

**UNIVERSIDADE ESTADUAL DE PONTA GROSSA
DEPARTAMENTO DE MATEMÁTICA E ESTATÍSTICA**

DANIEL JOSÉ SCHULMEISTER

**UM ESTUDO SOBRE O PROBLEMA DE CORTE DE ESTOQUE
UNIDIMENSIONAL**

TRABALHO DE CONCLUSÃO DE CURSO

PONTA GROSSA

2021

DANIEL JOSÉ SCHULMEISTER

**UM ESTUDO SOBRE O PROBLEMA DE CORTE DE ESTOQUE
UNIDIMENSIONAL**

Trabalho de Conclusão de Curso apresentado(a) como requisito para obtenção do título(grau) de Bacharel em Matemática Aplicada, do Departamento de Matemática e Estatística, da Universidade Estadual de Ponta Grossa (UEPG).

Orientadora: Prof(a). Dr(a). Scheila Valechenski Biehl

PONTA GROSSA

2021

Dedico este trabalho à minha família.

AGRADECIMENTOS

A Deus, que me acompanha em todos os momentos.

À minha orientadora, Scheila Valechenski Biehl, pela orientação, apoio, compreensão, confiança depositada e amizade desenvolvidas durante os últimos anos.

Aos meus pais, irmãs e amigos, pelo carinho, incentivo e total apoio em todos os momentos da minha vida.

A todos os professores e colegas do departamento, que ajudaram de forma direta e indireta na conclusão deste trabalho.

Ao PICME e CNPq, por todas as oportunidades ofertadas a mim.

Enfim, a todos os que de alguma forma contribuíram para a realização deste trabalho.

"É justo que muito custe o que muito vale."

(D'ÁVILLA, Santa Teresa).

RESUMO

SCHULMEISTER, Daniel José. **Um estudo sobre o Problema de Corte de Estoque Unidimensional**. Orientadora: Scheila Valechenski Biehl. Ponta Grossa, 2021. Trabalho de Conclusão de Curso (Bacharelado em Matemática Aplicada) – Universidade Estadual de Ponta Grossa. Ponta Grossa, 2021.

Nas últimas décadas, é notável uma crescente competitividade no mercado industrial e, por essa razão, uma das principais metas traçadas pelas empresas é o desenvolvimento de estratégias de redução de custos. O Problema de Corte de Estoque Unidimensional faz parte da categoria dos chamados Problemas de Corte e Empacotamento que surgem em muitas indústrias, como por exemplo, nas que se destinam ao corte de bobinas de aço, bobinas de alumínio, chapas de metal, de vidro, papel, madeira, entre outros materiais. O objetivo de tal problema é cortar objetos maiores, de dimensões conhecidas, em itens de comprimentos menores especificados, buscando otimizar os seus processos, seja no melhor aproveitamento da matéria prima, seja na redução dos custos, além de satisfazer a demanda existente por cada item. Neste trabalho, realizamos um estudo teórico desse importante problema da literatura, apresentado como um modelo de otimização linear inteira, bem como dos métodos utilizados em sua resolução. Testes computacionais foram realizados considerando um exemplo didático, comparando a resolução do problema de corte de estoque unidimensional com o Método Simplex padrão e com o Método Simplex com Geração de Colunas (geração de padrões de cortes). Os resultados mostraram a eficiência da técnica de geração de padrões de cortes na redução de desperdícios do material utilizado, indicando uma área promissora a ser explorada em pesquisas futuras.

Palavras-chave: Problema de Corte de Estoque Unidimensional. Geração de Colunas. Branch-and-Bound. Problema da Mochila. Método Simplex.

ABSTRACT

SCHULMEISTER, Daniel José. **Um estudo sobre o Problema de Corte de Estoque Unidimensional**. Orientadora: Scheila Valechenski Biehl. Ponta Grossa, 2021. Trabalho de Conclusão de Curso (Bacharelado in Matemática Aplicada) – Universidade Estadual de Ponta Grossa. Ponta Grossa, 2021.

In the last decades, there has been a notable increase in competitiveness in the industrial market and, for this reason, one of the main goals set by companies is the development of cost reduction strategies. The One-Dimensional Cutting Stock Problem is part of the category of the so-called Cutting and Packing Problems that arise in many industries, such as those that are destined to cutting steel coils, aluminium, metal sheets, glass, paper, wood and other materials. The objective of this problem is to cut larger objects, of known dimensions, into items of specified shorter lengths, seeking to optimize their processes, either in the best use of the raw material or in the reduction of costs, in addition to satisfying the existing demand for each item. In this work, we carried out a theoretical study of this important problem in the literature, presented as a linear optimization model, as well as of the methods used in its resolution. Computational tests were performed considering a didactic example, comparing the resolution of the one-dimensional cutting stock problem with the standard Simplex Method and with the Simplex Method with Column Generation (generation of cutting patterns). The results showed the efficiency of the technique of generating cutting patterns in reducing waste of the material used, indicating a promising area to be explored in future research.

Keywords: One-dimensional Cutting Stock Problem. Column Generation. Branch-and-Bound. Knapsack Problem. Simplex Method.

LISTA DE ILUSTRAÇÕES

Figura 1 – Corte em uma barra de 170cm produzindo dois itens de 30cm e dois itens de 55cm.	17
Figura 2 – Corte em duas dimensões em uma placa de 49cm x 52cm produzindo dois itens com dimensões 29cm x 26cm e três itens de dimensões 20cm x 16cm. O retângulo em branco significa desperdício de material.	17
Figura 3 – Uma ilustração geral de um problema de empacotamento.	18
Figura 4 – Uma ilustração geral de um problema de um PCE 1,5 dimensional	18
Figura 5 – Alguns padrões de corte para a barra de 10 m	21
Figura 6 – Padrões de corte com sobras intrínsecas.	23
Figura 7 – Alguns padrões de corte para a barra de 170 cm	24
Figura 8 – Exemplos de conjuntos convexos e não convexos	32
Figura 9 – Ilustração de um hiperplano conforme a caracterização dada pela Eq. (4.1.12)	33
Figura 10 – Região factível do problema em questão (em vermelho) e sua solução ótima.	49
Figura 11 – Árvore do primeiro <i>branch</i> do Exemplo 4.4.1.	51
Figura 12 – Árvore do Exemplo 4.4.1.	52
Figura 13 – Estratégias de divisão	53

LISTA DE TABELAS

Tabela 1 – Padrões de Corte para o Exemplo (3.2.3).	26
Tabela 2 – Padrões de corte homogêneos para o problema.	57
Tabela 3 – Padrões que reduzem as perdas dos padrões homogêneos.	58
Tabela 4 – Padrões aleatórios.	59
Tabela 5 – Padrões gerados com o Método de Geração de Colunas.	63
Tabela 6 – Conclusão dos Resultados.	66
Tabela 7 – Padrões de corte gerados pelo método de Geração de Colunas para o problema do Capítulo 5 - melhores padrões para o problema relaxado destacados em cinza.	74

SUMÁRIO

1	INTRODUÇÃO	11
1.1	OBJETIVOS GERAIS E ESPECÍFICOS	12
1.2	ORGANIZAÇÃO DO TEXTO	13
2	REVISÃO DA LITERATURA	14
3	O PROBLEMA DE CORTE DE ESTOQUE UNIDIMENSIONAL (PCEU)	16
3.1	CLASSIFICAÇÃO DOS PROBLEMAS DE CORTE DE ESTOQUE	16
3.2	DEFINIÇÃO	20
3.3	FORMULAÇÃO MATEMÁTICA DO PCEU	25
4	MÉTODOS EMPREGADOS NA RESOLUÇÃO DO PCEU	28
4.1	MÉTODO SIMPLEX	28
4.1.1	Algumas definições e o Teorema Fundamental da Programação linear	28
4.1.2	Relações com Convexidade	32
4.1.3	O Método Simplex	36
4.1.4	O algoritmo Simplex	42
4.2	MÉTODO DE GERAÇÃO DE COLUNAS (MGC)	43
4.2.1	Introdução	43
4.2.2	Formulação Matemática	44
4.2.3	Algoritmo Geração de Colunas para um Problema de Corte de Estoque Unidimensional	45
4.3	PROBLEMA DA MOCHILA (KNAPSACK PROBLEM - KP)	46
4.3.1	Introdução	46
4.3.2	Formulação Matemática	46
4.4	<i>BRANCH-AND-BOUND</i>	47
4.4.1	Introdução	47
4.4.2	Formulação Matemática	48
4.4.3	Algoritmo <i>Branch-and-Bound</i>	53
5	RESULTADOS COMPUTACIONAIS	56
5.1	METODOLOGIA	56
5.2	DEFINIÇÃO DO PROBLEMA	56
5.2.1	Solução 1 - Padrões de Corte Homogêneos e Variações	57
5.2.2	Solução 2 - Padrões de Corte aleatórios e em quantidade maior	59
5.2.3	Solução 3 - Padrões gerados pelo Método de Geração de Colunas	60
6	CONSIDERAÇÕES FINAIS	67
	REFERÊNCIAS	69

APÊNDICES	73
APÊNDICE A – PADRÕES DE CORTE GERADO PELO MÉTODO DE GERAÇÃO DE COLUNAS	74

1 INTRODUÇÃO

Nos últimos sessenta anos, a partir dos trabalhos publicados por Gilmore e Gomory (1961), o Problema de Corte de Estoque Unidimensional (PCEU) tem sido tema de pesquisas em diversas áreas, devido a sua grande importância teórica e prática, uma vez que busca modelar e solucionar uma situação que pode ocorrer na etapa de produção de bens manufaturados. Grande parte dos estudos que são realizados têm como motivação principal a otimização do fator custo-produção encontrado em diversas indústrias, como, por exemplo, as têxteis, moveleiras, siderúrgicas, entre outras. Além disso, na área acadêmica, pode ser abordado em disciplinas de vários cursos, tais como ciência da administração, engenharia, ciência da informação e computação, matemática e pesquisa operacional.

De forma geral, o Problema de Corte de Estoque (PCE) consiste em cortar objetos maiores de dimensões definidas, disponíveis em estoque, em itens menores, de modo que seja satisfeita a demanda existente associada a cada tipo de item. O objetivo é encontrar a melhor opção dentre todas as combinações possíveis de corte, de modo a suprir a demanda e minimizar o desperdício de material. Existem diversas variações do PCE quanto à dimensão dos cortes e demais condições que podem ser adicionadas, relacionadas à demanda, ao estoque e até mesmo aos padrões de corte.

O PCE pode ser modelado como um problema de programação linear inteira (PPI), que é um caso particular dos problemas de programação linear (PPL). Um PPL é caracterizado por possuir uma função objetivo e restrições lineares. Além disso, em sua forma padrão, condições de não negatividade são impostas às variáveis. Semelhantemente, um PPI possui as mesmas características quanto a linearidade; no entanto, as variáveis do problema pertencem ao conjunto dos números inteiros, possibilitando, dessa maneira, aprimorar a representação de problemáticas reais.

O método de Geração de Colunas, desenvolvido por Gilmore e Gomory (1961), permite que uma solução exata seja obtida para o problema de programação linear relaxado, ou seja, quando não é imposta a condição de que as variáveis sejam inteiras (integralidade) e posteriormente podem ser usadas heurísticas de arredondamento para obtenção de uma solução viável inteira. Assim, quando a demanda é suficientemente alta, esse método se torna eficiente, pois as soluções obtidas estão tolerantemente "próximas" da solução ótima do problema original, reduzindo consideravelmente o desperdício de matéria-prima.

Grande parte deste trabalho teve ênfase no estudo dos fundamentos matemáticos da teoria que levaram ao desenvolvimento do método de solução para o problema de corte de estoque unidimensional pelo método de geração de colunas. Também para uma exploração didática, abordamos um problema numérico, utilizando o software IBM ILOG CPLEX Optimization Studio. Diferentes maneiras de resolução foram abordadas, com o intuito de identificar as desvantagens do método exaustivo em contraponto com a eficiência do método de geração de colunas.

A finalidade desta pesquisa é exploratória e explicativa, pois foi abordado um problema de otimização da literatura pouco conhecido na área acadêmica, com intuito de aprofundar e explorar o conhecimento acerca do tema. A abordagem da pesquisa é quantitativa e de natureza aplicada pois objetiva gerar conhecimentos para aplicações práticas, na busca por soluções para um problema específico. Os procedimentos utilizados nesta pesquisa foram do tipo revisão bibliográfica e estudo de caso de um problema da literatura.

1.1 OBJETIVOS GERAIS E ESPECÍFICOS

- O objetivo geral do presente trabalho é apresentar uma visão geral do Problema de Corte de Estoque Unidimensional (PCEU) e de um método exato de resolução.

Os objetivos específicos são:

- Estudo da classificação dos problemas de corte e da teoria necessária para sua compreensão.
- Apresentar a formulação matemática do PCEU.
- Compreensão dos conceitos, resultados principais e teoria que embasam os métodos utilizados para a resolução de um PCEU, tais como, Método Simplex, Teoria dos Conjuntos Convexos, Método de Geração de Colunas, o Problema da Mochila e o Método *Branch-and-Bound*.
- Análise de resultados computacionais e comparação entre as diferentes soluções obtidas ao resolver o exemplo numérico sem a utilização do método de geração de colunas e com a utilização de tal método.

1.2 ORGANIZAÇÃO DO TEXTO

No Capítulo 2, apresentamos uma revisão de literatura com algumas publicações relevantes relacionadas ao tema do trabalho.

No Capítulo 3, apresentamos o Problema de Corte de Estoque Unidimensional e identificamos sua classificação em relação aos demais problemas de corte e empacotamento. Além disso, é exposta sua formulação matemática e a definição de alguns elementos necessários para seu entendimento.

No Capítulo 4, apresentamos todas as ferramentas matemáticas necessárias para a resolução do PCEU: definições básicas de programação linear, o método Simplex, o método de geração de colunas, o problema da mochila e o método *Branch-and-Bound*.

No Capítulo 5, apresentamos os resultados de uma aplicação feita utilizando um exemplo numérico da literatura e por fim, no Capítulo 6, apresentamos as considerações finais e propostas de trabalhos futuros.

2 REVISÃO DA LITERATURA

O problema de corte de estoque unidimensional (PCEU) é um dos problemas clássicos da classe de problemas de corte. De maneira geral, por apresentarem muitas semelhanças, os problemas de corte e empacotamento (ou *Cutting and Packing - C&P*) são agrupados e englobam uma grande quantidade de problemas combinatoriais, modelos e algoritmos.

Em Dyckhoff *et al.* (1997), os autores apresentaram uma relação de problemas e publicações relevantes que podem ser considerados integrantes dos C&P. Alguns exemplos citados foram os problemas de corte de estoque e perda de corte; problemas de empacotamento, empacotamento em tiras, da mochila; problemas de carregamento de veículos, paletes, contêineres, de design, de layout, aninhamento e particionamento; problemas de orçamento de capital, da mudança, balanceamento de linha, alocação de memória e escalonamento de multiprocessadores. Segundo os autores, é possível englobar estes problemas, uma vez que todos têm como objetivo principal o arranjo geométrico eficiente de peças em domínios maiores.

Em Dyckhoff (1990), foi realizada uma análise profunda a respeito das propriedades particulares que cada C&P pode apresentar e exibida quatro categorias classificatórias: dimensão, forma de alocação das unidades, tipos de objetos e tipos de itens. Dessa maneira, foi possível distinguir noventa e seis tipos distintos de problemas. Complementando o trabalho citado, Wäscher *et al.* (2007) incluiu mais uma categoria denominada forma dos itens.

Os primeiros trabalhos publicados na área do C&P foram os de Kantorovich (1960) (publicado pela primeira vez em 1939, em russo) e de Brooks *et al.* (1940), porém o grande marco histórico se dá com Gilmore e Gomory (1961), quando estes descreveram o problema de corte de estoque unidimensional e apresentaram técnicas matemáticas capazes de o solucionar em tempo computacional factível. Isso foi possível graças à elaboração do método de geração de colunas, introduzido ao método simplex. Além disso, constataram a utilização do problema da mochila como etapa fundamental de resolução dos subproblemas presentes em cada etapa do processo.

Em 1963, Gilmore e Gomory (1963) elaboraram um novo método para o problema da mochila e apresentaram uma aplicação em uma indústria de papel. Em 1965, Gilmore e Gomory (1965) estenderam a dimensão dos objetos em estoque e trataram da resolução do problema de corte de estoque bidimensional.

No trabalho de Haessler (1980), o autor expõe algumas alterações no método inicial-

mente proposto por Gilmore e Gomory, quanto à escolha da solução inicial e dos padrões que passam a compor a matriz básica em cada iteração. A ideia central é tornar o problema da mochila mais restritivo, porém menos efetivo, com o intuito de reduzir problemas de arredondamento e mudança de padrões de corte.

No trabalho de Hinxman (1980), o autor revisou os diferentes métodos de resolução presentes na literatura, como os métodos exatos *Branch-and-Bound*, programação dinâmica e heurísticas. Outros autores que apresentaram heurísticas para o PCEU foram Stadtler (1990), Wäscher e Gau (1996), Limeira e Yanasse (2001), Poldi e Arenales (2003), onde abordaram métodos heurísticos para obtenção de soluções inteiras, redução de perda de materiais e/ou redução do número de padrões de corte.

Em 2000, Foerster e Wäscher (2000) apresentaram um método que visava a redução do número de padrões de corte a serem utilizados. Isso, segundo os autores, é uma característica importante, pois representa uma limitação física de muitas máquinas industriais. Os resultados computacionais se mostraram muito superiores aos anteriores presentes na literatura, porém com um custo computacional superior para algumas classes de problemas.

Nos trabalhos de Cherri (2006) e Rios (2017), foram apresentadas revisões da literatura e resoluções computacionais para uma variante do PCEU, denominado Problema de Corte de Estoque com Sobras Aproveitáveis (PCESA).

Em 2017, Marques *et al.* (2017) apresenta formulações alternativas para o problema da mochila, com base nas observações feitas por Gilmore e Gomory em seus trabalhos da década de 60. Modificações na função objetivo e restrições adicionais permitiram produzir em boa parte dos testes computacionais realizados, um decréscimo no número de padrões de corte utilizados e no tempo computacional envolvido.

É interessante destacar trabalhos publicados que evidenciam a recorrência do PCEU em aplicações práticas, como por exemplo, nas indústrias de alumínio (Stadtler (1990)), de aço (Carvalho e Rodrigues (1995)), de papel (Goulimis (1990)), vidro, plástico, têxtil, entre outras.

3 O PROBLEMA DE CORTE DE ESTOQUE UNIDIMENSIONAL (PCEU)

O Problema de Corte de Estoque Unidimensional foi abordado pela primeira vez no trabalho de Kantorovich (1960), no qual abordou modelos matemáticos de programação linear e métodos de resolução para o planejamento e programação da produção. Alguns anos depois, Gilmore e Gomory (1961), (1963), (1965) publicaram trabalhos nos quais era possível obter soluções para o PCEU com a utilização do método de geração de colunas juntamente ao método Simplex e com a formulação matemática do problema da mochila para os subproblemas presentes nas etapas de resolução.

O problema, na sua forma mais simples consiste na efetuação de cortes unidimensionais em objetos (barras) em estoque que possuem o mesmo comprimento. Além disso, os cortes produzem itens de tamanhos menores (peças), procurando satisfazer a demanda existente de cada tipo de item. O objetivo é então cumprir toda a demanda, cortando a menor quantidade possível de objetos em estoque.

A formulação acima de um PCEU pode sofrer alterações, quanto à dimensão, variedade dos comprimentos dos materiais em estoque, restrições adicionais às demandas, como limitantes inferiores ou superiores, custos adicionais e, até mesmo, no objetivo em questão, como por exemplo, a redução dos padrões de corte utilizados. Tais características foram categorizadas e classificadas em grupos, inicialmente por Dyckhoff (1990) e depois complementadas por Wäscher *et al.* (2007).

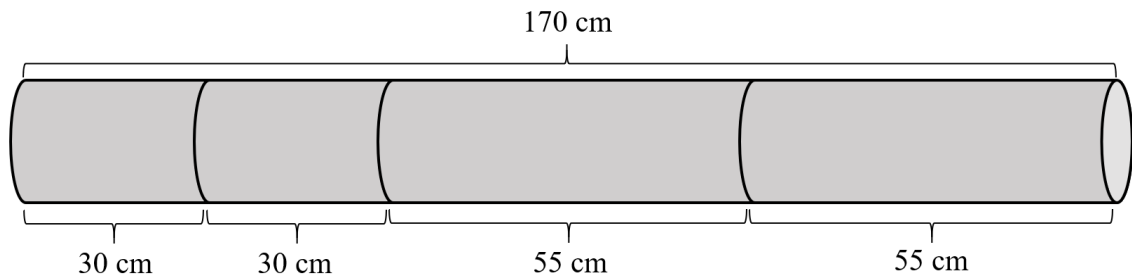
3.1 CLASSIFICAÇÃO DOS PROBLEMAS DE CORTE DE ESTOQUE

Nesta seção será abordada a classificação tipológica dada por Dyckhoff (1990) aos problemas de corte e empacotamento (*cutting and packing - C&P*), quanto à dimensão, tipo de atribuição, tipo de objeto e tipo de item. Também consideraremos a categoria acrescentada por Wäscher *et al.* (2007), que diz respeito à forma do item.

- **Dimensão:** esta característica classifica os problemas pela dimensão do corte que é considerada. Um C&E pode ser:
 - Unidimensional (1): neste caso os cortes serão feitos em apenas uma dimensão, ou seja, os itens produzidos possuirão comprimentos variáveis. Exemplos ocorrem nas

indústrias que produzem rolos de papel, bobinas de aço e de alumínio.

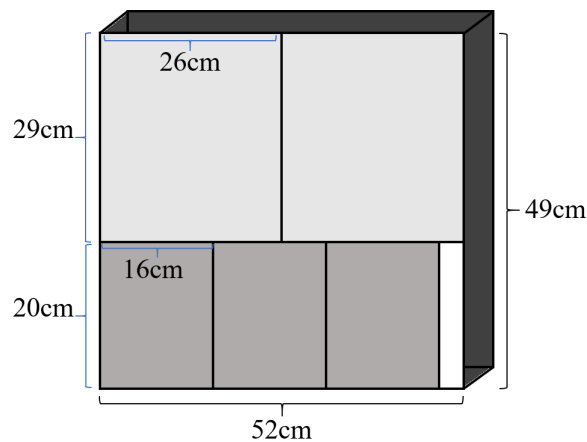
Figura 1 – Cortes em uma barra de 170cm produzindo dois itens de 30cm e dois itens de 55cm.



Fonte: Autoria própria.

- Bidimensional (2): cortes em duas dimensões, ou seja, alterações ocorrem no comprimento e largura dos itens. Exemplos são encontrados em indústrias de placas de vidro, madeira, etc.

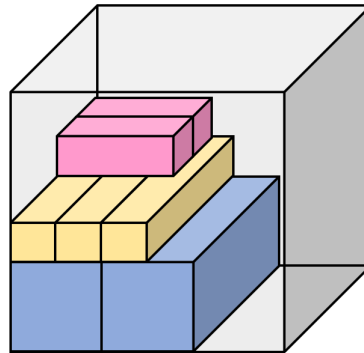
Figura 2 – Cortes em duas dimensões em uma placa de 49cm x 52cm produzindo dois itens com dimensões 29cm x 26cm e três itens de dimensões 20cm x 16cm. O retângulo em branco significa desperdício de material.



Fonte: Autoria própria.

- Tridimensional (3): quando três dimensões são consideradas: comprimento, largura e altura. Exemplos ocorrem nos problemas de empacotamento, que consistem em arranjar pequenos volumes em um volume maior.

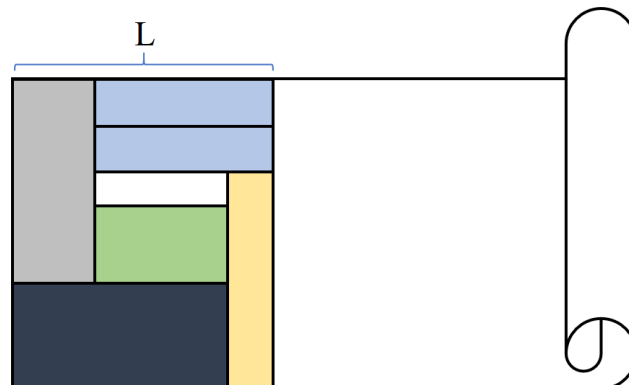
Figura 3 – Uma ilustração geral de um problema de empacotamento.



Fonte: Autoria própria.

- n -dimensional, com $n > 3$ (n): ocorre quando n dimensões são consideradas. Os problemas de alocação de tarefas se encaixam nessa categoria (Goldbarg e Luna (2005)). Os problema quadridimensionais podem ser interpretados como problemas de empacotamento com o tempo como quarta variável.
- 1,5 e 2,5 dimensionais: são problemas em que duas e três dimensões, respectivamente, são relevantes, porém uma delas é variável. O exemplo mais comum para o tipo 1,5 dimensional é o rolo de tecido que tem largura fixa e comprimento suficientemente grande para a confecção de roupas. Nesse caso, o objetivo é minimizar o comprimento L a ser cortado, como ilustra a figura abaixo:

Figura 4 – Uma ilustração geral de um problema de um PCE 1,5 dimensional



Fonte: Autoria própria.

Para a dimensão 2,5, um contêiner com altura e largura fixa e suficientemente comprido para comportar o volume da carga é um exemplo. Nessa situação, a figura 4 fornece uma visão lateral do problema, como apresenta Cherri (2006).

- **Forma de alocação das unidades:** essa característica é discorrida em Pileggi (2002) e exposta a seguir:
 - Seleção de unidades pequenas (em alemão, "*Belaproblem*") (B): Um conjunto dos itens menores é combinado para ocupar os objetos maiores, ou seja, as unidades maiores são insuficientes para alocar as menores. Um exemplo ocorre no Problema da Mochila.
 - Seleção de unidade grandes (em alemão, "*Verladeproblem*") (V): todos os itens menores são combinados para ocupar um conjunto de objetos maiores, ou seja, as unidades maiores são suficientes para alocar as menores. Um exemplo ocorre nos Problemas de Corte de Estoque.
- **Tipos de Objetos:** as unidades maiores podem ser divididas em:
 - Somente uma unidade (O): como é o caso do Problema da Mochila e do *Pallet Loading*.
 - Objetos de mesmo tamanho (I): como, por exemplo, nos problemas de empacotamento.
 - Objetos com tamanhos variados (D): como ocorre nos problemas de corte de estoque de peças com resíduos (sobras) de períodos anteriores.
- **Tipos de Itens:** as unidades menores podem ser divididas em:
 - Poucos itens com tamanhos distintos (F).
 - Muitos itens com tamanhos distintos (M).
 - Muitos itens com pequena variação do tamanho (R).
 - Itens com tamanhos iguais (C).

Com a categorização acima, criada por Dyckhoff (1990), é possível classificar os Problemas de Corte de Estoque Unidimensional como 1/ V/ I/ R, ou seja, unidimensional, com uma seleção das unidades maiores, cujo tamanho é fixo e os itens menores são em grande quantidade, possuindo pequena variação de tamanhos, em geral.

Quanto à forma dos itens introduzida por Wäscher *et al.* (2007), citamos

- **Forma dos itens:** permite classificar os problemas (C&P) em tipo básico, intermediário e refinado. Essa classificação é abordada por Rios (2017) e apresentada a seguir:

- Tipo básico: esse critério é obtido pelas combinações entre os tipos de itens e forma de alocações das unidades. Pode ser dividido em problemas com:
 - * Maximização de saída: o estoque dos objetos é limitado; como consequência, nem sempre é possível produzir todos os itens, de modo que o objetivo é a produção do máximo de itens possíveis.
 - * Minimização de entrada: em estoque há uma grande quantidade de objetos, de modo a produzir todos os itens. O objetivo é utilizar o mínimo possível de objetos.
- Tipo intermediário: obtidos por adicionar o critério de diversidade dos itens na classificação de problemas do tipo básico.
- Tipo refinado: considera, além dos critérios da classificação dos problemas do tipo intermediário, os critérios de dimensionalidade e da forma dos itens, regulares (itens convexos) ou irregulares.

E assim, com essa categorização, o Problema de Corte de Estoque Unidimensional em sua versão mais simples, aquele em que o estoque é ilimitado e os objetos (barras) possuem o mesmo tamanho, faz parte do conjunto dos chamados "Problemas de Corte de Estoque de tamanho único" (*Single Stock Size Cutting Stock Problem - SSSCSP*).

