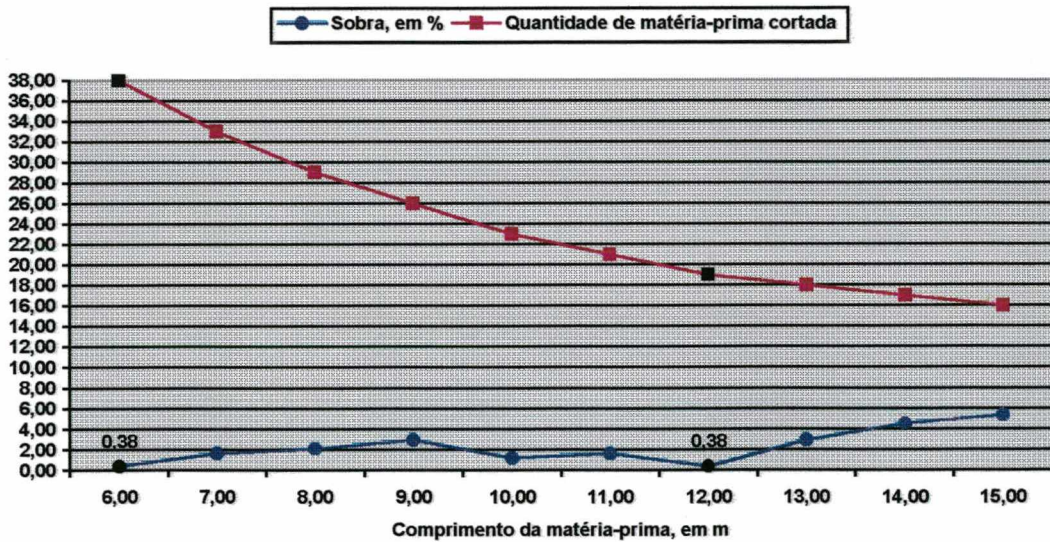


Figura 6-3: Variação no comprimento da matéria-prima de aço Tipo 1

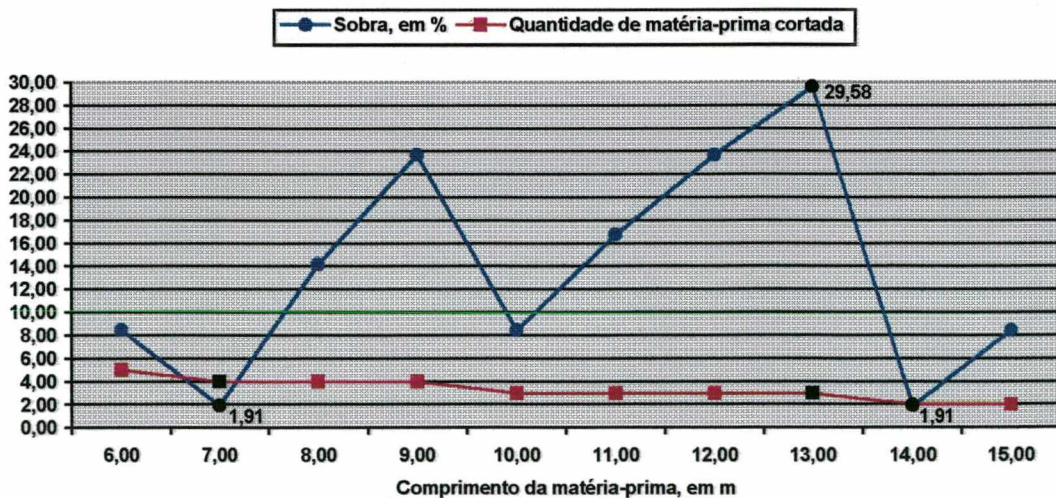


Figura 6-4: Variação no comprimento da matéria-prima de aço Tipo 2

Percebe-se pela Figura 6-3, que o pedido das 15 estruturas para o aço Tipo 1 pode ser melhor atendido com barras de comprimentos 6,00 m e 12,00 m. A sobra total é de 0,38% em ambos. Enquanto que no caso do aço Tipo 2 na Figura 6-4, o pedido pode ser melhor atendido para barras de comprimentos 7,00 m e 14,00 m. E em ambos, a sobra total é de 1,91%.

Para os respectivos dados do problema, observou-se ainda que a ocorrência dos pontos de mínima e de máxima sobra de matéria-prima ocorrem numa periodicidade dada por múltiplos do comprimento de 6,0 m para o aço Tipo 1, e por múltiplos do comprimento de 7,0 m para o aço Tipo 2.

Embora essa periodicidade não possa ser generalizada para todos os casos, em particular ele pode ser explicado. Considere a situação em que as peças no padrão de corte estão dispostas de forma tal que a sobra total seja a menor possível para aquelas peças. Fica lógico pensar que se for usado outros comprimentos de matéria-prima maiores, as sobras totais devem inicialmente aumentar até atingirem um máximo. Desse ponto em diante, os aumentos no comprimento da matéria-prima encaixarão outras combinações possíveis entre as peças da lista de pedido e aí as sobras totais começarão a diminuir até um mínimo. E essa periodicidade terá uma frequência maior ou menor dependendo do número de combinações possíveis (*padrões de corte possíveis e viáveis*) e dos valores dos comprimentos entre as peças da lista.

6.3 CASO TEÓRICO

Gilmore e Gomory [18] também estudaram o efeito da variação do comprimento da matéria-prima a partir de dados extraídos de um problema da indústria de papel. Esse problema pode ser classificado como do tipo 1/VI/M. Os dados do pedido estão mostrados na Tabela 6-9.

Os dados da Tabela 6-9 foram processados pelo algoritmo *XCut1D* com o comprimento nominal da matéria-prima variando de 1,00 pol, desde um comprimento nominal de 130,00 pol até um de 330,00 pol. O intervalo de investigação foi estendido a fim de observar o comportamento da variação de sobras de matéria-prima com o comprimento, para valores um pouco antes do comprimento de 168,00 pol e um pouco além do comprimento de 260,00 pol.

Tabela 6-9: Lista das peças do pedido

LISTA DE PEÇAS	Comprimento l, em pol	Quantidade, N	LISTA DE PEÇAS	Comprimento l, em pol	Quantidade, N
Peça 01	81,00	4415	Peça 16	51,00	212
Peça 02	70,00	291	Peça 17	50,00	720
Peça 03	68,25	4765	Peça 18	49,50	133
Peça 04	67,50	4827	Peça 19	46,50	529
Peça 05	66,75	90	Peça 20	45,50	185
Peça 06	66,00	691	Peça 21	44,50	94
Peça 07	64,00	263	Peça 22	41,25	393
Peça 08	63,75	141	Peça 23	38,50	47
Peça 09	63,00	133	Peça 24	38,00	95
Peça 10	60,00	390	Peça 25	35,00	411
Peça 11	56,25	459	Peça 26	33,50	36
Peça 12	56,00	343	Peça 27	33,00	273
Peça 13	52,50	766	Peça 28	32,00	56
Peça 14	52,00	58	Peça 29	31,50	171
Peça 15	51,75	2736	Peça 30	21,50	140

Os valores obtidos pelo algoritmo *XCut1D* para as sobras totais de matéria-prima foram plotados na série de gráficos abaixo. O aspecto do gráfico na Figura 6-5 reproduz bem o comportamento apresentado no trabalho de Gilmore e Gomory [18]. O objetivo era observar o comportamento das sobras totais com a variação do comprimento nominal da matéria-prima diante de um problema que apresenta uma diversidade de comprimentos de peças a serem cortadas e uma grande dispersão na quantidade de cada peça.

E em linhas gerais, comprovou-se que as sobras totais tendem a diminuir à medida que o comprimento nominal da matéria-prima aumenta. Gilmore e Gomory acharam como ponto de mínimo o comprimento de 217,00 pol, cuja sobra total foi de 0,11%. Esse valor da sobra total foi fornecido pelo algoritmo *XCut1D*, já que pelo gráfico mostrado no trabalho de Gilmore e Gomory só se constata que o mesmo se encontra bem próximo de zero.

