FUNDAÇÃO DE ENSINO “EURÍPIDES SOARES DA ROCHA”
CENTRO UNIVERSITÁRIO EURÍPIDES DE MARÍLIA – UNIVEM
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

LUCAS HENRIQUE SANTOS DO CARMO

MODELO DE OTIMIZAÇÃO PARA PROBLEMA DE CORTE EM TUBOS

MARÍLIA
2016

LUCAS HENRIQUE SANTOS DO CARMO

MODELO DE OTIMIZAÇÃO PARA PROBLEMA DE CORTE EM TUBOS

Trabalho de Curso apresentado ao Curso de Bacharelado em Ciência da Computação da Fundação de Ensino “Eurípides Soares da Rocha”, mantenedora do Centro Universitário Eurípides de Marília – UNIVEM, como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação.

Orientador
Profº. Ms. Ricardo José Sabatine

MARÍLIA
2016

Carmo, Lucas Henrique Santos do
Modelo de otimização para problema de corte em tubos / Lucas Henrique Santos do Carmo; orientador: Profº. Ms. Ricardo José Sabatine. Marília, SP: [s.n.], 2016.
60 páginas.

Trabalho de Curso (Graduação em Ciência da Computação) - Curso de Ciência da Computação da Fundação de Ensino "Eurípides Soares da Rocha", mantenedora do Centro Universitário Eurípides de Marília - UNIVEM, Marília, 2016.

1.Problema de corte unidimensional. 2.Otimização. 3.Pesquisa Operacional.

CENTRO UNIVERSITÁRIO EURÍPIDES DE MARÍLIA - UNIVEM
MANTIDO PELA FUNDAÇÃO DE ENSINO "EURÍPIDES SOARES DA ROCHA"

BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

Lucas Henrique Santos do Carmo

MODELO DE OTIMIZAÇÃO PARA PROBLEMA DE CORTE EM TUBOS.

Banca examinadora da monografia apresentada ao Curso de Bacharelado em Ciência da Computação do UNIVEM/F.E.E.S.R., para obtenção do Título de Bacharel em Ciência da Computação.

Nota: 10 (Dez)

Orientador:Ricardo José Sabatine Ricardo Sabatine

1º.Examinador:Renata Aparecida de Carvalho Paschoal Carvalho Paschoal

2º.Examinador:Jorge Luiz Barbosa Maciel Junior

Marília, 06 de dezembro de 2016.

AGRADECIMENTOS

Ao meu pai Valdeci do Carmo, sua presença significou segurança e certeza de que não estou sozinho nessa caminhada. Minha mãe Maria de Fátima Santos do Carmo, seu cuidado e dedicação deram momentos de esperança para seguir em frente.

A minha irmã Laís Regina Santos do Carmo e ao meu cunhado Peterson Ricardo Sampaio de Oliveira pela compreensão e companheirismo com tantos ensinamentos e palavras de motivação.

A Minha namorada Júlia Rojo Bezerra que de forma especial e carinhosa me deu força e coragem nesses meses de pesquisa e elaboração do trabalho de conclusão de curso.

Sendo assim, todos os nomes citados até o momento, formam a minha família e com muita honestidade conseguem transmitir a capacidade de acreditar nos meus sonhos.

Agradeço também aos meus amigos Caio, Diogo e Felipe, irmãos de amizade que fizeram parte da minha vida e continuarão presentes com toda certeza.

Ao meu orientador Profº. Ms. Ricardo José Sabatine, pelos incentivos e suporte nas correções necessárias desta monografia e, principalmente, pelo aprendizado e conhecimento passados.

As pessoas com quem convivi ao longo desses anos no curso de Ciência da Computação, a todos os professores que foram tão importantes na minha vida acadêmica.

E finalmente agradeço a Deus, por proporcionar estes agradecimentos a todos que me ajudaram a dar mais um grande passo na minha vida promissora, além de ter me dado uma família maravilhosa e amigos sinceros.

CARMO, Lucas Henrique Santos do. **Modelo de Otimização Para Modelos de Corte em Tubos**. 2016. 58f. Trabalho de Curso. (Graduação em Ciência da Computação) - Centro Universitário Eurípides de Marília, Fundação de Ensino "Eurípides Soares da Rocha", Marília, 2016.

RESUMO

Impulsionado pela crescente globalização, o mercado industrial é caracterizado pela sua competitividade, na qual as empresas, independentemente da sua estrutura, de pequeno ou grande porte, buscam sempre se manter sólidas, procurando meios administrativos mais eficientes, como uma boa lucratividade e altas taxas de crescimento, aliados a novas tecnologias e ofertas de produtos que atendam aos requisitos dos clientes. Neste trabalho, será abordado o Problema de Corte de Estoque Unidimensional que consiste no corte de tubos em estoque para produzir uma combinação de unidades menores (itens) dentro de unidades maiores (objetos), utilizando um modelo de otimização na busca de melhores soluções em termos de custo, aproveitamento de matéria-prima e redução do tempo. O estudo do caso possui vários tubos em estoque com dois tamanhos diferentes e em quantidade limitada. Para solucionar este problema será utilizado o método heurístico algoritmo genético baseado nos conceitos de programação evolutiva aliado ao método simplex, que é um algoritmo que permite resolver problemas de Programação Linear. Serão apresentados os resultados a partir de testes computacionais com base nos dados reais coletados por Moraes, aluno que cursou engenharia de produção na UNIVEM no ano de 2014.

Palavras-chave: Problema de Corte Unidimensional, Otimização, Pesquisa Operacional.

CARMO, Lucas Henrique Santos do. **Modelo de Otimização Para Modelos de Corte em Tubos**. 2016. 58f. Trabalho de Curso. (Graduação em Ciência da Computação) - Centro Universitário Eurípides de Marília, Fundação de Ensino "Eurípides Soares da Rocha", Marília, 2016.

ABSTRACT

Driven by growing globalization, the industrial market is characterized by its competitiveness, in which companies, regardless of their structure, small or large, always seek to remain solid, looking for more efficient administrative means, such as good profitability and high rates of Growth, allied to new technologies and offers of products that meet the requirements of customers. In this paper, we will approach the One-Dimensional Stock Cutting Problem that consists of cutting pipes in stock to produce a combination of smaller units (items) within larger units (objects), using an optimization model in search of better solutions in terms Of cost, use of raw material and reduction of time. The case study has several tubes in stock with two different sizes and in limited quantity. To solve this problem will be used the heuristic genetic algorithm method based on the concepts of evolutionary programming allied to the simplex method, which is an algorithm that allows solving Linear Programming problems. The results will be presented from computational tests based on the actual data collected by Moraes, a student who studied production engineering at UNIVEM in 2014.

Keywords: Cutting Problem Unidimensional, Optimization, Operations Research.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

PI	*Programação Inteira*
PL	*Programação Linear*
PK	*Knapsack Problem*
PO	*Pesquisa Operacional*
FFD	*First-Fit-Decreasing*
AG	*Algoritmo Genético*

SUMÁRIO

INTRODUÇÃO

Concentradas as atenções na integração do processo produtivo, a tomada de decisão está conceituada nas atividades de processos de produção, serviços terceirizados, movimentação de produtos, alocação de recursos e, como foco principal deste trabalho, a estratégia de corte da matéria-prima.

A escolha de uma solução “ótima” para um problema específico, interligado aos recursos financeiros resulta na redução de custos operacionais, proporcionando um ganho potencial na competitividade na área industrial (MORAIS, 2011).

As empresas, indiferentemente da sua estrutura, de pequeno ou grande porte visam manter-se no mercado industrial com uma boa lucratividade e altas taxas de crescimento. Para alcançar esses objetivos é necessário aplicar à redução de custos e/ou a otimização de alguns dos seus processos.

Os problemas de corte de estoque de acordo com Pinto (2004) são processos para resolução de problemas encontrados em produção de várias indústrias, tais como, indústrias metalúrgicas, têxtil, moveleiras, etc.

O mecanismo do processo de corte inclui decisões sobre como cortar a matéria-prima principal (objeto) em peças que compõe os produtos finais (itens) (Vanzela, Rangel e Araujo, 2012).

Para se chegar neste planejamento em cortar o objeto, até gerar a produção dos itens, de acordo com Lachtermacher (2009), é necessária uma tomada de decisão, onde uma das soluções básicas é a utilização de modelos matemáticos. Essa utilização atribui várias vantagens e bons resultados no processo de tomada de decisão.

Apesar do problema de corte ser antigo, a implementação de soluções sofisticadas nem sempre são adotadas no dia-a-dia da empresa (SILVA, 2008).

Na prática estes processos encontrados, em geral, são difíceis de serem tratados devido ao grande número de variáveis envolvidas e devido à restrição de integralidade das variáveis. O problema no processo de corte de tubos se torna cada vez mais preocupante, quando as perdas de matéria-prima produzem sobras (retalhos e/ou sucatas) que tendem a ser cada vez mais expressivas elevando o custo do produto para empresa.

Para reduzir estes possíveis prejuízos, a otimização provém da utilização de um método científico denominado Pesquisa Operacional, capaz de solucionar problemas de grande complexidade, auxiliando no processo de tomada de decisão (MORAIS, 2011), mediante a modelagem matemática do problema, buscando obter soluções na otimização do processo de corte.

Portanto, produzir soluções exatas em tempos computacionais razoáveis não é uma tarefa fácil. Desta forma, torna-se interessante o uso de heurísticas para sua resolução (HEIS et al., 2010).

Do ponto de vista econômico, trata-se de um problema extremamente importante devido às várias aplicações em diversos setores industriais. Do ponto de vista científico, trata-se de um problema muito interessante devido sua complexidade computacional (TEMPONI, 2007), buscando métodos eficientes que podem ser notadas pelo volume de publicações nos últimos anos.

O objetivo deste trabalho foi desenvolver uma heurística alternativa, que, posteriormente, aliada ao método simplex, resultará uma solução viável que possivelmente poderá ocasionar em uma solução ótima no modelo de otimização para o problema de corte de estoque com padrões de corte unidimensional relacionado ao corte de tubos, a fim de minimizar perda de matéria-prima que serão devidamente apresentados nesta literatura.

Objetivos

O objetivo principal foi desenvolver um software para resolução do problema de corte unidimensional para tubos a partir do algoritmo genético junto ao algoritmo simplex, baseada nas principais referências encontrada em literatura sobre o assunto.

Com o foco de desenvolver um modelo de otimização, gerando padrões de corte para facilitar no processo de corte.

O objetivo específico deste trabalho é realizar os testes computacionais com padrões de corte gerados pelo algoritmo genético juntamente ao método simplex, comparando com a solução desenvolvida por Moraes no ano de 2014, aluno que cursou engenharia de produção na UNIVEM, tendo apenas como base de resultados o método Solver (método simplex utilizada no Excel) para identificar se realmente era uma solução viável ou ótima para tomadas de decisão da empresa em que o aluno realizou os testes para reduzir os custos e o desperdício da matéria-prima.

1. PROBLEMA DE CORTE

Na literatura o problema de corte é estudado dentro da pesquisa operacional (PO) devido à importância nos contextos de aplicação industrial.