3.2 DEFINIÇÃO

Feita a apresentação das notações e classificações, passamos agora para a formulação matemática de um Problema de Corte de Estoque Unidimensional, bem como das definições necessárias para sua compreensão, baseando-se principalmente nos referenciais de Gilmore e Gomory (1961), Poldi e Arenales (2003) e Arenales *et al.* (2007).

Considere um conjunto de objetos em estoque de um determinado comprimento L e um conjunto de pedidos, com demanda conhecida d_i , $i = 1, \dots, m$ de itens de comprimentos l_i , $i = 1, \dots, m$, de modo que $l_i \leq L$. O objetivo então é determinar a produção das quantidades demandadas por cada item, a partir dos objetos em estoque, de maneira ótima, ou seja, gastando a menor quantidade de matéria prima possível.

Para que isso seja possível, é preciso determinar as melhores formas de se cortar os objetos em estoque, a fim de que as perdas de material sejam minimizadas. Para isso é preciso definir os chamados *padrões de corte*.

Definição 3.2.1 (*Padrão de Corte*). Um *padrão de corte* é uma maneira de como um objeto em estoque é cortado para a produção dos itens demandados. A um padrão de corte associamos um vetor m -dimensional que contabiliza os itens produzidos:

$$\mathbf{a}_j = (\alpha_{1j}, \alpha_{2j}, \dots, \alpha_{mj}), \quad j = 1, 2, \dots \quad (3.2.1)$$

em que α_{ij} é a quantidade de itens do tipo i no padrão de corte j . Vale destacar que a quantidade j de padrões de corte, em geral, é de uma ordem muito maior que m , o número de itens. Observe também que um vetor $\mathbf{a}_j = (\alpha_{1j}, \alpha_{2j}, \dots, \alpha_{mj})$ será um padrão de corte se e somente se satisfizer:

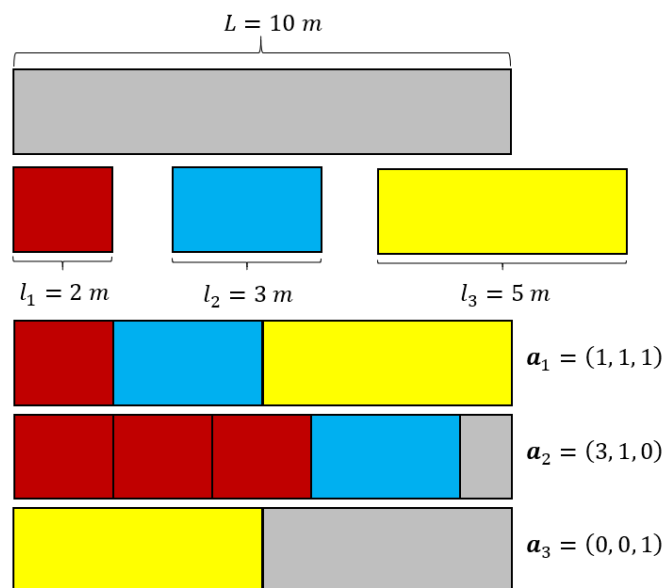
$$l_1\alpha_{1j} + l_2\alpha_{2j} + \dots + l_m\alpha_{mj} \leq L \quad (3.2.2)$$

$$0 \leq \alpha_{ij} \leq d_i, \quad \alpha_{ij} \in \mathbb{Z}_+, \quad i = 1, \dots, m \quad j = 1, 2, \dots \quad (3.2.3)$$

em que d_i é a demanda pelo item i .

Exemplo 3.2.1. Seja $L = 10$ m o tamanho das barras em estoque e as seguintes peças com seus respectivos comprimentos: $l_1 = 2$ m, $l_2 = 3$ m e $l_3 = 5$ m. Alguns padrões de corte possíveis estão representados na figura abaixo. O padrão \mathbf{a}_1 utiliza toda a barra sem produzir sobras. Já os padrões \mathbf{a}_2 e \mathbf{a}_3 geram desperdício de material, se utilizados. Observamos na prática o padrão \mathbf{a}_3 obviamente não é adequado, pois outras peças poderiam ser produzidas com a sobra resultante de 5 m.

Figura 5 – Alguns padrões de corte para a barra de 10 m



Fonte: Autoria própria.

Definição 3.2.2 (*Padrão de Corte Completo*). Consideremos $l_i \leq L$, $i = 1, \dots, m$, os tamanhos dos itens menores, e $l = \min_i l_i$ o menor entre eles. Um padrão de corte \mathbf{a}_j será dito *completo* quando

$$L - l_1\alpha_{1j} + l_2\alpha_{2j} + \dots + l_m\alpha_{mj} < l, \quad j = 1, 2, \dots, \quad (3.2.4)$$

ou seja, quando a sobra gerada pelo padrão de corte é menor que o menor tamanho de item a se produzir. Ou ainda, quando não é possível produzir nenhum item com a sobra gerada pelo padrão de corte \mathbf{a}_j . No Exemplo 3.2.1, os padrões de corte \mathbf{a}_1 e \mathbf{a}_2 são *completos*, enquanto que o padrão de corte \mathbf{a}_3 *não* é completo.

Para a formulação matemática de um problema de corte, utilizaremos apenas padrões de corte completos. Já em outros casos, como os problemas de corte de estoque com sobras aproveitáveis (PCESA), a utilização de padrões de corte que não são completos torna-se interessante para a construção de métodos de resolução (CHERRI, 2006; RIOS, 2017).

Definição 3.2.3 (*Perdas Intrínsecas*). Em situações reais, as serras que efetuarão os cortes nos objetos em estoque possuem uma determinada largura, normalmente na ordem de milímetros (mm). As sobras geradas pelo corte de cada serra são chamadas *perdas intrínsecas*.

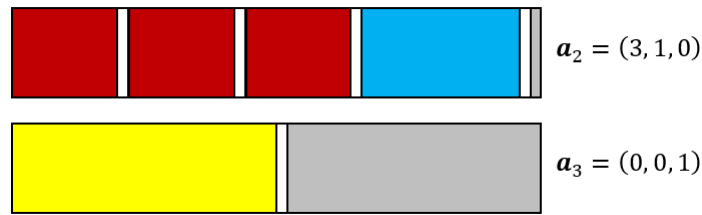
Quando consideradas as perdas intrínsecas causadas por serras de mesma largura σ , a Eq. (3.2.2), presente na Definição 3.2.1 de padrão de corte, é substituída por:

$$(l_1 + \sigma)\alpha_{1j} + (l_2 + \sigma)\alpha_{2j} + \dots + (l_m + \sigma)\alpha_{mj} \leq L + \sigma, \quad (3.2.5)$$

em que σ é a largura das serras. O comprimento l_i de cada item passa a ser $l_i + \sigma$, $i = 1, \dots, m$. Note que um termo σ também é acrescentado ao tamanho das barras em estoque, para contemplar os casos em que não é necessário cortar, de fato, o último item.

Exemplo 3.2.2. Considere a mesma situação problema do Exemplo 3.2.1, com a informação adicional de que as serras têm largura $\sigma = 2$ mm. Note que o padrão de corte $\mathbf{a}_1 = (1, 1, 1)$ não é mais viável nesse caso. Já o padrão \mathbf{a}_2 e \mathbf{a}_3 podem ser utilizados, como verificamos na figura abaixo:

Figura 6 – Padrões de corte com sobras intrínsecas.



Fonte: Autoria própria.

O material cortado pela serra é indicado pela cor branca. Verifica-se que \mathbf{a}_2 e \mathbf{a}_3 são padrão de corte pois da Eq. (3.2.5), temos

$$(2 + 0,2) \cdot 3 + (3 + 0,2) \cdot 1 + (5 + 0,2) \cdot 0 = 9,8 \leq 10 + 0,2 = 10,2;$$

$$(2 + 0,2) \cdot 0 + (3 + 0,2) \cdot 0 + (5 + 0,2) \cdot 1 = 5,2 \leq 10,2.$$

Entretanto, neste trabalho, os padrões de corte considerados não possuirão perdas intrínsecas, ou seja, em todas as demais construções matemáticas estaremos supondo a situação ideal em que a largura das serras é desprezível (nula).

Um outro tipo de padrão de corte importante é denominado de padrão homogêneo, que será apresentado após a seguinte definição:

Definição 3.2.4 (Função Piso e Função Teto). Seja x um número real. O *piso* de x , denotado por $\lfloor x \rfloor$, é o maior inteiro n tal que $n \leq x$. O *teto* de x , denotado por $\lceil x \rceil$, é o menor inteiro n tal que $n \geq x$.

Definição 3.2.5 (Padrão de Corte Homogêneo). Um padrão de corte que produz apenas um tipo de item é chamado padrão de corte homogêneo. Em outras palavras, \mathbf{a}_j é um padrão de corte *homogêneo* se o vetor associado tem apenas uma coordenada não-nula:

$$\mathbf{a}_j = (0, \dots, \alpha_{ij}, \dots, 0) \quad \alpha_{ij} \neq 0. \quad (3.2.6)$$

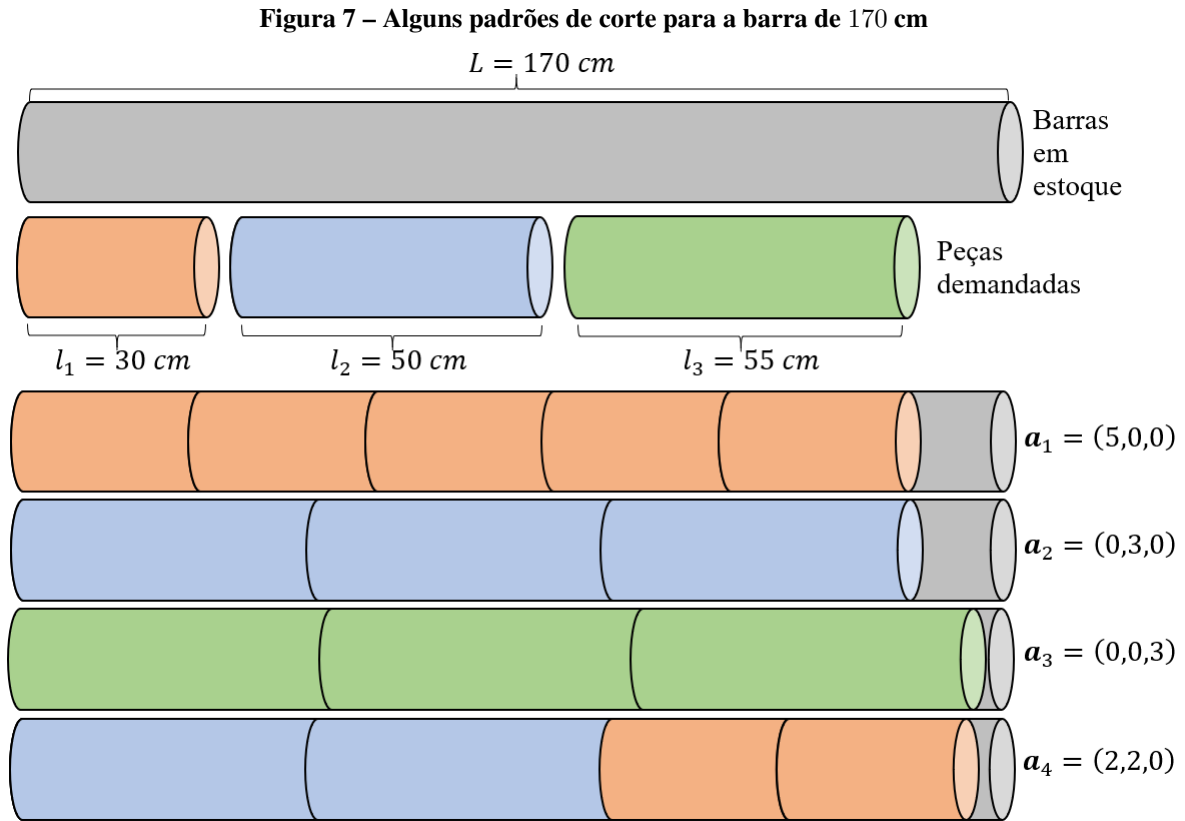
Uma maneira de produzir padrões de corte homogêneos é tomando

$$\alpha_{ij} = \left\lfloor \frac{L}{l_i} \right\rfloor, \quad i = 1, \dots, m. \quad (3.2.7)$$

Note que um padrão de corte homogêneo pode, ou não, ser um padrão de corte completo.

Exemplo 3.2.3. Considere que as barras em estoque têm $L = 170$ cm e que existem demandas por três itens: $l_1 = 30$ cm, $l_2 = 50$ cm e $l_3 = 55$ cm. Na figura abaixo, ilustramos as barras e as

peças e apresentamos alguns padrões de corte. Os padrões de corte \mathbf{a}_1 , \mathbf{a}_2 e \mathbf{a}_3 são homogêneos. O padrão \mathbf{a}_4 não é homogêneo. Utilizando esses padrões de corte ilustrativos, verificamos que a barra não foi totalmente utilizada, ou seja, os padrões geraram sobras.



Fonte: Autoria própria.

Exemplificamos assim, o objetivo de um PCEU, que é determinar os melhores padrões de corte, com o intuito de gerar menos desperdício de matéria-prima, ou equivalentemente, utilizar a menor quantidade de barras possível.

Observação 3.2.1. Podemos concluir que se existem m tipos de itens distintos a serem produzidos, haverá m padrões homogêneos, cujos vetores associados definem uma matriz diagonal, que é não singular. Essa observação é importante, pois poderemos considerar a matriz formada pelos padrões de corte homogêneos como a matriz básica da *Fase I* do Algoritmo Simplex, que será apresentado no Capítulo 4.

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

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

3.3 FORMULAÇÃO MATEMÁTICA DO PCEU

De posse destas informações, podemos enunciar o Problema de Corte de Estoque Unidimensional. A fim de facilitar futuras explicações, chamaremos, a partir dessa seção, os *objetos de tamanho maior apenas de barras* e os *itens de tamanho menor apenas de peças*.

Considere

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

os vetores associados aos possíveis padrões de corte completos, em que a_{ij} é o número de vezes que a peça de comprimento i , $i = 1, \dots, m$, aparece no padrão de corte j . Os vetores acima são todas as possíveis soluções do sistema formado pelas Eqs. (3.2.2) e (3.2.3), com a restrição adicional dada pela Eq. (3.2.4).

Como temos interesse apenas em padrões de corte completos, no restante do trabalho iremos nos referir aos mesmos somente como padrões de corte.

Seja x_j a variável que representa o número de barras cortadas conforme o padrão de corte j . O problema de corte de estoque, em que o número de barras a serem cortadas deve ser minimizado, é dado por:

$$\begin{aligned} \text{Minimizar} \quad & f(\mathbf{x}) = x_1 + x_2 + \dots + x_n \\ \text{sujeito a} \quad & \mathbf{a}_1 x_1 + \mathbf{a}_2 x_2 + \dots + \mathbf{a}_n x_n = \mathbf{d} \\ & x_j \geq 0, \quad x_j \in \mathbb{Z}_+, \quad j = 1, \dots, n. \end{aligned} \tag{3.3.1}$$

em que $\mathbf{a}_j \in \mathbb{Z}_+^m$, para $j = 1, \dots, n$, e $\mathbf{d} \in \mathbb{Z}_+^m$, onde $\mathbf{d} = (d_1, d_2, \dots, d_m)$ é o vetor com as demandas das peças.

Equivalentemente, o mesmo problema pode ser escrito de forma matricial:

$$\begin{aligned} \text{Minimizar} \quad & f(\mathbf{x}) = \sum_{i=1}^n x_j \\ \text{sujeito a} \quad & \mathbf{Ax} = \mathbf{d} \\ & \mathbf{x} \geq 0, \quad \mathbf{x} \in \mathbb{Z}_+ \end{aligned} \tag{3.3.2}$$

onde cada coluna da matriz $\mathbf{A} \in \mathbb{Z}_+^{m \times n}$ é um vetor associado a um padrão de corte e n é o número de padrões de corte.

Duas dificuldades são encontradas quando tentamos resolver este problema: a primeira é determinar todos os n padrões de corte, uma vez que em situações práticas a quantidade de peças m é da ordem de dezenas (ou centenas). Isso implica que o total de padrões de corte pode alcançar a ordem de milhar (ou até mesmo milhões). Para ilustrar, se considerarmos o simples Exemplo (3.2.3), que possui apenas 3 tipos de peças a serem produzidas, verifica-se que existem 27 padrões de corte distintos, o que tende a aumentar com a complexidade do problema. Listamos os padrões na tabela abaixo.

Tabela 1 – Padrões de Corte para o Exemplo (3.2.3).

$\mathbf{a}_1 = (0,0,1)$	$\mathbf{a}_{10} = (4,0,0)$	$\mathbf{a}_{19} = (2,0,2)$
$\mathbf{a}_2 = (0,0,2)$	$\mathbf{a}_{11} = (5,0,0)$	$\mathbf{a}_{20} = (1,1,0)$
$\mathbf{a}_3 = (0,0,3)$	$\mathbf{a}_{12} = (0,1,1)$	$\mathbf{a}_{21} = (2,1,0)$
$\mathbf{a}_4 = (0,1,0)$	$\mathbf{a}_{13} = (0,2,1)$	$\mathbf{a}_{22} = (3,1,0)$
$\mathbf{a}_5 = (0,2,0)$	$\mathbf{a}_{14} = (0,1,2)$	$\mathbf{a}_{23} = (4,1,0)$
$\mathbf{a}_6 = (0,3,0)$	$\mathbf{a}_{15} = (1,0,1)$	$\mathbf{a}_{24} = (1,2,0)$
$\mathbf{a}_7 = (1,0,0)$	$\mathbf{a}_{16} = (2,0,1)$	$\mathbf{a}_{25} = (2,2,0)$
$\mathbf{a}_8 = (2,0,0)$	$\mathbf{a}_{17} = (3,0,1)$	$\mathbf{a}_{26} = (1,1,1)$
$\mathbf{a}_9 = (3,0,0)$	$\mathbf{a}_{18} = (1,0,2)$	$\mathbf{a}_{27} = (2,1,1)$

Fonte: Cherri (2006).

A outra dificuldade está no fato do PCEU ser um problema de otimização inteira, que requer muito mais trabalho e tempo computacional para garantir a otimalidade.

Uma das maneiras de contornar essa questão é realizar a relaxação do problema, ou seja, reescrever o PPI (3.3.1) com variáveis reais. Dessa forma, teremos:

$$\begin{aligned} \text{Minimizar} \quad & f(\mathbf{x}) = x_1 + x_2 + \cdots + x_n \\ \text{sujeito a} \quad & \mathbf{a}_1x_1 + \mathbf{a}_2x_2 + \cdots + \mathbf{a}_nx_n = \mathbf{d} \\ & x_j \geq 0, \quad x_j \in \mathbb{R}_+, \quad j = 1, \dots, n. \end{aligned} \tag{3.3.3}$$

onde $\mathbf{a}_j, \mathbf{d} \in \mathbb{Z}_+^m$ e $\mathbf{d} = (d_1, d_2, \dots, d_m)$ é o vetor com as demandas das peças.

Para resolver esse problema, podemos aplicar o método de Geração de Colunas para o PCEU, desenvolvido por Gilmore e Gomory, que será abordado na Seção 4.2 e depois aplicar algum método heurístico para obter a solução inteira. Uma das maneiras mais simples de se fazer isso é simplesmente arredondando o valor real obtido para o maior inteiro menor ou igual, porém, dessa forma, a demanda não será satisfeita sob a restrição de igualdade, e sim, de maior ou igual, produzindo possíveis excessos de produção. Essa técnica é eficaz para problemas com grandes demandas, pois um simples arredondamento das componentes do vetor não irá gerar grandes desperdícios de matéria-prima. No caso de problemas com baixa demanda requerem outros métodos heurísticos para se determinar uma solução inteira razoável.

No próximo capítulo sintetizamos de maneira abrangente os métodos empregados na resolução do PCEU.

4 MÉTODOS EMPREGADOS NA RESOLUÇÃO DO PCEU

Para apresentar o método de resolução do problema de corte de estoque, chamado Método Simplex com Geração de Colunas, proposto por Gilmore e Gomory (1961), nas próximas seções abordamos os conceitos teóricos dos Métodos Simplex e de Geração de Colunas, o problema da mochila e sua resolução pelo método *Branch-and-Bound*.

4.1 MÉTODO SIMPLEX

A história da programação linear está fortemente relacionada com a Segunda Guerra Mundial, pois as nações envolvidas no conflito buscavam maneiras eficazes de planejar suas despesas, economizar recursos militares e aumentar o dano causado no inimigo. Por isso, se manteve em sigilo até o pós-guerra, quando as indústrias passaram a utilizar as técnicas de resolução de problemas desenvolvidas para fins comerciais. O russo Leonid Vitaliyevich Kantorovich e o norte americano George Bernard Dantzig foram dois dos nomes mais importantes dessa época.

Foi sob esse contexto bélico que o Método Simplex foi criado por Dantzig, em 1947, mais especificamente para solucionar problemas de planejamento que ocorriam na Força Aérea dos Estados Unidos da América. A partir de então, o método foi amplamente estudado e tornou-se popular pela aplicabilidade em diversos problemas que podem surgir nos mais variados contextos.

4.1.1 Algumas definições e o Teorema Fundamental da Programação linear

Para apresentar os conceitos básicos do Método Simplex as referências adotadas foram as de Luenberger e Ye (2008) e Arenales *et al.* (2007).

Consideremos um sistema de equações escritas na forma

$$\mathbf{Ax} = \mathbf{b}, \quad (4.1.1)$$

em que \mathbf{A} é uma matriz com m linhas e n colunas, tal que $m < n$, com posto m , ou seja, com as m linhas de \mathbf{A} formando um conjunto linearmente independente. Além disso, \mathbf{x} é um vetor n -dimensional e \mathbf{b} é um vetor m -dimensional.

Suponha que das n colunas de \mathbf{A} selecionamos um conjunto de m colunas linearmente independentes (esse conjunto existe se o posto de \mathbf{A} é m). Para simplificar, assumamos que

selecionamos as m primeiras colunas de \mathbf{A} e denotamos por \mathbf{B} a matriz $m \times m$ determinada por essas colunas. Então, \mathbf{B} é não singular e podemos resolver unicamente a equação

$$\mathbf{B}\mathbf{x}_B = \mathbf{b} \quad (4.1.2)$$

para um $\mathbf{x}_B \in \mathbb{R}^m$. Escrevendo $\mathbf{x} = (\mathbf{x}_B, \mathbf{x}_N) = (\mathbf{x}_B, \mathbf{0})$, obtemos uma solução para $\mathbf{A}\mathbf{x} = \mathbf{b}$.

Definição 4.1.1 (*Solução Básica, Variáveis Básicas*). Dado um sistema de m -equações lineares com n incógnitas como na Eq. (4.1.1) e uma matriz \mathbf{B} quadrada de ordem m inversível, formada por m colunas de \mathbf{A} . Uma *solução básica* para a Eq. (4.1.1) será todo vetor $\mathbf{x} \in \mathbb{R}^n$ que satisfaz à equação, cujas $n - m$ componentes não associadas com as colunas de \mathbf{B} são iguais a zero. As componentes de \mathbf{x} associadas com as colunas de \mathbf{B} são denominadas *variáveis básicas*.

Exemplo 4.1.1. Considere a Eq. (4.1.1) e a matriz \mathbf{B}' , formada pelas m -primeiras colunas de \mathbf{A} , como na Eq. (4.1.2). Então uma solução básica para a Eq. (4.1.1) é um vetor n -dimensional da forma

$$\mathbf{x} = (x_1, x_2, \dots, x_m, 0, \dots, 0) = (\mathbf{x}_B, \mathbf{0}) \quad (4.1.3)$$

Definição 4.1.2 (*Solução Básica Degenerada*). Uma solução básica degenerada é toda solução básica que possui alguma variável básica igual a zero.

Exemplo 4.1.2. No Exemplo (4.1.1), se algum $x_i = 0, i = 1, 2, \dots, m$, então \mathbf{x} é uma solução básica degenerada para a Eq. (4.1.1).

Definição 4.1.3 (*Solução Factível, Básica Factível*). Um vetor \mathbf{x} que satisfaz a Eq. (4.1.1) é dito ser uma *solução factível* em relação ao conjunto de restrições apresentado. Uma solução factível que também é uma solução básica é dita ser uma *solução básica factível*.

A fim de definirmos a otimalidade de uma solução, consideremos o seguinte problema de programação linear (PPL) em sua forma padrão:

$$\begin{aligned} &\text{Minimizar} && \mathbf{c}^T \mathbf{x} \\ &\text{sujeito a} && \mathbf{A}\mathbf{x} = \mathbf{b} \\ &&& \mathbf{x} \geq \mathbf{0}. \end{aligned} \quad (4.1.4)$$

Definição 4.1.4 (*Solução Ótima Factível, Ótima Básica Factível*). Uma solução factível que satisfaz a Eq. (4.1.4) e que alcança o valor mínimo da função objetivo é dita ser uma *solução ótima factível*. Se essa solução é básica, é dita ser uma *solução ótima básica factível*.

Teorema 4.1.1 (Teorema Fundamental da Programação Linear). *Dado um problema linear na forma padrão (4.1.4) em que \mathbf{A} é uma matriz $m \times n$ de posto m , então*

- (i) se existe uma solução factível, então existe uma solução básica factível;
- (ii) se existe uma solução ótima factível, então existe uma solução ótima básica factível.

Demonstração. (i) Denote as colunas de \mathbf{A} por $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n$. Suponha que $\mathbf{x} = (x_1, x_2, \dots, x_n)$ seja uma solução factível. Então, em termos das colunas de \mathbf{A} , essa solução satisfaz:

$$x_1 \mathbf{a}_1 + x_2 \mathbf{a}_2 + \dots + x_n \mathbf{a}_n = \mathbf{b}.$$

Assuma que exatamente p das variáveis x_i são maiores que zero, e por conveniência, que são as p primeiras variáveis. Assim,

$$x_1 \mathbf{a}_1 + x_2 \mathbf{a}_2 + \dots + x_p \mathbf{a}_p = \mathbf{b}. \quad (4.1.5)$$

Existem agora dois casos possíveis, correspondendo se o conjunto formado por $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_p$ é linearmente independente ou não.

- *Caso 1:* Assuma que $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_p$ seja linearmente independente. Então, claramente, $p \leq m$. Se $p = m$, a solução é básica e a prova está completa. Se $p < m$, então, uma vez que \mathbf{A} tem posto m , $m - p$ vetores podem ser encontrados dos $n - p$ vetores de forma que o conjunto resultante de m vetores seja linearmente independente. Assinalando o valor zero para as $m - p$ variáveis, produzimos uma solução básica factível (e degenerada).
- *Caso 2:* Assuma que $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_p$ é linearmente dependente. Então existe uma combinação linear não trivial desses vetores que é igual a zero. Assim, existem constantes y_1, y_2, \dots, y_p , em que pelo menos uma dessas podemos assumir ser positiva, tais que:

$$y_1 \mathbf{a}_1 + y_2 \mathbf{a}_2 + \dots + y_p \mathbf{a}_p = \mathbf{0}. \quad (4.1.6)$$

Multiplicando essa equação por um escalar ε e subtraindo da Eq. (4.1.5), obtemos:

$$(x_1 - \varepsilon y_1) \mathbf{a}_1 + (x_2 - \varepsilon y_2) \mathbf{a}_2 + \dots + (x_p - \varepsilon y_p) \mathbf{a}_p = \mathbf{b}. \quad (4.1.7)$$

Essa equação vale para todo ε , e para cada ε as componentes $x_i - \varepsilon y_i$ correspondem a uma solução das equações lineares (embora possam violar a restrição $x_i - \varepsilon y_i \geq 0$). Denotando

por $\mathbf{y} = (y_1, y_2, \dots, y_p, 0, \dots, 0) \in \mathbb{R}^n$, vemos que para cada ε , o vetor

$$\mathbf{x} - \varepsilon \mathbf{y} \quad (4.1.8)$$

é uma solução para as equações. Para $\varepsilon = 0$, isso se reduz a solução factível original. À medida que ε "cresce", as componentes do vetor aumentam, diminuem ou mantêm seu valor constante, dependendo se o correspondente y_i é positivo, negativo ou zero. Como assumimos que pelo menos um y_i é positivo, pelo menos uma das componentes irá diminuir seu valor. Então, aumentamos ε até o primeiro ponto onde uma das componente fica igual a zero. Especificamente, fazemos:

$$\varepsilon = \min \left\{ \frac{x_i}{y_i} \mid y_i > 0 \right\}. \quad (4.1.9)$$

Para esse valor de ε , a solução em (4.1.8) é factível e tem pelo menos $p - 1$ variáveis positivas. Repetindo esse processo, se necessário, podemos eliminar variáveis positivas até que tenhamos uma solução factível com as correspondentes colunas de \mathbf{A} formando um conjunto linearmente independente. Assim, a situação é estudado pelo Caso 1.