Alguns pontos dos gráficos apresentam valores interessantes e que vale a pena ser comentado. Os cinco primeiros pontos no gráfico da Figura 6-6, correspondentes aos comprimentos de 130,00 pol a 134,00 pol, são altamente dispendiosos. Eles apresentam percentuais de sobras totais de 22,34% em média. Entretanto, logo em seguida, para um comprimento de 136,00 pol, a sobra total correspondente é de

apenas 1,46%. A Tabela 6-10 dá uma idéia de economia de matéria-prima em termos de sobras totais transformadas para uma quantidade equivalente de matéria-prima.

Tabela 6-10: Exemplo de economia de matéria-prima

MUDANÇA DE COMPRIMENTO		ECONOMIA DE MATÉRIA-PRIMA
DE:	PARA:	
133,00 pol	134,00 pol	236,38 unidades
134,00 pol	135,00 pol	2351,76 unidades
135,00 pol	136,00 pol	491,61 unidades

O cálculo realizado para gerar os valores da tabela acima, foi o seguinte: determinou-se a sobra total individual para cada um dos comprimentos utilizados; subtraiu-se esses valores e dividiu-se o resultado dessa operação pelo maior dos comprimentos utilizados. Exemplificando com valores, tem-se que:

Considere a mudança de comprimento de $l_1 = 133,00$ pol para $l_2 = 134,00$ pol.

Para $l_1 = 133,00$ pol, corresponde 438545,50 pol de sobra total;

Para $l_2 = 134,00$ pol, corresponde 406870,50 pol de sobra total;

Subtraindo-se os valores das sobras totais, tomando-se o módulo e dividindo-se por 134, obtém-se como resposta uma economia de 236,38 unidades de matéria-prima.