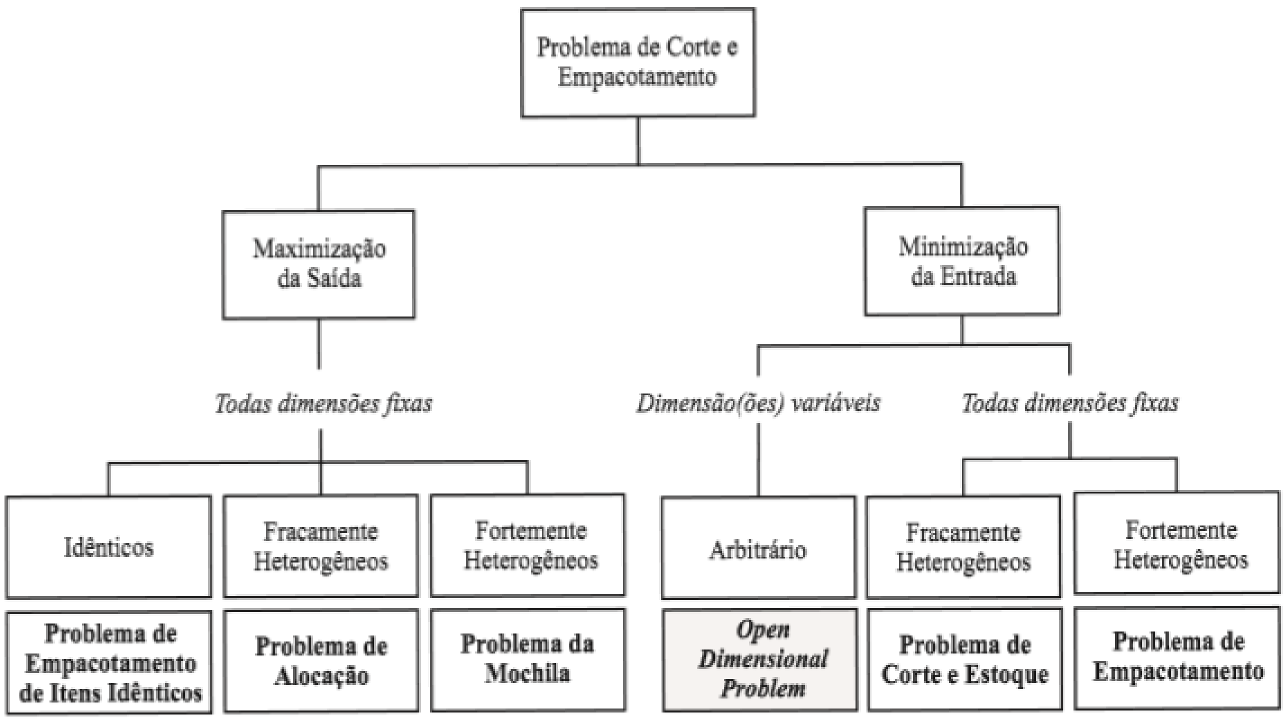
É de suma importância esclarecer que o problema de corte se divide em problema de corte e empacotamento e problema de corte de estoque, que é objeto de estudo deste trabalho.

O problema de corte de estoque é encontrado em indústrias de papéis, móveis, vidros, plásticos, tecidos, metalúrgicos, entre outros, que consiste em cortar um conjunto de itens a partir de objetos grandes disponíveis, visando atender a demanda do cliente (GHIDINI, 2008).

Este método é utilizado para combinar os itens que serão cortados (ou empacotados) do objeto, minimizando os efeitos negativos gerados, como por exemplo, custos de produção ou perda de matéria-prima, e maximizando benefícios positivos, ou seja, aumento de lucro (LEÃO, 2013).

Para exemplificar melhor a ideologia da classificação sobre o funcionamento desta tipologia, a Figura 1 demonstra a hierarquia da estrutura do problema de corte.

Figura 1 – Estrutura do Problema de Corte.



Fonte: TEMPONI, 2007.

Este critério está relacionado ao dimensionamento dos problemas de corte que podem ser classificados da seguinte maneira: Problema de Corte Unidimensional, Problema de Corte Bidimensional, Problema de Corte Tridimensional e Problema de Corte N-dimensional. As categorias e os objetivos de cada problema de corte serão apresentados nas seções 1.1, 1.2, 1.3 e 1.4.

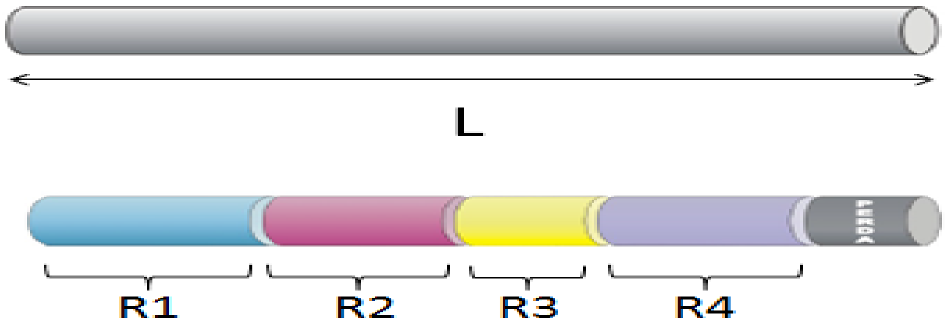
1.1. Problema de Corte Unidimensional

É considerado problema de corte unidimensional, quando apenas uma dimensão (comprimento) do objeto a ser cortado define um padrão de corte, associado ao processo do corte.

Neste tipo de problema o objetivo é determinar a melhor maneira de cortar os L tipos de objetos estocados para obter os diversos tipos de itens R com demandas sobre a necessidade do cliente para otimizar uma função objetivo que pode ser a minimização da perda de matéria prima ou a quantidade de objetos L cortados (GHIDINI, 2008).

O problema de corte unidimensional é aplicado em indústrias de papeis, tecidos, barras de aço, entre outros. Um exemplo do funcionamento deste problema é: uma barra de ferro com o tamanho padrão de comprimento L, atendendo uma demanda de itens representada em R1, R2, R3 e R4 apresentado na Figura 2.

Figura 2 – Modelo de Corte Unidimensional.



Fonte: Adaptado de JUNIOR, 2007.

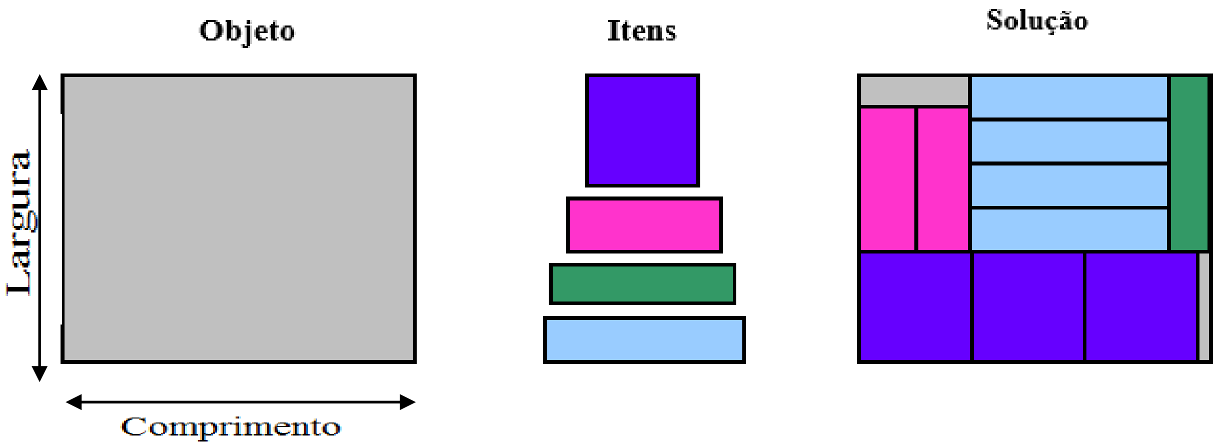
1.2. Problema de Corte Bidimensional

O problema de corte é considerado bidimensional quando duas dimensões (comprimento e largura) são relevantes para este processo de corte, tendo como objetivo principal obter todos os itens cortados na mesma espessura em relação à demanda do cliente.

No caso bidimensional, as restrições físicas são mais complexas de serem explicitadas como o problema de corte unidimensional, pois as características das máquinas de corte e os tipos de corte são importantes para a modelagem matemática (GRECCHO, 2013).

Um exemplo desta aplicação encontra-se em tipos de cortes, quais sejam os cortes retos (indústrias moveleiras no corte da placa de madeira e indústrias metalúrgicas no corte de chapas de aço) e cortes curvos (indústrias têxteis e de couro). É possível ver um exemplo de corte bidimensional na Figura 3.

Figura 3 – Exemplo de Corte Bidimensional.



Fonte: Adaptada de GHIDINI, 2008.

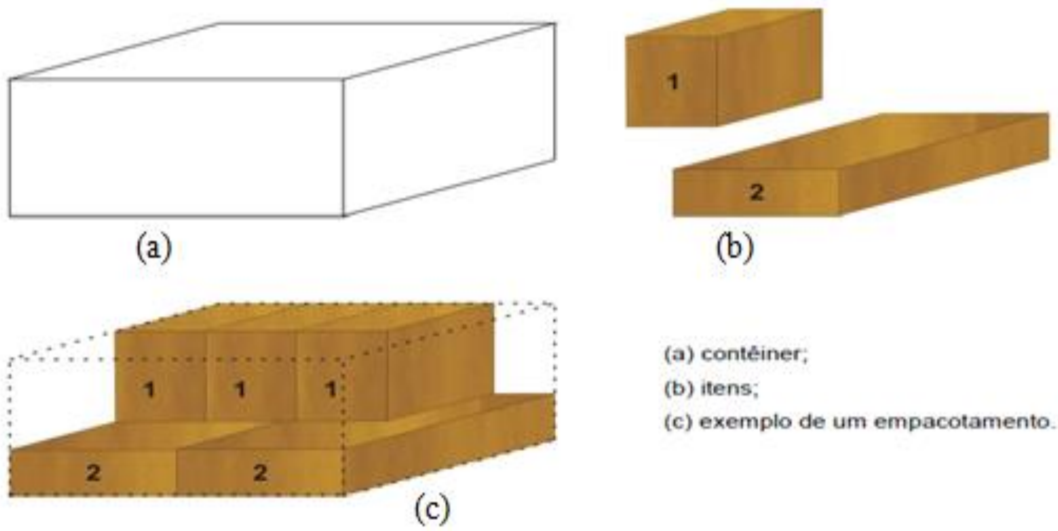
1.3. Problema de Corte Tridimensional

No problema de corte tridimensional três dimensões são relevantes (comprimento, largura e altura) no processo de corte. Basicamente trata-se em agrupar unidades de pequenas, sobrepondo dentro de unidades grandes, este cenário específico pode ser encontrado em transportadoras como: rodoviário, ferroviário, aéreo, entre outros meios de transporte.

É um problema que provavelmente não existam métodos que resolvam suas restrições e objetivos em um tempo computacional razoável, e por esta razão a maioria dos estudos concentra-se em desenvolver métodos heurísticos para resolução do problema. (ARAÚJO, 2006).

Um ambiente relacionado para este tipo de problema encontra-se na Figura 4, referente ao empacotamento de caixas em contêiner, com o objetivo de encaixar os itens dentro de um objeto da melhor maneira possível.

Figura 4 – Arranjo de empacotamento de contêiner.



Fonte: MORAIS, 2011.

1.4. Problema de Corte N-dimensional ou N > 3

Quando o problema de corte tem N como o número de dimensões relevantes na otimização do seu processo de corte, e sendo N > 3, será considerado como N-dimensional ou Multidimensional.

De acordo com Dyckhoff (1990) esses problemas são considerados como abstratos, pois fazem parte de problemas como, por exemplo, problema de alocação de tarefas, programação de rotas de veículos, entre outros.