(ii) Seja $\mathbf{x} = (x_1, x_2, \dots, x_n)$ uma solução ótima factível e, como na prova do Item 1, suponhamos que existam exatamente p variáveis positivas x_1, x_2, \dots, x_p . Novamente, existem dois casos: no Caso 1, correspondendo à independência linear do conjunto formado pelos p vetores, aplicamos o Caso 1 do Item 1.

No Caso 2, procedemos exatamente como antes, mas devemos mostrar que para qualquer ε a solução em (4.1.8) é ótima. Para mostrar isso, notemos que o valor da função associada à solução $\mathbf{x} - \varepsilon \mathbf{y}$ é

$$\mathbf{c}^T \mathbf{x} - \varepsilon \mathbf{c}^T \mathbf{y}. \quad (4.1.10)$$

Para ε suficientemente pequeno em magnitude, $\mathbf{x} - \varepsilon \mathbf{y}$ é uma solução factível para valores positivos ou negativos de ε . Assim, concluímos que $\mathbf{c}^T \mathbf{y} = 0$. De fato, se $\mathbf{c}^T \mathbf{y} \neq 0$, um ε de pequena magnitude e sinal próprio poderia ser determinado de forma que a solução (4.1.10) fosse menor que $\mathbf{c}^T \mathbf{x}$ e mantivesse a factibilidade. Isso violaria a suposição de otimalidade em \mathbf{x} . Logo, devemos ter $\mathbf{c}^T \mathbf{y} = 0$.

Tendo estabelecido que novas soluções factíveis com componentes positivas menores são também ótimas, o restante da prova pode ser completada exatamente como no item 1. ■

Esse teorema reduz a resolução de um PPL à tarefa de procurar soluções básicas factíveis. Para um problema tendo n variáveis e m restrições, existem pelo menos

$$\binom{n}{m} = \frac{n!}{m!(n-m)!} \quad (4.1.11)$$

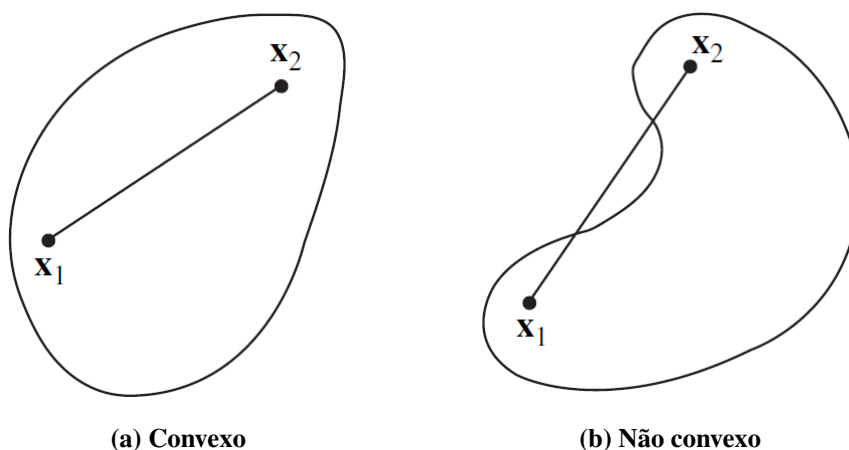
soluções básicas, ou seja, um número finito de possibilidades. Assim, o teorema fundamental produz uma técnica de busca finita, mas ineficiente.

4.1.2 Relações com Convexidade

Além de obtermos uma dedução alternativa do Teorema Fundamental da Programação Linear, poderemos, com o auxílio da teoria dos conjuntos convexos, fazer uma interpretação geométrica para o PPL (4.1.4), descrito na subseção anterior. Para que isso possa ser realizado, iremos apresentar as definições mais importantes com o objetivo de relacionar soluções básicas factíveis com pontos extremos de polítopos. A referência adotada foi a de Luenberger e Ye (2008).

Definição 4.1.5 (Conjunto Convexo). Seja C um conjunto no espaço euclidiano n -dimensional, que denotaremos por \mathbb{E}^n . Dizemos que C é um conjunto convexo, ou apenas convexo, se para quaisquer $x_1, x_2 \in C$ e para todo número real $\alpha \in [0, 1]$, o ponto $\alpha x_1 + (1 - \alpha) x_2 \in C$.

Figura 8 – Exemplos de conjuntos convexos e não convexos



Fonte: Luenberger e Ye (2008)

A definição a seguir é semelhante a de um conjunto convexo e nos permitirá, juntamente com as seguintes, definirmos o que é um *politopo*.

Definição 4.1.6 (*Variedade Linear*). Um conjunto $V \subset \mathbb{E}^n$ é uma *variedade linear* se para quaisquer $x_1, x_2 \in V$, a combinação linear $\alpha x_1 + (1 - \alpha) x_2 \in V$, para todo escalar $\alpha \in \mathbb{R}$.

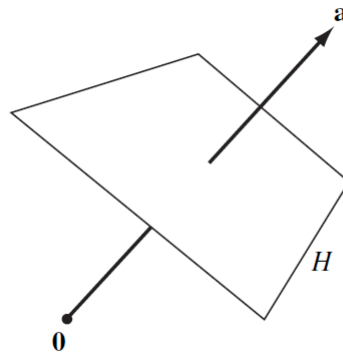
Definição 4.1.7 (*Hiperplano*). Um *hiperplano* é uma variedade linear de dimensão $(n - 1)$ em \mathbb{E}^n .

A definição acima nos permite caracterizar um hiperplano H em \mathbb{E}^n como:

$$H = \{\mathbf{x} \in \mathbb{E}^n \mid \mathbf{a}^T \mathbf{x} = k\}, \quad (4.1.12)$$

em que $\mathbf{a} \in \mathbb{E}^n$ é chamado vetor *normal* ou o *gradiente* para o hiperplano e $k \in \mathbb{R}$.

Figura 9 – Ilustração de um hiperplano conforme a caracterização dada pela Eq. (4.1.12)



Fonte: Luenberger e Ye (2008)

Um hiperplano (4.1.12) divide o espaço euclidiano \mathbb{E}^n em duas regiões, definidas abaixo:

Definição 4.1.8 (*Semi-espacos Fechados, Abertos*). Em relação ao hiperplano (4.1.12), os seguintes conjuntos

$$\begin{aligned} H_+ &= \{\mathbf{x} \in \mathbb{E}^n \mid \mathbf{a}^T \mathbf{x} \geq k\} \\ H_- &= \{\mathbf{x} \in \mathbb{E}^n \mid \mathbf{a}^T \mathbf{x} \leq k\} \end{aligned} \quad (4.1.13)$$

são denominados *semi-espaco fechado positivo* e *semi-espaco fechado negativo*, respectivamente.

Quando as desigualdades que definem os conjuntos acima são estritas, teremos os *semi-espaco aberto positivo* e *semi-espaco aberto negativo*, denotados por $\overset{\circ}{H}_+$ e $\overset{\circ}{H}_-$.

Definição 4.1.9 (*Politopo Convexo*). Um *politopo convexo* será todo conjunto que pode ser expresso como uma interseção finita de semi-espacos fechados.

Note que por essa definição, podem existir polítopos convexos vazios, limitados e ilimitados. Um caso especial ocorre quando o polítopo convexo é não vazio e limitado. Neste caso ele é chamado de *poliedro*.

De posse desses conceitos iniciais, podemos relacionar pontos extremos (Definição 4.1.10) e soluções básicas (Definição 4.1.1).

Definição 4.1.10 (*Ponto Extremo*). Um ponto \mathbf{x} em um conjunto convexo C é dito ser um *ponto extremo de C* se não existem pontos $\mathbf{x}_1, \mathbf{x}_2 \in C$, com $x_1 \neq x_2$, tais que $\mathbf{x} = \alpha \mathbf{x}_1 + (1 - \alpha) \mathbf{x}_2$ para todo α , $0 < \alpha < 1$.

Teorema 4.1.2 (Equivalência entre pontos extremos e soluções básicas). *Seja A uma matriz $m \times n$ de posto m e $\mathbf{b} \in \mathbb{R}^m$. Seja K um polítopo convexo consistindo de todos os $\mathbf{x} \in \mathbb{R}^n$ tais que*

$$\begin{aligned} \mathbf{Ax} &= \mathbf{b} \\ \mathbf{x} &\geq \mathbf{0}. \end{aligned} \tag{4.1.14}$$

Um vetor \mathbf{x} é um ponto extremo de K se, e somente se, \mathbf{x} é uma solução básica factível para o problema (4.1.14).

Demonstração. Suponha que $\mathbf{x} = (x_1, x_2, \dots, x_m, 0, \dots, 0)$ seja uma solução básica factível para o problema (4.1.14). Então

$$x_1 \mathbf{a}_1 + x_2 \mathbf{a}_2 + \dots + x_m \mathbf{a}_m = \mathbf{b},$$

em que $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_m$, as primeiras m colunas de \mathbf{A} , são linearmente independentes. Suponha que \mathbf{x} possa ser escrito como uma combinação convexa de dois pontos \mathbf{y} e \mathbf{z} em K , ou seja, $\mathbf{x} = \alpha \mathbf{y} + (1 - \alpha) \mathbf{z}$, $0 < \alpha < 1$. Como todas as componentes de $\mathbf{x}, \mathbf{y}, \mathbf{z}$ são não negativas e como $0 < \alpha < 1$, segue imediatamente que as últimas $n - m$ componentes de \mathbf{y} e \mathbf{z} são iguais a zero. Logo, como $\mathbf{y}, \mathbf{z} \in K$ e dada a conclusão anterior, temos que \mathbf{y} e \mathbf{z} também são soluções básicas factíveis para o problema (4.1.14). Logo, temos:

$$y_1 \mathbf{a}_1 + y_2 \mathbf{a}_2 + \dots + y_m \mathbf{a}_m = \mathbf{b} \tag{4.1.15}$$

$$z_1 \mathbf{a}_1 + z_2 \mathbf{a}_2 + \dots + z_m \mathbf{a}_m = \mathbf{b}. \tag{4.1.16}$$

Subtraindo a Eq. (4.1.16) na Eq. (4.1.15), obtemos:

$$(y_1 - z_1) \mathbf{a}_1 + (y_2 - z_2) \mathbf{a}_2 + \dots + (y_m - z_m) \mathbf{a}_m = \mathbf{0}.$$

Como $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_m$ são linearmente independentes, segue que $\mathbf{y} = \mathbf{z}$. Logo, \mathbf{x} não pode ser escrito como uma combinação convexa de dois pontos distintos de \mathbf{K} , isto é, \mathbf{x} é um ponto extremo do polítopo \mathbf{K} .

Agora, assumamos que \mathbf{x} é um ponto extremo de \mathbf{K} . Vamos assumir que as componentes não nulas de \mathbf{x} são as k primeiras componentes. Então, devemos ter:

$$x_1 \mathbf{a}_1 + x_2 \mathbf{a}_2 + \dots + x_k \mathbf{a}_k = \mathbf{b},$$

com $x_i > 0, i = 1, 2, \dots, k$. Para mostrar que \mathbf{x} é uma solução básica factível, devemos mostrar que os vetores $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_k$ são linearmente independentes. Por contradição, suponhamos que não sejam, ou seja, existe uma combinação linear não trivial desses vetores dada por

$$y_1 \mathbf{a}_1 + y_2 \mathbf{a}_2 + \dots + y_k \mathbf{a}_k = \mathbf{0}.$$

Consideremos o vetor $\mathbf{y} = (y_1, y_2, \dots, y_k, 0, \dots, 0) \in \mathbb{R}^n$. Como $x_i > 0, 1 \leq i \leq k$, é possível determinar um ε tal que:

$$\mathbf{x} + \varepsilon \mathbf{y} \geq \mathbf{0}, \quad \mathbf{x} - \varepsilon \mathbf{y} \geq \mathbf{0}.$$

Portanto

$$\mathbf{x} = \frac{1}{2}(\mathbf{x} + \varepsilon \mathbf{y}) + \frac{1}{2}(\mathbf{x} - \varepsilon \mathbf{y}).$$

Ou seja, \mathbf{x} é escrito como uma combinação convexa de dois pontos distintos de \mathbf{K} . Porém, isso não pode ocorrer, uma vez que \mathbf{x} é uma solução básica factível. Logo, $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_m$ são linearmente independentes. ■

Dessa maneira, mostramos a equivalência entre pontos extremos e soluções básicas.

A seguir apresentamos alguns corolários, para complementação dos conceitos sobre relações de convexidade.

Corolário 4.1.2.1. *Se um conjunto convexo \mathbf{K} correspondente ao problema (4.1.14) é não vazio, então possui pelo menos um ponto extremo.*

Corolário 4.1.2.2. *Se existe uma solução ótima finita para um problema de programação linear, existe uma solução ótima finita que é um ponto extremo do conjunto formado pelas restrições.*

Corolário 4.1.2.3. *O conjunto de restrições \mathbf{K} , correspondente ao problema (4.1.14), possui pelo menos um número finito de pontos extremos.*

Corolário 4.1.2.4. *Se um politopo convexo \mathbf{K} correspondente ao problema (4.1.14) é limitado, então \mathbf{K} é um poliedro convexo, ou seja, \mathbf{K} consiste de pontos que são combinações convexas de um número finito de pontos.*

4.1.3 O Método Simplex

Vimos na Subseção 4.1.1 alguns conceitos importantes, como por exemplo, o de uma solução básica. Relembremos que, sob a hipótese de que o posto de \mathbf{A} é igual à m , é possível construir uma matriz $\mathbf{B}_{m \times m}$ ao selecionar m colunas linearmente independentes de \mathbf{A} e determinar um vetor $\mathbf{x}_B \in \mathbb{R}^m$ que soluciona $\mathbf{B}\mathbf{x}_B = \mathbf{b}$. Além disso, escrevendo $\mathbf{x} = (\mathbf{x}_B, \mathbf{x}_N) = (\mathbf{x}_B, \mathbf{0})$, tem-se a igualdade $\mathbf{A}\mathbf{x} = \mathbf{b}$.

Com o intuito de auxiliar na visão matemática da estrutura e da construção do algoritmo Simplex, o qual será apresentado posteriormente, combinado com a técnica de geração de colunas, para a resolução do problema de corte de estoque, apresentamos a seguir algumas definições e teoremas, baseados em Arenales *et al.* (2007) e Bazaraa *et al.* (2009).

Definição 4.1.11 (Partição Básica). *Seja a matriz $\mathbf{A}_{m \times n}$. Consideremos a seguinte reorganização de suas colunas:*

$$\mathbf{A} = [\mathbf{B} \quad \mathbf{N}]$$

em que:

- $\mathbf{B}_{m \times m}$, chamada *matriz básica* é formada por m colunas da matriz \mathbf{A} e é não singular. É dada por:

$$\mathbf{B} = [\mathbf{a}_{B_1} \quad \mathbf{a}_{B_2} \quad \dots \quad \mathbf{a}_{B_m}]$$

isto é, B_1, B_2, \dots, B_m são os índices das colunas de \mathbf{A} que pertencem a \mathbf{B} , denominados *índices básicos*.

- $\mathbf{N}_{m \times (n-m)}$, chamada *matriz não-básica*, é formada pelas $n - m$ colunas restantes de \mathbf{A} que não estão em \mathbf{B} . É dada por:

$$\mathbf{N} = [\mathbf{a}_{N_1} \quad \mathbf{a}_{N_2} \quad \dots \quad \mathbf{a}_{N_{n-m}}]$$

isto é, N_1, N_2, \dots, N_{n-m} são os índices das colunas de \mathbf{A} que pertencem a \mathbf{N} .

Essa partição nas colunas de \mathbf{A} é chamada *partição básica* e se relaciona com a partição do vetor $\mathbf{x} = (\mathbf{x}_B, \mathbf{x}_N)$ que foi construída na primeira subseção. Vamos chamar as componentes x_{B_i} , $1 \leq i \leq m$, do vetor \mathbf{x}_B de *variáveis básicas* e as componentes x_{N_i} , $1 \leq i \leq n - m$, do vetor \mathbf{x}_N de *variáveis não-básicas*.

Dessa forma, o sistema $\mathbf{Ax} = \mathbf{b}$ é reescrito como

$$\mathbf{Ax} = \mathbf{b} \iff [\mathbf{B} \quad \mathbf{N}] \begin{bmatrix} \mathbf{x}_B \\ \mathbf{x}_N \end{bmatrix} = \mathbf{B}\mathbf{x}_B + \mathbf{N}\mathbf{x}_N = \mathbf{b}, \quad (4.1.17)$$

e assim, a *solução geral do sistema* será:

$$\mathbf{x}_B = \mathbf{B}^{-1}\mathbf{b} - \mathbf{B}^{-1}\mathbf{N}\mathbf{x}_N. \quad (4.1.18)$$

Como vimos, o Teorema Fundamental da Programação Linear garante um método finito de busca da solução básica factível ótima, porém ineficiente, pois o número de possibilidades pode ser muito grande em problemas práticos. Utilizando as relações que definimos entre soluções básicas factíveis e pontos extremos (vértices) de um politopo gerado pelas restrições de um problema linear, constrói-se um método que inicia com uma solução básica factível e pesquisa apenas outras soluções básicas melhores que a corrente, ou seja, dado $S = \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{Ax} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$, é pesquisado apenas um subconjunto (em geral, pequeno) dos p vértices de S , e, dada uma solução básica factível, o processo de busca se resume a verificar se tal solução é ótima e caso não seja, determinar outra solução básica factível melhor. Este método se chama *Método Simplex*.

Seja então, a solução básica factível $\hat{\mathbf{x}} = (\hat{\mathbf{x}}_B, \hat{\mathbf{x}}_N) = (\mathbf{B}^{-1}\mathbf{b}, \mathbf{0})$, onde $\mathbf{B}^{-1}\mathbf{b} \geq \mathbf{0}$, e a solução geral do sistema (4.1.18), usando a mesma partição básica. A função objetivo $f(\mathbf{x})$ pode ser expressa considerando a partição básica:

$$f(\mathbf{x}) = \mathbf{c}^T \mathbf{x} = (\mathbf{c}_B^T \quad \mathbf{c}_N^T) \begin{bmatrix} \mathbf{x}_B \\ \mathbf{x}_N \end{bmatrix} = \mathbf{c}_B^T \mathbf{x}_B + \mathbf{c}_N^T \mathbf{x}_N, \quad (4.1.19)$$

onde

\mathbf{c}_B^T são os coeficientes das variáveis básicas na função objetivo;

\mathbf{c}_N^T são os coeficientes das variáveis não-básicas na função objetivo.

Utilizando a expressão (4.1.18) para \mathbf{x}_B , vem:

$$f(\mathbf{x}) = \mathbf{c}_B^T (\mathbf{B}^{-1}\mathbf{b} - \mathbf{B}^{-1}\mathbf{N}\mathbf{x}_N) + \mathbf{c}_N^T \mathbf{x}_N = \mathbf{c}_B^T \mathbf{B}^{-1}\mathbf{b} - \mathbf{c}_B^T \mathbf{B}^{-1}\mathbf{N}\mathbf{x}_N + \mathbf{c}_N^T \mathbf{x}_N \quad (4.1.20)$$

Também, podemos observar que $f(\hat{\mathbf{x}}) = \mathbf{c}_B^T \hat{\mathbf{x}}_B = \mathbf{c}_B^T \mathbf{B}^{-1} \mathbf{b}$. Assim, temos que:

$$f(\mathbf{x}) = f(\hat{\mathbf{x}}) - \mathbf{c}_B^T \mathbf{B}^{-1} \mathbf{N} \mathbf{x}_N + \mathbf{c}_N^T \mathbf{x}_N$$

Para auxiliar na notação, façamos a seguinte definição:

Definição 4.1.12 (*Vetor Multiplicador Simplex*). O vetor m -dimensional $\boldsymbol{\lambda}$, dado por:

$$\boldsymbol{\lambda}^T = \mathbf{c}_B^T \mathbf{B}^{-1}$$

é chamado *vetor multiplicador simplex* ou também *vetor de variáveis duais*.

Note que $\boldsymbol{\lambda}^T = \mathbf{c}_B^T \mathbf{B}^{-1} \iff \mathbf{B}^T \boldsymbol{\lambda} = \mathbf{c}_B$.

Utilizando o vetor multiplicador simplex na Eq. (4.1.20), temos:

$$\begin{aligned} f(\mathbf{x}) &= f(\hat{\mathbf{x}}) - \boldsymbol{\lambda}^T \mathbf{N} \mathbf{x}_N + \mathbf{c}_N^T \mathbf{x}_N \\ &= f(\hat{\mathbf{x}}) + (\mathbf{c}_N^T - \boldsymbol{\lambda}^T \mathbf{N}) \mathbf{x}_N. \end{aligned} \quad (4.1.21)$$

Desenvolvendo os termos de $\mathbf{c}_N^T - \boldsymbol{\lambda}^T \mathbf{N}$ para $N = N_1, N_2, \dots, N_{n-m}$, escrevemos a Eq. (4.1.21) como

$$f(\mathbf{x}) = f(\hat{\mathbf{x}}) + (c_{N_1} - \boldsymbol{\lambda}^T \mathbf{a}_{N_1}) x_{N_1} + (c_{N_2} - \boldsymbol{\lambda}^T \mathbf{a}_{N_2}) x_{N_2} + \dots + (c_{N_{n-m}} - \boldsymbol{\lambda}^T \mathbf{a}_{N_{n-m}}) x_{N_{n-m}} \quad (4.1.22)$$

Na expressão acima, os termos (coeficientes) de x_{N_j} , $j = 1, \dots, n - m$, são associados aos chamados *custos reduzidos*, que serão usados para a análise da condição de otimalidade da solução, e apresentados a seguir.

Definição 4.1.13 (*Custos Reduzidos*). Os coeficientes $\hat{c}_{N_j} = (c_{N_j} - \boldsymbol{\lambda}^T \mathbf{a}_{N_j})$ das variáveis não-básicas na função objetivo descrita pela Eq. (4.1.22) são chamados *custos reduzidos* ou *custos relativos*.

Com essa notação, a Eq. (4.1.22) é reescrita por:

$$f(\mathbf{x}) = f(\hat{\mathbf{x}}) + \hat{c}_{N_1} x_{N_1} + \hat{c}_{N_2} x_{N_2} + \dots + \hat{c}_{N_{n-m}} x_{N_{n-m}}. \quad (4.1.23)$$

Se os todos os custos reduzidos são não negativos, ou seja,

$$(c_{N_j} - \boldsymbol{\lambda}^T \mathbf{a}_{N_j}) \geq 0, \quad j = 1, \dots, n - m, \quad (4.1.24)$$

temos que $f(\mathbf{x}) \geq f(\hat{\mathbf{x}})$, para todo $\mathbf{x}_N \geq \mathbf{0}$. E assim, se a solução básica é factível e todos os custos reduzidos são positivos (para problemas de minimização) então a solução é ótima, como afirma a propriedade a seguir.

Propriedade 4.1.1 (*Condição de otimalidade*). Considere uma partição básica $\mathbf{A} = [\mathbf{B} \ \mathbf{N}]$ em que a solução básica associada $\hat{\mathbf{x}}_B = \mathbf{B}^{-1}\mathbf{b} \geq \mathbf{0}$, e seja $\boldsymbol{\lambda}^T = \mathbf{c}_B^T \mathbf{B}^{-1}$ o vetor multiplicador simplex. Se $(c_{N_j} - \boldsymbol{\lambda}^T \mathbf{a}_{N_j}) \geq 0, j = 1, \dots, n - m$, então a solução básica é ótima.

Considere uma solução básica factível e suponha que a condição de otimalidade seja violada (caso contrário, a solução é ótima), isto é, supondo que exista k tal que:

$$\hat{c}_{N_k} = c_{N_k} - \boldsymbol{\lambda}^T \mathbf{a}_{N_k} < 0,$$

ou seja, o custo relativo da variável não-básica x_{N_k} é negativo, é preciso definir uma estratégia para analisar os efeitos de uma perturbação na solução básica factível com intuito de diminuir o valor da função objetivo.

Definição 4.1.14 (*Estratégia Simplex*). Denomina-se por *estratégia simplex* a perturbação de uma solução básica factível que consiste em alterar as variáveis não-básicas por:

$$\begin{cases} \mathbf{x}_{N_k} = \varepsilon \geq 0 & \text{(variável com custo reduzido negativo)} \\ \mathbf{x}_{N_j} = 0 & j = 1, 2, \dots, n - m, \quad \text{e } i \neq k \end{cases} \quad (4.1.25)$$

Com essa alteração, a função objetivo da Eq. (4.1.23) pode ser escrita como:

$$\begin{aligned} f(\mathbf{x}) &= f(\hat{\mathbf{x}}) + \hat{c}_{N_1} 0 + \dots + \hat{c}_{N_k} \varepsilon + \dots + \hat{c}_{N_{n-m}} 0 \\ &= f(\hat{\mathbf{x}}) + \hat{c}_{N_k} \varepsilon < f(\hat{\mathbf{x}}). \end{aligned}$$

Note que a função objetivo decresce quando ε cresce, com taxa negativa \hat{c}_{N_k} . Quanto menor o valor de \hat{c}_{N_k} , mais rápido a função objetivo decresce. Isso justifica a escolha da variável não-básica a ser perturbada com o menor custo relativo (esta escolha é conhecida na literatura como *regra de Dantzig*), isto é:

$$\hat{c}_{N_k} = \text{Mínimo} \{ \hat{c}_{N_j} \} = \text{Mínimo} \{ c_{N_j} - \boldsymbol{\lambda}^T \mathbf{a}_{N_j} \}, \quad j = 1, 2, \dots, n - m. \quad (4.1.26)$$

↓

a variável não básica x_{N_k} será perturbada.

Como a função objetivo decresce quando ε cresce, determinamos o maior valor possível para ε que mantém factível a solução perturbada:

Tamanho do passo ε : a estratégia simplex é equivalente a alterar as variáveis não-básicas para:

$$\mathbf{x}_N = \begin{bmatrix} x_{N_1} \\ \vdots \\ x_{N_k} \\ \vdots \\ x_{N_{n-m}} \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ \varepsilon \\ \vdots \\ 0 \end{bmatrix} \leftarrow k \quad (4.1.27)$$

portanto, as variáveis básicas são modificadas por

$$\mathbf{x}_B = \mathbf{B}^{-1}\mathbf{b} - \mathbf{B}^{-1}\mathbf{N}\mathbf{x}_N = \hat{\mathbf{x}}_B - \mathbf{B}^{-1}\mathbf{a}_{N_k}\varepsilon = \hat{\mathbf{x}}_B - \mathbf{y}\varepsilon \quad (4.1.28)$$

em que $\mathbf{y} = \mathbf{B}^{-1}\mathbf{a}_{N_k}$. A identidade anterior decorre de definição da matriz não básica \mathbf{N} e da definição de \mathbf{x}_N dada pela Eq. (4.1.27):

$$\mathbf{N}\mathbf{x}_N = \mathbf{N}(0, \dots, \varepsilon, \dots, 0) = [\mathbf{a}_{N_1} \dots \mathbf{a}_{N_k} \mathbf{a}_{N_{n-m}}](0, \dots, \varepsilon, \dots, 0) = \mathbf{a}_{N_k}\varepsilon. \quad (4.1.29)$$

Nesta altura, determinada a variável que vai entrar na base (se tornar básica) é preciso analisar, pela estratégia da direção simplex, qual variável sairá da base.

Definição 4.1.15 (*Direção Simplex*). Denomina-se por *direção simplex* o vetor $\mathbf{y} = \mathbf{B}^{-1}\mathbf{a}_{N_k}$, o qual fornece os coeficientes de como as variáveis básicas são alteradas pela estratégia simplex. A direção simplex é solução do sistema de equações lineares $\mathbf{B}\mathbf{y} = \mathbf{a}_{N_k}$.