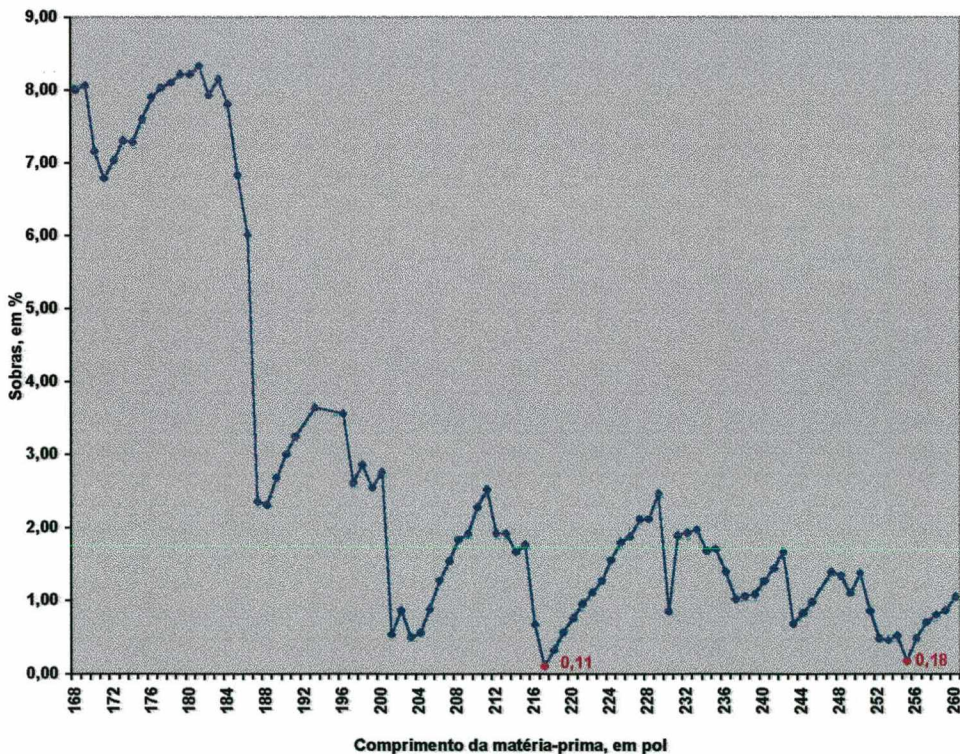


Figura 6-5: Variação no comprimento da matéria-prima — 168,00 a 260,00 pol

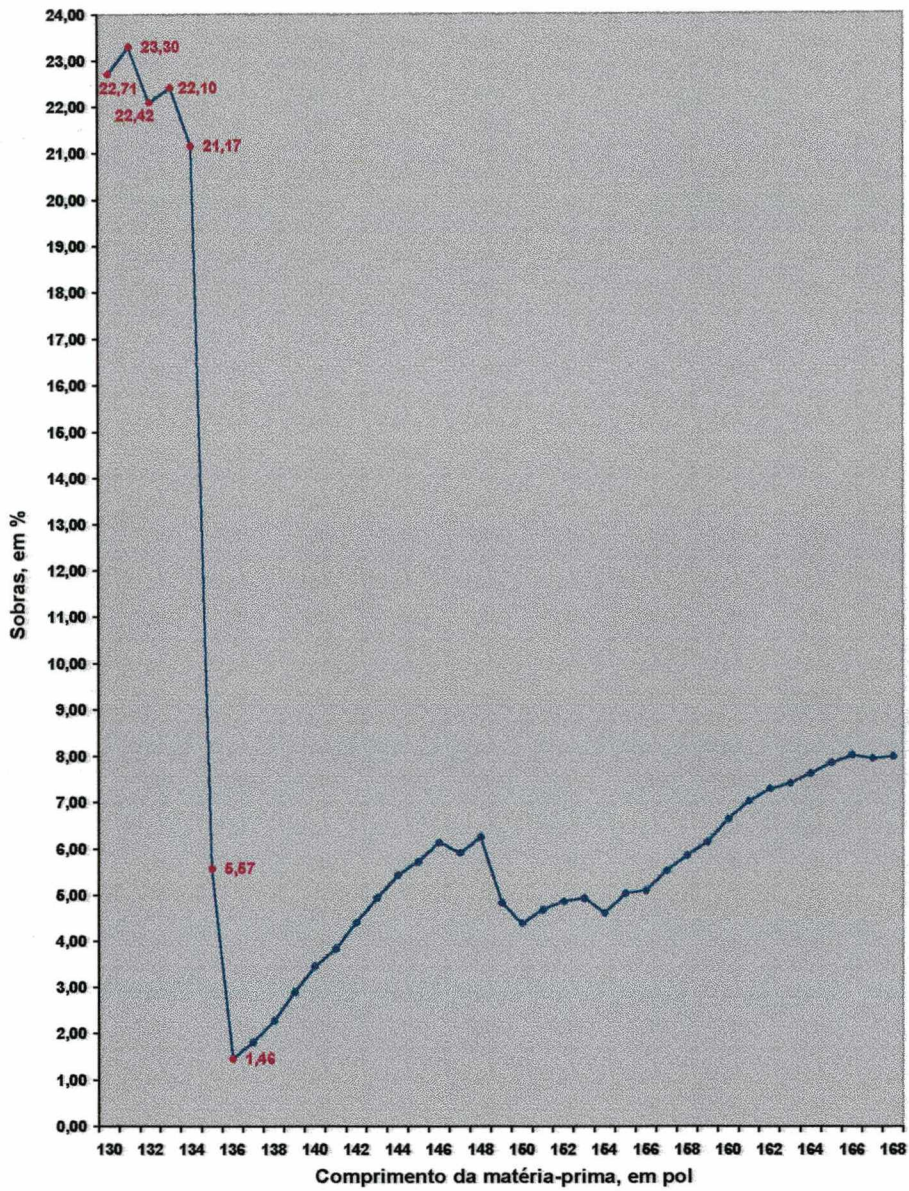


Figura 6-6: Variação no comprimento da matéria-prima — 130,00 a 168,00 pol

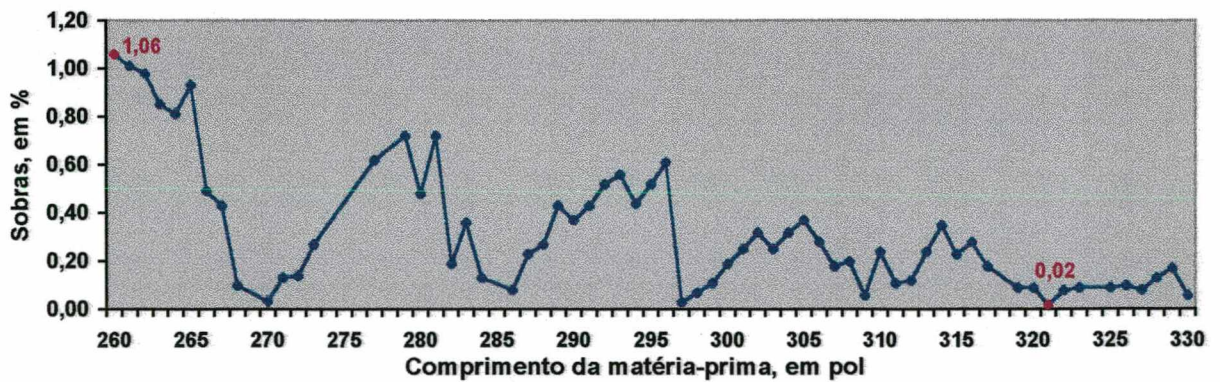


Figura 6-7: Variação no comprimento da matéria-prima — 260,00 a 330,00 pol

O gráfico da Figura 6-7 mostra que as sobras totais ficam abaixo de 1,00% no intervalo de valores investigado, com exceção apenas do valor de 1,06% de sobra total verificada no comprimento de 260,00 pol. Neste intervalo ainda, verificou-se a existência de diversos pontos com valores de sobras totais abaixo do mínimo apontado pelo gráfico da Figura 6-5, o qual reproduz os resultados do trabalho de Gilmore e Gomory [18]. O mínimo verificado ocorreu para um comprimento de matéria-prima de 321,00 pol, com 0,02% de sobra total. Isso equivale a apenas 349,50 pol; ou seja, um encaixe quase exato, conforme se pode verificar nas tabelas abaixo.

Tabela 6-11: Padrões de corte — Parte inteira da solução do *XCut1D*

PADRÃO DE CORTE		PC 01	PC 02	PC 03	PC 04	PC 05	PC 06	PC 07	PC 08	PC 09	PC 10	PC 11
QUANT. DO P.C.		1	1	1	1	1	1	1	1	1	1	1
COMPRIMENTO, EM pol	81,00	3	3	2								
	70,00	1										
	68,25		1	2	1							
	67,50				3	4	3					
	66,75						1					
	66,00							3				
	64,00							1				
	63,75											
	63,00											
	60,00								1			
	56,25							1	1			
	56,00								2			
	52,50											
	52,00								1			
	51,75						1			1		
	51,00											
	50,00				1	1				4		
	49,50											
	46,50									1	1	
	45,50										2	
44,50										1		
41,25												
38,50								1				
38,00										1		
35,00										2	3	
33,50												
33,00											2	
32,00											1	
31,50											1	
21,50												
SOBRA, EM pol		8,0	9,75	22,5	0,25	1,0	0,0	2,75	2,25	22,75	31,0	86,5

Capítulo 7

CONSIDERAÇÕES FINAIS

Os exemplos mostrados permitem extrair algumas conclusões sobre a performance do algoritmo proposto neste trabalho, o qual sob a influência das diversas variáveis que compõem este tipo de problema, possibilitou comprovar alguns comportamentos evidenciados em outros trabalhos afins. Objetivou-se também mostrar e reforçar a importância prática associada ao algoritmo desenvolvido.

7.1 CONCLUSÕES

O Problema do Corte Industrial apresenta uma forte sensibilidade às variações nos valores de suas variáveis principais ou seja, dados de entrada do problema, como: comprimento nominal da matéria-prima, comprimento e quantidade de cada uma das peças a serem cortadas e o número de comprimentos distintos a serem cortados. Os exemplos mostrados no Capítulo 6 levam a comprovar que:

1. Quando não existe uma grande variação nos valores dos comprimentos e nas quantidades de peças a serem cortadas, observa-se uma periodicidade de ocorrência dos valores de sobra mínima total. Esta periodicidade é observada em intervalos de variação curtos. Isso foi verificado tanto na variação do comprimento nominal da matéria-prima (ver 6.2.2) quanto na variação da quantidade de peças a serem cortadas (ver 6.2.1). Já no caso onde diversidade dos valores das variáveis é maior, esse comportamento periódico fica menos evidente, embora ainda se perceba que os períodos de ocorrência dos valores de sobra mínima ocorram em intervalos de variação um pouco maior, conforme se constata no exemplo extraído do trabalho de Gilmore e Gomory [18] (ver 6.3).

2. O fator periodicidade permite ainda sugerir tamanhos de pedidos ótimos ao sistema produtivo, pois o parcelamento de um pedido em outros de menor tamanho mostrou ser mais dispendioso, isto é, corta-se mais matérias-primas para atender o mesmo pedido (ver Tabela 6-8).
3. O tempo de processamento para diferentes tamanhos de pedidos foi bastante satisfatório do ponto de vista prático. Os exemplos foram processados num computador IBM-PC compatível, equipado com um processador Pentium Intel 133 MHz. A disponibilidade de memória RAM foi de 64 MB. O mesmo exemplo analisado por Gilmore e Gomory [18] foi reproduzido aqui pelo algoritmo *XCut1D* e os tempos de processamento variaram de acordo com o gráfico da Figura 7-1. O maior tempo computacional ocorreu quando o comprimento da matéria-prima foi de 326,00 pol, gastando 90 s de processamento. A causa mais determinante no desempenho computacional do algoritmo está vinculada ao número de Problemas de Knapsack a serem resolvidos para se achar o padrão maximal em cada coluna gerada da matriz de padrões de corte (*matriz A da formulação*). Da mesma forma, a performance de solução do Problema de Knapsack está vinculada à performance do algoritmo de enumeração Branch-and-Bound.
4. Embora o algoritmo *XCut1D* implementado nesse trabalho não garanta a otimalidade da solução, os resultados obtidos no encaixe de peças para problemas unidimensionais foram bastante satisfatórios.
5. O algoritmo *XCut1D* constitui-se numa ferramenta auxiliar na decisão para escolher um comprimento nominal de matéria-prima ou atender um pedido de corte que resulte numa economia considerável e relevante do ponto de vista econômico.
6. A possibilidade de se poder simular situações diversas de corte, capacita o analista de produção em obter uma solução conveniente e adequada dentro das necessidades de peças estipuladas.