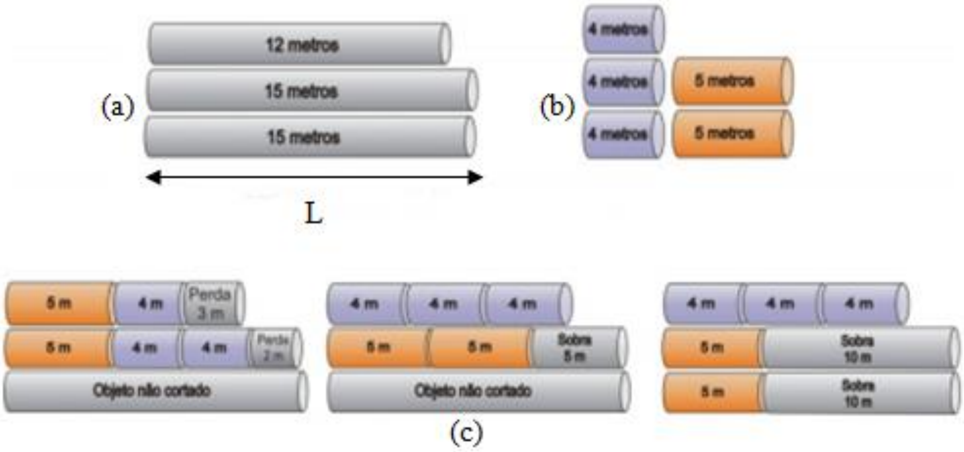
1.5. Formulação Matemática

O problema de corte unidimensional será utilizado para exemplificar a definição sobre formulação matemática.

Suponha que no estoque há uma quantidade suficiente de objetos (barras em estoque) de comprimento L para cortar itens l_i (peças menores) em comprimentos onde $l_i \leq L$, i = 1,..., n, de modo que atenda as demandas de d_i, i = 1,..., n (PINTO, 2008).

Portanto, o problema de corte consiste em cortar as barras em estoque para otimizar este processo, minimizando o número de objetos a serem usados e/ou maximizando os lucros.

Figura 5 – Exemplo de Padrão de Corte.



Fonte: Adaptado de CHERRI, 2010.

Onde (a) determina o comprimento L da peça, (b) denomina os tipos de itens l_i que serão cortados e (c) são exemplos de possíveis d_i para padrões de corte.

Na figura 5 ocorre o que é chamado de padrão de corte, e de acordo com Junior (2007) o padrão de corte é a maneira de como um objeto disponível em estoque é cortado para a produção dos itens que atenderam a linha de produção da indústria.

Um padrão de corte é associado a um vetor m-dimensional para contabilizar as peças produzidas (c), conforme mostrado na Figura 1.5. Pode ser observado também que α_i é a quantidade de peças do tipo i contidas no padrão de corte:

$$\alpha = (\alpha_1, \alpha_2, ..., \alpha_m) \tag{1.1}$$

Outro fator importante a ser notado é que tendo dois padrões de corte com o mesmo vetor associado, eles serão chamados de equivalentes (MORAES, 2014).

Apenas corresponderá um Vetor A = $(\alpha_1, \alpha_2, ..., \alpha_m)$ a um padrão de corte somente se corresponder às restrições contidas no problema.

$$l_1\, \alpha_1 + l_2\, \alpha_2 + ... + l_m\, \alpha_m + \leq L \tag{1.2}$$

$$\alpha_1 \geq 0,\ i = 1,...,m \text{ e inteiro.} \tag{1.3}$$

1.6. Classificação dos Problemas de Corte

Existe uma grande diversidade nos problemas de corte de estoque. Assim, Dyckhoff (1990) apresentou uma tipologia para classificá-los conforme algumas características, tais como: dimensionamento, modo de alocação das unidades, sorteamento de unidades grandes e sorteamento de unidades pequenas.

(Wäscher et al., 2006) propôs uma extensão da tipologia de Dyckhoff, tendo sido criado um novo critério de categorização e organização dos problemas de corte e empacotamento.

Neste trabalho será abordada a tipologia proposta por Dyckhoff, que será detalhada na tabela 1. Cada uma das características apresenta suas subdivisões representadas pelos símbolos entre parênteses.

Tabela 1 – Tipologia de Dyckoff.

1. Dimensionamento	(1) Unidimensional
	(2) Bidimensional
	(3) Tridimensional
	(N) N-dimensional, com N > 3
2. Modo de Alocação das Unidades	(V) Seleção das unidades grandes
	(B) Seleção das unidades pequenas
3. Sorteamento de Unidades Grandes	(O) Uma unidade
	(I) Unidades de tamanhos iguais
	(D) Unidades de tamanhos diferentes
4. Sorteamento de Unidades Pequenas	(F) Poucas unidades de tamanho diferentes
	(M) Muitas unidades de muitos tamanhos diferentes
	(R) Muitas unidades de poucos tamanhos diferentes
	(C) Unidades de tamanhos iguais

Fonte: BRANDÃO, 2009.

A tipologia é indicada pela quádrupla (α / β / γ / δ), que corresponde diretamente aos critérios definidos anteriormente no 1, 2, 3 e N (BRANDÃO, 2009).

1.7. Programação Inteira

Este método surge por meio das limitações encontradas na Programação Linear (PL) no tratamento de alguns problemas que demandam o uso de variáveis inteiras.

As primeiras propostas de metodologias que tratavam o problema como variáveis inteiras eram mais adaptações do método simplex do que novas teorias. O principal arquiteto dos primórdios da PI foi Ralph E. Gomory em 1929 (COLIN, 2013).

O conceito de modelagem de problemas de PI é semelhante com a modelagem de problemas de PL. A diferença é que pelo menos parte das variáveis é de números inteiros, ou seja, números genéricos (1, 2, 3,...) ou números binários (0 ou 1).

Atualmente o problema de PI apresenta três tipos de conjunto:

1. Conjunto: São os problemas de programação inteira pura, onde todas as variáveis de decisão são do tipo genérico.
2. Conjunto: São os problemas de programação inteira mista, onde apenas uma parte das variáveis é do tipo inteiro, enquanto outras são do tipo real.
3. Conjunto: São os problemas de programação inteiras com variáveis 0 e 1, onde todas assumem valores 0 ou 1.

Uma característica importante no cenário de PI é avaliar a real necessidade de se modelar uma variável como inteira. De acordo com Oliveira (2010), para problemas de grande porte, isso geralmente gerará uma resolução aceitável (próximo do ótimo real) sem a violação de nenhuma das restrições. Já para problemas menores, esse tipo de procedimento poderá nos levar a soluções inviáveis ou não ótimas.

Em tese, todo problema de PI está associado a um problema com a mesma função objetivo e as mesmas restrições, com exceção da condição de variáveis inteiras. A esse problema se dá o nome de Problema Relaxado (OLIVEIRA, 2010). Este conceito define as limitantes da solução do problema de PI, em outras palavras, para um problema de maximização:

$$\begin{pmatrix} \text{Valor ótimo da relaxação de PL} \\ \text{de um problema de PI} \end{pmatrix} \geq \begin{pmatrix} \text{Valor ótimo do} \\ \text{problema de PI} \end{pmatrix}$$

Enquanto para um problema de minimização:

$$\begin{pmatrix} \text{Valor ótimo da relaxação de PL} \\ \text{de um problema de PI} \end{pmatrix} \leq \begin{pmatrix} \text{Valor ótimo do} \\ \text{problema de PI} \end{pmatrix}$$

Nos próximos tópicos serão avaliados as técnicas, os conceitos e as definições de alguns métodos de algoritmos utilizados para a solução dos problemas de PI, com associação ao problema de corte, nas seções 1.7.1, 1.7.2 e 1.7.3.

1.7.1. O Problema da Mochila (Knapsack Problem - PK)

O Problema da mochila (ou *Knapsack Problem* - PK) é conhecido por este nome devido ao seu modelo de situação, onde é necessário preencher uma mochila com itens de diferentes pesos e valores.

Assim toda mochila possui uma restrição fundamental, onde a soma do peso dos itens selecionados deve respeitar a capacidade da mochila não ultrapassando o seu peso máximo (MARQUES, 2004).

Quanto maior a quantidade de restrições maior será a dificuldade de encontrar a solução exata para o problema.

Atualmente, o problema da mochila contém cinco tipos de conjunto, representado na Tabela 2.

Tabela 2 – Tipos de conjuntos relacionados ao problema da mochila.

Tipos	Conceitos
Problema da Mochila 0/1	Cada item pode ser escolhido no máximo uma vez
Problema da Mochila limitada	Há uma quantidade limitada para cada tipo de item
Problema da Mochila com múltipla escolha	Os itens devem ser escolhidos de classes disjuntas, ou seja, caso não tenham nenhum item em comum
Problema da Mochila múltiplo	Várias mochilas são preenchidas simultaneamente
Problema da mochila com multi-restrições	Problema de PI geral com coeficientes positivos

Fonte: Elaborado pelo autor.

O problema da mochila é um problema de otimização combinatória que pode ser aplicado em problemas como corte de estoque, corte de investimento, capital, transporte, produção, entre outros (MARQUES, 2004).

1.7.2. Branch and Bound

A principal dificuldade dos problemas de PI está associada com o número excessivamente grande de soluções possíveis (soluções candidatas). A ideia essencial sobre o algoritmo de Branch and Bound (B&B) é resolver um grande número de subproblemas lineares, na busca de uma solução ótima.

É uma metodologia aplicada para soluções de PI e PI mista, onde diversas variantes tratam diversos tipos de problemas específicos através de um problema combinatório de minimização ou maximização (OLIVEIRA, 2010).

O termo Branch em B&B refere-se ao processo de particionamento do problema original em sucessivos subproblemas de PL, enquanto o Bound refere-se aos limites superiores e inferiores que são utilizados para a construção da prova de otimização sem pesquisas exaustivas.

O algoritmo B&B é representado por uma árvore (o problema) que contém nós (o subproblema), onde estão representando todas as soluções que podem ser obtidas respeitando as soluções que já existem, e as folhas que representam as soluções completas.

Um problema P é dividido em um conjunto de subproblemas {SPk} de forma que a solução de P possa ser obtida através da solução dos subproblemas. No final, obtém-se a solução do problema de interesse (CAMPONOGARA 2006).

De acordo com o conceito de B&B, as divisões são feitas iterativamente, sempre observando que os subproblemas devem ser mais fáceis de serem resolvidos que o problema original (Camponogara, 2006).

O uso dos limitantes torna o B&B mais eficiente ao permitir descartar nós da árvore de pesquisa que ainda não foram completamente explorados, pela certeza de que nunca originarão soluções melhores do que as que já existem e também permitem medir a distância (em termos de valor da função objetivo) na procura de uma possível solução ótima.

1.7.3. Enumeração Implícita

O algoritmo de enumeração implícita é uma versão do algoritmo B&B adaptado para resolver problemas com variáveis binárias.

É uma ideia de desenvolver uma enumeração de pontos candidatos (nós) em busca de uma solução ótima e inteira do problema, por meio da participação do espaço e avaliação progressiva das soluções obtidas.

Para encontrar uma possível solução em um problema de PI, embora todas as enumerações sejam testadas, nem todas são feitas explicitamente, visto que muitas delas são realizadas implicitamente por meio de testes e conclusões lógicas (COLIN, 2013).

De forma exaustiva, todos os valores possíveis para a função-objetivo são calculados sendo escolhido aquele que apresentar o maior valor (no caso de maximização) ou o menor valor (no caso de minimização) (OLIVEIRA, 2010).

A intenção do problema com essa tática de solução está no fato de que o algoritmo de enumeração implícita só consegue ser aplicada a problemas pequenos. Outra característica importante é que o algoritmo trabalha apenas com operações de soma e subtração e o número de combinações possíveis de soluções cresce de forma exponencial.