Reescrevendo a Eq. (4.1.28) acima em termos de cada uma de suas coordenadas, e considerando a não-negatividade das variáveis básicas, temos:

$$x_{B_i} = \hat{x}_{B_i} - y_i\varepsilon \geq 0, \quad i = 1, \dots, m.$$

Assim:

- se $y_i \leq 0$, então $x_{B_i} \geq 0$, para todo $\varepsilon \geq 0$;
- se $y_i > 0$, como $x_{B_i} = \hat{x}_{B_i} - y_i\varepsilon \geq 0$, então, $\varepsilon \leq \frac{\hat{x}_{B_i}}{y_i}$.

Logo, o maior valor de ε é dado por

$$\hat{\varepsilon} = \frac{\hat{x}_{B_l}}{y_l} = \text{mínimo} \left\{ \frac{\hat{x}_{B_i}}{y_i} \text{ tal que } y_i > 0 \right\}. \quad (4.1.30)$$

Com o valor $\hat{\varepsilon} = \frac{\hat{x}_{B_l}}{y_l}$, a variável básica x_{B_l} se anula e a variável não-básica x_{N_k} torna-se positiva. De fato, da Eq. (4.1.28), temos:

- l -ésima variável básica: $x_{B_l} = \hat{x}_{B_l} - y_l \hat{\varepsilon} = \hat{x}_{B_l} - y_l \left(\frac{\hat{x}_{B_l}}{y_l} \right) = 0$;
- k -ésima variável não-básica: $x_{N_k} = \hat{\varepsilon}$,

isto é, a nova solução tem a seguinte característica:

$$(x_{B_1} \dots \underbrace{x_{B_l}}_{=0} \dots x_{B_m} \mid 0 \dots \underbrace{x_{N_k}}_{=\hat{\varepsilon}} \dots 0),$$

ou seja, $n - m$ variáveis não nulas: $x_{N_1}, \dots, x_{B_l}, \dots, x_{N_{n-m}}$, as quais podem ser consideradas não-básicas. Isso resulta em uma nova partição básica com os índices trocados:

$$B_l \longleftrightarrow N_k$$

isto é, a variável x_{N_k} torna-se básica e a variável x_{B_l} não-básica. As matrizes básicas e não-básicas são alteradas por apenas uma coluna:

$$\mathbf{B} = [\mathbf{a}_{B_1} \dots \mathbf{a}_{B_l} \dots \mathbf{a}_{B_m}] \longrightarrow \mathbf{B}' = [\mathbf{a}_{B_1} \dots \underbrace{\mathbf{a}_{N_k}}_{l\text{-ésima coluna}} \dots \mathbf{a}_{B_m}] \quad (4.1.31)$$

$$\mathbf{N} = [\mathbf{a}_{N_1} \dots \mathbf{a}_{N_k} \dots \mathbf{a}_{N_{n-m}}] \longrightarrow \mathbf{N}' = [\mathbf{a}_{N_1} \dots \underbrace{\mathbf{a}_{B_l}}_{k\text{-ésima coluna}} \dots \mathbf{a}_{N_{n-m}}]. \quad (4.1.32)$$

Dizemos que x_{N_k} entra na base e x_{B_l} sai da base. A seguinte propriedade garante que a nova partição gerada é básica:

Propriedade 4.1.2. A matriz \mathbf{B}' , definida anteriormente, é invertível, de modo que $\mathbf{A} = [\mathbf{B}' \quad \mathbf{N}']$ é uma partição básica.

Com isso, mostramos que a estratégia simplex produz uma nova solução básica factível (um novo vértice, em termos de teoria de conjuntos convexas), para a qual a função objetivo tem um valor menor:

$$f(\mathbf{x}) = f(\hat{\mathbf{x}}) + \hat{c}_{N_k} \hat{\varepsilon} < f(\hat{\mathbf{x}}), \quad (4.1.33)$$

e podemos repetir o procedimento, isto é: *encontramos outra solução básica melhor a partir daquela em mãos*, enquanto a condição de otimalidade não for verificada. Este procedimento basicamente consiste na estrutura do método simplex.

4.1.4 O algoritmo Simplex

Após a apresentação dos conceitos teóricos do Método Simplex na subsecção anterior, iremos reunir as informações e expor a seguir o algoritmo simplex, conforme Arenales *et al.* (2007). Para tanto, consideremos um PPL em sua forma padrão (4.1.4). Observamos que o algoritmo estará dividido em duas fases, onde para a *Fase I* se faz necessário determinar uma solução básica factível inicial. Considerando a formulação matemática de um PCEU, dada pelo PPI (3.3.1) e a Observação (3.2.1), podemos tomar a matriz formada pelos padrões de corte homogêneos como a matriz básica e calcular a solução básica factível inicial a partir dela. Já o Método Simplex é iniciado na *Fase II*. Descrevemos ambas as fases a seguir:

Algoritmo: Método Simplex

Fase I

Determine inicialmente uma partição básica factível $\mathbf{A} = [\mathbf{B} \quad \mathbf{N}]$. A rigor, precisamos de dois vetores de índices básicos e não básicos:

$$(B_1, B_2, \dots, B_m) \quad \text{e} \quad (N_1, N_2, \dots, N_{n-m}).$$

Os vetores das variáveis básicas e não básicas são, respectivamente:

$$\mathbf{x}_{\mathbf{B}}^T = (x_{B_1} \ x_{B_2} \ \dots \ x_{B_m}) \quad \text{e} \quad \mathbf{x}_{\mathbf{N}}^T = (x_{N_1} \ x_{N_2} \ \dots \ x_{N_{n-m}}).$$

Faça Iteração = 1.

Fase II:

{Início da iteração simplex}

Passo 1: {Cálculo da solução básica}

$$\begin{cases} \mathbf{x}_{\mathbf{B}}^T = \mathbf{B}^{-1} \mathbf{b} & (\text{equivalentemente, resolva o sistema } \mathbf{B} \mathbf{x}_{\mathbf{B}} = \mathbf{b}) \\ \mathbf{x}_{\mathbf{N}}^T = \mathbf{0} \end{cases}$$

Passo 2: {Cálculo dos custos reduzidos}

2.1) *{Vetor multiplicador simplex}*

$$\boldsymbol{\lambda}^T = \mathbf{c}_{\mathbf{B}}^T \mathbf{B}^{-1} \quad (\text{equivalentemente, resolva o sistema } \mathbf{B}^T \boldsymbol{\lambda} = \mathbf{c}_{\mathbf{B}})$$

2.2) *{Custos reduzidos}*

$$\hat{c}_{N_j} = c_{N_j} - \boldsymbol{\lambda}^T \mathbf{a}_{N_j} \quad j = 1, 2, \dots, n - m$$

2.3) *{Determinação da variável a entrar na base }*

$$\hat{c}_{N_k} = \text{Mínimo} \{ \hat{c}_{N_j}, \quad j = 1, 2, \dots, n - m \} \quad (\text{a variável } x_{N_j} \text{ entra na base})$$

Passo 3: {Teste de Otimalidade }

Se $\hat{c}_{N_k} \geq 0$, então: *pare {solução na iteração atual é ótima }*

Caso contrário, prossiga com

3.1) *{Cálculo da direção simplex}*

$$\mathbf{y} = \mathbf{B}^{-1} \mathbf{a}_{N_k} \quad (\text{equivalentemente, resolva o sistema: } \mathbf{B}\mathbf{y} = \mathbf{a}_{N_k})$$

3.2) *{Determinação do passo e variável a sair da base }*

Se $\mathbf{y} \leq \mathbf{0}$, então: *pare: {problema não tem solução ótima finita: } $f(\mathbf{x}) \rightarrow -\infty$*

Caso contrário, determine a variável a sair da base pela razão mínima:

$$\hat{\varepsilon} = \frac{\hat{x}_{B_l}}{y_l} = \text{Mínimo} \left\{ \frac{\hat{x}_{B_i}}{y_i} \mid y_i > 0, \quad i = 1, \dots, m \right\} \quad (\text{a variável } x_{B_l} \text{ sai da base}).$$

3.3) *{Atualização}*: nova partição básica, troque a l -ésima coluna de \mathbf{B} pela k -ésima coluna de \mathbf{N} .

$$\text{Matriz básica: } \mathbf{B} = [\mathbf{a}_{B_1} \dots \mathbf{a}_{B_{l-1}} \mathbf{a}_{N_k} \mathbf{a}_{B_{l+1}} \dots \mathbf{a}_{B_m}]$$

$$\text{Matriz não básica: } \mathbf{N} = [\mathbf{a}_{N_1} \dots \mathbf{a}_{N_{k-1}} \mathbf{a}_{B_l} \mathbf{a}_{N_{k+1}} \dots \mathbf{a}_{N_{n-m}}]$$

Iteração = Iteração + 1

Retorne ao *Passo 1*.

{Fim da iteração simplex}

4.2 MÉTODO DE GERAÇÃO DE COLUNAS (MGC)

4.2.1 Introdução

Os princípios do Método de Geração de Colunas (MGC) foram estabelecidos no final dos anos 50 e início dos anos 60, nos trabalhos publicados por Ford e Fulkerson (1958), Manne (1958), Dantzig e Wolfe (1960). Em 1961, Gilmore e Gomory (1961), propuseram e desenvolveram o MGC para a resolução do Problema de Corte de Estoque Unidimensional, sendo esse um dos marcos mais importantes na história dos Problema de Corte de Estoque.

Como destacamos no Capítulo 3, a grande dificuldade em resolver um PCEU utilizando apenas o Método Simplex está em listar todos os padrões de corte possíveis e determinar qual

coluna (padrão) irá "entrar" na matriz básica \mathbf{B} , após o cálculo do custo reduzido das colunas que compõem a matriz não básica \mathbf{N} , em cada iteração. Para problemas reais, estas duas tarefas são muito árduas, pois, em geral, o número de variáveis (colunas) que determinam os padrões de corte é muito maior que o número de restrições (linhas), o que torna o simplex ineficiente.

Nesse contexto, seria muito conveniente elaborar um método em que não fosse necessário elencar todos os n padrões de corte, ou seja, que fosse possível trabalhar apenas com uma pequena quantidade deles. E ainda melhor, que produzisse de forma direta quais são os melhores padrões que constituirão a matriz básica. Todas estas características fazem parte do chamado *Método de Geração de Colunas*, que será descrito a seguir, conforme Poldi (2003) e Arenales *et al.* (2007).

4.2.2 Formulação Matemática

Consideremos o método Simplex e seu algoritmo, ambos apresentado na seção anterior. A Eq. (4.1.26) (ou os Passos 2.2 e 2.3 do algoritmo) determina o menor entre os custos reduzidos das variáveis não básicas. Observamos que pela formulação matemática de um PCEU, dada pelo PPI (3.3.1), os coeficientes de todas variáveis na função objetivo são iguais a 1. Logo, o vetor \mathbf{c} , inserido na Eq. (4.1.19) é tal que

$$\mathbf{c} = (\mathbf{c}_{\mathbf{B}}^T, \mathbf{c}_{\mathbf{N}}^T) = (1, 1, \dots, 1). \quad (4.2.1)$$

Assim, se solucionássemos um PCEU pelo método Simplex (ou seja, se dispuséssemos de todos os n padrões de corte existentes) a Regra de Dantzig ficaria

$$\hat{c}_{N_k} = \text{Mínimo} \{1 - \boldsymbol{\lambda}^T \mathbf{a}_{N_j}\} = 1 - \text{Máximo} \{\boldsymbol{\lambda}^T \mathbf{a}_{N_j}\}, \quad j = 1, 2, \dots, n - m. \quad (4.2.2)$$

Como não pretendemos listar todos os padrões para fazer essa verificação em cada iteração, podemos gerar a coluna com o menor custo reduzido possível interpretando a Regra de Dantzig como o seguinte problema de maximização:

$$\begin{aligned} &\text{Maximizar } \phi(\mathbf{a}) = \lambda_1 \alpha_1 + \lambda_2 \alpha_2 + \dots + \lambda_m \alpha_m \\ &\text{sujeito a } l_1 \alpha_1 + l_2 \alpha_2 + \dots + l_m \alpha_m \leq L \\ &\mathbf{a} \in \mathbb{Z}_+^m \quad i = 1, 2, \dots, m. \end{aligned} \quad (4.2.3)$$

A função objetivo é justamente $\boldsymbol{\lambda}^T \mathbf{a}$, onde $\mathbf{a} = (\alpha_1, \alpha_2, \dots, \alpha_m)$ é a coluna gerada. As restrições do problema (4.2.3) representam as propriedades que permitem considerar \mathbf{a} como um padrão de corte.

É a substituição dos Passos 2.2 e 2.3 pelo subproblema acima que consiste o Método de Geração de Colunas para o PCEU. O subproblema é conhecido na literatura como *Problema da Mochila*, o qual será abordado na Seção 4.3. A seguir, o algoritmo do MGC será apresentado, segundo Poldi (2003).

4.2.3 Algoritmo Geração de Colunas para um Problema de Corte de Estoque Unidimensional

Algoritmo: *MGC para PCEU*

Fase I: Determine a matriz \mathbf{B} , formada pelos padrões de corte homogêneos, dados pela Eq. (3.2.7):

$$\mathbf{B} = [\mathbf{a}_1 \dots \mathbf{a}_l \dots \mathbf{a}_m] = \begin{bmatrix} \alpha_{11} & \dots & 0 & \dots & 0 \\ \vdots & \ddots & \dots & \dots & \vdots \\ 0 & \dots & \alpha_{ll} & \dots & 0 \\ \vdots & \dots & \dots & \ddots & \vdots \\ 0 & \dots & 0 & \dots & \alpha_{mm} \end{bmatrix}$$

Faça Iteração = 1.

Fase II:

Passo 1: Cálculo da solução básica atual: $\mathbf{B}\mathbf{x}_B = \mathbf{d}$.

Passo 2: Cálculo do vetor multiplicador simplex: $\mathbf{B}^T \boldsymbol{\lambda} = (1, 1, \dots, 1)$.

Passo 3: Resolução do *Problema da Mochila*, a fim de gerar uma coluna para entrar na matriz básica \mathbf{B} :

$$\text{Maximizar } \phi(\mathbf{a}) = \lambda_1 \alpha_1 + \lambda_2 \alpha_2 + \dots + \lambda_m \alpha_m$$

$$\text{sujeito a } l_1 \alpha_1 + l_2 \alpha_2 + \dots + l_m \alpha_m \leq L$$

$$\mathbf{a} \in \mathbb{Z}_+^m \quad i = 1, 2, \dots, m.$$

Passo 4: {Teste de Otimalidade}

Se $1 - \phi(\mathbf{a}) \geq 1$, então: *pare {solução na iteração atual é ótima}*.

Caso contrário, prossiga com

4.1) Cálculo da direção simplex: $\mathbf{B}\mathbf{y} = \mathbf{a}$

4.2) Determinação do passo e variável a sair da base:

Se $\mathbf{y} \leq \mathbf{0}$, então: *pare: {problema não tem solução ótima finita: } $f(\mathbf{x}) \rightarrow -\infty$*

Caso contrário, faça $\mathbf{a} = \mathbf{a}_{m+1}$ e determine a variável a sair da base pela razão mínima:

$$\hat{\varepsilon} = \frac{\hat{x}_{B_i}}{y_i} = \text{Mínimo} \left\{ \frac{\hat{x}_{B_i}}{y_i} \mid y_i > 0, i = 1, \dots, m \right\} \text{ (a variável } x_{B_i} \text{ sai da base).}$$

3.3) {Atualização}

Matriz básica: $\mathbf{B} = [\mathbf{a}_1 \dots \mathbf{a}_{m+1} \dots \mathbf{a}_m]$

Iteração = Iteração + 1

retorne ao *Passo 1*

4.3 PROBLEMA DA MOCHILA (KNAPSACK PROBLEM - KP)

4.3.1 Introdução

O Problema da Mochila (KP) trata-se de um problema de Otimização Combinatória que está intimamente relacionado com um grande número de outros modelos de programação. Como o nome sugere, podemos interpretar como o desafio de encher uma mochila sem ultrapassar um determinado limite de peso, otimizando o valor do produto carregado.

Na seção anterior, vimos que o MGC utiliza-se da resolução de um subproblema para a geração de um novo padrão de corte que diminua o valor da função objetivo. Tal subproblema, descrito pelo problema (4.2.3) é um *Problema da Mochila*, onde as barras em estoque de tamanho L podem ser entendidas como mochilas que devem ser preenchidas de maneira ótima pelas peças de comprimento menor.

4.3.2 Formulação Matemática

Definimos o Problema da mochila, conforme Rios (2017), da seguinte forma:

Definição 4.3.1. Dado um conjunto de m itens, em que cada item i , $i = 1, 2, \dots, m$, tem um peso l_i e um valor de utilidade c_i , determinar um subconjunto tal que a soma dos pesos dos elementos seja a maior possível dentro de um certo limite (capacidade).

Interpretando a definição para o PCEU, os pesos l_i de cada item são os tamanhos de cada peça, os valores de utilidade c_i são os coeficientes do vetor multiplicador simplex, ou seja, as componentes da solução básica dual. Também, a capacidade é vista como o comprimento L

das barras em estoque. Escrevemos as informações acima como o PPI abaixo:

$$\begin{aligned} \text{Maximizar } \phi(\mathbf{a}) &= \lambda_1 \alpha_1 + \lambda_2 \alpha_2 + \cdots + \lambda_m \alpha_m \\ \text{sujeito a } l_1 \alpha_1 + l_2 \alpha_2 + \cdots + l_m \alpha_m &\leq L \\ \mathbf{a} &\in \mathbb{Z}_+^m \quad i = 1, 2, \dots, m. \end{aligned} \tag{4.3.1}$$

Ao restringirmos o domínio das variáveis para o conjunto dos números inteiros, o problema (4.3.1) se trata de um *Problema da Mochila Inteiro*, que é classificado, segundo Garey e Johnson (1979), como um problema NP-difícil. Já o *Problema da Mochila Contínuo* pode ser solucionado com exatidão pelo Método Simplex ou pelo método de pontos interiores. Como discorrem Goldberg e Luna (2005), os KP's provavelmente são os melhores exemplos quando são comparadas as dificuldades para se obter a solução ótima de um problema inteiro e sua versão contínua, uma vez que somente uma restrição é considerada.

Na literatura, existem vários métodos para a resolução do Problema da Mochila Inteiro, como por exemplo: enumeração implícita, programação dinâmica e heurísticas. Na próxima seção, iremos abordar o método da enumeração implícita, também conhecido como *Branch-and-Bound*. Outras variações do PK podem ser consultadas em Goldberg e Luna (2005).

4.4 BRANCH-AND-BOUND

4.4.1 Introdução

O *Branch-and-Bound* é uma ferramenta muito poderosa para a resolução de problemas na área de Otimização Inteira. Em muitas situações práticas, a solução de um problema de programação linear (PPL) não consegue reproduzir com fidelidade o contexto no qual a problemática se desenvolveu. Isso se deve pelo fato das componentes do vetor solução pertencerem ao domínio dos números reais, o que pode fazer com que a resposta obtida, eventualmente, não possua uma interpretação lógica. Podemos restringir o domínio das variáveis do problema para o conjunto dos números inteiros e assim, teremos um problema de programação inteira (PPI), que possui diversos métodos de resolução, dentre eles o *Branch-and-Bound*, também conhecido como algoritmo de separação (divisão ou ramificação) e avaliação.

Em 1954, Dantzig *et al.* (1954) publicou seu trabalho, no qual resolvia o problema do caixeiro viajante (ou *Travelling salesman problem - TSP*) no caso em que haviam 49 cidades de parada. Esse foi o grande marco da Programação Inteira. A primeira versão completa do

método *Branch-and-Bound* ocorreu no algoritmo para um TSP descrito em 1958, na tese de Ph.D de Eastman (1958). A partir de 1960, com o trabalho de Land e Doig (1960), publicado pela *Econometrica*, o *Branch-and-Bound* ganhou notoriedade como uma ferramenta para a resolução de PPI's. Uma curiosidade é que o nome como conhecido na atualidade surgiu apenas em 1963, com o trabalho de Little *et al.* (1963). Mais informações podem ser consultadas em Cook (2012).

Antes de apresentarmos a formulação matemática do método *Branch-and-Bound*, vamos fazer a seguinte definição:

Definição 4.4.1 (*Envoltória Convexa*). Seja $S = \{x_1, x_2, \dots, x_n\} \subset \mathbb{E}^n$. A *envoltória convexa* de S , ou $\text{Conv } S$, é a interseção de todos os conjuntos convexos que contêm S , ou seja, é o conjunto de todas as combinações convexas dos elementos de S . Podemos representar em notação de conjuntos por:

$$\text{Conv } S = \left\{ x \mid x = \sum_{j=1}^n \lambda_j x_j, \text{ em que } \sum_{j=1}^n \lambda_j = 1, \lambda_j \geq 0, x_j \in S \right\}.$$

4.4.2 Formulação Matemática

O método denominado de *Branch-and-Bound* (B&B) baseia-se na ideia de desenvolver uma enumeração inteligente dos pontos candidatos à solução ótima inteira de um problema. O termo *branch* refere-se ao fato de que o método efetua partições no espaço das soluções e o termo *bound* ressalta que a prova da otimalidade da solução utiliza-se de limites calculados ao longo da enumeração. A seguir são descritos alguns conceitos necessários para a compreensão do método conforme Goldbarg e Luna (2005) e Arenales *et al.* (2007).

Sejam $V^*(P)$ e $V^*(\bar{P})$ os valores ótimos das funções objetivo dos problemas (P) e (\bar{P}) , respectivamente dados por

$$(P) = \text{Maximizar } \{ \mathbf{c}^T \mathbf{x} \mid \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}, \mathbf{x} \in \mathbb{Z}^+ \} \quad (4.4.1)$$

$$(\bar{P}) = \text{Maximizar } \{ \mathbf{c}^T \mathbf{x} \mid \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}, \mathbf{x} \in \mathbb{R}^+ \} \quad (4.4.2)$$

Note que (\bar{P}) se trata da relaxação do problema (P) e por isso o valor ótimo do problema inteiro não pode ser maior que o valor ótimo do problema relaxado, ou seja,

$$V^*(P) \leq V^*(\bar{P}). \quad (4.4.3)$$

Considerando ainda qualquer solução factível \mathbf{x} de (P) e chamando de $V(\mathbf{x})$ o valor da

função objetivo no ponto \mathbf{x} , temos

$$V(\mathbf{x}) \leq V^*(P) \leq V^*(\bar{P}) \quad (4.4.4)$$

e dessa forma, $V^*(\bar{P})$ é um limite superior para (P) .

Se $\bar{\mathbf{x}}$ é a solução ótima de (\bar{P}) tal que sua j -ésima componente, \bar{x}_j , é não inteira, podemos estabelecer as seguintes desigualdades para toda solução factível \mathbf{x} de (P) :

$$x_j \leq \lfloor \bar{x}_j \rfloor \quad (4.4.5)$$

$$x_j \geq \lfloor \bar{x}_j \rfloor + 1 = \lceil \bar{x}_j \rceil. \quad (4.4.6)$$

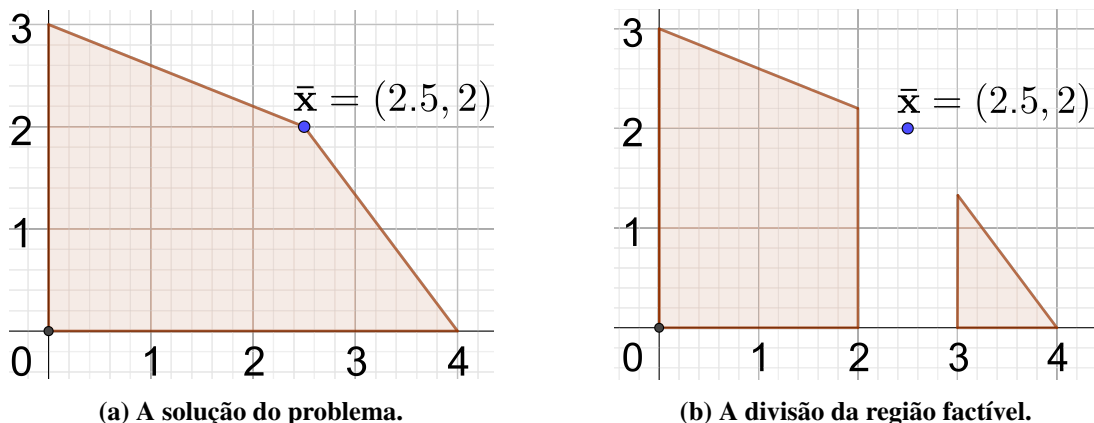
A desigualdade (4.4.5), quando acrescentada ao conjunto de restrições do problema (P) , gera um novo problema, que será chamado *subproblema* (P_1) de (P) . Neste caso, fazemos com que a j -ésima coordenada de x , x_j , seja menor ou igual ao maior inteiro menor ou igual a \bar{x}_j .

Por outro lado, a desigualdade (4.4.6), quando acrescentada ao conjunto de restrições de (P) , gera o subproblema (P_2) de (P) , no qual fazemos com que x_j seja maior ou igual ao menor inteiro maior ou igual a \bar{x}_j .

Dessa maneira, o problema (P) é subdividido em (P_1) e (P_2) . Observemos que as regiões factíveis desses subproblemas são disjuntas e que a envoltória convexa de $(P^1) \cup (P^2)$ está estritamente contida na envoltória de (P) , ou seja, $\text{Conv}((P^1) \cup (P^2)) \subset \text{Conv}((P))$.

Nas figuras abaixo, apresentamos um exemplo simples de como ocorre a subdivisão de um problema. Suponha que para um dado problema relaxado, como em (4.4.2), seja encontrada a solução ótima $\bar{\mathbf{x}} = (2.5, 2)$. Então, aplicamos as Eqs. (4.4.5) e (4.4.6) em relação à $x_1 = 2.5$, ou seja, fazemos $x_1 \leq 2$ e $x_1 \geq 3$ para produzir dois novos subproblemas P_1 e P_2 , cujas regiões viáveis de suas respectivas relaxações são ilustradas a seguir:

Figura 10 – Região factível do problema em questão (em vermelho) e sua solução ótima.



Fonte: Autoria própria

Agora, vamos apresentar um exemplo numérico, proposto em Chvátal (1983), onde aplicaremos os conceitos do método *Branch-and-Bound*, seguidos de suas explicações.

Exemplo 4.4.1. Considere o seguinte Problema da Mochila Inteiro:

$$\begin{aligned} (P) \text{ Maximizar } & f(\mathbf{x}) = 4x_1 + 5x_2 + 5x_3 + 2x_4 \\ \text{sujeito a } & 33x_1 + 49x_2 + 51x_3 + 22x_4 \leq 120 \\ & x_1, x_2, x_3, x_4 \in \mathbb{Z}^+, \end{aligned} \quad (4.4.7)$$

A solução ótima para (\bar{P}) , ou seja, o problema (P) relaxado, é $\bar{\mathbf{x}} = (3.636, 0, 0, 0)$, com $f(\bar{\mathbf{x}}) = 14.545$. Desenvolvemos a ideia de separação da envoltória convexa em relação à variável x_1 , a partir das equações

$$x_1 \leq \lfloor 3.636 \rfloor = 3 \quad \text{ou} \quad x_1 \geq \lceil 3.636 \rceil = 4. \quad (4.4.8)$$

Com a inclusão destas restrições, o problema original será reduzido então a dois novos subproblemas, a saber:

$$\begin{aligned} (P_1) \text{ Maximizar } & f(\mathbf{x}) = 4x_1 + 5x_2 + 5x_3 + 2x_4 \\ \text{sujeito a } & 33x_1 + 49x_2 + 51x_3 + 22x_4 \leq 120 \\ & x_1 \leq 3 \\ & x_1, x_2, x_3, x_4 \in \mathbb{Z}^+, \end{aligned}$$

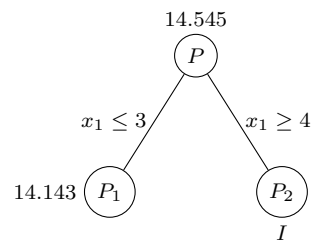
$$\begin{aligned} (P_2) \text{ Maximizar } & f(\mathbf{x}) = 4x_1 + 5x_2 + 5x_3 + 2x_4 \\ \text{sujeito a } & 33x_1 + 49x_2 + 51x_3 + 22x_4 \leq 120 \\ & x_1 \geq 4 \\ & x_1, x_2, x_3, x_4 \in \mathbb{Z}^+, \end{aligned}$$

A solução ótima de (\bar{P}_1) é o vetor $\bar{\mathbf{x}}_1 = (3, 0.429, 0, 0)$, cujo valor da função objetivo é $f(\bar{\mathbf{x}}_1) = 14.143$. Já (P_2) não tem solução, uma vez que se impusermos $x_1 \geq 4$, a restrição $33x_1 + 49x_2 + 51x_3 + 22x_4 \leq 120$ não pode ser satisfeita.