Variação do tempo de processamento com o comprimento da matéria-prima

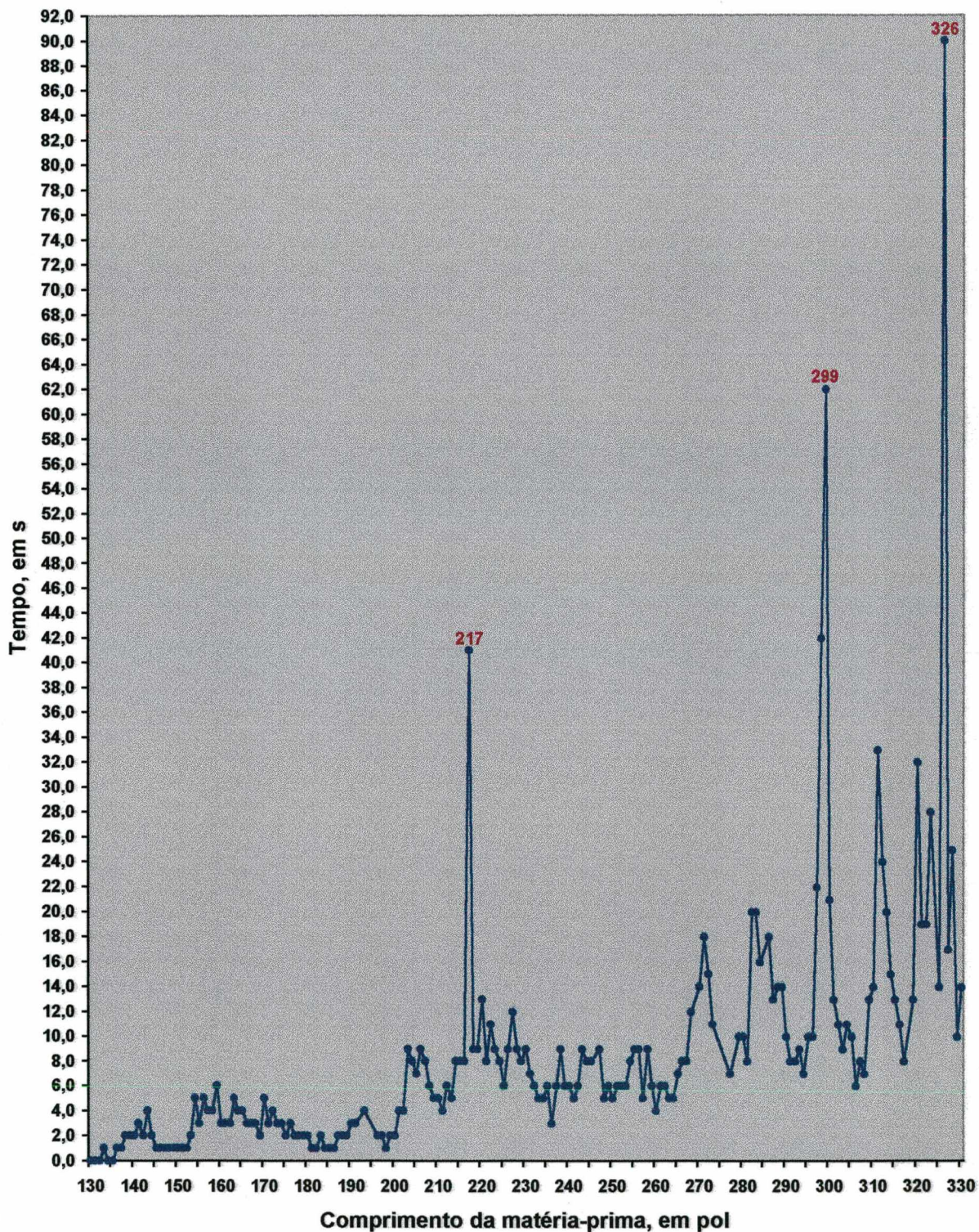


Figura 7-1: Desempenho computacional do algoritmo XCut1D

7.2 SUGESTÕES PARA TRABALHOS FUTUROS

7.2.1 Problema com Diferentes Comprimentos de Matéria-prima

A formulação proposta nesse trabalho contempla apenas os casos onde se tem apenas um comprimento nominal de matéria-prima disponível para o corte. Toda a implementação do algoritmo de otimização foi baseada nesta linha de solução do Problema do Corte Industrial, isto é, a cada solução obtida pelo algoritmo ele usa apenas um comprimento.

Considerando-se como uma extensão do algoritmo, pode-se propor e sugerir uma formulação do Problema do Corte Industrial que englobe casos práticos onde são disponíveis mais de um comprimento nominal de matéria-prima. Ou seja, tem-se comprimentos L_1, L_2, \dots, L_k disponíveis para o corte e para cada comprimento um custo c_1, c_2, \dots, c_k correspondente. O objetivo seria atender a um pedido de peças tal que a condição de minimização dos custos seja satisfeita.

Analiticamente, tem-se:

$$\text{minimizar } \sum_{j=1}^n c_j \cdot x_j,$$

$$\text{sujeito a } \sum_{j=1}^n a_{ij} \cdot x_j \geq N_i, \text{ com } i = 1, 2, \dots, m;$$

$$\text{e } \sum_{i=1}^m l_i \cdot a_{ij} \leq L_k, \text{ com } k = 1, 2, \dots, k.$$

Neste caso, tem-se que resolver k Problemas de Knapsack da forma

$$\text{maximizar } z = \sum_{i=1}^n a_i \cdot y_i > c_k$$

$$\text{sujeito a } \sum_{i=1}^m l_i \cdot a_{ij} \leq L_k.$$

O vetor-coluna \mathbf{a} que apresentar a maior diferença z_k^* ótimo menos o custo c_k do correspondente comprimento L_k deverá ser o padrão de corte a entrar na matriz \mathbf{A} .

7.2.2 Encaixe de Retângulos em Corte Tipo Guilhotina

Um tipo de problema, em que se poderia utilizar a rotina de Knapsack implementada neste trabalho, seria o do encaixe de retângulos em corte guilhotina. Este tipo de corte já foi definido anteriormente na seção 2.2 (Ver Figura 2-9).

Considere uma região contínua de área retangular $W \times L$, a qual deverá ser cortada em peças retangulares menores $w_i \times l_i$, $i=1, 2, \dots, m$; de forma a atender uma demanda de peças N_i . Na Figura 7-2, tem-se que:

L , representa o comprimento da área de encaixe;

W , representa a largura da área de corte;

l_i , representa o comprimento da peça i ;

w_i , representa a largura da peça i .

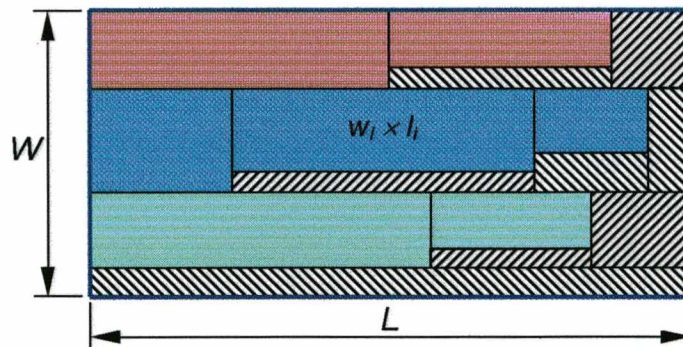


Figura 7-2: Corte tipo guilhotina

A solução desse problema pode ser dividida em dois passos:

- (i). Resolver o problema unidimensional em L .

Para cada uma das larguras de peças w_i , obtém-se o padrão maximal que encaixa os retângulos numa faixa retangular dada por $w_i \times L$. Os retângulos pertencentes à lista de peças em cada iteração i , são tais que $w_j \leq w_i$. Para tanto, ordenam-se as larguras em ordem crescente e resolve-se m diferentes Problemas de Knapsack dado por:

$$\begin{aligned} &\text{maximizar } \sum_{i=1}^m c_i \cdot a_i, \\ &\text{sujeito a } \sum_{i=1}^m l_i \cdot a_i \leq L, \\ &a_i \in \mathbb{Z}_+, i = 1, 2, \dots, m. \end{aligned}$$

onde:

a_i , representa o número de vezes que a peça $w_i \times l_i$ repete-se no padrão maximal obtido;

(ii). Resolver o problema unidimensional em W .

Utilizando os m padrões de corte gerados no passo anterior, pode-se obter o padrão maximal resolvendo mais um Problema de Knapsack, dado por:

$$\begin{aligned} &\text{maximizar } \sum_{i=1}^m c_i \cdot b_i, \\ &\text{sujeito a } \sum_{i=1}^m w_i \cdot b_i \leq W. \end{aligned}$$

onde:

b_i , representa o número de vezes que a peça $w_i \times l_i$ repete-se no padrão maximal obtido;

No passo (i), percebe-se que o resultado são faixas retangulares constituídas de peças encaixadas dentro do comprimento L da área retangular. No passo (ii), as faixas geradas no passo (i) agora são encaixadas na dimensão W da área retangular.

Com algumas modificações no código do algoritmo *XCut1D* e utilizando as demais rotinas já implementadas, pode-se resolver os problemas unidimensionais acima, e obter uma solução para problema do encaixe de retângulos.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] ARENALES, Marcos; MORABITO, Reinaldo. **An AND/OR-Graph Approach to the Solution of Two-dimensional Non-guillotine Cutting Problems.** European Journal of Operational Research, v. 84, p. 599-617, 1995.
- [2] BEASLEY, J. E.. **Algorithms for Unconstrained Two-dimensional Guillotine Cutting.** Journal of the Operational Research, v. 36, n. 4, p. 297-306, 1985.
- [3] BEASLEY, J. E.. **An Exact Two-dimensional Non-guillotine Cutting Tree Search Procedure.** Operations Research, v. 33, p. 49-64, 1985.
- [4] CAMPELLO, R. Eduardo; MACULAN, Nelson. **Algoritmos e Heurísticas – Desenvolvimento e Avaliação de Performance.** Niterói – RJ: Editora da Universidade Federal Fluminense, 1994.
- [5] CARVALHO, J. M. Valério de; RODRIGUES, A. J. Guimarães. **An LP-based Approach to a Two-stage Cutting Stock Problem.** European Journal of Operational Research, v. 84. p. 580-589, 1995.
- [6] CHVÁTAL, Vašek. **Linear Programming.** New York: W. H. Freeman and Company, cap. 13 - The Cutting-Stock Problem, p. 195-212, 1983
- [7] CUNHA, Raimundo R. M.; DIAS, Altamir. **Desenvolvimento de um Sistema Computacional para Solução do Problema de Recorte Unidimensional e Bidimensional de Peças.** Anais do III Congresso Ibero-americano de Engenharia Mecânica – CIDIM, Havana – Cuba, Setembro – 1997.
- [8] CUNHA, Raimundo R. M.; DIAS, Altamir. **Implementação de um Sistema Computacional para Otimização do Corte de Retângulos.** Anais do XIV Congresso Brasileiro de Engenharia Mecânica – COBEM, Bauru – SP, Dezembro – 1997.