1.8. Considerações Finais

Neste capítulo, foi esclarecida a classificação, a dimensão e a formulação matemática de um problema de corte unidimensional, sendo o foco do presente trabalho. Também foram apresentados de forma breve, os outros tipos de problemas de corte em relação a suas categorias e objetivos no processo de corte.

Um dos fatores mais importante do problema de corte unidimensional estudado até o momento foi à criação dos padrões de corte, uma vez que se faz necessário levar em consideração algumas restrições específicas ao problema em questão.

Para se chegar à resolução final do problema proposto, será utilizado o método da mochila limitada para auxiliar nas restrições aliado ao algoritmo heurístico genético e o método simplex, resultando em uma solução viável, ou que chegue o mais próximo possível de uma solução ótima, com o intuito de minimizar a perda de matéria-prima.

2. MÉTODOS HEURÍSTICOS

As heurísticas atuam como um processo iterativo ou de refinamento para solução do problema que se organiza pela combinação de diferentes conceitos, podendo manipular uma solução completa, incompleta ou um conjunto de soluções, tentando evitar uma parada prematura.

Estes métodos englobam estratégias, procedimentos e modelos que visam encontrar resultados que satisfaçam um problema específico, mesmo que não se obtenha soluções exatas, perfeitas e definitivas, em um tempo computacional razoável.

Portanto, pode ser considerada como um desenvolvimento de métodos e regras baseadas em processos não dedutivos, podendo ser compreendida como um caso especial de tentativa e erro (COSTA, 2011).

A classificação dos métodos heurísticos está previamente descrito na tabela 3 com as definições de cada padrão referente às suas soluções heurísticas.

Tabela 3 – Classificação dos Métodos Heurísticos.

Métodos	Definições
Busca	Inicia-se em uma solução (podendo ser obtida a partir de outra heurística) e caminha sobre as soluções vizinhas.
Construtivos	Processo iterativo que inicia com uma solução vazia e adiciona um novo elemento a cada iteração até a obtenção de uma solução
Decomposição	Dividi-se o problema em subproblemas menores, de modo que a resolução de todos os subproblemas possa compor uma solução para o problema maior.
Redução	Identificam algumas características para a resolução do problema relaxado, obtendo uma solução inteira.

Fonte: Elaborado pelo autor.

Nesta seção, foi descrita de forma rápida uma introdução sobre a definição do método heurístico.

Ainda, serão abordadas com ênfase as heurísticas divididas em duas categorias: heurísticas construtivas e heurísticas de busca, que estarão devidamente apresentadas nas seções 2.1 e 2.2.

2.1. Heurísticas Construtivas

A aplicação desta heurística é através de repetições exaustivas, ou seja, para construir um bom padrão de corte será utilizado o máximo de vezes possível. Isto é o que chamamos de heurísticas construtivas (POLDI e ARENALES, 2006).

Elas são empregadas na geração de uma ou mais soluções viáveis, eventualmente de boa qualidade, com baixo esforço computacional.

De acordo com Queiroz (2011), podem ser utilizadas de maneira isolada, mas de forma geral, são combinados com outras técnicas mais elaboradas, como algoritmos de busca local, meta-heurísticas e métodos exatos.

Há dois métodos clássicos da literatura sobre problema de corte unidimensional inteiro para se construir um bom padrão de corte: As heurísticas construtivas FFD e Gulosa, que serão discutidas nas seções 2.1.1 e 2.1.2.

2.1.1. Heurística FFD

A heurística FFD (First-Fit-Decreasing) trata-se, de acordo com Cherri (2006), colocar o maior item em um padrão de corte, onde consiga ser executado ao máximo de vezes possível até que este item não ultrapasse seu limite de tamanho ou até que sua demanda já tenha sido completada.

Todas às vezes quando não for mais possível ou necessário à inclusão do maior item, o segundo maior item é considerado e assim por diante. Quando nenhum novo item for incluído, um padrão de corte é construído.

Para cada objeto L em estoque, um padrão de corte é construído e aquele que apresentar a menor perda é o escolhido, até que toda a demanda seja atendida (CHERRI, 2006).

O algoritmo FFD utiliza apenas a restrição física do problema, não fazendo mais nenhuma restrição aos itens que serão utilizados na solução. Por exemplo, suponha quatro itens, cuja unidade são metros, ordenados da seguinte forma:

L = [4,50; 4,25; 4,00; 0,75]

Quando estes itens forem colocados em barras de 5 metros, o algoritmo funcionará da seguinte forma:

- Inicia-se uma barra B1 como solução.
- Coloca-se o primeiro item, 4,50 metros em, B1.

Solução parcial: **B1 = [4,50]**

- Verifica se o próximo item, 4,25 metros, pode ser colocado em B1, como não pode, é iniciada uma nova solução.

Solução parcial: **B1 = [4,50] B2 = [4,25]**

- Verifica se o próximo item, 4,00 metro, pode ser encaixado em uma das soluções correntes, como não cabe em B1 e também não cabe em B2, é aberta uma nova solução.

Solução parcial: **B1 = [4,50] B2 = [4,25] B3 = [4,00]**

- Verifica se o próximo item, 0,75 metro, pode ser encaixado em uma das soluções, não cabe em B1, mas cabe em B2, então é colocado em B2.

Solução parcial: **B1 = [4,50] B2 = [4,25; 0,75] B3 = [4,00]**

Seguindo o raciocínio chegasse a seguinte solução:

Solução Final: B1 = [4,50] B2 = [4,25; 0,75] B3 = [4,00]

Verificasse que o único trabalho do algoritmo é o de verificar se o item pode ser utilizado em alguma das soluções já existentes. Caso não possa ser utilizado, é aberta uma nova solução.

2.1.2. Heurística Gulosa

Uma estratégia simples e que obtêm bons resultados para uma variedade de problemas de otimização combinatória são os denominados algoritmos gulosos.

Correspondente à heurística FFD, a heurística Gulosa produz uma solução inteira. Sendo assim, a cada iteração, mantém uma demanda (di) até a geração de um padrão de corte aplicada ao problema original.

É a partir desta demanda que o algoritmo consiste em resolver o determinado cenário com uma função objetivo apropriada, por exemplo, um valor α_i de cada item em uma mochila como sendo o próprio comprimento do item l_i, de modo que a perda no padrão seja minimizada (POLDI, 2002).

Observa-se que a heurística gulosa é um procedimento heurístico de repetição exaustivo para gerar um padrão de corte resolvendo um determinando cenário em relação ao problema da mochila, onde:

Definições:

$d_i \rightarrow$ Demanda

l_i, $i = 1,..., m \rightarrow$ Comprimento dos itens

$\alpha_i \rightarrow$ É a quantidade de itens do tipo i no padrão de corte G(a)

$$\text{Maximizar } G(a) = l_1\,\alpha_1 + l_2\,\alpha_2 + ... + l_m\,\alpha_m \tag{2.1}$$

$$\text{sujeito a: } l_1\,\alpha_1 + l_2\,\alpha_2 + ... + l_m\,\alpha_m \leq L \tag{2.2}$$

$$0 \leq \alpha_i \leq d_i,\ i = 1,..., m \text{ e inteiro} \tag{2.3}$$

o qual tem demanda d_i atualizada após ter sido escolhido o padrão de corte com a menor perda. O algoritmo encerra sua execução quando uma solução viável é obtida.

2.2. Heurísticas de Busca

As heurísticas de busca partem de uma solução inicial conhecida, e procuram explorar a vizinhança desta solução por meio de movimentos, com o objetivo de melhorá-la, chegando a ótimos locais.

Desta forma, é criada a vizinhança da solução inicial e feita a escolha da solução nesta vizinhança com o menor valor de função objetivo. A solução escolhida torna-se, então, a nova solução de referência e o processo continua até ser encontrado um ótimo local (QUEIROZ, 2011).

No próximo tópico, serão analisados os principais conceitos, técnicas e definições sobre um método heurístico de busca, algoritmo genético, inserido na solução final do presente trabalho, conforme seção 2.2.1.

2.2.1. Algoritmo Genético

Publicados inicialmente em 1975, pelo professor John Holland, da Universidade de Michigam, os algoritmos genéticos (AG) procuram solucionar problemas complexos de otimização e aprendizado computacional.

Tratam-se de algoritmos de busca baseados em uma técnica de inteligência artificial que pertence à classe particular dos processos de seleção natural e reprodução genética da evolução sobre a Teoria de Darwin.

Os AG's procuram imitar, de uma forma computacional, algumas etapas desse processo evolutivo das espécies, e de acordo com Costa (2011), todos os indivíduos com características melhores têm maiores chances de sobrevivência e de produzir filhos cada vez mais fortes (mais aptos), enquanto os indivíduos mais fracos (menos aptos) tendem a desaparecer.

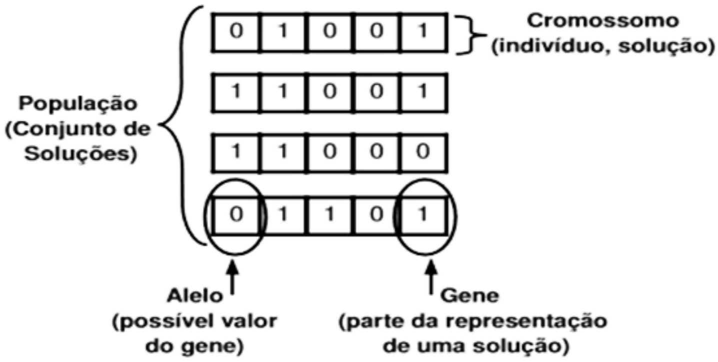
A utilização do AG está representada por uma estrutura com métodos fundamentais mostrado na Figura 6.

A **população** é formada por um conjunto de indivíduos, que está representada nos **cromossomos**. Segundo COSTA (2011), cada cromossomo equivale a uma solução do problema. Dessa forma, uma população é um conjunto de soluções. O cromossomo é dividido em componentes, chamados de **alelos**, que são os possíveis valores (uma cadeia de bits 0 e 1) que cada componente da solução pode assumir.

O valor que cada alelo possui é chamado de **gene** (representação de cada parâmetro de acordo com o alfabeto utilizado). Um mecanismo de reprodução baseado em processos evolutivos é aplicado sobre a população, com o objetivo de explorar o espaço de busca (população) e encontrar as melhores soluções para o problema (COSTA, 2011).

Cada indivíduo é avaliado pelo **fitness** (função de aptidão), que resultará seu grau de adaptação. Quanto maior for o valor da função de aptidão, significa que o indivíduo está apto a participar da nova população.

Figura 6 – Estrutura do Algoritmo Genético.