A estratégia desenvolvida acima pode ser aplicada novamente ao problema (P_1) , dando origem a dois novos subproblemas, que denotaremos por (P_3) . A essência do método é esta e podemos repetir o processo até obtermos a solução ótima (com componentes inteiras). Se, ao resolvermos o problema relaxado, tivermos uma solução com mais de uma variável não inteira, podemos escolher aquela que em módulo tem menor diferença com o inteiro mais próximo para realizar a divisão da região factível.

Observamos a natureza combinatória do B&B, já que cada problema gera dois novos subproblemas. Uma maneira de representarmos as etapas de resolução do método é por meio de uma árvore, onde cada nó corresponde a um problema e cada nível indica uma ramificação ou *branch* em relação a uma das variáveis. Na figura 11, ilustramos a árvore do exemplo até o momento. No nó P_2 indicamos a infactibilidade pela letra I . Os nós P_1 e P_2 estão um nível abaixo do nó P , ou seja, ocorreu um *branch* do problema (P).

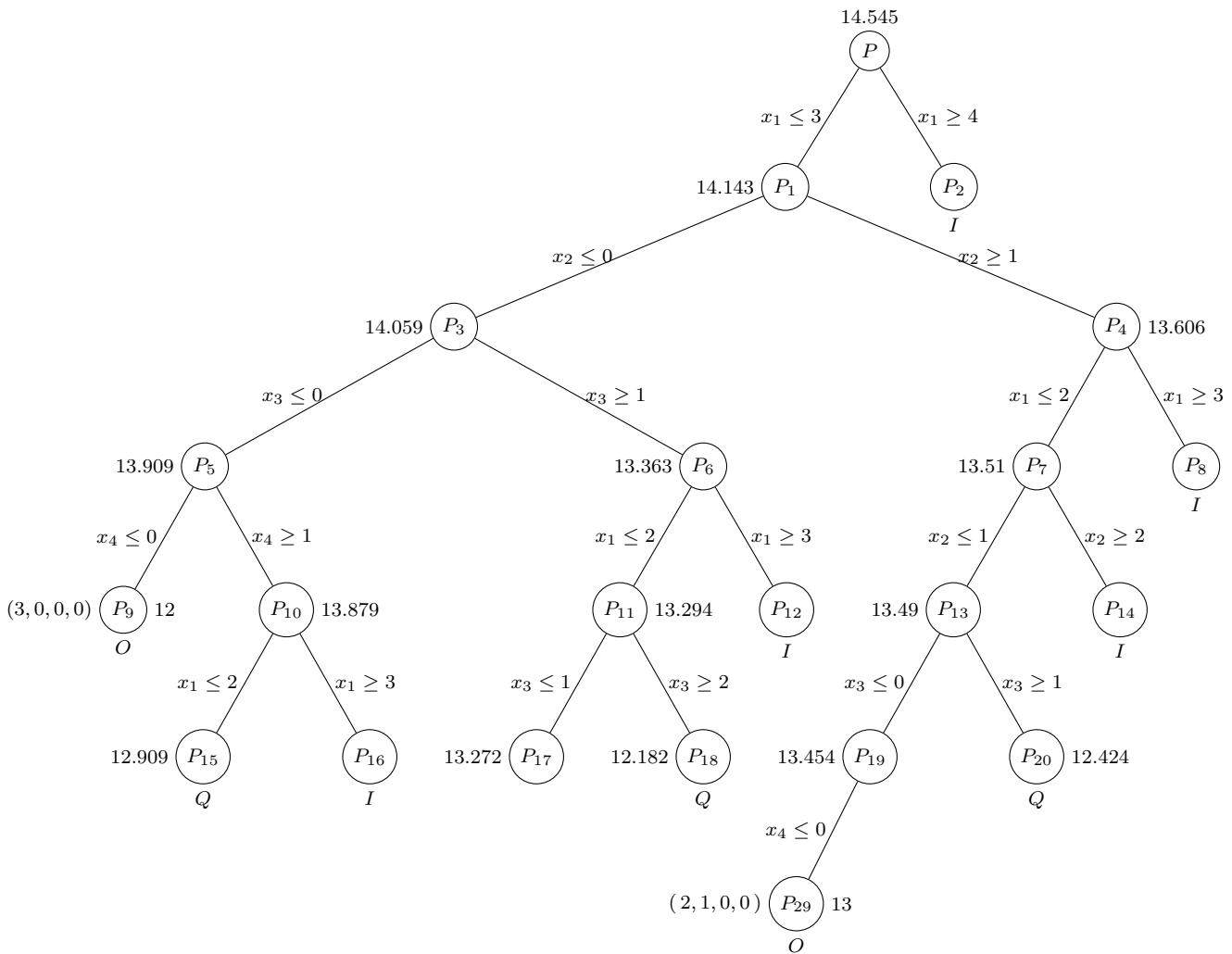
Figura 11 – Árvore do primeiro *branch* do Exemplo 4.4.1.



Fonte: Autoria Própria.

Apresentamos abaixo a árvore parcialmente completa do Exemplo 4.4.1, onde as letras I , Q , O , indicam, respectivamente, eliminação do nó por infactibilidade da solução, eliminação do nó por qualidade da solução e eliminação do nó por otimalidade da solução.

Figura 12 – Árvore do Exemplo 4.4.1.



Fonte: Autoria Própria.

Até o quinto nível de *branches*, realizamos uma busca pela solução do problema conhecida como *busca em largura*. Repare que o nó P_{29} é assim indexado, pois este seria seu índice caso o mesmo procedimento continuasse sendo empregado. A escolha por ramificar o nó P_{13} é para exibir uma propriedade muito importante do B&B, chamada *bound*, ou seja, a existência de um limite inferior.

Para entendermos o efeito do *bound*, consideremos a sequência de *branches*, dada pelos problemas $(P) \rightarrow (P_1) \rightarrow (P_4) \rightarrow (P_7) \rightarrow (P_{13}) \rightarrow (P_{19}) \rightarrow (P_{29})$. Se essa lista fosse escolhida a partir de P , então após 7 ramificações resolveríamos o problema (\bar{P}_{29}) e obteríamos a solução inteira $\bar{x}_{29} = (2, 1, 0, 0)$, com a solução objetivo valendo $f(\bar{x}_{29}) = 13$.

Ou seja, teríamos um vetor candidato à solução ótima com a função objetivo associada valendo 13. Podemos afirmar que este valor é um limite inferior para as demais soluções, uma vez que qualquer outro vetor (inteiro ou contínuo) cuja função objetivo correspondente menor

que 13 não é considerado um vetor "atrativo".

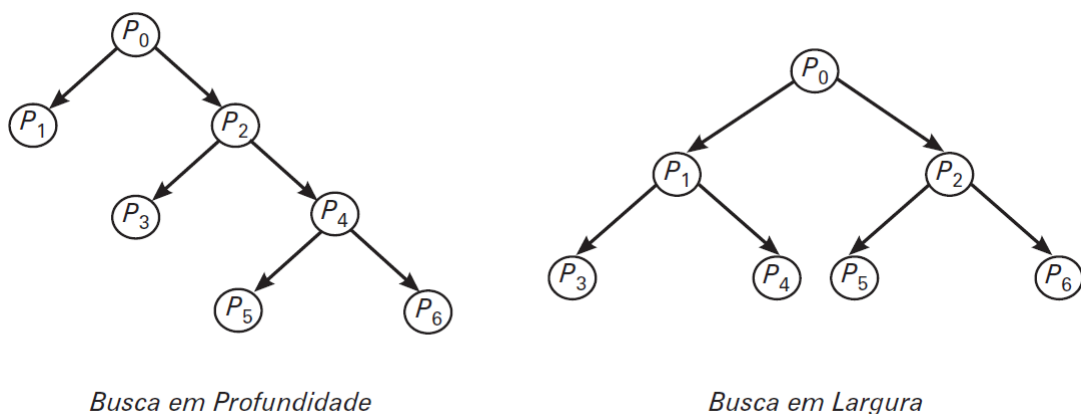
Com esta conclusão, podemos interromper as ramificações nos nós P_{15} , P_{18} e P_{19} , já que o valor da função objetivo nesses nós já é inferior à 13. Além disso, concluímos que a solução obtida em P_9 , embora inteira, não é superior que a obtida em P_{29} .

Percebemos a eficácia do *bound* ao notarmos que não precisaremos continuar com as ramificações, até mesmo em P_{17} , que ainda tem função objetivo maior que 13, pois o maior inteiro menor que o valor da função objetivo em P_{17} é justamente 13.

Em problemas práticos, o uso de limitantes inferiores para a resolução do Problema da Mochila é fundamental pois permite que muitas operações não precisem ser realizadas, reduzindo o custo computacional envolvido.

Um dos pontos fundamentais para o sucesso do B&B é a qualidade do limite gerado pela solução inteira. Fazer o desenvolvimento completo do método pode ser uma tarefa trabalhosa, já que o Problema da Mochila Inteiro é do tipo NP-difícil. Porém, abdicando da otimalidade da solução, pode-se determinar uma solução inteira razoável, aplicando estratégias de busca na árvore do B&B. Existem basicamente duas grandes estratégias de divisão ou *branch*. A figura abaixo apresenta o aspecto das árvores desenvolvidas pela *busca em profundidade* e pela *busca em largura*.

Figura 13 – Estratégias de divisão



Fonte: Goldberg e Luna (2005)

4.4.3 Algoritmo *Branch-and-Bound*

Nesta subseção, apresentaremos o algoritmo do método B&B, conforme Arenales *et al.* (2007). Antes, porém, consideremos a seguinte notação:

- $S(P_i)$: região factível do problema (P_i) ;
- $S(\bar{P}_i)$: região factível do problema (\bar{P}_i) ;
- $f(\mathbf{x}_i)$: valor ótimo do problema (P_i) , ou seja, um limitante inferior para os demais valores de \mathbf{x} ;
- $f(\bar{\mathbf{x}}_i)$: valor ótimo do problema (\bar{P}_i) , ou seja, um limitante superior para os demais valores de \mathbf{x} ;
- \mathbf{x}^* : a melhor solução até o momento presente;
- $f(\mathbf{x}^*)$: valor da função objetivo avaliada em \mathbf{x}^* ;

O nó referente ao problema (P_i) é eliminado se satisfizer alguns dos testes de eliminação abaixo:

- $S(\bar{P}_i) = \emptyset$: (P_i) é eliminado por *infectibilidade*;

O resultado decorre da relação $S(P_i) \subseteq S(\bar{P}_i)$.

- $f(\bar{\mathbf{x}}_i) \leq f(\mathbf{x}^*)$: (P_i) é eliminado por *qualidade*;

Como $f(\mathbf{x}_i) \leq f(\bar{\mathbf{x}}_i) \leq f(\mathbf{x}^*)$, segue o resultado.

- Se a solução ótima de (\bar{P}_i) é inteira: (P_i) é eliminado por *otimalidade*.

Como a solução ótima de (\bar{P}_i) é inteira, segue que $f(\bar{\mathbf{x}}_i) \leq f(\mathbf{x}_i)$.

Além disso, $f(\mathbf{x}_i) \leq f(\bar{\mathbf{x}}_i)$, portanto, $f(\mathbf{x}_i) = f(\bar{\mathbf{x}}_i)$.

Além disso, definimos o nó 0 da árvore do método como sendo o próprio problema inicial (P) e armazenaremos em uma lista L os nós *ativos*, isto é, aqueles que ainda não sofreram *branch* e que não foram eliminados pelos testes acima. Concluídas as designações, damos início ao algoritmo abaixo, conforme Arenales *et al.* (2007).

Algoritmo: {B&B}

Passo 0: {Inicialização}: Faça $f(\bar{\mathbf{x}}) = \infty$, $f(\mathbf{x}^*) = -\infty$, $\mathbf{x}^* = \emptyset$, $S = \{(P)\}$.

Passo 1: {Seleção do nó}: Selecione o nó ativo i , associado ao problema (P_i) , da lista de nós ativos L . Se a lista estiver vazia, vá para o *Passo 6*.

Passo 2: {Teste de eliminação 1}: Se a região factível de $S(\bar{P}_i)$ for vazia, vá para o *Passo 1*.

Passo 3: {Teste de eliminação 2}: Se o valor $f(\bar{\mathbf{x}}_i)$ da solução ótima de $S(\bar{P})$ é tal que $f(\bar{\mathbf{x}}_i) \leq f(\mathbf{x}^*)$, vá para o *Passo 1*.

Passo 4: {Teste de eliminação 3}: Se a solução ótima $\bar{\mathbf{X}}_i$ de $S(\bar{P})$ é inteira com valor $f(\bar{\mathbf{x}}_i)$, e se $f(\bar{\mathbf{x}}_i) > f(\mathbf{x}^*)$, atualize \mathbf{x}^* e $f(\mathbf{x}^*)$. Elimine nós ativos i da lista L , tais que $f(\bar{\mathbf{x}}_i) \leq f(\mathbf{x}^*)$, e volte para o *Passo 1*.

Passo 5: {Ramificação}: Selecione uma variável da solução ótima $\bar{\mathbf{X}}_i$ de $S(\bar{P})$ com valor não inteiro e divida P_i em dois problemas. Adicione estes problemas à lista L e vá para o *Passo 1*.

Passo 6: {Fim}: Se $f(\mathbf{x}^*) = -\infty$, não existe solução factível; caso contrário, a melhor solução encontrada até o momento \mathbf{x}^* é uma solução ótima.

5 RESULTADOS COMPUTACIONAIS

Neste capítulo apresentaremos os os resultados computacionais obtidos na resolução de um Problema de Corte de Estoque Unidimensional encontrado na literatura, com auxílio do software IBM ILOG CPLEX Optimization Studio, cuja modelagem é feita em uma linguagem própria denominada OPL (Optimization Programming Language). O algoritmo "Cutstock", disponível na biblioteca de exemplos do CPLEX, foi usado para solucionar as abordagens propostas. Na realização dos testes computacionais foi utilizado um computador com processador Intel® Core™ i7-7500U 2.70GHz, com 8 GB de memória RAM, em um sistema de 64 bits

5.1 METODOLOGIA

A metodologia desenvolvida para a obtenção dos resultados numéricos consistiu em solucionar o PCEU de três formas distintas. Em cada uma, os padrões de corte foram escolhidos para serem:

1. Os homogêneos e duas de suas variações;
2. Aleatórios e em quantidade maior que o item anterior;
3. Os gerados pelo Método de Geração de Colunas.

Nos baseamos na teoria proposta nos capítulos anteriores para não apenas exibir os resultados numéricos, mas para também identificar na construção de cada maneira de resolução as possíveis adversidades encontradas na prática e quais reajustes poderiam ser introduzidos a fim de se alcançar um método eficaz e computacionalmente operante.

Por fim, comparamos a qualidade da solução obtida em cada modo e concluímos com as considerações computacionais finais.

5.2 DEFINIÇÃO DO PROBLEMA

Valendo-se da disponibilidade de padrões de corte fixos, bem como na geração de padrões de corte, utilizaremos como exemplo numérico a seguinte situação problema:

Considere uma empresa que realiza o corte de barras de alumínio, cujo comprimento é $L = 2500$ mm, e que exista um pedido solicitando a produção de seis peças de tamanhos

distintos, $l_1 = 182 \text{ mm}$, $l_2 = 256 \text{ mm}$, $l_3 = 331 \text{ mm}$, $l_4 = 350 \text{ mm}$, $l_5 = 410 \text{ mm}$ e $l_6 = 435 \text{ mm}$, com demandas $d_1 = 500$, $d_2 = 1800$, $d_3 = 800$, $d_4 = 1000$, $d_5 = 500$ e $d_6 = 2000$, respectivamente.

5.2.1 Solução 1 - Padrões de Corte Homogêneos e Variações

Assumimos neste exemplo o caso de estoque ilimitado, ou seja, existe matéria-prima para atender a todas as demandas, os padrões de corte desejados são *completos* e não consideraremos perdas intrínsecas geradas pela espessura das lâminas (ou laser, em alguns casos). Assim, dado o enunciado acima, podemos identificar inicialmente quais são os seis padrões de corte homogêneos. Para isso, devemos fazer

$$\left\lfloor \frac{2500}{l_i} \right\rfloor, \quad i = 1, \dots, 6, \quad (5.2.1)$$

o que resulta nos seguintes padrões, listados na tabela a seguir:

Tabela 2 – Padrões de corte homogêneos para o problema.

Padrão homogêneo	Sobra
$\mathbf{a}_1 = (13, 0, 0, 0, 0, 0)$	134
$\mathbf{a}_2 = (0, 9, 0, 0, 0, 0)$	196
$\mathbf{a}_3 = (0, 0, 7, 0, 0, 0)$	183
$\mathbf{a}_4 = (0, 0, 0, 7, 0, 0)$	50
$\mathbf{a}_5 = (0, 0, 0, 0, 6, 0)$	40
$\mathbf{a}_6 = (0, 0, 0, 0, 0, 5)$	325

Fonte: Autoria Própria.

Utilizando apenas estes padrões de corte, poderíamos escrever o problema por meio do seguinte PPL

$$\text{Minimizar } f(x_1, x_2, x_3, x_4, x_5, x_6) = x_1 + x_2 + x_3 + x_4 + x_5 + x_6$$

$$\begin{bmatrix} 13 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} x_1 + \begin{bmatrix} 0 \\ 9 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} x_2 + \begin{bmatrix} 0 \\ 0 \\ 7 \\ 0 \\ 0 \\ 0 \end{bmatrix} x_3 + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 7 \\ 0 \\ 0 \end{bmatrix} x_4 + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 6 \\ 0 \end{bmatrix} x_5 + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 5 \end{bmatrix} x_6 = \begin{bmatrix} 500 \\ 1800 \\ 800 \\ 1000 \\ 500 \\ 2000 \end{bmatrix} \quad (5.2.2)$$

$$x_j \in \mathbb{R}_+ \quad j = 1, \dots, 6.$$

A solução encontrada é $\mathbf{x} = (38.462, 200, 114.29, 142.86, 83.333, 400)$ e o valor da função objetivo calculada em \mathbf{x} é 978.938. Para evitar possíveis confusões com as componente dos vetores, substituiremos a vírgula pelo ponto quando escrevermos números decimais.

Observando a coluna "Sobra" na Tabela 2, vemos que os padrões de corte homogêneos \mathbf{a}_2 , \mathbf{a}_3 e \mathbf{a}_6 não são padrões de corte completos, segundo a Definição 3.2.2, uma vez que a peça de tamanho $l_1 = 182$ mm poderia ser produzida com a sobra gerada por estes padrões. Além disso, para \mathbf{a}_6 , uma outra possibilidade seria a produção uma de tamanho $l_2 = 256$ mm.

Percebamos o efeito causado pela escolha de padrões que reduzem a perda de material ao calcular o valor da função objetivo nos dois casos explicados acima. Os resultados são exibidos na tabela abaixo, com destaque para o decréscimo do valor da função objetivo (redução da perda de material)

Tabela 3 – Padrões que reduzem as perdas dos padrões homogêneos.

Caso 1	Sobras	Caso 2	Sobras
$\mathbf{b}_1 = (13, 0, 0, 0, 0, 0)$	134	$\mathbf{c}_1 = (13, 0, 0, 0, 0, 0)$	134
$\mathbf{b}_2 = (1, 9, 0, 0, 0, 0)$	14	$\mathbf{c}_2 = (1, 9, 0, 0, 0, 0)$	14
$\mathbf{b}_3 = (1, 0, 7, 0, 0, 0)$	1	$\mathbf{c}_3 = (1, 0, 7, 0, 0, 0)$	1
$\mathbf{b}_4 = (0, 0, 0, 7, 0, 0)$	50	$\mathbf{c}_4 = (0, 0, 0, 7, 0, 0)$	50
$\mathbf{b}_5 = (0, 0, 0, 0, 6, 0)$	40	$\mathbf{c}_5 = (0, 0, 0, 0, 6, 0)$	40
$\mathbf{b}_6 = (1, 0, 0, 0, 0, 5)$	143	$\mathbf{c}_6 = (0, 1, 0, 0, 0, 5)$	69
solução $\mathbf{x} = (17.705, 155.56, 114.29, 142.86, 83.333, 400)$		solução $\mathbf{x} = (0, 200, 114.29, 142.86, 83.333, 400)$	
Valor da função objetivo = 940.476		Valor da função objetivo = 913.736	

Fonte: Autoria Própria.

Os dados acima nos indicam duas características importantes de um "bom" padrão de corte. A primeira, mais evidente, é que o uso de padrões que produzem menores sobras obviamente será mais eficiente para a minimização da perda de material do que aqueles que geram sobras maiores. Esta é uma observação muito natural, pois podemos evitar uma perda desnecessária produzindo outras peças, o que configura um melhor aproveitamento do comprimento das barras em estoque.

O segundo aspecto fica claro quando defrontamos o Caso 1 e o Caso 2. Mais especificamente, quando notamos que a única distinção entre eles é a presença do padrão \mathbf{b}_6 em um e do padrão \mathbf{c}_6 no outro. Essa sutil alteração surtiu um decréscimo de aproximadamente 26,74 barras no processo de corte. Isso se deve ao fato de que \mathbf{c}_6 produz peças de comprimentos l_2 , enquanto que \mathbf{b}_6 produz peças de comprimentos l_1 , mas como $l_1 \leq l_2$ e a demanda pelas peças do

tipo l_2 é maior do que a demanda pelas peças do tipo l_1 , o uso do padrão c_6 produziu a economia obtida.

As considerações acima motivam a identificar quais os melhores padrões de corte para o problema, isto é, aqueles que reduzem a quantidade de barras necessárias e satisfazem a demanda por cada peça. Como mencionamos previamente neste trabalho, o Método Simplex é capaz de realizar essa identificação apenas quando dispomos de todos os padrões de corte possíveis, o que é uma tarefa de difícil execução, pois na prática, n é muito maior que m . Também, no *Passo 2* do Algoritmo Simplex, mais precisamente, em 2.2, é necessário o cálculo dos $n - m$ custos reduzidos das $n - m$ colunas que constituem a matriz não básica \mathbf{N} , porém essa avaliação torna-se inviável computacionalmente. Para o problema relativamente pequeno em questão, já é possível identificar esse percalço, uma vez que existem 676 padrões de corte completos (ver Apêndice A).

5.2.2 Solução 2 - Padrões de Corte aleatórios e em quantidade maior

De posse dos 676 vetores e das sobras associadas a cada um, escolhemos arbitrariamente 20 deles e calculamos a solução e valor da função objetivo correspondente. Os dados obtidos estão expostos abaixo, na Tabela 4.

Tabela 4 – Padrões aleatórios.

$\mathbf{a}_{672} = (11, 0, 0, 1, 0, 0)$	$\mathbf{a}_{341} = (2, 1, 1, 3, 0, 1)$	$\mathbf{a}_{155} = (0, 5, 1, 1, 0, 1)$	$\mathbf{a}_{350} = (2, 1, 3, 0, 1, 1)$
$\mathbf{a}_{59} = (0, 1, 0, 5, 1, 0)$	$\mathbf{a}_{69} = (0, 1, 2, 2, 1, 1)$	$\mathbf{a}_{320} = (2, 0, 2, 0, 0, 3)$	$\mathbf{a}_{377} = (2, 3, 0, 0, 3, 0)$
$\mathbf{a}_{117} = (0, 3, 0, 0, 4, 0)$	$\mathbf{a}_{49} = (0, 0, 6, 1, 0, 0)$	$\mathbf{a}_{40} = (0, 0, 4, 2, 1, 0)$	$\mathbf{a}_{393} = (2, 5, 0, 0, 2, 0)$
$\mathbf{a}_{366} = (2, 2, 1, 1, 0, 2)$	$\mathbf{a}_{528} = (4, 2, 2, 0, 0, 1)$	$\mathbf{a}_{192} = (1, 0, 2, 2, 0, 2)$	$\mathbf{a}_{144} = (0, 4, 3, 1, 0, 0)$
$\mathbf{a}_{488} = (4, 0, 0, 1, 2, 1)$	$\mathbf{a}_{2750} = (1, 4, 0, 0, 2, 1)$	$\mathbf{a}_{358} = (2, 2, 0, 2, 0, 2)$	$\mathbf{a}_{350} = (1, 4, 1, 1, 0, 1)$
solução $\mathbf{x} = (0, 0, 20, 0, 0, 93.333, 296.67, 0, 0, 0, 0, 0, 0, 0, 0, 0, 203.33, 396.67, 0)$			
Valor da função objetivo = 1010			

Fonte: Autoria Própria.

Embora o uso de uma quantidade maior de padrões de corte à primeira vista pareça ser um bom critério para a redução do número de barras usadas, concluímos que este não é um fato necessário, pois, mesmo tomando vinte padrões de corte, o valor da função objetivo foi maior em relação às situações anteriores, quando foram utilizados apenas seis. É certo que à medida que se considera mais padrões, a probabilidade de escolher um dos melhores é maior, mas, em troca, há o aumento do número de operações computacionais a se realizar.

Convencidos assim da necessidade de um método mais eficiente, apresentamos a seguir a solução obtida pelo método de geração de colunas. A primeira iteração do método será exposta em detalhes, segundo o algoritmo da Seção 4.2.3 para uma maior compreensão da técnica, e o restante das iterações serão apresentadas na Tabela 5.

5.2.3 Solução 3 - Padrões gerados pelo Método de Geração de Colunas

Sejam \mathbf{B} a matriz não singular, constituída pelos padrões de corte homogêneos, dados na Tabela 2, \mathbf{d} o vetor demanda e \mathbf{c}_B o vetor formado pelos coeficientes da função objetivo do PPI (3.3.1):

$$\mathbf{B} = [\mathbf{a}_1 \ \mathbf{a}_2 \ \mathbf{a}_3 \ \mathbf{a}_4 \ \mathbf{a}_5 \ \mathbf{a}_6] = \begin{bmatrix} 13 & 0 & 0 & 0 & 0 & 0 \\ 0 & 9 & 0 & 0 & 0 & 0 \\ 0 & 0 & 7 & 0 & 0 & 0 \\ 0 & 0 & 0 & 7 & 0 & 0 \\ 0 & 0 & 0 & 0 & 6 & 0 \\ 0 & 0 & 0 & 0 & 0 & 5 \end{bmatrix}; \quad \mathbf{d} = \begin{bmatrix} 500 \\ 1800 \\ 800 \\ 1000 \\ 500 \\ 2000 \end{bmatrix}; \quad \mathbf{c}_B = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}.$$

Damos início ao método Simplex, calculando a solução básica \mathbf{x}_B , por meio da resolução do sistema

$$\mathbf{B}\mathbf{x}_B = \mathbf{d} \implies \mathbf{x}_B = (38.461, 200, 114.286, 142.857, 83.333, 400),$$

O valor da função objetivo em \mathbf{x}_B é 978.938. Também obtemos a solução dual $\boldsymbol{\lambda}$, computando

$$\mathbf{B}^T \boldsymbol{\lambda} = \mathbf{c}_B \implies \boldsymbol{\lambda} = (0.077, 0.11, 0.143, 0.143, 0.167, 0.2).$$

Vamos gerar uma nova coluna com custo reduzido negativo (caso contrário, a solução corrente é ótima), a fim de que o valor da função objetivo decresça se esta coluna for adicionada à matriz \mathbf{B} . Fazemos isso, solucionando o seguinte subproblema:

$$\begin{aligned} \text{Minimizar} \quad & \phi(\mathbf{a}) = 0.077 \alpha_1 + 0.11 \alpha_2 + 0.143 \alpha_3 + 0.143 \alpha_4 + 0.167 \alpha_5 + 0.2 \alpha_6 \\ \text{sujeito a} \quad & 500 \alpha_1 + 1800 \alpha_2 + 800 \alpha_3 + 1000 \alpha_4 + 500 \alpha_5 + 2000 \alpha_6 \leq 2500 \\ & \alpha_i \in \mathbb{Z}_+, \quad i = 1, \dots, 6. \end{aligned}$$

No Capítulo 4, abordamos um dos principais e mais conhecidos métodos de resolução do problema da mochila, como o proposto acima. Se trata do *Branch-and-Bound*, que realiza

uma enumeração implícita de todas as soluções do problema e faz uso de limitantes que permite o descarte de soluções piores do que a atual. Dessa forma, o método fornece o vetor mais *atrativo*, ou seja, aquele com componentes inteiras capaz de reduzir o valor da função objetivo.