- [9] Curso de Pós-graduação em Engenharia Mecânica da Universidade Federal de Santa Catarina. **Normas para Apresentação de Dissertações e Teses.**
- [10] DAZA, Víctor P.; ALVARENGA, Arlindo G.; DIEGO, José de. **Exact Solutions for Constrained Two-dimensional Cutting Problems.** European Journal of Operational Research, v. 84, p. 633-644, 1995.
- [11] DIAS, Altamir. **Encaixe Geral de Figuras Planas.** Rio de Janeiro, 1991. Tese (Doutorado em Engenharia Mecânica) - Departamento de Engenharia Mecânica, Pontifícia Universidade Católica – RJ.
- [12] DEGREAVE, Z.; BEMELMANS, R.. **Management Science Cases - Cutting Stock Problem at Metal Aarschot.** Department of Applied Economic Sciences, Katholieke Universiteit Leuven. http://www.econ.kuleuven.Ac.be/e_frameset.htm
- [13] DYCKHOFF, Harald; SCHEITHAUER, Guntram; TERNO, Johannes. **Cutting and Packing: An Annotated Bibliography.** Department of Mathematics, Dresden University of Technology, MATH-NM 08, April -1996.
- [14] DYCKHOFF, Harald; **A Typology of Cutting and Packing Problems.** European Journal of Operational Research, v. 44, p. 145-159, 1990.
- [15] FARLEY, A. A.. **Practical Adaptations of the Gilmore-Gomory Approach to Cutting Stock Problems.** OR Spektrum, v. 10, p. 113-123, 1988.
- [16] GAU, T.; WÄSCHER, G.. **CUTGEN1: A Problem Generator for the Standard One-dimensional Cutting Stock Problem.** European Journal of Operational Research, v. 84, p. 572-579, 1995.
- [17] GILMORE, P. C.; GOMORY, R. E.. **A Linear Programming Approach to the Cutting-Stock Problem.** Operations Research, v. 9, p. 849-859, 1961.
- [18] GILMORE, P. C.; GOMORY, R. E.. **A Linear Programming Approach to the Cutting-Stock Problem - Part II.** Operations Research, v. 11, p. 863-888, 1963.

- [19] GILMORE, P. C.; GOMORY, R. E.. **Multistage Cutting Stock Problems of Two and More Dimensions**. *Operations Research*, v. 13, p. 94-120, 1965.
- [20] GILMORE, P. C.; GOMORY, R. E.. **The Theory and Computation of Knapsack Functions**. *Operations Research*, v. 14, p. 1045-1074, 1966.
- [21] GUREWICH, Ori; GUREWICH, Nathan. **Teach yourself Visual C++ 4 in 21 days**. Indianapolis: SAMS Publishing, 1ª edição, 1996.
- [22] HAESSLER, R. W.; SWEENEY, P. E.. **Cutting Stock Problems and Solution Procedures**. *European Journal of Operational Research*, v. 54, p. 141-150, 1991.
- [23] LINS, Sóstenes. **CONJUG: Um Sistema para Diminuir Perdas no Corte Industrial de Chapas, Barras e Bobinas**. Proposta de trabalho para o XXI Congresso Nacional de Informática, 1988.
- [24] MONTENEGRO, Fernando; PACHECO, Roberto. **Orientação a Objetos em C++**. Rio de Janeiro: Editora Ciência Moderna Ltda, 1994.
- [25] MURTY, Katta G.. **Linear Programming**. New York: John Wiley & Sons Inc., 1983.
- [26] NICA, V.. **General Counterexample to the Integer Round-up Property**. *Dep. Econ. Cyb., Academy of Econ. Studies*, Bucharest, 1994.
- [27] OLIVEIRA, José F.; FERREIRA, José S.. **An Improved Version of Wang's Algorithm for Two-dimensional Cutting Problems**. *European Journal of Operational Research*, v. 44, p. 256-266, 1990.
- [28] RAO, Singiresu S.. **Engineering Optimization – Theory and Practice**. New York: John Wiley & Sons, Inc., 3ª edição, 1996.
- [29] RIEHME, Jan; SCHEITHAUER, Guntram; TERNO, Johannes. **The Solution of Two-stage Guillotine Cutting Stock Problems Having Extremely Varying Order Demands**. *European Journal of Operational Research*, v. 91, p. 543-552, 1996.

- [30] SAATY, Thomas L.. **Optimization in Integers and Related Extremal Problems**. New York: McGraw-Hill Book Company, cap. 5 - The Knapsack Problem, p. 211-215, 1970.
- [31] SALKIN, Harvey M.. **Integer Programming**. Cleveland: Addison-Wesley Publishing Company, cap. 10 - The Knapsack Problem, p. 383-438, 1975.
- [32] SCHEITHAUER, Guntram. **On the MAXGAP Problem for Cutting Stock Problems**. J. Inf. Proc. Cyb. EIK 30, p. 111-117, 1994.
- [33] SCHEITHAUER, Guntram; TERNO, Johannes. **About the Gap between the Optimal Values of the Integer and Continuous Relaxation One-dimensional Cutting Stock**. Oper. Research. Proc, Springer-Verlag Berlin Heidelberg, p. 439-444, 1991.
- [34] SCHEITHAUER, Guntram; TERNO, Johannes. **A Branch&Bound Algorithm for Solving One-dimensional Cutting Stock Problems Exactly**. Aplicationes Mathematicae, v. 23, p. 151-167, 1995.
- [35] SCHEITHAUER, Guntram; TERNO, Johannes. **The Modified Integer Round-up Property of the One-dimensional Cutting Stock Problem**. European Journal of Operational Research, v. 84, p. 562-571, 1995.
- [36] SHAMMAS, Namir C.. **C/C++ Mathematical Algorithms for Scientists and Engineers**. New York: McGraw-Hill Inc., 1995.
- [37] STROUSTRUP, Bjarne. **The C++ Programming Language**. New Jersey: Addison-Wesley Publishing Company, 2ª edição, 1994.
- [38] Universidade Federal do Paraná – UFPR – Biblioteca Central. **Normas para Apresentação de Trabalhos**. Curitiba: Editora da UFPR, parte 6: Referências Bibliográficas, 5ª edição, 1995.
- [39] VINADÉ, Cesar A. C.. **Problema de Encaixe de Figuras Não Convexas em um Contorno Não Convexo**. Florianópolis, 1997. Dissertação (Mestrado em Engenharia Mecânica) – Departamento de Engenharia Mecânica, Universidade Federal de Santa Catarina – SC.

-
- [40] WANG, P. Y.. **Two Algorithms for Constrained Two-dimensional Cutting Stock Problems.** Operations Research, v. 31, p. 573-586, 1983.
- [41] WÄSCHER, G; GAU, T.. **Generating Almost Optimal Solutions for the Integer One-dimensional Cutting Stock Problem.** Artigo 94/06, TU Braunschweig, 1994.

APÊNDICES

Algumas considerações que nortearam a implementação computacional do algoritmo de otimização *XCut1D* são importantes de serem mencionadas neste trabalho. Elas servirão como fonte de informação e de dicas úteis, tanto do ponto de vista de implementação computacional como também de ajuda para se entender a estrutura lógica de solução do Problema do Corte Industrial. E até mesmo, para auxiliar em outras implementações semelhantes ou de extensões destas, as quais no futuro possam ser usadas na solução de outros problemas.