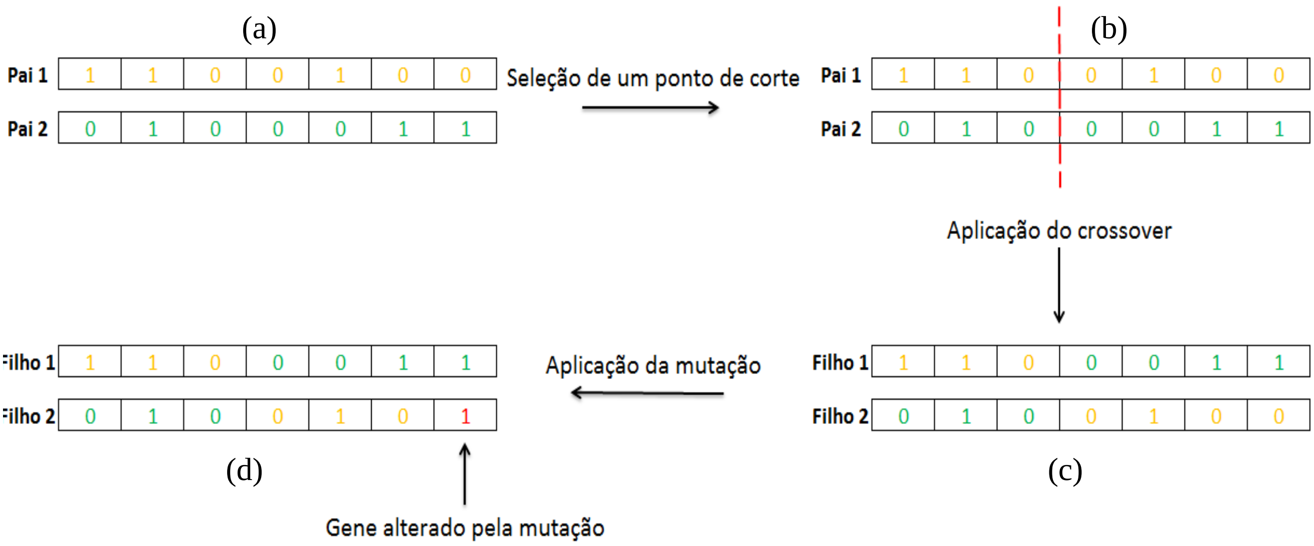


Fonte: COSTA, 2011.

Na Figura 7 é ilustrado o ciclo principal do Algoritmo Genético, que após a população gerada e a validação do fitness concluída, são selecionados dois indivíduos desta população, chamados pais (a), que passam por uma determinada etapa, onde acontecerá a reprodução através da **seleção** em um dado ponto de corte (b).

Em seguida, encaminhado ao processo de recombinação **crossover** (c), são gerados os filhos, passando pelo método de **mutação** que altera aleatoriamente uma parte dos genes de cada cromossomo (d), em um determinado gene pode ser modificado, fazendo com que os descendentes gerados herdem características dos indivíduos mais aptos (os fortes) da população, consequentemente descartando os menos aptos (os fracos) para a próxima geração conforme o tamanho da população inicial.

Figura 7 – Ciclo principal do Algoritmo Genético.



Fonte: Elaborado pelo autor.

Este processo de otimização também será apresentado como um pseudocódigo de um AG básico.

Algoritmo Genético

```
1.  Início
2.    Gerar uma população inicial
3.    Avaliar o fitness dos indivíduos da população
4.      Faça
5.        Início
6.            Selecionar um conjunto de pais na população
7.            Cruzar os pais de modo que se reproduzam
8.            Avaliar o fitness dos filhos gerados
9.            Substituir os filhos julgados inadequados
10.       Fim
11.     Enquanto os critérios de parada não estiverem satisfeitos
12. Fim
```

Os principais parâmetros do algoritmo são o seu tamanho da população, o ponto de corte da operação crossover, a probabilidade de mutação, a quantidade de gerações e a quantidade de gerações sem melhora.

O AG realiza uma busca usando um conjunto de soluções (população) e através de um processo iterativo (linhas 6 a 9) são obtidos novos candidatos à resolução final. De acordo com JÚNIOR (2003), este processo de busca deve possuir um número reduzido de soluções, respeitando uma estratégia adequada para satisfazer critérios de parada.

2.3. Considerações Finais

Foram apresentadas as heurísticas em duas categorias: heurísticas construtivas e heurísticas de busca.

As heurísticas construtivas partem inicialmente sua programação como vazia, adicionando as tarefas uma a uma, até concluir uma solução viável. Já as heurísticas de busca iniciam-se a partir de uma heurística construtiva, realizando etapas no processo de operações para transformar a resolução obtida, em uma solução melhor. Estas iterações são repetidas até que uma ótima população seja encontrada, ou se atinja um critério de parada.

Este capítulo procurou mostrar as definições e técnicas importantes para a contextualização deste trabalho.

Partindo deste enfoque, foi estudado e apresentado de maneira minuciosa o ciclo do algoritmo genético, que fará parte do desenvolvimento da solução final. O próximo capítulo irá propor o uso deste método para geração dos padrões de corte que serão aplicados juntamente ao método simplex na resolução do estudo de caso sobre o problema de corte de estoque na busca de melhores soluções em termos de aproveitamento de matéria-prima e redução do tempo.

3. DESENVOLVIMENTO DE UMA SOLUÇÃO COMPUTACIONAL PARA O PROBLEMA DE CORTE

Neste trabalho foi desenvolvido um modelo de otimização para o problema de corte de estoque com padrões de corte unidimensional relacionado ao corte de tubos, a fim de minimizar perda de matéria-prima. Será explicado também o contexto onde a solução foi aplicada, relatando a arquitetura e suas funcionalidades.

3.1. Contextualização

A pesquisa operacional de acordo com Cardoso (2011) está designada na área do conhecimento que consiste em desenvolver métodos científicos de sistemas complexos, com a finalidade de prever e comparar estratégias ou decisões alternativas, cujo objetivo é dar suporte à determinadas ações.

Um aspecto importante de problemas envolvendo decisões é o de otimização; quando se procura estabelecer maneiras eficientes utilizando recursos disponíveis, atingindo critérios que possibilitam aumentar o rendimento da produtividade e diminuir o desperdício de matéria prima. Tais características serão abordadas no cenário relacionado ao problema de corte de estoque unidimensional.

A Figura 8 demonstra o fluxograma com as principais operações das etapas para uma boa tomada de decisão. O ciclo está divido em cinco fases, apontando o fluxo de dados ao longo do processo:

Definição do problema: Uma indústria do setor metalúrgico, situada em Pompéia – SP fabrica produtos tanto na área fitness como no setor agrícola. A produção é feita internamente e por isso a empresa corta uma grande quantidade de tubos.

O comprimento padrão do tubo é de 6000 mm. A empresa metalúrgica utiliza-se de vários tipos de tubo para a produção de seus itens, dentre os mais usados atualmente são tubos obilongo e retangular.

Construção do modelo: O problema consiste em como otimizar de maneira rápida o processo de cada barra, sendo que são vários os perfis de tubo e cada perfil de tubo possui uma grande variedade de itens (peça).

Uma solução viável pela empresa foi obtida juntamente com Moraes, aluno que cursou engenharia de produção, onde a maioria dos resultados encontrados era por força bruta, ou seja, sem auxílios computacionais.

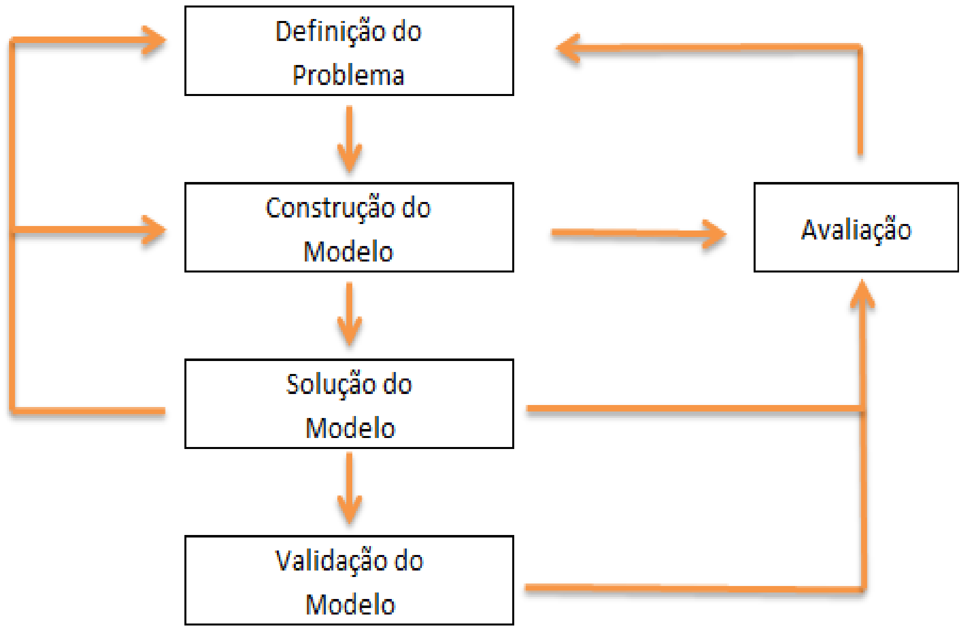
Solução do modelo: É a identificação dos padrões de corte no cenário proposto, desenvolvendo uma solução computacional utilizando a linguagem de programação Python na versão 2.7.12 para as seguintes técnicas:

1. Programação inteira: Problema da mochila, sua principal função será de restringir a quantidade de itens que estará contido no tamanho padrão do tubo que são de 5835 mm.
2. Método heurístico: Algoritmo Genético será responsável pela geração dos padrões de corte.
3. Gerador de arquivos com extensão .ltx utilizado para adicionar os valores que serão automaticamente representados no software LINDO na versão 6.1, que tem como objetivo aplicar o método simplex com base nos padrões de corte definidos anteriormente pelo AG.

Validação do modelo: Obter uma solução viável que possivelmente poderá ocasionar em uma solução ótima.

Avaliação: É a realização de testes em dois cenários, onde ocorrerá a verificação na comparação dos resultados obtidos, para identificar a existência de possíveis melhoramentos na otimização do processo de corte em relação à perda de matéria-prima.

Figura 8 - Fluxograma para tomadas de decisão.



Fonte: Adaptado de ANDRADE, 2009.

O primeiro capítulo descreve de forma geral, o estudo do problema de corte com as soluções encontradas no mercado industrial para resolver situações cotidianas no consumo de matéria-prima. Os itens 1.1 até 1.4 enfatizam os critérios de dimensionalidade interligada a suas características existentes no processo de corte.

Já no item 1.6 foi demonstrada na Tabela 1 a tipologia utilizada por Dickhoff (1990) para classificar o dimensionamento, modo de alocação das unidades, sorteamento de unidades grandes e sorteamento de unidades pequenas em relação ao problema de corte. Neste caso podemos definir o estudo de caso da seguinte maneira:

- (1) Trata-se de um Problema de Corte Unidimensional;
- (B) Produção grande de itens;
- (I) Unidades do mesmo formato, porém tamanhos diferentes; e
- (M) Muitos itens de tamanhos diferentes.

Setores de produção, onde a matéria-prima consequentemente resultará em um desperdício significativo para empresa, demonstra que nestas ocasiões, a necessidade de uma boa estratégia para tomadas de decisão possibilita a otimização do processo de corte.

3.2. Modelo de otimização

O modelo de otimização a seguir será baseado a partir de duas funções objetivas que podem ser representadas por métodos matemáticos, sendo que a primeira função deverá obter os padrões de corte através do AG com o PK, enquanto a segunda função será responsável de minimizar as perdas geradas pelos padrões utilizando o método simplex.

Função Objetivo

1. Definindo os padrões de corte utilizando o AG com o PK:

$$\text{F.O.} = (\alpha_{11} + \alpha_{12} + \ldots + \alpha_{1n}) \leq \text{TamB} \tag{3.1}$$

ou ainda:

$$\text{F.O.} = \sum_{i=1}^{n} (\alpha_{11} + \alpha_{1n}) \leq \text{TamB} \tag{3.2}$$

Onde:

TamB = Tamanho da barra.

α_{11} ... α_{1n} = São os comprimentos de cada item i existentes associados ao padrão de corte.