A árvore de solução do problema acima apresenta dois vetores igualmente satisfatórios, a saber $\hat{\mathbf{a}}_1 = (0, 1, 0, 0, 0, 5)$ e $\hat{\mathbf{a}}_2 = (0, 0, 0, 1, 1, 4)$, cujo valor da função objetivo para ambos é 1.111. Como pontuado na Seção 4.2, heurísticas são adotadas tanto na maneira da busca operada pelo método como na escolha do vetor, quando existir mais de uma solução. Como os algoritmos computacionais em geral realizam a *busca por profundidade*, escolhemos para solução do problema da mochila o vetor $\hat{\mathbf{a}}_1 = (0, 1, 0, 0, 0, 5)$. Para auxiliar na notação, iremos denotar este vetor por \mathbf{a}_7 .

Agora, procedemos com o teste de otimalidade, isto é, verificamos que

$$\hat{c}_7 = 1 - \phi(\mathbf{a}_7) = 1 - 1.111 = -0.111 < 0.$$

Portanto, o vetor \mathbf{a}_7 deve entrar na matriz \mathbf{B} .

Para determinar qual coluna deve deixar a matriz \mathbf{B} , faremos uso dos passos 4, 5 e 6 do algoritmo Simplex. O Passo 4 determina a direção simplex, ou seja, o vetor \mathbf{y} tal que $\mathbf{B}\mathbf{y} = \mathbf{a}_7$. Nesse caso, $\mathbf{y} = (0, 0.111, 0, 0, 0, 1)$.

No Passo 5 obtemos o tamanho do passo e a coluna que deixa de compor a matriz básica fazendo

$$\begin{aligned} \hat{\varepsilon} &= \frac{\hat{x}_{B_i}}{y_i} = \text{mínimo} \left\{ \frac{\hat{x}_{B_i}}{y_i} \quad \text{tal que } y_i > 0, \quad i = 1, \dots, 6 \right\} \\ &= \text{mínimo} \left\{ \frac{200}{0.111}, \frac{400}{1} \right\} \\ &= \text{mínimo} \{1800, 400\} = 400 = \frac{\hat{x}_{B_6}}{y_6}, \end{aligned}$$

ou seja, a variável x_6 , que corresponde à coluna \mathbf{a}_6 , sai da base.

Por fim, no sétimo passo, atualizamos a matriz básica \mathbf{B} , trocando a coluna \mathbf{a}_6 pela coluna gerada \mathbf{a}_7 . Portanto,

$$\mathbf{B} = [\mathbf{a}_1 \quad \mathbf{a}_2 \quad \mathbf{a}_3 \quad \mathbf{a}_4 \quad \mathbf{a}_5 \quad \mathbf{a}_7] = \begin{bmatrix} 13 & 0 & 0 & 0 & 0 & 0 \\ 0 & 9 & 0 & 0 & 0 & 1 \\ 0 & 0 & 7 & 0 & 0 & 0 \\ 0 & 0 & 0 & 7 & 0 & 0 \\ 0 & 0 & 0 & 0 & 6 & 0 \\ 0 & 0 & 0 & 0 & 0 & 5 \end{bmatrix}.$$

Assim, conclui-se a primeira iteração do método e devemos proceder desta maneira até que a condição de otimalidade seja satisfeita. O poder do método pode ser percebido na segunda iteração, quando calculamos a solução básica \mathbf{x}_B e o valor da função objetivo associada, já que $\mathbf{x}_B = (38.461, 155.556, 114.286, 142.857, 83.333, 400)$ e $f(\mathbf{x}_B) = 934.493$, isto é, houve uma economia de aproximadamente 44.445 barras no processo de produção.

As demais iterações e resultados correspondentes estão resumidos abaixo, onde:

- B é a matriz básica;
- \mathbf{x}_B é a solução básica;
- λ é a solução dual;
- $f(\mathbf{x}_B)$ é o valor da função objetivo em \mathbf{x}_B ;
- C.G é a coluna gerada;
- \hat{c} é o custo reduzido da coluna gerada e
- Δf é a alteração no número de barras utilizadas na iteração anterior em relação à atual.

Tabela 5 – Padrões gerados com o Método de Geração de Colunas.

It	2	3
B	$[a_1 a_2 a_3 a_4 a_5 a_7]$	$[a_1 a_2 a_8 a_4 a_5 a_7]$
x_B	(38.461, 155.556, 114.286, 142.857, 83.333, 400)	(38.461, 125.926, 133.333, 142.857, 83.333, 400)
λ	(0.077, 0.111, 0.143, 0.143, 0.167, 0.178)	(0.077, 0.111, 0.129, 0.143, 0.167, 0.178)
f(x_B)	934.493	923.911
C.G.	$a_8 = (0, 2, 6, 0, 0, 0)$	$a_9 = (1, 9, 0, 0, 0, 0)$
\hat{c}	$\hat{c}_8 = -0.079$	$\hat{c}_9 = -0.077$
Δf	44.445	10.582

It	4	5
B	$[a_1 a_9 a_8 a_4 a_5 a_7]$	$[a_{10} a_9 a_8 a_4 a_5 a_7]$
x_B	(28.775, 125.926, 133.333, 142.857, 83.333, 400)	(47.285, 121.723, 133.333, 136.102, 83.333, 390.543)
λ	(0.077, 0.103, 0.132, 0.143, 0.167, 0.179)	(0.072, 0.103, 0.132, 0.143, 0.167, 0.179)
f(x_B)	914.225	912.319
C.G.	$a_{10} = (8, 1, 0, 1, 0, 1)$	$a_{11} = (0, 0, 0, 1, 1, 4)$
\hat{c}	$\hat{c}_{10} = -0.04$	$\hat{c}_{11} = -0.027$
Δf	9.686	1.906

It	6	7
B	$[a_{10} a_9 a_8 a_4 a_5 a_{11}]$	$[a_{10} a_9 a_8 a_{12} a_5 a_{11}]$
x_B	(41.784, 165.728, 133.333, 66.952, 1.741, 489.554)	(46.154, 130.769, 113.942, 116.346, 1.923, 488.462)
λ	(0.073, 0.103, 0.132, 0.143, 0.167, 0.173)	(0.073, 0.103, 0.132, 0.14, 0.167, 0.173)
f(x_B)	899.092	897.596
C.G.	$a_{12} = (0, 3, 1, 4, 0, 0)$	$a_{13} = (1, 1, 0, 0, 5, 0)$
\hat{c}	$\hat{c}_{12} = -0.013$	$\hat{c}_{13} = -0.009$
Δf	13.227	1.496

It	8	9
B	$[a_{10} a_9 a_8 a_{12} a_{13} a_{11}]$	$[a_{10} a_9 a_8 a_{12} a_{14} a_{11}]$
x_B	(45.897, 130.529, 113.934, 116.394, 2.295, 488.526)	(42.997, 129.153, 113.844, 116.938, 5.37459, 489.251)
λ	(0.073, 0.103, 0.132, 0.14, 0.165, 0.174)	(0.073, 0.103, 0.132, 0.14, 0.163, 0.174)
f(x_B)	897.575	897.557
C.G.	$a_{14} = (5, 3, 0, 0, 2, 0)$	$a_{15} = (0, 5, 0, 1, 0, 2)$
\hat{c}	$\hat{c}_{13} = -0.003$	$\hat{c}_{14} = -0.003$
Δf	0.021	0.018

It	10	11
B	$[\mathbf{a}_{10} \mathbf{a}_{15} \mathbf{a}_8 \mathbf{a}_{12} \mathbf{a}_{14} \mathbf{a}_{11}]$	$[\mathbf{a}_{16} \mathbf{a}_{15} \mathbf{a}_8 \mathbf{a}_{12} \mathbf{a}_{14} \mathbf{a}_{11}]$
\mathbf{x}_B	(26.484, 217.26, 117.854, 92.877, 57.626, 384.749)	(25.664, 215.221, 118.053, 91.681, 53.805, 392.389)
λ	(0.073, 0.102, 0.132, 0.14, 0.164, 0.174)	(0.073, 0.102, 0.132, 0.14, 0.164, 0.174)
$f(\mathbf{x}_B)$	896.849	896.814
C.G.	$\mathbf{a}_{16} = (9, 2, 0, 1, 0, 0)$	$\mathbf{a}_{17} = (0, 0, 5, 0, 1, 1)$
\hat{c}	$\hat{c}_{16} = -0.001$	$\hat{c}_{17} = -0.001$
Δf	0.708	0.035

It	12	13
B	$[\mathbf{a}_{16} \mathbf{a}_{15} \mathbf{a}_{17} \mathbf{a}_{12} \mathbf{a}_{14} \mathbf{a}_{11}]$	$[\mathbf{a}_{16} \mathbf{a}_{15} \mathbf{a}_{17} \mathbf{a}_{12} \mathbf{a}_{18} \mathbf{a}_{11}]$
\mathbf{x}_B	(46.474, 279.167, 142.521, 87.393, 16.346, 324.786)	(55.556, 286.763, 138.615, 80.644, 13.140, 321.965)
λ	(0.073, 0.102, 0.132, 0.14, 0.164, 0.174)	(0.073, 0.102, 0.132, 0.14, 0.164, 0.174)
$f(\mathbf{x}_B)$	896.688	896.683
C.G.	$\mathbf{a}_{18} = (0, 1, 2, 1, 3, 0)$	$\mathbf{a}_{19} = (1, 1, 2, 4, 0, 0)$
\hat{c}	$\hat{c}_{18} = -0.0004$	$\hat{c}_{19} = -0.0003$
Δf	0.126	0.005

It	14	15
B	$[\mathbf{a}_{16} \mathbf{a}_{15} \mathbf{a}_{17} \mathbf{a}_{19} \mathbf{a}_{18} \mathbf{a}_{11}]$	$[\mathbf{a}_{16} \mathbf{a}_{15} \mathbf{a}_{17} \mathbf{a}_{19} \mathbf{a}_{20} \mathbf{a}_{11}]$
\mathbf{x}_B	(47.255, 321.48, 120.764, 74.702, 23.389, 309.069)	(9.44, 348.48, 106.88, 38.72, 94.08, 299.04)
λ	(0.073, 0.102, 0.132, 0.14, 0.164, 0.174)	(0.073, 0.102, 0.132, 0.14, 0.164, 0.174)
$f(\mathbf{x}_B)$	896.659	896.64
C.G.	$\mathbf{a}_{20} = (4, 0, 2, 2, 1, 0)$	
\hat{c}	$\hat{c}_{20} = -0.0001$	$\nexists \mathbf{a}_k; \hat{c}_k \leq 1.0 \cdot 10^{-4}$
Δf	0.024	0.019

Fonte: Autoria Própria.

No software utilizado para os cálculos acima, impôs-se que as iterações fossem interrompidas assim que o custo reduzido de uma nova coluna gerada fosse menor que $1.0 \cdot 10^{-4}$. Assim sendo, a solução ótima relaxada para o problema proposto é

$$\bar{\mathbf{x}} = (9.44, 348.48, 106.88, 38.72, 94.08, 299.04)$$

e serão necessárias 896.64 barras para que toda a demanda seja atendida. De fato,

$$9.44 \mathbf{a}_{16} + 348.48 \mathbf{a}_{15} + 106.88 \mathbf{a}_{17} + 38.72 \mathbf{a}_{19} + 94.08 \mathbf{a}_{20} + 299.04 \mathbf{a}_{11} =$$

$$9.44 \begin{bmatrix} 9 \\ 2 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} + 348.48 \begin{bmatrix} 0 \\ 5 \\ 0 \\ 1 \\ 0 \\ 2 \end{bmatrix} + 106.88 \begin{bmatrix} 0 \\ 0 \\ 5 \\ 0 \\ 1 \\ 1 \end{bmatrix} + 38.72 \begin{bmatrix} 1 \\ 1 \\ 2 \\ 4 \\ 0 \\ 0 \end{bmatrix} + 94.08 \begin{bmatrix} 4 \\ 0 \\ 2 \\ 2 \\ 1 \\ 0 \end{bmatrix} + 299.04 \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 4 \end{bmatrix} = \begin{bmatrix} 500 \\ 1800 \\ 800 \\ 1000 \\ 500 \\ 2000 \end{bmatrix}.$$

Observamos neste ponto que o enfoque do trabalho não está na obtenção das soluções inteiras e sim na importância do uso de melhores padrões de corte para minimização da perda de material e conseqüentemente minimização dos custos envolvidos no processo. Em casos mais simples, em que restrições físicas e operacionais possam ter um pouco de flexibilidade, uma solução inteira pode ser obtida com uma heurística de arredondamento da solução obtida, como é o caso desse problema. Se tomarmos para cada componente contínua do vetor solução o menor inteiro maior ou igual à componente, teremos uma aproximação muito satisfatória para o problema inteiro. Assim, temos um novo vetor, denotado por \mathbf{x}_{int} , dado por:

$$\mathbf{x}_{int} = (10, 349, 107, 39, 95, 300),$$

cujo valor da função objetivo correspondente é $f(\mathbf{x}_{int}) = 900$.

Efetuada a mesma operação matricial acima, obtemos:

$$10 \mathbf{a}_{16} + 349 \mathbf{a}_{15} + 107 \mathbf{a}_{17} + 39 \mathbf{a}_{19} + 95 \mathbf{a}_{20} + 300 \mathbf{a}_{11} = (509, 1804, 803, 1005, 502, 2005),$$

isto é, produziremos algumas peças além da demanda. Mais especificamente,

- 9 peças de 182 mm.
- 4 peças de 256 mm.
- 3 peças de 331 mm.
- 5 peças de 350 mm.
- 2 peças de 410 mm.
- 5 peças de 435 mm.

Nesse caso, não obtivemos a solução ótima, mas muito aproximada da encontrada anteriormente, sendo também muitas vezes utilizadas como um estoque extra para aproveitamento posterior.

Finalizamos a análise, reunindo na Tabela 6 a seguir, as soluções obtidas pelas maneiras distintas de resolução um PCEU apresentadas neste capítulo.

Tabela 6 – Conclusão dos Resultados.

Abordagem	Valor da Função objetivo
Padrões Homogêneos	978.938
Caso 1 (variação dos padrões homogêneos)	940.476
Caso 2 (variação dos padrões homogêneos)	913.736
Padrões Aleatórios (20 unidades)	1010
Padrões gerados pelo Método Geração de Coluna	896.64
Solução arredondada do Método de Geração de Colunas	900

Fonte: Autoria Própria.

Podemos notar com os resultados apresentados o poder, eficiência e qualidade de solução que o Método de Geração de Colunas proporciona. Uma pequena alteração no Método Simplex, substituindo a etapa de determinação dos custos reduzidos das variáveis não básicas por um problema da mochila, que busca dentre todos os padrões de corte existentes para o modelo aquele que se adicionado à matriz das variáveis básicas, reduz a função objetivo, permitiu que muitos problemas, antes tidos como inviáveis de se solucionar, passassem a serem resolvidos em tempo computacional factível. A teoria que permeia o método e os demais subproblemas que surgem junto a ele é riquíssima e é por essa razão que existem muitos grupos de pesquisa na área.

6 CONSIDERAÇÕES FINAIS

O Problema de Corte de Estoque Unidimensional é um dos problemas clássicos da Programação Linear Inteira e, desde que foi formulado e resolvido em tempo computacional por Gilmore e Gomory, no início dos anos 60, tem sido tema constante de grupos de pesquisa em Otimização Inteira e Combinatória, Pesquisa Operacional, Ciência da Computação e demais Engenharias.

Além das complexidades teóricas que o problema apresenta, outro aspecto que o diferencia é a sua aplicabilidade na resolução de problemas industriais. Fábricas que contam com etapas de produção em que se faz necessário cortar objetos a fim de produzir itens menores, recorrem à modelagem do Problema de Corte Estoque com o intuito de aumentar seus lucros, reduzir a perda de matéria-prima ou determinar a melhor maneira que uma máquina de serras pode operar.

Discorreremos nos capítulos iniciais do presente trabalho a sobre formulação matemática do problema, em sua versão unidimensional, com estoque ilimitado, cuja função objetivo era minimizar a quantidade de barras em estoque que deveriam ser cortadas para a produção de uma certa quantidade de peças distintas de comprimento menor, de modo a cumprir a demanda por cada tipo. Identificamos que a questão fundamental para solucionar este problema, por meio de algum método, seria a identificação dos melhores padrões de corte, que, ao serem usados para cortar as barras, gerariam o menor desperdício possível de material.

Entretanto, como vimos, o método até então existente na literatura, o Simplex, não era capaz de resolvê-lo em tempo computacional factível, uma vez que era necessário listar previamente todos os padrões de corte existentes para o problema, e esta era uma tarefa muito árdua, pois em geral, nos problemas reais, dezenas ou centenas de itens diferentes a se produzir implicam em uma quantidade de padrões de cortes permissíveis na ordem de milhares ou até mesmo milhões.

O Método de Geração de Colunas, desenvolvido para o PCEU, permitiu que fosse necessário o conhecimento de apenas uma pequena quantidade de padrões de corte para se determinar a solução ótima do problema relaxado. Tomando os padrões de corte homogêneos para compor a matriz básica, em cada iteração do método, é possível gerar a melhor coluna a entrar na base, substituindo um dos passos do algoritmo Simplex, o qual retornava o valor mínimo dos custos reduzidos das $n - m$ variáveis não básicas, por um Problema da Mochila,

que por sua vez gera o padrão (coluna) com menor custo reduzido possível e que se acrescido na matriz básica, reduz o valor da função objetivo, caso o custo seja, de fato, negativo.

Para a obtenção de uma solução inteira do Problema da Mochila, foi empregado um dos métodos exatos conhecidos da literatura, o chamado método *Branch-and-Bound*, um método da enumeração implícita que se baseia na ideia de desenvolver um processo de enumeração e identificação das soluções ótimas candidatas à solução inteira do problema.

Em processos de produção na área industrial, onde surgem problemas de corte de estoque unidimensional, é imprescindível a busca por redução de custos e de desperdícios de matéria-prima. Os resultados computacionais obtidos de um exemplo numérico evidenciam a importância de se trabalhar com os melhores padrões de corte, pois em comparação com outras abordagens feitas, o valor da função objetivo tem grande redução quando solucionamos o PCEU proposto pelo Método de geração de colunas.

Atualmente, os problemas e métodos abordados neste trabalho são constantemente temas de pesquisa, devido às dificuldades em se resolver problemas inteiros de maneira eficiente e exata e também por causa das diversas variantes existentes dos temas apresentados. Na literatura, muitas heurísticas são desenvolvidas para solucionar o PCEU sem a necessidade de relaxação das variáveis. Além disso, reformulações no Problema da Mochila são estudadas a fim de melhorar a qualidade da coluna gerada. E também, existem muitas variações do problema de corte, quanto à dimensão e restrições impostas sobre os objetos em estoque e os itens a se produzir.

Ressaltamos nesse ponto que a ideia inicial para execução do trabalho seria um estudo de caso em alguma empresa, procurando modelar o processo de produção como um Problema de Corte de Estoque Unidimensional e fazer uso dos métodos apresentados neste trabalho, a fim de comparar as soluções obtidas com as soluções adotadas pela empresa, identificando possíveis melhorias nas etapas de fabricação. No entanto, devido ao contexto da pandemia, a aplicação do método ficou restrita ao estudo teórico dos conceitos matemáticos relacionados ao PCEU e com aplicação em um exemplo numérico.

Algumas perspectivas para trabalhos futuros seria desenvolver efetivamente um estudo de caso em situações reais, aprofundar o estudo de outros métodos para o Problema da Mochila, em investigações na literatura por heurísticas de resolução do Método *Branch- and-Bound* e do Problema de Corte de Estoque Unidimensional, bem como a extensão do PCEU às demais reformulações, quanto à minimização de padrões de corte e restrições adicionais ao estoque.

REFERÊNCIAS

ARENALES, M.; ARMENTANO, V. A.; MORABITO, R.; YANASSE, H. H. **Pesquisa Operacional**. 1. ed. Rio de Janeiro: Elsevier, 2007.

BAZARAA, M. S.; JARVIS, J. J.; SHERALI, H. D. **Linear programming and network flows**. 4. ed. New York: Wiley-Interscience, 2009.

BROOKS, R. L.; SMITH, C. A. B.; STONE, A. H.; TUTTE, W. T. The dissection of rectangles into squares. **Duke Mathematical Journal**, Duke University Press, v. 7, n. 1, p. 312–340, 1940.

CARVALHO, J. M. V. de; RODRIGUES, A. J. G. An lp-based approach to a two-stage cutting stock problem. **European Journal of Operational Research**, v. 84, n. 3, p. 580–589, 1995. Cutting and Packing. Disponível em: <https://www.sciencedirect.com/science/article/pii/S037722179500024K>.

CHERRI, A. C. **O problema de corte de estoque com reaproveitamento das sobras de material**. 2006. 134 p. Dissertação (Mestrado) — Instituto de Ciências Matemáticas e de Computação, São Carlos, 2006.

CHVÁTAL, V. **Linear Programming (Series of Books in the Mathematical Sciences)**. 1. ed. W. H. Freeman, 1983. Disponível em: <http://gen.lib.rus.ec/book/index.php?md5=b733680a78d62d05d78a7ab32d1f7b38>.

COOK, W. Markowitz and manne + eastman + land and doig = branch and bound. *In: Documenta Mathematica. Extra Volume ISMP. [S.l.: s.n.]*, 2012. p. 227–238.

DANTZIG, G.; FULKERSON, R.; JOHNSON, S. Solution of a large-scale traveling-salesman problem. **Journal of the Operations Research Society of America**, INFORMS, v. 2, n. 4, p. 393–410, 1954. Disponível em: <http://www.jstor.org/stable/166695>.

DANTZIG, G. B.; WOLFE, P. Decomposition principle for linear programs. **Operations Research**, v. 8, n. 1, p. 101–111, 1960. Disponível em: <https://doi.org/10.1287/opre.8.1.101>.

DYCKHOFF, H. A typology of cutting and packing problems. **European Journal of Operational Research**, v. 44, n. 2, p. 145–159, 1990. Cutting and Packing. Disponível em: <https://www.sciencedirect.com/science/article/pii/S037722179090350K>.

DYCKHOFF, H.; SCHEITHAUER, G.; TERNO, J. Cutting and packing. *In: Annotated Bibliographies in Combinatorial Optimization. [S.l.]: John Wiley and Sons*, 1997. p. 393–412.

EASTMAN, W. L. **Linear Programming with Pattern Constraints**. 1958. Tese (Doutorado) — Harvard University, Cambridge, Massachusetts, EUA, 1958.

FOERSTER, H.; WÄSCHER, G. Pattern reduction in one-dimensional cutting stock problems. **International Journal of Production Research**, Taylor & Francis, v. 38, n. 7, p. 1657–1676, 2000. Disponível em: <https://doi.org/10.1080/002075400188780>.

FORD, L. R.; FULKERSON, D. R. A suggested computation for maximal multi-commodity network flows. **Management Science**, v. 5, n. 1, p. 97–101, 1958. Disponível em: <https://doi.org/10.1287/mnsc.5.1.97>.

GAREY, M. R.; JOHNSON, D. S. **Computers and Intractability: A Guide to the Theory of NP-Completeness (Series of Books in the Mathematical Sciences)**. First edition. W. H. Freeman, 1979. Disponível em: <http://www.amazon.com/Computers-Intractability-NP-Completeness-Mathematical-Sciences/dp/0716710455>.

GILMORE, P.; GOMORY, Ralph. Multi-stage cutting stock problems of two or more dimensions. **Operations Research**, v. 13, 02 1965.

GILMORE, P. C.; GOMORY, R. E. A linear programming approach to the cutting-stock problem. **Operations Research**, v. 9, n. 6, p. 849–859, 1961. Disponível em: <https://doi.org/10.1287/opre.9.6.849>.

GILMORE, P. C.; GOMORY, R. E. A linear programming approach to the cutting stock problem—part ii. **Operations Research**, v. 11, n. 6, p. 863–888, 1963. Disponível em: <https://doi.org/10.1287/opre.11.6.863>.

GOLDBARG, M.; LUNA, H. **Otimização Combinatória e Programação Linear**. 2. ed. [S.l.]: Elsevier, 2005.

GOULIMIS, C. Optimal solutions for the cutting stock problem. **European Journal of Operational Research**, v. 44, n. 2, p. 197–208, 1990. Cutting and Packing. Disponível em: <https://www.sciencedirect.com/science/article/pii/037722179090355F>.

HAESSLER, R. W. Technical note—a note on computational modifications to the gilmore-gomory cutting stock algorithm. **Operations Research**, v. 28, n. 4, p. 1001–1005, 1980. Disponível em: <https://doi.org/10.1287/opre.28.4.1001>.

HINXMAN, A. I. The trim-loss and assortment problems: A survey. **European Journal of Operational Research**, v. 5, n. 1, p. 8–18, 1980. Disponível em: <https://www.sciencedirect.com/science/article/pii/0377221780900685>.

KANTOROVICH, Leonid V. Mathematical methods of organizing and planning production (translated from a report in russian, dated 1939). **Management Science**, v. 6, n. 4, p. 366–422, 1960. Disponível em: <https://doi.org/10.1287/mnsc.6.4.366>.

LAND, A. H.; DOIG, A. G. An automatic method of solving discrete programming problems. **Econometrica**, [Wiley, Econometric Society], v. 28, n. 3, p. 497–520, 1960. Disponível em: <http://www.jstor.org/stable/1910129>.

LIMEIRA, M. S.; YANASSE, H. H. Uma heurística para o problema de redução de padrões de corte. *In: Anais da V Oficina Nacional de Problema de Corte e Empacotamento*. São José dos Campos, SP: [s.n.], 2001. p. 137–145.

LITTLE, J. D. C.; MURTY, K. G.; SWEENEY, D. W.; KAREL, C. An algorithm for the traveling salesman problem. **Oper. Res.**, INFORMS, Linthicum, MD, USA, v. 11, n. 6, p. 972–989, dez. 1963. Disponível em: <https://doi.org/10.1287/opre.11.6.972>.

LUENBERGER, D. G.; YE, Y. **Linear and nonlinear programming**. 3. ed. [S.l.]: Springer, 2008. (International series in operations research and management science).

MANNE, A. S. Programming of economic lot sizes. **Management science**, INFORMS, v. 4, n. 2, p. 115–135, 1958.

MARQUES, J. G. O.; OLIVEIRA, W. A.; LIMEIRA, S. P.; MORETTI, A. C.; MARTIN, M. P. Variantes do método de geração de colunas de Gilmore e Gomory. *In: XLIX Simpósio Brasileiro de Pesquisa Operacional*. [S.l.: s.n.], 2017. p. 2968–2975.

PILEGGI, G. C. F. **Abordagens para otimização integrada dos problemas de geração e sequenciamento de padrões de corte**. 2002. Tese (Doutorado) — Instituto de Ciências Matemáticas e de Computação, São Carlos, 2002.

POLDI, K.C.; ARENALES, M.N. Sobre uma heurística de redução de padrões de corte para o problema de corte de estoque. **Revista TEMA – Tendências em Matemática Aplicada e Computacional**, v. 4, n. 2, p. 227–236, 2003. Disponível em: <https://doi.org/10.5540/tema.2003.04.02.0227>.

POLDI, K. C. **Algumas extensões do problema de corte de estoque**. 2003. 141 p. Dissertação (Mestrado) — Instituto de Ciências Matemáticas e de Computação, São Carlos, SP, 2003.

RIOS, A. R. **Problema de corte de estoque unidimensional com aproveitamento de sobras: resolução via meta-heurística GRASP**. 2017. 79 p. Dissertação (Mestrado) — Universidade Estadual de Campinas, Campinas, 2017.

STADTLER, H. A one-dimensional cutting stock problem in the aluminium industry and its solution. **European Journal of Operational Research**, v. 44, n. 2, p. 209–223, 1990. Cutting and Packing. Disponível em: <https://www.sciencedirect.com/science/article/pii/037722179090356G>.