Nos apêndices seguintes, fazem-se comentários sobre estas questões e também são mostrados os respectivos fluxogramas das rotinas implementadas na solução do Problema do Corte Industrial.

Apêndice A. TÓPICOS DE PROGRAMAÇÃO E IMPLEMENTAÇÃO COMPUTACIONAL

A preocupação na implementação do algoritmo se dividiram em:

1. Escolher uma ferramenta de desenvolvimento computacional para auxiliar na montagem da interface visual para o código do algoritmo.

No início do trabalho, dispúnhamos de duas opções: a ferramenta matemática Matlab v5.1, da *Mathworks Inc.*; e o ambiente Visual C++ v4.0 da *Microsoft Corp.*.

Dentre os vários módulos de implementação computacional para fins numéricos, o Matlab dispõe de dois módulos adequados para o que se pensava implementar no algoritmo aqui proposto. Um deles, denominado de *Optimization Toolbox*, constitui-se de códigos implementados dos principais métodos de otimização, e os quais podem ser adequados à solução de problemas específicos, a exemplo do Problema do Corte Industrial. Um outro módulo, denominado de *Visualization Toolbox*, oferece as ferramentas necessárias à criação de ambientes interativos visuais.

No caso do Visual C++, dispõe-se de um ambiente de edição integrado com um compilador C++ e a biblioteca de funções de objetos, denominada *Microsoft Foundation*

Class – MFC, com todas as ferramentas necessárias para criação de interface interativas visuais padrão compatível com Windows 95/98.

A definição da escolha favorável ao ambiente do Visual C++ foram muito mais por motivos externos à solução do problema do corte. Os motivos estão comentados a seguir:

- *Portabilidade do código*: o Matlab não gera um programa executável independente do seu ambiente. A idéia, embora ainda não concretizada neste trabalho, é criar um sistema computacional maior que possa ser executado independentemente de qualquer outro ambiente.
- *Compatibilidade do código*: embora o Matlab permita a criação de interfaces visuais interativas, as mesmas possuem um aspecto visual diferente do apresentado por uma programa inteiramente compatível com a plataforma Windows 95/98.

Ou seja, dentre as opções, o Visual C++ satisfaz plenamente esses requisitos.

As rotinas propostas para solução do Problema do Corte Industrial foram implementadas em compilador C++ e foram agrupadas em seis arquivos fontes:

XCut1D.cxx: É o arquivo que guarda rotina principal de solução. Ele chama as demais rotinas que solucionam o Problema do Corte Industrial.

InitialSolution.cxx: É o arquivo da rotina que gera uma solução inicial real.

FFDSolution.cxx: É o arquivo da rotina que gera uma solução inicial inteira.

GaussJordan.cxx: É o arquivo da rotina que soluciona o sistema de equações lineares.

BranchBound.cxx: É o arquivo da rotina que soluciona o Problema de Knapsack.

Sort.cxx: É o arquivo da rotina que faz o ordenamento dos valores de algumas variáveis.

Arrays.cxx: É o arquivo das classes de matrizes e vetores dinâmicos.

2. Desejava-se implementar um código rápido, ou seja, de desempenho computacional que solucionasse instâncias do Problema do Corte Industrial dentro de um tempo considerado prático.

Para conseguir este intento, atacou-se o problema de duas formas:

- Por se tratar de um problema interativo, incorporamos ferramentas de geração de soluções iniciais para o problema. A idéia era reduzir o número de iterações. Por isso foram implementadas as rotinas Initial-Solution e FFD-Solution. Ainda na

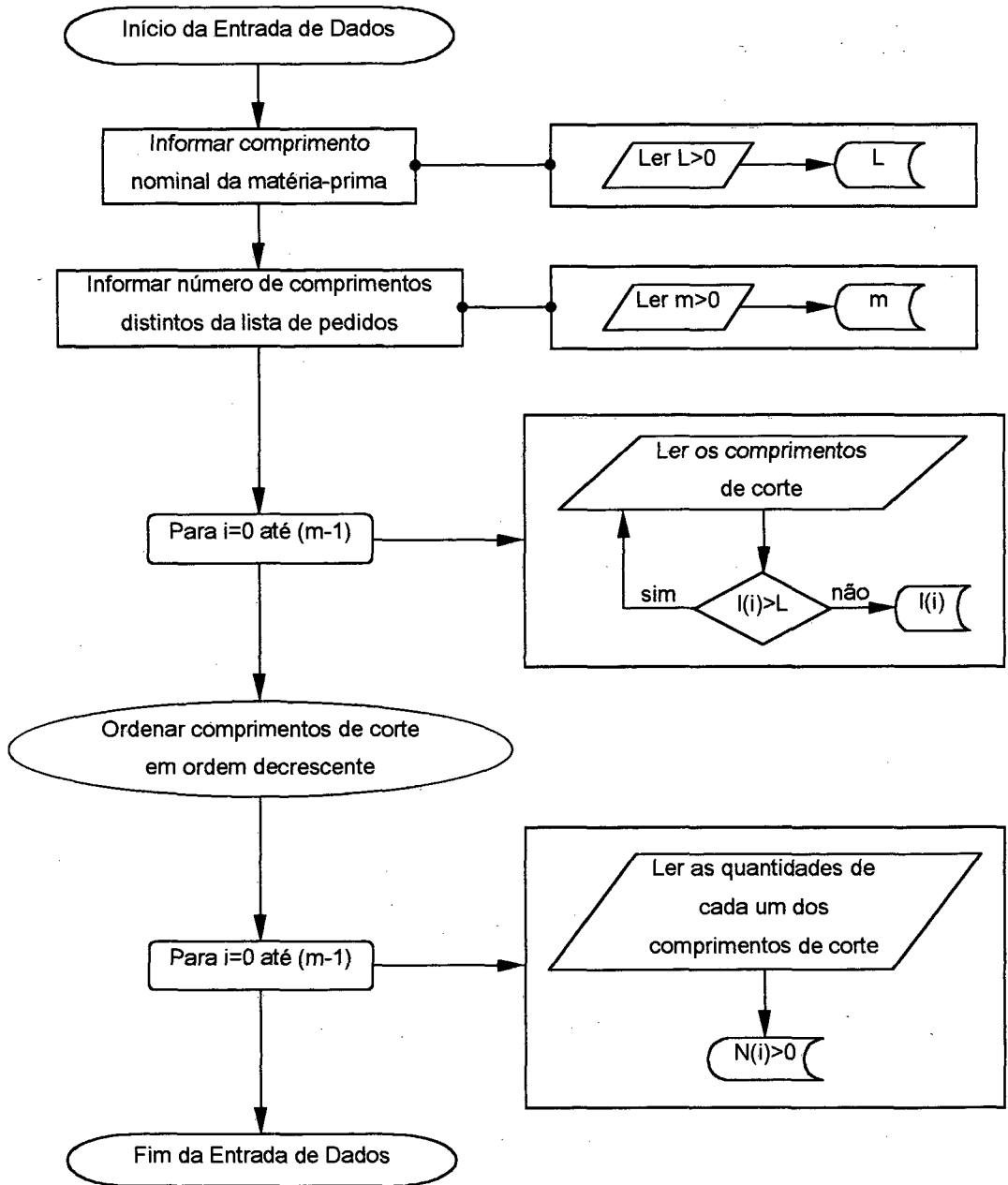
questão das iterações, utilizou-se o algoritmo de Branch-and-Bound na solução do Problema de Knapsack pois é sabido que o mesmo é cerca de 5 vezes mais rápido quando comparado a uma formulação usando Programação Dinâmica [6][18].

- Na parte computacional, foi implementada uma classe para geração de matrizes e vetores com endereçamento por ponteiros dinâmicos. Objetivou-se com isso, além de tornar mais rápido o acesso à memória, também otimizar a memória disponível do computador, alocando e desalocando espaço quando necessário. Os objetos e métodos estão no arquivo fonte *Arrays.cxx*. Na questão do ordenamento decrescente, imposto aos valores do comprimento das peças a serem cortadas e da variável de eficiência (c_i / a_i) no Problema de Knapsack, adotou-se o algoritmo Quicksort, o qual apresenta o melhor desempenho computacional dentre os algoritmos de ordenamento existentes.