2. Minimiza as perdas geradas pelos padrões de corte com o método simplex, maximizando a utilização de cada barra, sendo o principal objetivo deste presente trabalho.

$$\text{F.O. Min } Z = \sum_{i=1}^{n} (X_1 * Y_1 + X_2 * Y_2 + ... + X_n * Y_n) \tag{3.3}$$

ou ainda:

$$\text{F.O. Min } Z = \sum_{i=1}^{n} (X_1 * Y_1) \tag{3.4}$$

Sendo:

X_1 = Número de vezes que o item i será utilizado pelo padrão de corte.

Y_1 = Comprimento do item i utilizado.

O primeiro conjunto da equação F.O. (3.1 ou 3.2) garante a quantidade de itens produzida em relação ao comprimento dos objetos do estoque. Já o segundo conjunto é dado pela equação F.O. (3.3 ou 3.4), que certificará se toda a demanda será atendida, resultando em uma solução viável que, consequentemente, poderá ocasionar em uma solução ótima para otimização na perda de matéria-prima.

3.3. Solução desenvolvida

Após esclarecer a definição do modelo de otimização, é necessário enfatizar de maneira específica o uso das técnicas computacionais para identificar possíveis melhorias ao problema estudado.

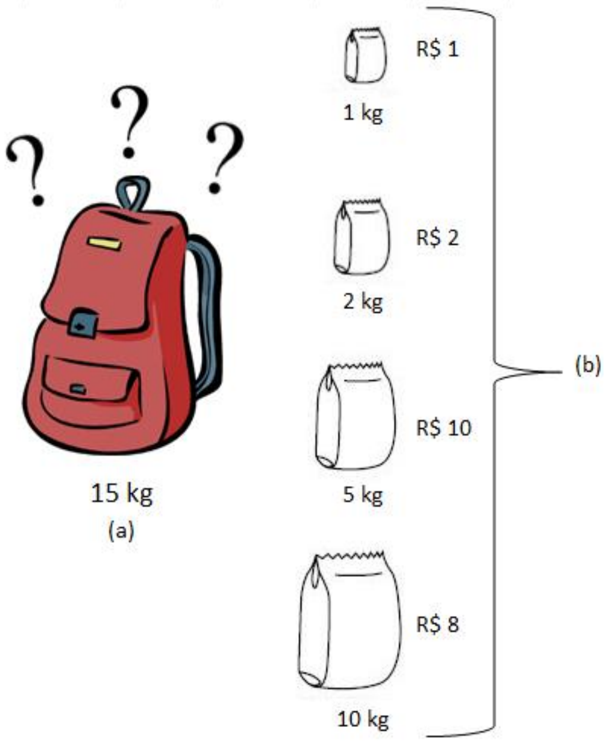
As funcionalidades da solução propostas estarão descritas a seguir nas seções 3.3.1, 3.3.2 e 3.3.3.

3.3.1. Problema da mochila aplicado ao Problema de Corte Inteiro

No capítulo 1 do item 1.7.1 foi apresentado de forma geral os principais conjuntos e características relacionados ao PK.

Este método de PI consiste em determinar a capacidade adequada do compartimento (a), de modo que deva carregar o valor de utilidade total (b) somado aos pesos de todos os itens e os seus custos selecionados seja viável, ou seja, não ultrapasse o limite permitido da mochila. A Figura 9 ilustra um cenário clássico sobre o Problema da Mochila.

Figura 9 – Cenário clássico sobre o Problema da Mochila.



Fonte: Elaborado pelo autor.

Segundo MORAES (2014), se associarmos ao problema de corte unidimensional, teremos então uma barra que deverá ser cortada ao longo de seu comprimento L em itens de tamanho $l_1, l_2, \ldots, l_m$.

Ao considerar uma quantidade x de itens a ser selecionado, (situação que surge especialmente em problemas de corte com baixa demanda), o problema é denominado problema da mochila limitado (MORAES, 2014).

A seguir o cenário ilustrado pela Figura 10 demonstra o ambiente de interface do usuário, onde é possível definir o comprimento da barra, a perda programada, o *presset*, o limite mínimo para o processo de corte, a quantidade de itens para atender a demanda e, por fim, a inserção do tamanho e o peso de cada item para obter os padrões de corte.

Figura 10 – Interface do usuário.

```
Python 2.7.12 (v2.7.12:d33e0cf91556, Jun 27 2016, 15:19:22) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
 RESTART: C:\Users\Lucas\Desktop\TesteAlgoritmo\Algoritmos TCC prontos\algoritmo_genetico.py

Tamanho da barra: 6000
Perda prgrogamada: 160
Presset: 5

Comprimento final da barra: 5835

Digite o limite minimo do tamanho da barra para o processo de corte: 5735

Digite a quantidade de itens que serao inseridos: 5


Digite o tamanho do comprimento e o peso do item separados por um espaco !
Item 1: 1194 1
Item 2: 689 1
Item 3: 590 1
Item 4: 508 1
Item 5: 370 1
```

Fonte: Elaborado pelo autor.

O peso de cada item será o mesmo valor, para não gerar critérios de preferência pelo software na escolha de qualquer um deles utilizados.

3.3.2. Algoritmo Genético

O AG tem a função de gerar os padrões de corte, sendo que para chegar ao desenvolvimento desta solução, haverá um conjunto de regras pré-estabelecidas. Em conformidade com (MORABITO 1998), os seguintes passos são:

- Passo 1: Construir um bom padrão de corte;
- Passo 2: Repetir este padrão (Passo 1) o máximo número de vezes possível, sem gerar excessos de itens cortados; e
- Passo 3: Atualizar a demanda e o estoque.

Para realizar este fluxo corretamente, foi apresentado no item 2.2.1 do capítulo 2 a estrutura do pseudocódigo utilizado para implementar o AG.

Os padrões de corte na Figura 11 demostram o banco de soluções possíveis que armazenam os itens contidos e a capacidade total de cada padrão de corte obtido para auxiliar na solução final do usuário.

Figura 11 – Banco de soluções possíveis.

```
Itens contidos: [(1194, 1), (689, 1), (689, 1), (689, 1), (590, 1), (590, 1), (508, 1), (508, 1), (370, 1)]
Capacidade Total: 5827


Itens contidos: [(1194, 1), (689, 1), (689, 1), (590, 1), (508, 1), (508, 1), (508, 1), (370, 1), (370, 1), (370, 1)]
Capacidade Total: 5796


Itens contidos: [(1194, 1), (1194, 1), (689, 1), (590, 1), (508, 1), (508, 1), (370, 1), (370, 1), (370, 1)]
Capacidade Total: 5793


Itens contidos: [(1194, 1), (689, 1), (590, 1), (590, 1), (508, 1), (370, 1), (370, 1), (370, 1), (370, 1), (370, 1), (370, 1)]
Capacidade Total: 5791


Itens contidos: [(1194, 1), (689, 1), (590, 1), (590, 1), (590, 1), (508, 1), (508, 1), (370, 1), (370, 1), (370, 1)]
Capacidade Total: 5779


Itens contidos: [(1194, 1), (689, 1), (590, 1), (508, 1), (508, 1), (508, 1), (508, 1), (508, 1), (370, 1), (370, 1)]
Capacidade Total: 5753


Itens contidos: [(1194, 1), (689, 1), (689, 1), (689, 1), (590, 1), (508, 1), (508, 1), (508, 1), (370, 1)]
Capacidade Total: 5745
```

Fonte: Elaborado pelo autor.

Considerando o tempo de resposta para que o algoritmo genético chegasse às possíveis soluções em relação a gerações de padrões de corte, foram realizados 10 testes com um tempo de execução entre 6 min e 15 seg até 6 min e 25 seg.

Este período considerado "longo" em resultados computacionais acontece pelo fato do AG rodar 200 vezes para encontrar uma boa quantidade de padrões de corte e obter o melhor banco de soluções possíveis.

As ilustrações da Figura 11 estão representadas da seguinte forma:

- O retângulo contornado em azul apresenta todas as possíveis soluções geradas no problema de corte.
- O retângulo contornado em vermelho, define quais padrões de corte foram utilizados para continuar o processo de otimização, sendo que a próxima etapa será adicionar todos os itens contidos no método simplex.

3.3.3. Método Simplex

É uma técnica que consiste em resolver repetidas vezes um sistema de equações lineares para se determinar, numericamente, a resolução de um modelo de Programação Linear.

Colin (2013, p. 28) constatou que “para um problema colocado na forma padrão, o algoritmo simplex caminha de uma solução viável para outra, de modo que o valor da função-objetivo é diminuído até o ponto ótimo ser alcançado”.

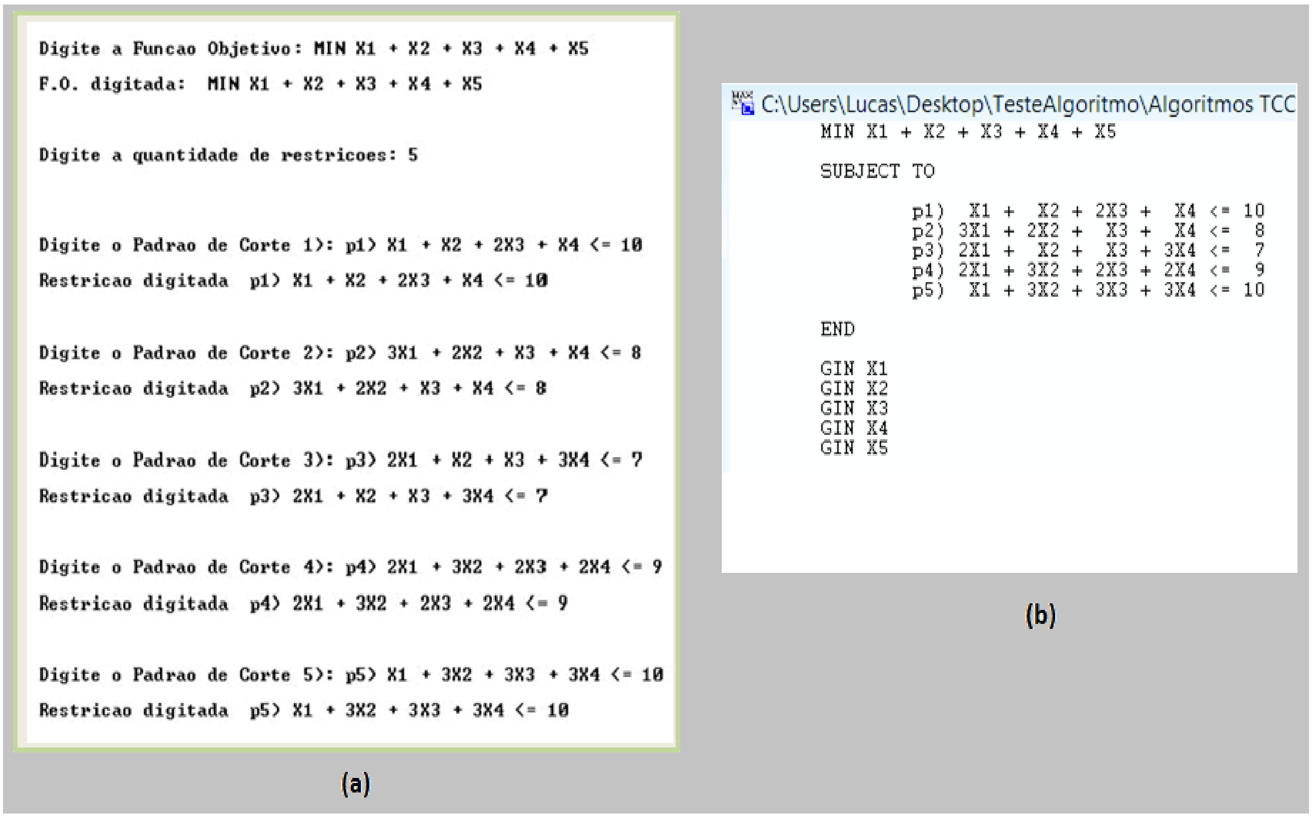
Assim, para solucionarmos o problema de corte de estoque unidimensional após a geração dos padrões de corte será necessário utilizar o método simplex.