WÄSCHER, G.; GAU, T. Heuristics for the integer one-dimensional cutting stock problem: A computational study. **OR Spectrum**, Springer, v. 18, p. 131–144, 1996.

WÄSCHER, G.; HAUBNER, H.; SCHUMANN, H. An improved typology of cutting and packing problems. **European Journal of Operational Research**, v. 183, n. 3, p. 1109–1130, 2007. Disponível em: <https://www.sciencedirect.com/science/article/pii/S037722170600292X>.

APÊNDICES

APÊNDICE A – PADRÕES DE CORTE GERADO PELO MÉTODO DE GERAÇÃO DE COLUNAS

Tabela 7 – Padrões de corte gerados pelo método de Geração de Colunas para o problema do Capítulo 5 - melhores padrões para o problema relaxado destacados em cinza.

$\mathbf{a}_1 = (0, 0, 0, 0, 5, 1)$	$\mathbf{a}_2 = (0, 0, 0, 0, 6, 0)$	$\mathbf{a}_3 = (0, 0, 0, 1, 1, 4)$	$\mathbf{a}_4 = (0, 0, 0, 1, 2, 3)$
$\mathbf{a}_5 = (0, 0, 0, 1, 3, 2)$	$\mathbf{a}_6 = (0, 0, 0, 1, 4, 1)$	$\mathbf{a}_7 = (0, 0, 0, 1, 5, 0)$	$\mathbf{a}_8 = (0, 0, 0, 2, 0, 4)$
$\mathbf{a}_9 = (0, 0, 0, 2, 1, 3)$	$\mathbf{a}_{10} = (0, 0, 0, 2, 2, 2)$	$\mathbf{a}_{11} = (0, 0, 0, 2, 3, 1)$	$\mathbf{a}_{12} = (0, 0, 0, 2, 4, 0)$
$\mathbf{a}_{13} = (0, 0, 0, 3, 0, 3)$	$\mathbf{a}_{14} = (0, 0, 0, 3, 1, 2)$	$\mathbf{a}_{15} = (0, 0, 0, 7, 0, 0)$	$\mathbf{a}_{16} = (0, 0, 1, 0, 1, 4)$
$\mathbf{a}_{17} = (0, 0, 1, 0, 2, 3)$	$\mathbf{a}_{18} = (0, 0, 1, 0, 3, 2)$	$\mathbf{a}_{19} = (0, 0, 1, 0, 4, 1)$	$\mathbf{a}_{20} = (0, 0, 1, 0, 5, 0)$
$\mathbf{a}_{21} = (0, 0, 1, 1, 0, 4)$	$\mathbf{a}_{22} = (0, 0, 1, 1, 1, 3)$	$\mathbf{a}_{23} = (0, 0, 1, 1, 2, 2)$	$\mathbf{a}_{24} = (0, 0, 1, 1, 3, 1)$
$\mathbf{a}_{25} = (0, 0, 1, 1, 4, 0)$	$\mathbf{a}_{26} = (0, 0, 1, 2, 0, 3)$	$\mathbf{a}_{27} = (0, 0, 1, 5, 1, 0)$	$\mathbf{a}_{28} = (0, 0, 1, 6, 0, 0)$
$\mathbf{a}_{29} = (0, 0, 2, 0, 0, 4)$	$\mathbf{a}_{30} = (0, 0, 2, 0, 1, 3)$	$\mathbf{a}_{31} = (0, 0, 2, 0, 2, 2)$	$\mathbf{a}_{32} = (0, 0, 2, 0, 3, 1)$
$\mathbf{a}_{33} = (0, 0, 2, 4, 0, 1)$	$\mathbf{a}_{34} = (0, 0, 2, 4, 1, 0)$	$\mathbf{a}_{35} = (0, 0, 2, 5, 0, 0)$	$\mathbf{a}_{36} = (0, 0, 3, 3, 0, 1)$
$\mathbf{a}_{37} = (0, 0, 3, 3, 1, 0)$	$\mathbf{a}_{38} = (0, 0, 3, 4, 0, 0)$	$\mathbf{a}_{39} = (0, 0, 4, 1, 2, 0)$	$\mathbf{a}_{40} = (0, 0, 4, 2, 0, 1)$
$\mathbf{a}_{41} = (0, 0, 4, 2, 1, 0)$	$\mathbf{a}_{42} = (0, 0, 4, 3, 0, 0)$	$\mathbf{a}_{43} = (0, 0, 5, 0, 1, 1)$	$\mathbf{a}_{44} = (0, 0, 5, 0, 2, 0)$
$\mathbf{a}_{45} = (0, 0, 5, 1, 0, 1)$	$\mathbf{a}_{46} = (0, 0, 5, 1, 1, 0)$	$\mathbf{a}_{47} = (0, 0, 5, 2, 0, 0)$	$\mathbf{a}_{48} = (0, 0, 6, 0, 0, 1)$
$\mathbf{a}_{49} = (0, 0, 6, 0, 1, 0)$	$\mathbf{a}_{50} = (0, 0, 6, 1, 0, 0)$	$\mathbf{a}_{51} = (0, 1, 0, 0, 0, 5)$	$\mathbf{a}_{52} = (0, 1, 0, 0, 1, 4)$
$\mathbf{a}_{53} = (0, 1, 0, 0, 2, 3)$	$\mathbf{a}_{54} = (0, 1, 0, 0, 3, 2)$	$\mathbf{a}_{55} = (0, 1, 0, 0, 4, 1)$	$\mathbf{a}_{56} = (0, 1, 0, 1, 0, 4)$
$\mathbf{a}_{57} = (0, 1, 0, 1, 1, 3)$	$\mathbf{a}_{58} = (0, 1, 0, 4, 2, 0)$	$\mathbf{a}_{59} = (0, 1, 0, 5, 0, 1)$	$\mathbf{a}_{60} = (0, 1, 0, 5, 1, 0)$
$\mathbf{a}_{61} = (0, 1, 0, 6, 0, 0)$	$\mathbf{a}_{62} = (0, 1, 1, 0, 0, 4)$	$\mathbf{a}_{63} = (0, 1, 1, 3, 1, 1)$	$\mathbf{a}_{64} = (0, 1, 1, 3, 2, 0)$
$\mathbf{a}_{65} = (0, 1, 1, 4, 0, 1)$	$\mathbf{a}_{66} = (0, 1, 1, 4, 1, 0)$	$\mathbf{a}_{67} = (0, 1, 1, 5, 0, 0)$	$\mathbf{a}_{68} = (0, 1, 2, 1, 3, 0)$
$\mathbf{a}_{69} = (0, 1, 2, 2, 0, 2)$	$\mathbf{a}_{70} = (0, 1, 2, 2, 1, 1)$	$\mathbf{a}_{71} = (0, 1, 2, 2, 2, 0)$	$\mathbf{a}_{72} = (0, 1, 2, 3, 0, 1)$
$\mathbf{a}_{73} = (0, 1, 2, 3, 1, 0)$	$\mathbf{a}_{74} = (0, 1, 3, 0, 3, 0)$	$\mathbf{a}_{75} = (0, 1, 3, 1, 0, 2)$	$\mathbf{a}_{76} = (0, 1, 3, 1, 1, 1)$
$\mathbf{a}_{77} = (0, 1, 3, 1, 2, 0)$	$\mathbf{a}_{78} = (0, 1, 3, 2, 0, 1)$	$\mathbf{a}_{79} = (0, 1, 3, 2, 1, 0)$	$\mathbf{a}_{80} = (0, 1, 4, 0, 0, 2)$
$\mathbf{a}_{81} = (0, 1, 4, 0, 1, 1)$	$\mathbf{a}_{82} = (0, 1, 4, 0, 2, 0)$	$\mathbf{a}_{83} = (0, 1, 4, 1, 0, 1)$	$\mathbf{a}_{84} = (0, 1, 4, 1, 1, 0)$
$\mathbf{a}_{85} = (0, 1, 5, 0, 0, 1)$	$\mathbf{a}_{86} = (0, 1, 5, 0, 1, 0)$	$\mathbf{a}_{87} = (0, 2, 0, 2, 1, 2)$	$\mathbf{a}_{88} = (0, 2, 0, 2, 2, 1)$
$\mathbf{a}_{89} = (0, 2, 0, 2, 3, 0)$	$\mathbf{a}_{90} = (0, 2, 0, 3, 0, 2)$	$\mathbf{a}_{91} = (0, 2, 0, 3, 1, 1)$	$\mathbf{a}_{92} = (0, 2, 0, 3, 2, 0)$
$\mathbf{a}_{93} = (0, 2, 0, 4, 0, 1)$	$\mathbf{a}_{94} = (0, 2, 0, 4, 1, 0)$	$\mathbf{a}_{95} = (0, 2, 1, 0, 4, 0)$	$\mathbf{a}_{96} = (0, 2, 1, 1, 0, 3)$
$\mathbf{a}_{97} = (0, 2, 1, 1, 1, 2)$	$\mathbf{a}_{98} = (0, 2, 1, 1, 2, 1)$	$\mathbf{a}_{99} = (0, 2, 1, 1, 3, 0)$	$\mathbf{a}_{100} = (0, 2, 1, 2, 0, 2)$
$\mathbf{a}_{101} = (0, 2, 1, 2, 1, 1)$	$\mathbf{a}_{102} = (0, 2, 1, 2, 2, 0)$	$\mathbf{a}_{103} = (0, 2, 1, 3, 0, 1)$	$\mathbf{a}_{104} = (0, 2, 2, 0, 0, 3)$
$\mathbf{a}_{105} = (0, 2, 2, 0, 1, 2)$	$\mathbf{a}_{106} = (0, 2, 2, 0, 2, 1)$	$\mathbf{a}_{107} = (0, 2, 2, 0, 3, 0)$	$\mathbf{a}_{108} = (0, 2, 2, 1, 0, 2)$
$\mathbf{a}_{109} = (0, 2, 2, 1, 1, 1)$	$\mathbf{a}_{110} = (0, 2, 2, 1, 2, 0)$	$\mathbf{a}_{111} = (0, 2, 3, 0, 0, 2)$	$\mathbf{a}_{112} = (0, 2, 3, 0, 1, 1)$
$\mathbf{a}_{113} = (0, 2, 3, 0, 2, 0)$	$\mathbf{a}_{114} = (0, 2, 6, 0, 0, 0)$	$\mathbf{a}_{115} = (0, 3, 0, 0, 1, 3)$	$\mathbf{a}_{116} = (0, 3, 0, 0, 2, 2)$
$\mathbf{a}_{117} = (0, 3, 0, 0, 3, 1)$	$\mathbf{a}_{118} = (0, 3, 0, 0, 4, 0)$	$\mathbf{a}_{119} = (0, 3, 0, 1, 0, 3)$	$\mathbf{a}_{120} = (0, 3, 0, 1, 1, 2)$
$\mathbf{a}_{121} = (0, 3, 0, 1, 2, 1)$	$\mathbf{a}_{122} = (0, 3, 0, 1, 3, 0)$	$\mathbf{a}_{123} = (0, 3, 0, 2, 0, 2)$	$\mathbf{a}_{124} = (0, 3, 1, 0, 0, 3)$
$\mathbf{a}_{125} = (0, 3, 1, 0, 1, 2)$	$\mathbf{a}_{126} = (0, 3, 1, 0, 2, 1)$	$\mathbf{a}_{127} = (0, 3, 1, 0, 3, 0)$	$\mathbf{a}_{128} = (0, 3, 1, 1, 0, 2)$
$\mathbf{a}_{129} = (0, 3, 1, 4, 0, 0)$	$\mathbf{a}_{130} = (0, 3, 2, 3, 0, 0)$	$\mathbf{a}_{131} = (0, 3, 3, 2, 0, 0)$	$\mathbf{a}_{132} = (0, 3, 4, 1, 0, 0)$
$\mathbf{a}_{133} = (0, 3, 5, 0, 0, 0)$	$\mathbf{a}_{134} = (0, 4, 0, 0, 0, 3)$	$\mathbf{a}_{135} = (0, 4, 0, 3, 1, 0)$	$\mathbf{a}_{136} = (0, 4, 0, 4, 0, 0)$
$\mathbf{a}_{137} = (0, 4, 1, 2, 0, 1)$	$\mathbf{a}_{138} = (0, 4, 1, 2, 1, 0)$	$\mathbf{a}_{139} = (0, 4, 1, 3, 0, 0)$	$\mathbf{a}_{140} = (0, 4, 2, 1, 0, 1)$
$\mathbf{a}_{141} = (0, 4, 2, 1, 1, 0)$	$\mathbf{a}_{142} = (0, 4, 2, 2, 0, 0)$	$\mathbf{a}_{143} = (0, 4, 3, 0, 0, 1)$	$\mathbf{a}_{144} = (0, 4, 3, 0, 1, 0)$

Tabela 7 – continuação da página anterior.

$\mathbf{a}_{145} = (0, 4, 3, 1, 0, 0)$	$\mathbf{a}_{146} = (0, 4, 4, 0, 0, 0)$	$\mathbf{a}_{147} = (0, 5, 0, 1, 0, 2)$	$\mathbf{a}_{148} = (0, 5, 0, 1, 1, 1)$
$\mathbf{a}_{149} = (0, 5, 0, 1, 2, 0)$	$\mathbf{a}_{150} = (0, 5, 0, 2, 0, 1)$	$\mathbf{a}_{151} = (0, 5, 0, 2, 1, 0)$	$\mathbf{a}_{152} = (0, 5, 0, 3, 0, 0)$
$\mathbf{a}_{153} = (0, 5, 1, 0, 0, 2)$	$\mathbf{a}_{154} = (0, 5, 1, 0, 1, 1)$	$\mathbf{a}_{155} = (0, 5, 1, 0, 2, 0)$	$\mathbf{a}_{156} = (0, 5, 1, 1, 0, 1)$
$\mathbf{a}_{157} = (0, 5, 1, 1, 1, 0)$	$\mathbf{a}_{158} = (0, 5, 2, 0, 0, 1)$	$\mathbf{a}_{159} = (0, 5, 2, 0, 1, 0)$	$\mathbf{a}_{160} = (0, 6, 0, 0, 0, 2)$
$\mathbf{a}_{161} = (0, 6, 0, 0, 1, 1)$	$\mathbf{a}_{162} = (0, 6, 0, 0, 2, 0)$	$\mathbf{a}_{163} = (0, 6, 0, 1, 0, 1)$	$\mathbf{a}_{164} = (0, 7, 0, 2, 0, 0)$
$\mathbf{a}_{165} = (0, 7, 1, 1, 0, 0)$	$\mathbf{a}_{166} = (0, 7, 2, 0, 0, 0)$	$\mathbf{a}_{167} = (0, 8, 0, 0, 0, 1)$	$\mathbf{a}_{168} = (0, 8, 0, 0, 1, 0)$
$\mathbf{a}_{169} = (0, 8, 0, 1, 0, 0)$	$\mathbf{a}_{170} = (0, 8, 1, 0, 0, 0)$	$\mathbf{a}_{171} = (1, 0, 0, 0, 0, 5)$	$\mathbf{a}_{172} = (1, 0, 0, 0, 1, 4)$
$\mathbf{a}_{173} = (1, 0, 0, 3, 2, 1)$	$\mathbf{a}_{174} = (1, 0, 0, 3, 3, 0)$	$\mathbf{a}_{175} = (1, 0, 0, 4, 0, 2)$	$\mathbf{a}_{176} = (1, 0, 0, 4, 1, 1)$
$\mathbf{a}_{177} = (1, 0, 0, 4, 2, 0)$	$\mathbf{a}_{178} = (1, 0, 0, 5, 0, 1)$	$\mathbf{a}_{179} = (1, 0, 0, 5, 1, 0)$	$\mathbf{a}_{180} = (1, 0, 1, 2, 1, 2)$
$\mathbf{a}_{181} = (1, 0, 1, 2, 2, 1)$	$\mathbf{a}_{182} = (1, 0, 1, 2, 3, 0)$	$\mathbf{a}_{183} = (1, 0, 1, 3, 0, 2)$	$\mathbf{a}_{184} = (1, 0, 1, 3, 1, 1)$
$\mathbf{a}_{185} = (1, 0, 1, 3, 2, 0)$	$\mathbf{a}_{186} = (1, 0, 1, 4, 0, 1)$	$\mathbf{a}_{187} = (1, 0, 1, 4, 1, 0)$	$\mathbf{a}_{188} = (1, 0, 2, 0, 4, 0)$
$\mathbf{a}_{189} = (1, 0, 2, 1, 0, 3)$	$\mathbf{a}_{190} = (1, 0, 2, 1, 1, 2)$	$\mathbf{a}_{191} = (1, 0, 2, 1, 2, 1)$	$\mathbf{a}_{192} = (1, 0, 2, 1, 3, 0)$
$\mathbf{a}_{193} = (1, 0, 2, 2, 0, 2)$	$\mathbf{a}_{194} = (1, 0, 2, 2, 1, 1)$	$\mathbf{a}_{195} = (1, 0, 2, 2, 2, 0)$	$\mathbf{a}_{196} = (1, 0, 2, 3, 0, 1)$
$\mathbf{a}_{197} = (1, 0, 3, 0, 0, 3)$	$\mathbf{a}_{198} = (1, 0, 3, 0, 1, 2)$	$\mathbf{a}_{199} = (1, 0, 3, 0, 2, 1)$	$\mathbf{a}_{200} = (1, 0, 3, 0, 3, 0)$
$\mathbf{a}_{201} = (1, 0, 3, 1, 0, 2)$	$\mathbf{a}_{202} = (1, 0, 3, 1, 1, 1)$	$\mathbf{a}_{203} = (1, 0, 3, 1, 2, 0)$	$\mathbf{a}_{204} = (1, 0, 4, 0, 0, 2)$
$\mathbf{a}_{205} = (1, 0, 4, 0, 1, 1)$	$\mathbf{a}_{206} = (1, 0, 4, 0, 2, 0)$	$\mathbf{a}_{207} = (1, 0, 7, 0, 0, 0)$	$\mathbf{a}_{208} = (1, 1, 0, 0, 5, 0)$
$\mathbf{a}_{209} = (1, 1, 0, 1, 2, 2)$	$\mathbf{a}_{210} = (1, 1, 0, 1, 3, 1)$	$\mathbf{a}_{211} = (1, 1, 0, 1, 4, 0)$	$\mathbf{a}_{212} = (1, 1, 0, 2, 0, 3)$
$\mathbf{a}_{213} = (1, 1, 0, 2, 1, 2)$	$\mathbf{a}_{214} = (1, 1, 0, 2, 2, 1)$	$\mathbf{a}_{215} = (1, 1, 0, 2, 3, 0)$	$\mathbf{a}_{216} = (1, 1, 0, 3, 0, 2)$
$\mathbf{a}_{217} = (1, 1, 0, 3, 1, 1)$	$\mathbf{a}_{218} = (1, 1, 1, 0, 1, 3)$	$\mathbf{a}_{219} = (1, 1, 1, 0, 2, 2)$	$\mathbf{a}_{220} = (1, 1, 1, 0, 3, 1)$
$\mathbf{a}_{221} = (1, 1, 1, 0, 4, 0)$	$\mathbf{a}_{222} = (1, 1, 1, 1, 0, 3)$	$\mathbf{a}_{223} = (1, 1, 1, 1, 1, 2)$	$\mathbf{a}_{224} = (1, 1, 1, 1, 2, 1)$
$\mathbf{a}_{225} = (1, 1, 1, 1, 3, 0)$	$\mathbf{a}_{226} = (1, 1, 1, 2, 0, 2)$	$\mathbf{a}_{227} = (1, 1, 2, 0, 0, 3)$	$\mathbf{a}_{228} = (1, 1, 2, 0, 1, 2)$
$\mathbf{a}_{229} = (1, 1, 2, 0, 2, 1)$	$\mathbf{a}_{230} = (1, 1, 2, 0, 3, 0)$	$\mathbf{a}_{231} = (1, 1, 2, 1, 0, 2)$	$\mathbf{a}_{232} = (1, 1, 2, 4, 0, 0)$
$\mathbf{a}_{233} = (1, 1, 3, 3, 0, 0)$	$\mathbf{a}_{234} = (1, 1, 4, 2, 0, 0)$	$\mathbf{a}_{235} = (1, 1, 5, 1, 0, 0)$	$\mathbf{a}_{236} = (1, 1, 6, 0, 0, 0)$
$\mathbf{a}_{237} = (1, 2, 0, 0, 0, 4)$	$\mathbf{a}_{238} = (1, 2, 0, 0, 1, 3)$	$\mathbf{a}_{239} = (1, 2, 0, 0, 2, 2)$	$\mathbf{a}_{240} = (1, 2, 0, 0, 3, 1)$
$\mathbf{a}_{241} = (1, 2, 0, 0, 4, 0)$	$\mathbf{a}_{242} = (1, 2, 0, 1, 0, 3)$	$\mathbf{a}_{243} = (1, 2, 0, 1, 1, 2)$	$\mathbf{a}_{244} = (1, 2, 0, 5, 0, 0)$
$\mathbf{a}_{245} = (1, 2, 1, 0, 0, 3)$	$\mathbf{a}_{246} = (1, 2, 1, 3, 1, 0)$	$\mathbf{a}_{247} = (1, 2, 1, 4, 0, 0)$	$\mathbf{a}_{248} = (1, 2, 2, 2, 0, 1)$
$\mathbf{a}_{249} = (1, 2, 2, 2, 1, 0)$	$\mathbf{a}_{250} = (1, 2, 2, 3, 0, 0)$	$\mathbf{a}_{251} = (1, 2, 3, 1, 0, 1)$	$\mathbf{a}_{252} = (1, 2, 3, 1, 1, 0)$
$\mathbf{a}_{253} = (1, 2, 3, 2, 0, 0)$	$\mathbf{a}_{254} = (1, 2, 4, 0, 0, 1)$	$\mathbf{a}_{255} = (1, 2, 4, 0, 1, 0)$	$\mathbf{a}_{256} = (1, 2, 4, 1, 0, 0)$
$\mathbf{a}_{257} = (1, 2, 5, 0, 0, 0)$	$\mathbf{a}_{258} = (1, 3, 0, 2, 1, 1)$	$\mathbf{a}_{259} = (1, 3, 0, 2, 2, 0)$	$\mathbf{a}_{260} = (1, 3, 0, 3, 0, 1)$
$\mathbf{a}_{261} = (1, 3, 0, 3, 1, 0)$	$\mathbf{a}_{262} = (1, 3, 0, 4, 0, 0)$	$\mathbf{a}_{263} = (1, 3, 1, 1, 1, 1)$	$\mathbf{a}_{264} = (1, 3, 1, 1, 2, 0)$
$\mathbf{a}_{265} = (1, 3, 1, 2, 0, 1)$	$\mathbf{a}_{266} = (1, 3, 1, 2, 1, 0)$	$\mathbf{a}_{267} = (1, 3, 1, 3, 0, 0)$	$\mathbf{a}_{268} = (1, 3, 2, 0, 0, 2)$
$\mathbf{a}_{269} = (1, 3, 2, 0, 1, 1)$	$\mathbf{a}_{270} = (1, 3, 2, 0, 2, 0)$	$\mathbf{a}_{271} = (1, 3, 2, 1, 0, 1)$	$\mathbf{a}_{272} = (1, 3, 2, 1, 1, 0)$
$\mathbf{a}_{273} = (1, 3, 3, 0, 0, 1)$	$\mathbf{a}_{274} = (1, 3, 3, 0, 1, 0)$	$\mathbf{a}_{275} = (1, 4, 0, 0, 1, 2)$	$\mathbf{a}_{276} = (1, 4, 0, 0, 2, 1)$
$\mathbf{a}_{277} = (1, 4, 0, 0, 3, 0)$	$\mathbf{a}_{278} = (1, 4, 0, 1, 0, 2)$	$\mathbf{a}_{279} = (1, 4, 0, 1, 1, 1)$	$\mathbf{a}_{280} = (1, 4, 0, 1, 2, 0)$
$\mathbf{a}_{281} = (1, 4, 0, 2, 0, 1)$	$\mathbf{a}_{282} = (1, 4, 1, 0, 0, 2)$	$\mathbf{a}_{283} = (1, 4, 1, 0, 1, 1)$	$\mathbf{a}_{284} = (1, 4, 1, 0, 2, 0)$
$\mathbf{a}_{285} = (1, 4, 1, 1, 0, 1)$	$\mathbf{a}_{286} = (1, 5, 0, 0, 0, 2)$	$\mathbf{a}_{287} = (1, 5, 1, 2, 0, 0)$	$\mathbf{a}_{288} = (1, 5, 2, 1, 0, 0)$
$\mathbf{a}_{289} = (1, 5, 3, 0, 0, 0)$	$\mathbf{a}_{290} = (1, 6, 0, 1, 1, 0)$	$\mathbf{a}_{291} = (1, 6, 0, 2, 0, 0)$	$\mathbf{a}_{292} = (1, 6, 1, 0, 0, 1)$
$\mathbf{a}_{293} = (1, 6, 1, 0, 1, 0)$	$\mathbf{a}_{294} = (1, 6, 1, 1, 0, 0)$	$\mathbf{a}_{295} = (1, 6, 2, 0, 0, 0)$	$\mathbf{a}_{296} = (1, 7, 0, 0, 0, 1)$
$\mathbf{a}_{297} = (1, 7, 0, 0, 1, 0)$	$\mathbf{a}_{298} = (1, 7, 0, 1, 0, 0)$	$\mathbf{a}_{299} = (1, 9, 0, 0, 0, 0)$	$\mathbf{a}_{300} = (2, 0, 0, 0, 2, 3)$
$\mathbf{a}_{301} = (2, 0, 0, 0, 3, 2)$	$\mathbf{a}_{302} = (2, 0, 0, 0, 4, 1)$	$\mathbf{a}_{303} = (2, 0, 0, 0, 5, 0)$	$\mathbf{a}_{304} = (2, 0, 0, 1, 0, 4)$

Tabela 7 – continuação da página anterior.