Embora não se possa quantificar o quão mais lento o código tornar-se-ia caso as medidas acima não fossem tomadas, acredita-se que o tempo de processamento computacional teria sido bem maior do que o verificado nos testes realizados aqui neste trabalho.

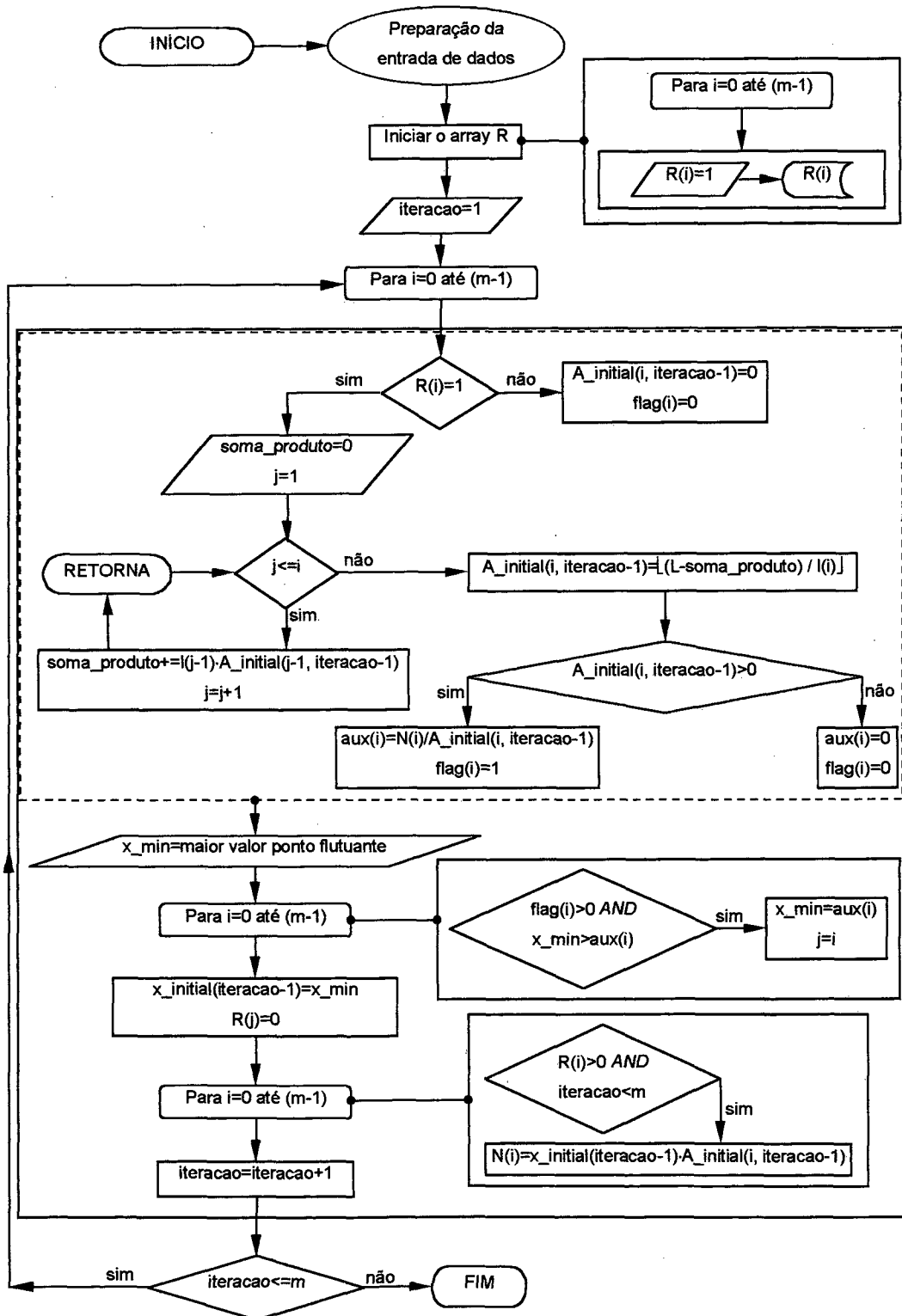
Nos apêndices a seguir, são mostrados os fluxogramas de todos os códigos dos arquivos, citados anteriormente, e utilizados neste trabalho para resolver o Problema do Corte Industrial descrito no Capítulo 2, e cuja definição e formulação estão descritas na seção 5.1.