Segundo Colin (2013), pode ser definido em três etapas:

- Dados de entrada: Adicionar a Função Objetivo (F.O.) e as variáveis de restrição;
- Iteração: O algoritmo repete diversas vezes todo procedimento de otimização do modelo até que seja alcançada uma solução; e
- Regra de parada: Avalia se a solução ótima foi obtida, ou se é impossível obtê-la.

A Figura 12 ilustra todo procedimento do software a partir do gerador de arquivos com extensão .ltx, que, como mencionado anteriormente, é utilizado para adicionar os valores que serão automaticamente representados no LINDO, tendo como escopo aplicar o método simplex com fundamento nos padrões de corte já definidos pelo AG.

Figura 12 – Aplicação do método simplex.



Fonte: Elaborado pelo autor.

Ao lado esquerdo da interface do usuário (a), foi demonstrada a sequência da execução aplicada no software, que possui as seguintes características:

- Função Objetivo: É a representação de cada tamanho do comprimento usado na Figura 10 sendo convertido em uma função linear de variáveis.
- Quantidade de restrições: Este critério é obtido pela demanda usada em produção (está informação estará presente no próximo capítulo.).
- Variáveis de Decisão: São encontradas nos padrões de corte pelo retângulo destacado em vermelho da Figura 11. Verifica-se que é necessário transformar as colunas de cada padrão de corte em linhas, que se tornarão em variáveis de decisão, exemplificado da seguinte maneira:

Figura 13 – Representação numérica dos padrões de corte.

```
PC1 :  1194 +  1194 + 2*1194 +  1194 <= 10
PC2 : 3*689 + 2*689 +    689 +   689 <=  8
PC3 : 2*590 +   590 +    590 + 3*590 <=  7
PC4 : 2*508 + 3*508 +  2*508 + 2*508 <=  9
PC5 :   370 + 3*370 +  3*370 + 3*370 <= 10
```

Fonte: Elaborado pelo autor.

Após progredir o preenchimento das etapas em relação ao gerador de arquivos com extensão .ltx, automaticamente, todos os valores estarão representados no software LINDO (b) que basicamente será responsável por resolver o mecanismo do algoritmo simplex para obter a frequência de utilização do padrão de corte no modelo estudado.

3.4. Considerações Finais

Foi apresentado neste capítulo a importância da tomada de decisão na otimização em critérios que possibilitam aumentar o rendimento da produtividade e diminuir o desperdício de matéria prima no processo de corte.

Partindo deste enfoque, o tema proposto faz referência ao problema de corte unidimensional, utilizando fundamentos computacionais, sendo que a solução final está divida em duas partes.

A primeira resolução consiste em dois métodos, o PK que delimita a quantidade itens em uma determinada barra de maneira que não ultrapasse o seu comprimento total, enquanto o AG tem como função gerar os possíveis padrões de corte. Já a segunda resolução é obtida através do gerador de arquivos com extensão .ltx, desenvolvido para adicionar os valores que serão automaticamente representados no software LINDO, tendo como finalidade aplicar o método simplex com fulcro nos padrões de corte definidos anteriormente pelo AG.

Para uma maior organização do projeto, serão mostrados alguns trechos de código desenvolvidos para solução proposta. Os algoritmos estarão listados no Apêndice A.

No próximo capítulo, serão descritos os resultados e as comparações da solução proposta em dois cenários reais.

4. ANÁLISE E RESULTADOS

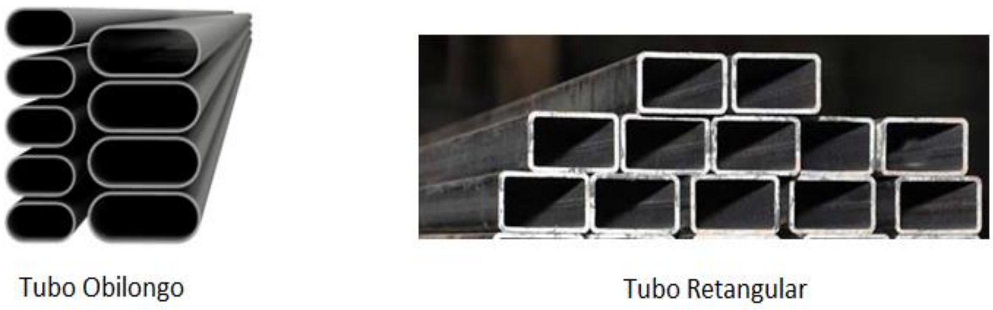
Esse capítulo refere-se aos resultados computacionais, com a finalidade de analisar e comparar as soluções obtidas pelo aluno que cursou engenharia de produção na UNIVEM em 2014. Para esse estudo em particular, a característica de interesse do problema é a otimização da perda de matéria-prima no processo de corte.

As próximas seções detalham os testes realizados em dois cenários reais, nomeados de A e B, nos quais ambos registram os resultados alcançados na proposta de solução do problema de corte unidimensional.

4.1. Ambiente de Teste

Para realizar os testes deste trabalho foram utilizados dois perfis de barra: o tubo obilongo e tubo retangular.

Figura 14 Tubos usados em produção.



Fonte: Elaborado pelo autor.

Na Figura 14 são representados os tubos usados em produção com comprimento padrão de 6000 mm.

A partir destes cenários, as análises e comparações de resultados estarão a seguir nas próximas seções 4.1.1 e 4.1.2. Os processos para o desenvolvimentos das tabelas de solução são apresentados a seguir:

- C_i = Padrão de Corte (i = 1 até 5).
- P/P AC. = Perda por Padrão Acumulada.
- P/P TOTAL = Perda por Padrão Total.
- DEMANDA AT. = Demanda Atendida.
- DEMANDA PRE. = Demanda Prevista.

Outro ponto importante a ser destacado em relação ao padrão de corte do problema estudado é:

- Perda programada: Consiste na perda de 160 mm em todas as barras, onde o corte não se faz possível, devido ao comprimento da máquina.
- *Presset*: Sua função é realizar um corte de 5 mm no início de cada barra antes da produção, para obter o alinhamento correto das peças finais do produto.

Portanto, a perda programada quanto o *presset* estarão inclusos no modelo de otimização. A Figura 15 ilustra a perda final na área de corte que passa a ter agora 5835 mm numa barra de 6000 mm.

Figura 15 – Perda final na área de corte.

Fonte: MORAES, 2014.

Ressalta-se que a perda estará sendo representada pela cor vermelha, tanto na imagem a cima como nas próximas ilustrações.

4.1.1. Cenário A

Neste cenário o perfil de barra estudado é o tubo obilongo com a medida de 120 x 60 x 6000 mm. Para representação da matéria-prima usada em produção, a Tabela 4 ilustra o comprimento (tamanho do item) e a demanda (quantidade de peças produzidas para cada item).

Tabela 4 – Dados do Cenário A.

ITEM	COMPRIMENTO	DEMANDA
1	1194	10
2	689	8
3	590	7
4	508	9
5	370	10

Fonte: MORAES, 2014.

A partir desse cenário, serão mostrados os resultados obtidos por Moraes (2014) e o método proposto no presente trabalho.

Projeto desenvolvido por Moraes

Segundo o trabalho de Moraes (2014), sua proposta de solução resultou no desperdício de 2 barras do estoque para atender a demanda prevista, alcançando uma P/P TOTAL de 226 mm. A Tabela 5 apresenta os valores referentes ao seu processo de execução.

Tabela 5 – Solução proposta por Moraes no Cenário A.

Padrão de Corte executado	ITENS	Frequência utilizada por padrão	P1	P2	P3	P4	P5	DEMANDA AT.	DEMANDA PRE.
C1	ITEM 1	1	4	1	2	1	2	10	10
C2	ITEM 2	1	0	4	3	0	1	8	8
C3	ITEM 3	1	0	3	1	3	0	7	7
C4	ITEM 4	1	2	0	0	4	3	9	9
C5	ITEM 5	1	0	0	2	2	3	7	10
P/P AC			13	75	10	49	79	P/P TOTAL	226

Fonte: Adaptado de MORAES, 2014.

Método proposto

Para chegar até solução proposta, é necessário encontrar os padrões de corte e aplica-los no gerador de arquivos com extensão .ltx, referenciados no capítulo 3 nas Figuras 3.4 e 3.5.

Neste cenário, foram desperdiçadas 1 barra e mais 20% da segunda barra do estoque para atender a demanda prevista, sendo que a P/P TOTAL é de 145 mm, contendo uma diferença de 81 mm em comparação ao método apresentado por Moraes (2014). A Tabela 6 representa os resultados a seguir.

Tabela 6 – Solução proposta no Cenário A.

Padrão de Corte executado	ITENS	Frequência utilizada por padrão	P1	P2	P3	P4	DEMANDA AT.	DEMANDA PRE.
C1	ITEM 1	2	1	1	2	1	10	10
C2	ITEM 2	1	3	2	1	1	7	8
C3	ITEM 3	1	2	1	1	3	7	7
C4	ITEM 4	1	2	3	2	2	9	9
C5	ITEM 5	1	1	3	3	3	10	10
P/P AC			8	39	42	56	P/P TOTAL	145

Fonte: Elaborado pelo autor.

4.1.2. Cenário B

O Cenário B corresponde ao plano de corte efetuado no perfil de barra retangular 100 x 50 x 6000 mm. A Tabela 7 ilustra o comprimento (tamanha do item) e a demanda (quantidade de peças produzidas para cada item) no processo de execução da matéria-prima aplicado em produção.

Tabela 7 – Dados do Cenário B.

ITEM	COMPRIMENTO	DEMANDA
1	1560	11
2	889	15
3	750	10
4	655	10
5	165	12

Fonte: MORAES, 2014.

Assim, como demonstrado no item 4.1.1, serão apresentados também no cenário B os resultados adquiridos por Moraes (2014) e método proposto do projeto.

Projeto desenvolvido por Moraes

A Tabela 8 corresponde os valores pertencentes ao processo de execução encontrado por Moraes (2014). Sua proposta de solução, resulta na perda de 5 barras do estoque para atender a demanda prevista, alcançando uma P/P TOTAL de 620 mm.

Tabela 8 – Solução proposta por Moraes no Cenário B.

Padrão de Corte executado	ITENS	Frequência utilizada por padrão	P1	P2	P3	P4	P5	DEMANDA AT.	DEMANDA PRE.
C1	ITEM 1	1	2	6	3	0	0	11	11
C2	ITEM 2	2	3	0	0	8	4	15	15
C3	ITEM 3	1	0	0	1	2	6	9	10
C4	ITEM 4	2	0	2	0	4	4	10	10
C5	ITEM 5	2	0	4	2	2	4	12	12
P/P AC			23	280	45	28	244	P/P TOTAL	620

Fonte: Adaptado de MORAES, 2014.

Seguindo a mesma lógica do item 3.3.2, obteve-se um tempo de execução entre 7 min e 10 seg até 7 min e 20 seg.

Método Proposto

Seguindo o mesmo critério de execução realizado no método proposto do item 4.1.1 a Tabela 9 ilustra a solução obtida.