$\mathbf{a}_{305} = (2, 0, 0, 1, 1, 3)$	$\mathbf{a}_{306} = (2, 0, 0, 1, 2, 2)$	$\mathbf{a}_{307} = (2, 0, 0, 1, 3, 1)$	$\mathbf{a}_{308} = (2, 0, 0, 1, 4, 0)$
$\mathbf{a}_{309} = (2, 0, 0, 2, 0, 3)$	$\mathbf{a}_{310} = (2, 0, 0, 2, 1, 2)$	$\mathbf{a}_{311} = (2, 0, 0, 2, 2, 1)$	$\mathbf{a}_{312} = (2, 0, 0, 6, 0, 0)$
$\mathbf{a}_{313} = (2, 0, 1, 0, 0, 4)$	$\mathbf{a}_{314} = (2, 0, 1, 0, 1, 3)$	$\mathbf{a}_{315} = (2, 0, 1, 0, 2, 2)$	$\mathbf{a}_{316} = (2, 0, 1, 0, 3, 1)$
$\mathbf{a}_{317} = (2, 0, 1, 0, 4, 0)$	$\mathbf{a}_{318} = (2, 0, 1, 1, 0, 3)$	$\mathbf{a}_{319} = (2, 0, 1, 1, 1, 2)$	$\mathbf{a}_{320} = (2, 0, 1, 5, 0, 0)$
$\mathbf{a}_{321} = (2, 0, 2, 0, 0, 3)$	$\mathbf{a}_{322} = (2, 0, 2, 3, 1, 0)$	$\mathbf{a}_{323} = (2, 0, 2, 4, 0, 0)$	$\mathbf{a}_{324} = (2, 0, 3, 2, 0, 1)$
$\mathbf{a}_{325} = (2, 0, 3, 2, 1, 0)$	$\mathbf{a}_{326} = (2, 0, 3, 3, 0, 0)$	$\mathbf{a}_{327} = (2, 0, 4, 1, 0, 1)$	$\mathbf{a}_{328} = (2, 0, 4, 1, 1, 0)$
$\mathbf{a}_{329} = (2, 0, 4, 2, 0, 0)$	$\mathbf{a}_{330} = (2, 0, 5, 0, 0, 1)$	$\mathbf{a}_{331} = (2, 0, 5, 0, 1, 0)$	$\mathbf{a}_{332} = (2, 0, 5, 1, 0, 0)$
$\mathbf{a}_{333} = (2, 0, 6, 0, 0, 0)$	$\mathbf{a}_{334} = (2, 1, 0, 0, 0, 4)$	$\mathbf{a}_{335} = (2, 1, 0, 0, 1, 3)$	$\mathbf{a}_{336} = (2, 1, 0, 3, 2, 0)$
$\mathbf{a}_{337} = (2, 1, 0, 4, 0, 1)$	$\mathbf{a}_{338} = (2, 1, 0, 4, 1, 0)$	$\mathbf{a}_{339} = (2, 1, 0, 5, 0, 0)$	$\mathbf{a}_{340} = (2, 1, 1, 2, 1, 1)$
$\mathbf{a}_{341} = (2, 1, 1, 2, 2, 0)$	$\mathbf{a}_{342} = (2, 1, 1, 3, 0, 1)$	$\mathbf{a}_{343} = (2, 1, 1, 3, 1, 0)$	$\mathbf{a}_{344} = (2, 1, 1, 4, 0, 0)$
$\mathbf{a}_{345} = (2, 1, 2, 1, 1, 1)$	$\mathbf{a}_{346} = (2, 1, 2, 1, 2, 0)$	$\mathbf{a}_{347} = (2, 1, 2, 2, 0, 1)$	$\mathbf{a}_{348} = (2, 1, 2, 2, 1, 0)$
$\mathbf{a}_{349} = (2, 1, 2, 3, 0, 0)$	$\mathbf{a}_{350} = (2, 1, 3, 0, 0, 2)$	$\mathbf{a}_{351} = (2, 1, 3, 0, 1, 1)$	$\mathbf{a}_{352} = (2, 1, 3, 0, 2, 0)$
$\mathbf{a}_{353} = (2, 1, 3, 1, 0, 1)$	$\mathbf{a}_{354} = (2, 1, 3, 1, 1, 0)$	$\mathbf{a}_{355} = (2, 1, 4, 0, 0, 1)$	$\mathbf{a}_{356} = (2, 1, 4, 0, 1, 0)$
$\mathbf{a}_{357} = (2, 2, 0, 1, 2, 1)$	$\mathbf{a}_{358} = (2, 2, 0, 1, 3, 0)$	$\mathbf{a}_{359} = (2, 2, 0, 2, 0, 2)$	$\mathbf{a}_{360} = (2, 2, 0, 2, 1, 1)$
$\mathbf{a}_{361} = (2, 2, 0, 2, 2, 0)$	$\mathbf{a}_{362} = (2, 2, 0, 3, 0, 1)$	$\mathbf{a}_{363} = (2, 2, 0, 3, 1, 0)$	$\mathbf{a}_{364} = (2, 2, 1, 0, 1, 2)$
$\mathbf{a}_{365} = (2, 2, 1, 0, 2, 1)$	$\mathbf{a}_{366} = (2, 2, 1, 0, 3, 0)$	$\mathbf{a}_{367} = (2, 2, 1, 1, 0, 2)$	$\mathbf{a}_{368} = (2, 2, 1, 1, 1, 1)$
$\mathbf{a}_{369} = (2, 2, 1, 1, 2, 0)$	$\mathbf{a}_{370} = (2, 2, 1, 2, 0, 1)$	$\mathbf{a}_{371} = (2, 2, 2, 0, 0, 2)$	$\mathbf{a}_{372} = (2, 2, 2, 0, 1, 1)$
$\mathbf{a}_{373} = (2, 2, 2, 0, 2, 0)$	$\mathbf{a}_{374} = (2, 2, 2, 1, 0, 1)$	$\mathbf{a}_{375} = (2, 3, 0, 0, 0, 3)$	$\mathbf{a}_{376} = (2, 3, 0, 0, 1, 2)$
$\mathbf{a}_{377} = (2, 3, 0, 0, 2, 1)$	$\mathbf{a}_{378} = (2, 3, 0, 0, 3, 0)$	$\mathbf{a}_{379} = (2, 3, 0, 1, 0, 2)$	$\mathbf{a}_{380} = (2, 3, 0, 1, 1, 1)$
$\mathbf{a}_{381} = (2, 3, 1, 0, 0, 2)$	$\mathbf{a}_{382} = (2, 3, 2, 2, 0, 0)$	$\mathbf{a}_{383} = (2, 3, 3, 1, 0, 0)$	$\mathbf{a}_{384} = (2, 3, 4, 0, 0, 0)$
$\mathbf{a}_{385} = (2, 4, 0, 2, 1, 0)$	$\mathbf{a}_{386} = (2, 4, 0, 3, 0, 0)$	$\mathbf{a}_{387} = (2, 4, 1, 1, 1, 0)$	$\mathbf{a}_{388} = (2, 4, 1, 2, 0, 0)$
$\mathbf{a}_{389} = (2, 4, 2, 0, 0, 1)$	$\mathbf{a}_{390} = (2, 4, 2, 0, 1, 0)$	$\mathbf{a}_{391} = (2, 4, 2, 1, 0, 0)$	$\mathbf{a}_{392} = (2, 4, 3, 0, 0, 0)$
$\mathbf{a}_{393} = (2, 5, 0, 0, 1, 1)$	$\mathbf{a}_{394} = (2, 5, 0, 0, 2, 0)$	$\mathbf{a}_{395} = (2, 5, 0, 1, 0, 1)$	$\mathbf{a}_{396} = (2, 5, 0, 1, 1, 0)$
$\mathbf{a}_{397} = (2, 5, 0, 2, 0, 0)$	$\mathbf{a}_{398} = (2, 5, 1, 0, 0, 1)$	$\mathbf{a}_{399} = (2, 5, 1, 0, 1, 0)$	$\mathbf{a}_{400} = (2, 5, 1, 1, 0, 0)$
$\mathbf{a}_{401} = (2, 6, 0, 0, 0, 1)$	$\mathbf{a}_{402} = (2, 7, 1, 0, 0, 0)$	$\mathbf{a}_{403} = (2, 8, 0, 0, 0, 0)$	$\mathbf{a}_{404} = (3, 0, 0, 2, 3, 0)$
$\mathbf{a}_{405} = (3, 0, 0, 3, 0, 2)$	$\mathbf{a}_{406} = (3, 0, 0, 3, 1, 1)$	$\mathbf{a}_{407} = (3, 0, 0, 3, 2, 0)$	$\mathbf{a}_{408} = (3, 0, 0, 4, 0, 1)$
$\mathbf{a}_{409} = (3, 0, 0, 4, 1, 0)$	$\mathbf{a}_{410} = (3, 0, 1, 1, 2, 1)$	$\mathbf{a}_{411} = (3, 0, 1, 1, 3, 0)$	$\mathbf{a}_{412} = (3, 0, 1, 2, 0, 2)$
$\mathbf{a}_{413} = (3, 0, 1, 2, 1, 1)$	$\mathbf{a}_{414} = (3, 0, 1, 2, 2, 0)$	$\mathbf{a}_{415} = (3, 0, 1, 3, 0, 1)$	$\mathbf{a}_{416} = (3, 0, 1, 3, 1, 0)$
$\mathbf{a}_{417} = (3, 0, 2, 0, 1, 2)$	$\mathbf{a}_{418} = (3, 0, 2, 0, 2, 1)$	$\mathbf{a}_{419} = (3, 0, 2, 0, 3, 0)$	$\mathbf{a}_{420} = (3, 0, 2, 1, 0, 2)$
$\mathbf{a}_{421} = (3, 0, 2, 1, 1, 1)$	$\mathbf{a}_{422} = (3, 0, 2, 1, 2, 0)$	$\mathbf{a}_{423} = (3, 0, 2, 2, 0, 1)$	$\mathbf{a}_{424} = (3, 0, 3, 0, 0, 2)$
$\mathbf{a}_{425} = (3, 0, 3, 0, 1, 1)$	$\mathbf{a}_{426} = (3, 0, 3, 0, 2, 0)$	$\mathbf{a}_{427} = (3, 0, 3, 1, 0, 1)$	$\mathbf{a}_{428} = (3, 1, 0, 0, 2, 2)$
$\mathbf{a}_{429} = (3, 1, 0, 0, 3, 1)$	$\mathbf{a}_{430} = (3, 1, 0, 0, 4, 0)$	$\mathbf{a}_{431} = (3, 1, 0, 1, 0, 3)$	$\mathbf{a}_{432} = (3, 1, 0, 1, 1, 2)$
$\mathbf{a}_{433} = (3, 1, 0, 1, 2, 1)$	$\mathbf{a}_{434} = (3, 1, 0, 1, 3, 0)$	$\mathbf{a}_{435} = (3, 1, 0, 2, 0, 2)$	$\mathbf{a}_{436} = (3, 1, 0, 2, 1, 1)$
$\mathbf{a}_{437} = (3, 1, 0, 2, 2, 0)$	$\mathbf{a}_{438} = (3, 1, 1, 0, 0, 3)$	$\mathbf{a}_{439} = (3, 1, 1, 0, 1, 2)$	$\mathbf{a}_{440} = (3, 1, 1, 0, 2, 1)$
$\mathbf{a}_{441} = (3, 1, 1, 0, 3, 0)$	$\mathbf{a}_{442} = (3, 1, 1, 1, 0, 2)$	$\mathbf{a}_{443} = (3, 1, 1, 1, 1, 1)$	$\mathbf{a}_{444} = (3, 1, 2, 0, 0, 2)$
$\mathbf{a}_{445} = (3, 1, 3, 2, 0, 0)$	$\mathbf{a}_{446} = (3, 1, 4, 1, 0, 0)$	$\mathbf{a}_{447} = (3, 1, 5, 0, 0, 0)$	$\mathbf{a}_{448} = (3, 2, 0, 0, 0, 3)$
$\mathbf{a}_{449} = (3, 2, 0, 0, 1, 2)$	$\mathbf{a}_{450} = (3, 2, 0, 4, 0, 0)$	$\mathbf{a}_{451} = (3, 2, 1, 2, 1, 0)$	$\mathbf{a}_{452} = (3, 2, 1, 3, 0, 0)$
$\mathbf{a}_{453} = (3, 2, 2, 1, 1, 0)$	$\mathbf{a}_{454} = (3, 2, 2, 2, 0, 0)$	$\mathbf{a}_{455} = (3, 2, 3, 0, 0, 1)$	$\mathbf{a}_{456} = (3, 2, 3, 0, 1, 0)$
$\mathbf{a}_{457} = (3, 2, 3, 1, 0, 0)$	$\mathbf{a}_{458} = (3, 2, 4, 0, 0, 0)$	$\mathbf{a}_{459} = (3, 3, 0, 1, 2, 0)$	$\mathbf{a}_{460} = (3, 3, 0, 2, 0, 1)$
$\mathbf{a}_{461} = (3, 3, 0, 2, 1, 0)$	$\mathbf{a}_{462} = (3, 3, 0, 3, 0, 0)$	$\mathbf{a}_{463} = (3, 3, 1, 0, 1, 1)$	$\mathbf{a}_{464} = (3, 3, 1, 0, 2, 0)$

Tabela 7 – continuação da página anterior.

$\mathbf{a}_{465} = (3, 3, 1, 1, 0, 1)$	$\mathbf{a}_{466} = (3, 3, 1, 1, 1, 0)$	$\mathbf{a}_{467} = (3, 3, 1, 2, 0, 0)$	$\mathbf{a}_{468} = (3, 3, 2, 0, 0, 1)$
$\mathbf{a}_{469} = (3, 3, 2, 0, 1, 0)$	$\mathbf{a}_{470} = (3, 3, 2, 1, 0, 0)$	$\mathbf{a}_{471} = (3, 4, 0, 0, 0, 2)$	$\mathbf{a}_{472} = (3, 4, 0, 0, 1, 1)$
$\mathbf{a}_{473} = (3, 4, 0, 0, 2, 0)$	$\mathbf{a}_{474} = (3, 4, 0, 1, 0, 1)$	$\mathbf{a}_{475} = (3, 4, 0, 1, 1, 0)$	$\mathbf{a}_{476} = (3, 4, 1, 0, 0, 1)$
$\mathbf{a}_{477} = (3, 5, 2, 0, 0, 0)$	$\mathbf{a}_{478} = (3, 6, 0, 0, 1, 0)$	$\mathbf{a}_{479} = (3, 6, 0, 1, 0, 0)$	$\mathbf{a}_{480} = (3, 6, 1, 0, 0, 0)$
$\mathbf{a}_{481} = (3, 7, 0, 0, 0, 0)$	$\mathbf{a}_{482} = (4, 0, 0, 0, 0, 4)$	$\mathbf{a}_{483} = (4, 0, 0, 0, 1, 3)$	$\mathbf{a}_{484} = (4, 0, 0, 0, 2, 2)$
$\mathbf{a}_{485} = (4, 0, 0, 0, 3, 1)$	$\mathbf{a}_{486} = (4, 0, 0, 0, 4, 0)$	$\mathbf{a}_{487} = (4, 0, 0, 1, 0, 3)$	$\mathbf{a}_{488} = (4, 0, 0, 1, 1, 2)$
$\mathbf{a}_{489} = (4, 0, 0, 1, 2, 1)$	$\mathbf{a}_{490} = (4, 0, 0, 5, 0, 0)$	$\mathbf{a}_{491} = (4, 0, 1, 0, 0, 3)$	$\mathbf{a}_{492} = (4, 0, 1, 0, 1, 2)$
$\mathbf{a}_{493} = (4, 0, 1, 4, 0, 0)$	$\mathbf{a}_{494} = (4, 0, 2, 2, 1, 0)$	$\mathbf{a}_{495} = (4, 0, 2, 3, 0, 0)$	$\mathbf{a}_{496} = (4, 0, 3, 1, 1, 0)$
$\mathbf{a}_{497} = (4, 0, 3, 2, 0, 0)$	$\mathbf{a}_{498} = (4, 0, 4, 0, 0, 1)$	$\mathbf{a}_{499} = (4, 0, 4, 0, 1, 0)$	$\mathbf{a}_{500} = (4, 0, 4, 1, 0, 0)$
$\mathbf{a}_{501} = (4, 0, 5, 0, 0, 0)$	$\mathbf{a}_{502} = (4, 1, 0, 3, 0, 1)$	$\mathbf{a}_{503} = (4, 1, 0, 3, 1, 0)$	$\mathbf{a}_{504} = (4, 1, 0, 4, 0, 0)$
$\mathbf{a}_{505} = (4, 1, 1, 1, 2, 0)$	$\mathbf{a}_{506} = (4, 1, 1, 2, 0, 1)$	$\mathbf{a}_{507} = (4, 1, 1, 2, 1, 0)$	$\mathbf{a}_{508} = (4, 1, 1, 3, 0, 0)$
$\mathbf{a}_{509} = (4, 1, 2, 0, 1, 1)$	$\mathbf{a}_{510} = (4, 1, 2, 0, 2, 0)$	$\mathbf{a}_{511} = (4, 1, 2, 1, 0, 1)$	$\mathbf{a}_{512} = (4, 1, 2, 1, 1, 0)$
$\mathbf{a}_{513} = (4, 1, 2, 2, 0, 0)$	$\mathbf{a}_{514} = (4, 1, 3, 0, 0, 1)$	$\mathbf{a}_{515} = (4, 1, 3, 0, 1, 0)$	$\mathbf{a}_{516} = (4, 1, 3, 1, 0, 0)$
$\mathbf{a}_{517} = (4, 2, 0, 0, 2, 1)$	$\mathbf{a}_{518} = (4, 2, 0, 0, 3, 0)$	$\mathbf{a}_{519} = (4, 2, 0, 1, 0, 2)$	$\mathbf{a}_{520} = (4, 2, 0, 1, 1, 1)$
$\mathbf{a}_{521} = (4, 2, 0, 1, 2, 0)$	$\mathbf{a}_{522} = (4, 2, 0, 2, 0, 1)$	$\mathbf{a}_{523} = (4, 2, 0, 2, 1, 0)$	$\mathbf{a}_{524} = (4, 2, 1, 0, 0, 2)$
$\mathbf{a}_{525} = (4, 2, 1, 0, 1, 1)$	$\mathbf{a}_{526} = (4, 2, 1, 0, 2, 0)$	$\mathbf{a}_{527} = (4, 2, 1, 1, 0, 1)$	$\mathbf{a}_{528} = (4, 2, 1, 1, 1, 0)$
$\mathbf{a}_{529} = (4, 2, 2, 0, 0, 1)$	$\mathbf{a}_{530} = (4, 3, 0, 0, 0, 2)$	$\mathbf{a}_{531} = (4, 3, 0, 0, 1, 1)$	$\mathbf{a}_{532} = (4, 3, 3, 0, 0, 0)$
$\mathbf{a}_{533} = (4, 4, 0, 2, 0, 0)$	$\mathbf{a}_{534} = (4, 4, 1, 0, 1, 0)$	$\mathbf{a}_{535} = (4, 4, 1, 1, 0, 0)$	$\mathbf{a}_{536} = (4, 4, 2, 0, 0, 0)$
$\mathbf{a}_{537} = (4, 5, 0, 0, 0, 1)$	$\mathbf{a}_{538} = (4, 5, 0, 0, 1, 0)$	$\mathbf{a}_{539} = (4, 5, 0, 1, 0, 0)$	$\mathbf{a}_{540} = (4, 5, 1, 0, 0, 0)$
$\mathbf{a}_{541} = (5, 0, 0, 1, 3, 0)$	$\mathbf{a}_{542} = (5, 0, 0, 2, 0, 2)$	$\mathbf{a}_{543} = (5, 0, 0, 2, 1, 1)$	$\mathbf{a}_{544} = (5, 0, 0, 2, 2, 0)$
$\mathbf{a}_{545} = (5, 0, 0, 3, 0, 1)$	$\mathbf{a}_{546} = (5, 0, 0, 3, 1, 0)$	$\mathbf{a}_{547} = (5, 0, 1, 0, 2, 1)$	$\mathbf{a}_{548} = (5, 0, 1, 0, 3, 0)$
$\mathbf{a}_{549} = (5, 0, 1, 1, 0, 2)$	$\mathbf{a}_{550} = (5, 0, 1, 1, 1, 1)$	$\mathbf{a}_{551} = (5, 0, 1, 1, 2, 0)$	$\mathbf{a}_{552} = (5, 0, 1, 2, 0, 1)$
$\mathbf{a}_{553} = (5, 0, 1, 2, 1, 0)$	$\mathbf{a}_{554} = (5, 0, 2, 0, 0, 2)$	$\mathbf{a}_{555} = (5, 0, 2, 0, 1, 1)$	$\mathbf{a}_{556} = (5, 0, 2, 0, 2, 0)$
$\mathbf{a}_{557} = (5, 0, 2, 1, 0, 1)$	$\mathbf{a}_{558} = (5, 0, 2, 1, 1, 0)$	$\mathbf{a}_{559} = (5, 0, 3, 0, 0, 1)$	$\mathbf{a}_{560} = (5, 1, 0, 0, 0, 3)$
$\mathbf{a}_{561} = (5, 1, 0, 0, 1, 2)$	$\mathbf{a}_{562} = (5, 1, 0, 0, 2, 1)$	$\mathbf{a}_{563} = (5, 1, 0, 0, 3, 0)$	$\mathbf{a}_{564} = (5, 1, 0, 1, 0, 2)$
$\mathbf{a}_{565} = (5, 1, 0, 1, 1, 1)$	$\mathbf{a}_{566} = (5, 1, 0, 1, 2, 0)$	$\mathbf{a}_{567} = (5, 1, 1, 0, 0, 2)$	$\mathbf{a}_{568} = (5, 1, 1, 0, 1, 1)$
$\mathbf{a}_{569} = (5, 1, 4, 0, 0, 0)$	$\mathbf{a}_{570} = (5, 2, 0, 3, 0, 0)$	$\mathbf{a}_{571} = (5, 2, 1, 2, 0, 0)$	$\mathbf{a}_{572} = (5, 2, 2, 0, 1, 0)$
$\mathbf{a}_{573} = (5, 2, 2, 1, 0, 0)$	$\mathbf{a}_{574} = (5, 2, 3, 0, 0, 0)$	$\mathbf{a}_{575} = (5, 3, 0, 0, 2, 0)$	$\mathbf{a}_{576} = (5, 3, 0, 1, 0, 1)$
$\mathbf{a}_{577} = (5, 3, 0, 1, 1, 0)$	$\mathbf{a}_{578} = (5, 3, 0, 2, 0, 0)$	$\mathbf{a}_{579} = (5, 3, 1, 0, 0, 1)$	$\mathbf{a}_{580} = (5, 3, 1, 0, 1, 0)$
$\mathbf{a}_{581} = (5, 3, 1, 1, 0, 0)$	$\mathbf{a}_{582} = (5, 3, 2, 0, 0, 0)$	$\mathbf{a}_{583} = (5, 4, 0, 0, 0, 1)$	$\mathbf{a}_{584} = (5, 4, 0, 0, 1, 0)$
$\mathbf{a}_{585} = (5, 6, 0, 0, 0, 0)$	$\mathbf{a}_{586} = (6, 0, 0, 0, 0, 3)$	$\mathbf{a}_{587} = (6, 0, 0, 0, 1, 2)$	$\mathbf{a}_{588} = (6, 0, 0, 0, 2, 1)$
$\mathbf{a}_{589} = (6, 0, 0, 0, 3, 0)$	$\mathbf{a}_{590} = (6, 0, 0, 4, 0, 0)$	$\mathbf{a}_{591} = (6, 0, 1, 3, 0, 0)$	$\mathbf{a}_{592} = (6, 0, 2, 2, 0, 0)$
$\mathbf{a}_{593} = (6, 0, 3, 0, 1, 0)$	$\mathbf{a}_{594} = (6, 0, 3, 1, 0, 0)$	$\mathbf{a}_{595} = (6, 0, 4, 0, 0, 0)$	$\mathbf{a}_{596} = (6, 1, 0, 2, 0, 1)$
$\mathbf{a}_{597} = (6, 1, 0, 2, 1, 0)$	$\mathbf{a}_{598} = (6, 1, 0, 3, 0, 0)$	$\mathbf{a}_{599} = (6, 1, 1, 0, 2, 0)$	$\mathbf{a}_{600} = (6, 1, 1, 1, 0, 1)$
$\mathbf{a}_{601} = (6, 1, 1, 1, 1, 0)$	$\mathbf{a}_{602} = (6, 1, 1, 2, 0, 0)$	$\mathbf{a}_{603} = (6, 1, 2, 0, 0, 1)$	$\mathbf{a}_{604} = (6, 1, 2, 0, 1, 0)$
$\mathbf{a}_{605} = (6, 1, 2, 1, 0, 0)$	$\mathbf{a}_{606} = (6, 1, 3, 0, 0, 0)$	$\mathbf{a}_{607} = (6, 2, 0, 0, 0, 2)$	$\mathbf{a}_{608} = (6, 2, 0, 0, 1, 1)$
$\mathbf{a}_{609} = (6, 2, 0, 0, 2, 0)$	$\mathbf{a}_{610} = (6, 2, 0, 1, 0, 1)$	$\mathbf{a}_{611} = (6, 2, 0, 1, 1, 0)$	$\mathbf{a}_{612} = (6, 2, 1, 0, 0, 1)$
$\mathbf{a}_{613} = (6, 2, 1, 0, 1, 0)$	$\mathbf{a}_{614} = (6, 4, 0, 1, 0, 0)$	$\mathbf{a}_{615} = (6, 4, 1, 0, 0, 0)$	$\mathbf{a}_{616} = (6, 5, 0, 0, 0, 0)$
$\mathbf{a}_{617} = (7, 0, 0, 1, 0, 2)$	$\mathbf{a}_{618} = (7, 0, 0, 1, 1, 1)$	$\mathbf{a}_{619} = (7, 0, 0, 1, 2, 0)$	$\mathbf{a}_{620} = (7, 0, 0, 2, 0, 1)$
$\mathbf{a}_{621} = (7, 0, 0, 2, 1, 0)$	$\mathbf{a}_{622} = (7, 0, 0, 3, 0, 0)$	$\mathbf{a}_{623} = (7, 0, 1, 0, 0, 2)$	$\mathbf{a}_{624} = (7, 0, 1, 0, 1, 1)$

Tabela 7 – continuação da página anterior.

$\mathbf{a}_{625} = (7, 0, 1, 0, 2, 0)$	$\mathbf{a}_{626} = (7, 0, 1, 1, 0, 1)$	$\mathbf{a}_{627} = (7, 0, 1, 1, 1, 0)$	$\mathbf{a}_{628} = (7, 0, 2, 0, 0, 1)$
$\mathbf{a}_{629} = (7, 0, 2, 0, 1, 0)$	$\mathbf{a}_{630} = (7, 1, 0, 0, 0, 2)$	$\mathbf{a}_{631} = (7, 1, 0, 0, 1, 1)$	$\mathbf{a}_{632} = (7, 1, 0, 0, 2, 0)$
$\mathbf{a}_{633} = (7, 2, 0, 2, 0, 0)$	$\mathbf{a}_{634} = (7, 2, 1, 1, 0, 0)$	$\mathbf{a}_{635} = (7, 2, 2, 0, 0, 0)$	$\mathbf{a}_{636} = (7, 3, 0, 0, 0, 1)$
$\mathbf{a}_{637} = (7, 3, 0, 0, 1, 0)$	$\mathbf{a}_{638} = (7, 3, 0, 1, 0, 0)$	$\mathbf{a}_{639} = (7, 3, 1, 0, 0, 0)$	$\mathbf{a}_{640} = (8, 0, 0, 0, 0, 2)$
$\mathbf{a}_{641} = (8, 0, 1, 2, 0, 0)$	$\mathbf{a}_{642} = (8, 0, 2, 1, 0, 0)$	$\mathbf{a}_{643} = (8, 0, 3, 0, 0, 0)$	$\mathbf{a}_{644} = (8, 1, 0, 1, 0, 1)$
$\mathbf{a}_{645} = (8, 1, 0, 1, 1, 0)$	$\mathbf{a}_{646} = (8, 1, 0, 2, 0, 0)$	$\mathbf{a}_{647} = (8, 1, 1, 0, 0, 1)$	$\mathbf{a}_{648} = (8, 1, 1, 0, 1, 0)$
$\mathbf{a}_{649} = (8, 1, 1, 1, 0, 0)$	$\mathbf{a}_{650} = (8, 1, 2, 0, 0, 0)$	$\mathbf{a}_{651} = (8, 2, 0, 0, 0, 1)$	$\mathbf{a}_{652} = (8, 2, 0, 0, 1, 0)$
$\mathbf{a}_{653} = (8, 4, 0, 0, 0, 0)$	$\mathbf{a}_{654} = (9, 0, 0, 0, 1, 1)$	$\mathbf{a}_{655} = (9, 0, 0, 0, 2, 0)$	$\mathbf{a}_{656} = (9, 0, 0, 1, 0, 1)$
$\mathbf{a}_{657} = (9, 0, 0, 1, 1, 0)$	$\mathbf{a}_{658} = (9, 0, 0, 2, 0, 0)$	$\mathbf{a}_{659} = (9, 0, 1, 0, 0, 1)$	$\mathbf{a}_{660} = (9, 0, 1, 0, 1, 0)$
$\mathbf{a}_{661} = (9, 0, 1, 1, 0, 0)$	$\mathbf{a}_{662} = (9, 1, 0, 0, 0, 1)$	$\mathbf{a}_{663} = (9, 2, 0, 1, 0, 0)$	$\mathbf{a}_{664} = (9, 2, 1, 0, 0, 0)$
$\mathbf{a}_{665} = (9, 3, 0, 0, 0, 0)$	$\mathbf{a}_{666} = (10, 0, 2, 0, 0, 0)$	$\mathbf{a}_{667} = (10, 1, 0, 0, 1, 0)$	$\mathbf{a}_{668} = (10, 1, 0, 1, 0, 0)$
$\mathbf{a}_{669} = (10, 1, 1, 0, 0, 0)$	$\mathbf{a}_{670} = (10, 2, 0, 0, 0, 0)$	$\mathbf{a}_{671} = (11, 0, 0, 0, 0, 1)$	$\mathbf{a}_{672} = (11, 0, 0, 0, 1, 0)$
$\mathbf{a}_{673} = (11, 0, 0, 1, 0, 0)$	$\mathbf{a}_{674} = (11, 0, 1, 0, 0, 0)$	$\mathbf{a}_{675} = (12, 1, 0, 0, 0, 0)$	$\mathbf{a}_{676} = (13, 0, 0, 0, 0, 0)$

Fonte: Autoria Própria