Apêndice B. FLUXOGRAMA DA ENTRADA DE DADOS

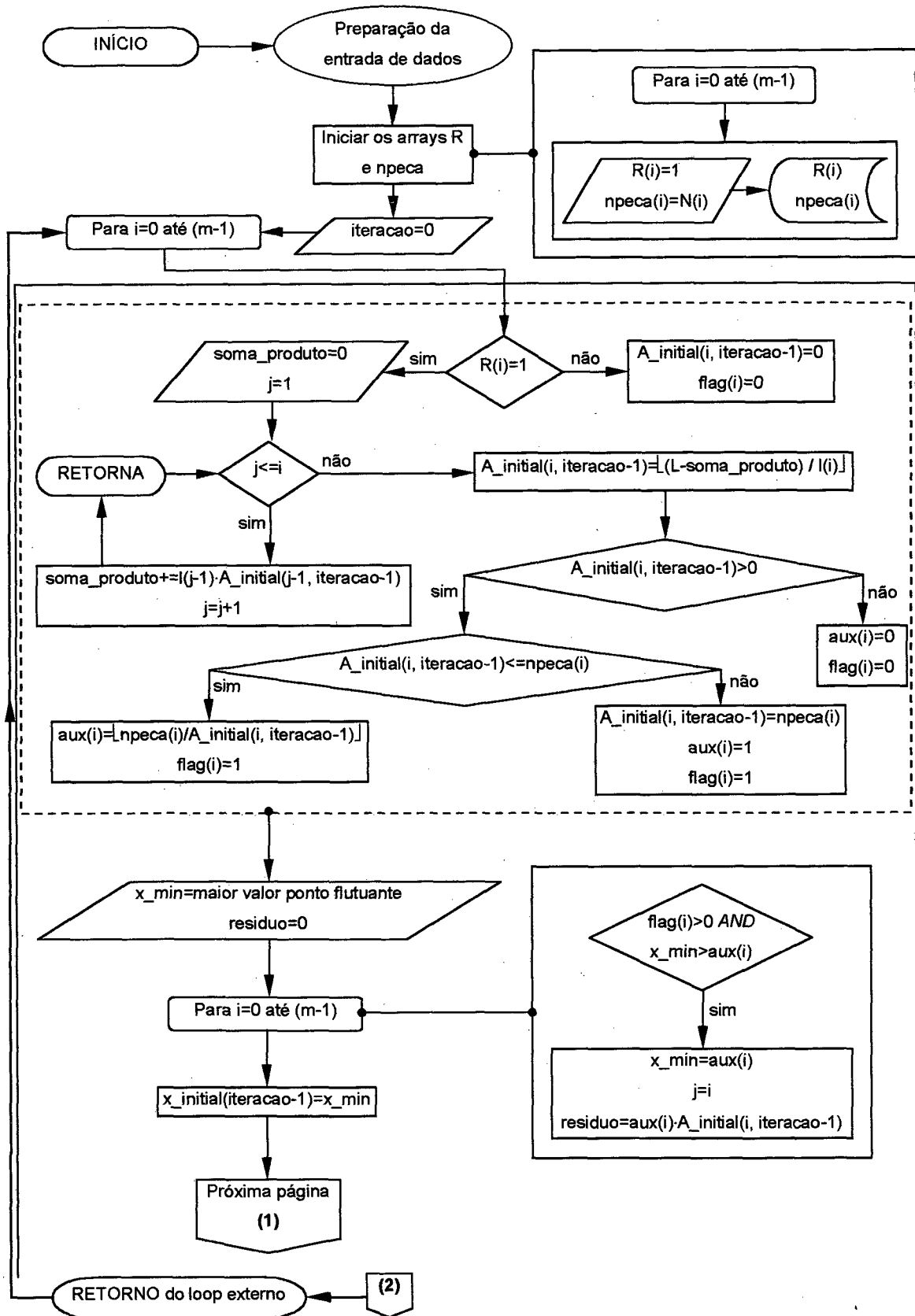


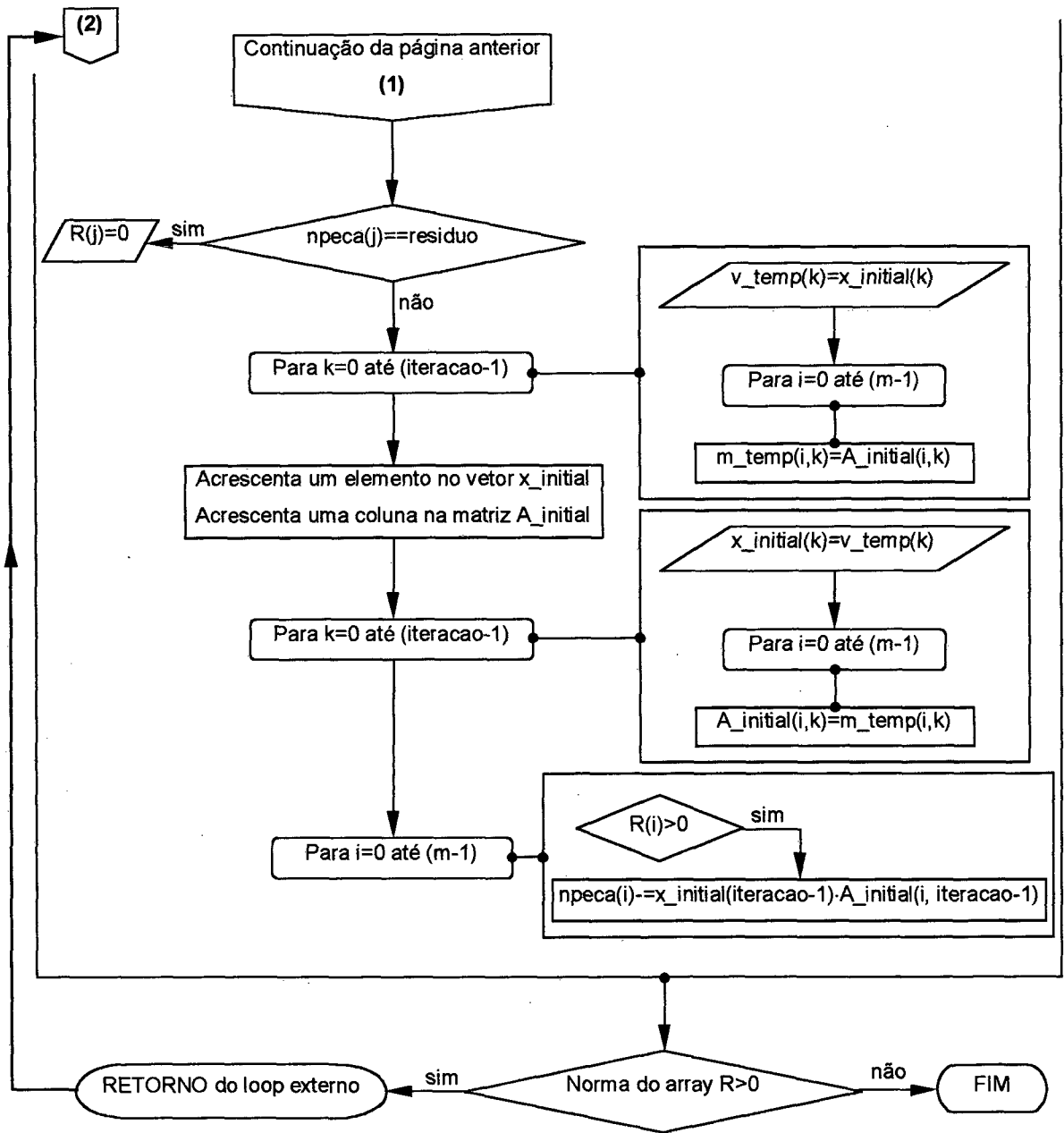
Apêndice C. FLUXOGRAMAS DOS ALGORITMOS DE SOLUÇÃO INICIAL

C.1 Algoritmo Initial-Solution

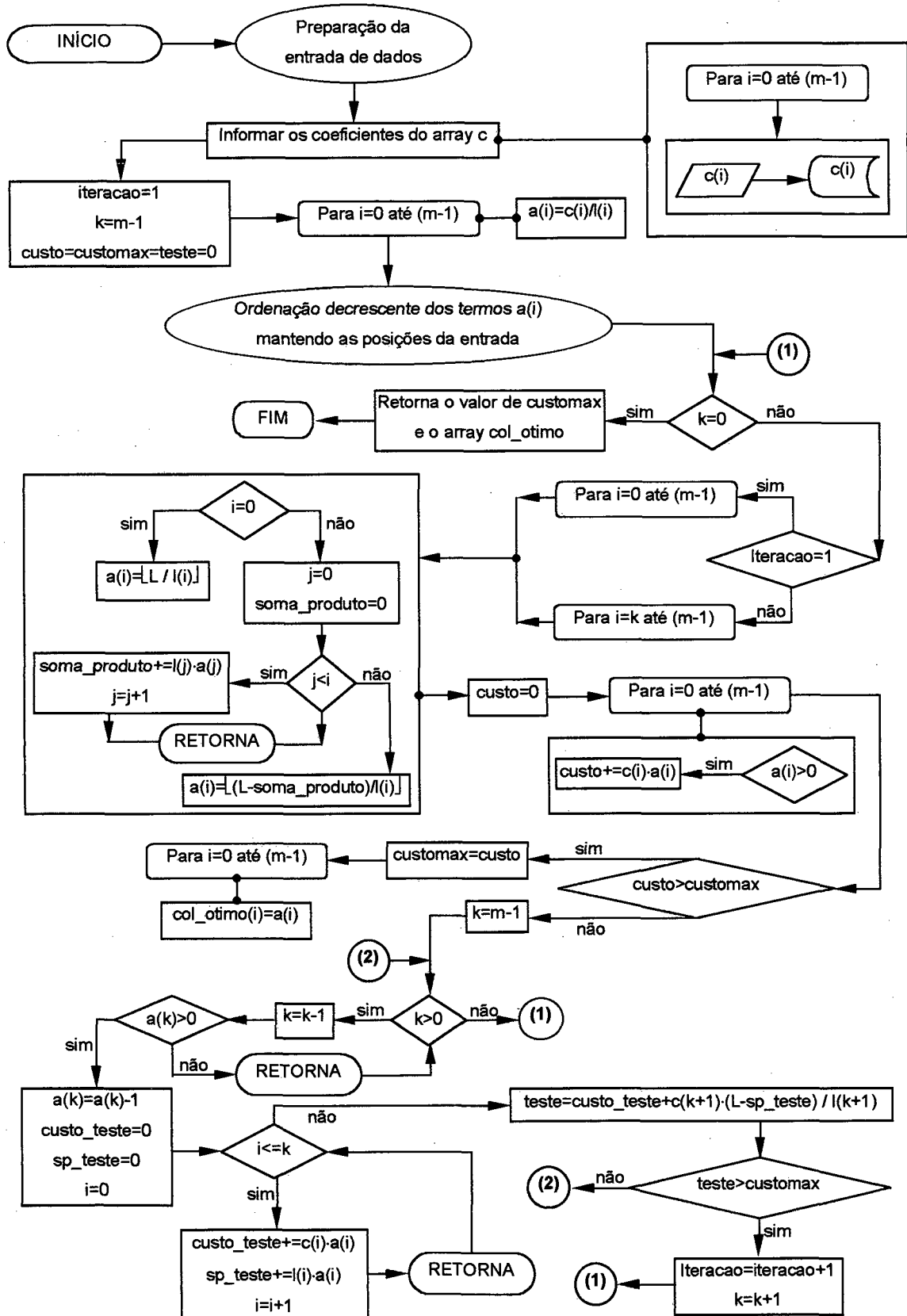


C.2 Algoritmo FFD-Solution





Apêndice D. FLUXOGRAMA DO ALGORITMO DE BRANCH-AND-BOUND



Apêndice E. FLUXOGRAMA DO ALGORITMO PRINCIPAL – XCut1D