Neste cenário, foram desperdiçadas 1 barra e 1/2 do estoque para atender a demanda prevista, observando-se que a P/P TOTAL é de 163 mm, contendo uma diferença de 457 mm a menos que o método apresentado por Moraes (2014).

Tabela 9 – Solução proposta no Cenário B.

Padrão de Corte executado	ITENS	Frequência utilizada por padrão	P1	P2	DEMANDA AT.	DEMANDA PRE.
C1	ITEM 1	5	1	1	10	11
C2	ITEM 2	5	2	1	15	15
C3	ITEM 3	2	2	3	10	10
C4	ITEM 4	5	1	1	10	10
C5	ITEM 5	3	2	2	12	12
P/P AC			12	151	P/P TOTAL	163

Fonte: Elaborado pelo autor.

4.2. Análise dos resultados

Os experimentos e cenários apresentados no presente trabalho abordam todas as situações citadas no contexto descrito no item 3.1.

Visando um detalhamento para análise dos resultados, procurou-se resolver o problema referente ao desperdício da matéria-prima com comparações das P/P TOTAL aplicados em cálculos de porcentagem (%) nos cenários A e B. Desta forma, a diferença entre a solução encontrada por Moraes (2014) e o método computacional proposto será perfeitamente visível.

É importante ressaltar, também, que a P/P TOTAL se dá pela soma das sobras no final de cada barra, lembrando que no método apresentado foi utilizada uma barra a menos que no atual.

Para uma maior organização na representação das tabelas, todos os dados ilustrados serão apurados da seguinte forma:

- Execução da produção (Dias).
- Desperdício de barra (%).
- Perda de barra (Qtde).

Na Tabela 10, são mostrados os resultados encontrados em cada projeção de produção no cenário A com as duas soluções referentes ao processo de corte para tubos obilongos.

Tabela 10 – Projeção de produção no Cenário A.

Projeto desenvolvido por Moraes

Cenário A		
Execução da produção (Dias)	Desperdício de Barra (%)	Perda de Barra (Qtde)
1 Semana	3,9	
1 Mês	15,6	
1 Semestre	93,6	
1 Ano	187,2	

Método proposto

Cenário A		
Soft. Execução da produção (Dias)	Desperdício de Barra (%)	Perda de Barra (Qtde)
1 Semana	2,5	
1 Mês	10	
1 Semestre	60	
1 Ano	120	

Fonte: Elaborado pelo autor.

Realizando o mecanismo proposto por Moraes (2014), nota-se que na Tabela 10, quando a execução da produção (Dias) é aplicada apenas 1 dia por Semana, gerando um desperdício de 3,9% por barra utilizada em produção, enquanto em 1 Ano está margem de porcentagem pode significar na perda de quase 2 barras para atender a demanda.

Pela solução computacional, quando o método proposto é executado apenas 1 dia por Semana, gera-se 2,5% de desperdício por barra na produção, denotando a princípio uma economia de 1,4%, determinando, portanto, um consumo de 67,2% em relação a 1 Ano no processo produtivo apresentado por Moraes.

Na Tabela 11 a seguir, serão mostrados os resultados alcançados em cada projeção de produção no cenário B com as duas soluções referentes ao processo de corte para tubos retangulares.

Tabela 11 – Projeção de produção no Cenário B.

Projeto desenvolvido por Moraes

Cenário B		
Execução da produção (Dias)	Desperdício de Barra (%)	Perda de Barra (Qtde)
1 Semana	10,6	
1 Mês	42,4	
1 Semestre	254,4	
1 Ano	508,8	

Método proposto

Cenário B		
Soft. Execução da produção (Dias)	Desperdício de Barra (%)	Perda de Barra (Qtde)
1 Semana	2,8	
1 Mês	11,2	
1 Semestre	67,2	
1 Ano	134,4	

Fonte: Elaborado pelo autor.

Efetuando a resolução proposta por Moraes (2014), nota-se que na Tabela 11, quando a execução da produção (Dias) é aplicada apenas 1 dia por Semana, gerando um desperdício de 10,6% por barra utilizada em produção, enquanto em 1 Ano está margem de porcentagem pode significar na perda de 5 barras para atender a demanda.

Entretanto na solução computacional, quando o método proposto é executado apenas 1 dia por Semana, gera-se 2,8% de desperdício por barra na produção, obtendo inicialmente uma economia de 7,8%, demostrando, assim, um consumo de 374,4% em relação a 1 Ano no processo produtivo de Moraes.

4.3. Considerações Finais

Após a análise dos resultados pode-se perceber que nos dois cenários mencionados, houve a possibilidade de avaliar e comparar as soluções propostas para o contexto apresentado neste capítulo

Com a implementação dos algoritmos é visível comprovar as grandes vantagens em questão de desempenhos, como:

- A corretude do problema resolvido pelo algoritmo.
- O melhoramento na qualidade das soluções.
- A otimização no ganho de consumo de matéria-prima.

As resoluções colhetadas permitem concluir que as metodologias escolhidas e as soluções apresentadas se mostraram apropriadas para a otimização no processo de corte em tubos.

CONCLUSÃO

O trabalho foi desenvolvido com a finalidade de implementar uma solução computacional para minimizar o desperdício de tubos. Necessário se faz mencionar que foi possível verificar todos os resultados obtidos e, consequentemente, determinar uma boa estratégia de corte, sobre como cortar a matéria-prima principal (objetos) em peças que irão compor os produtos finais (itens).

Os algoritmos desenvolvidos apresentaram uma condição satisfatória para resolução aplicada no problema de corte de estoque unidimensional.

A heurística de busca utilizada como principal fundamento para se chegar a solução final foi o algoritmo genético, que mostrou ser uma alternativa viável para encontrar os padrões de corte, apesar de possuir pequenos números de variáveis do problema proposto. O desempenho deste algoritmo é influenciado, pelos seguintes parâmetros: **População Inicial**, sendo formada por um conjunto de indivíduos. O **Fitness**, que significa que o indivíduo está apto ou não para participar da nova população. A **Seleção**, responsável por realizar a reprodução. O **Crossover**, que obtém a taxa de reprodução, ou seja, os filhos gerados pela seleção e a **Mutação**, responsável por fazer a alteração na taxa de combinação entre os genes de cada cromossomo.

Já o gerador de arquivos com extensão .ltx é usado para inserir os valores que serão automaticamente representados pelo software LINDO, na versão 6.1, tendo como principal função resolver o método simplex. Além de atender as necessidades previstas, o software também apresentou resultados com elevada confiabilidade e rapidez, observando-se que para a realização deste estudo foi utilizada a versão demonstrativa do referido software, que se mostrou suficiente para fins acadêmicos.

De acordo com os dois cenários testados, as comparações e análises feitas no projeto desenvolvido por Moraes (2014) e o estudo detalhado das soluções do método proposto comprovam que as resoluções computacionais possuem uma boa eficiência, alcançando um desempenho melhor nas soluções viáveis para o problema de corte de estoque unidimensional.

Desta forma, é possível perceber que a pesquisa operacional por meio de suas técnicas de solução como a programação linear e o método heurístico, com os devidos conhecimentos necessários para aplicação de cada um dos métodos específicos, consegue estabelecer boas escolhas referentes a tomadas de decisão sobre o problema de corte em tubos, a fim de otimizar todo o processo produtivo.

Trabalhos Futuros

Apesar dos resultados promissores, muitos estudos ainda devem ser realizados a respeito do tema analisado. Para trabalhos futuros, buscando o aprimoramento do modelo de otimização para problemas de corte em tubos, sugere-se:

- Melhorar a forma de como encontrar os padrões de corte usados na resolução final, desenvolvendo um algoritmo capaz de otimizar a busca no banco de soluções possíveis;
- Testar a eficiência de outras heurísticas em relação ao tempo de execução e ao grau de exaustão dos experimentos para que os parâmetros possam ser configurados com a máxima precisão; e
- A necessidade de comparar as soluções obtidas neste trabalho com implementações similares, a fim de demonstrar um maior detalhamento quanto à qualidade das soluções encontradas.

REFERÊNCIAS

ANDRADE, Eduardo Leopoldino de. Introdução à Pesquisa Operacional: métodos e modelos para análise de decisões. 4.ed. – Rio de Janeiro: LTC, 2009.

ARAÚJO, Olindo César Bassi de. Problemas de Corte e Empacotamento Tridimensional e Integração com Roteamento de Veículos, 2006. Tese de doutorado, Faculdade de Engenharia Elétrica e de Computação da Universidade de Campinas – UNICAMP.

BRANDÃO, Julliany Sales. Aplicação de algoritmos genéricos para minimização do número de objetos processados e o setup num problema de corte unidimensional, 2009. Dissertação de mestrado, Universidade do Estado do Rio de Janeiro – UERJ. Disponível em: <http://www.bdtd.uerj.br/tde_busca/arquivo.php?codArquivo=957> Acesso em 04 mai 2016.

CAMPONOGARA, Eduardo. Métodos de Otimização: teoria e prática, 2006. Tese de doutorado, Universidade Federal de Santa Catarina.

CARDOSO, Andréa. Fundamentos da Pesquisa Operacional. Monografia, 2011. UNIFAL. Disponível em: <http://www.unifal-mg.edu.br/matematica/files/file/po.pdf>. Acesso em 08 set 2016.

CHERRI, Adriana Cristina. O problema de corte de estoque com reaproveitamento das sobras de material, 2006. Dissertação de mestrado, Instituto de Ciências Matemáticas e de Computação – ICMC – USP.

COLIN, Emerson Carlos. Pesquisa Operacional: 170 Aplicações em estratégia, finanças, logística, produção, marketing e vendas. [Reimpr.] – Rio de Janeiro: LTC, 2013.

COSTA, Carine Rodrigues da. Condução e Experimentos Computacionais com Métodos Heurísticos, 2011. Dissertação de mestrado, Instituto de Informática da Universidade Federal de Goiás.

Dyckhoff, H. (1990). A Topology of Cutting and Packing Problems. European Journal of Operational Research, v. 44, p. 145-159.

GRECCHO, Thiago Xavier. Aplicação de técnicas de decomposição em problema de corte de estoque, 2013. Dissertação de mestrado, Universidade Estadual Paulista – UNESP.

GHIDINI, Carla Taviane Lucke da Silva. Otimização de processos acoplados: programação da produção e corte de estoque, 2009. Tese de doutorado, Instituto de Ciências Matemáticas e de Computação – ICMC – USP. Disponível em: <http://www.teses.usp.br/teses/disponiveis/55/55134/tde-13022009-102119/pt-br.php>. Acesso em 22 mar 2016.

HEIS, Adriano. et al. Um algoritmo genético para o problema de corte unidimensional inteiro. Simpósio Brasileiro de Pesquisa Operacional, 2010.

JUNIOR, Douglas José Alem. O problema de corte de estoque com demanda estocástica, 2007 Dissertação de mestrado, Instituto de Ciências Matemáticas e de Computação – ICMC – USP.

JÚNIOR, Sérgio Maurício Trad. Problemas de otimização, 2003. Monografia, Universidade Presidente Antônio Carlos.

LACHTERMACHER, Gerson. Pesquisa Operacional na Tomada de Decisões. 4 ed. São Paulo: Pearson Prentice Hall, 2009.

LEÃO, Aline Aparecida de Souza. Extensões em problema de corte: padrões compartimentos e problemas acoplados, 2013. Tese de doutorado, Instituto de Ciências Matemáticas e de Computação – ICMC – USP.

MARQUES, Fabiano do Prado. O problema da mochila compartimentada e aplicações, 2004. Tese de doutorado, Instituto de Ciências Matemáticas e de Computação – ICMC – USP. Disponível em: <http://www.teses.usp.br/teses/disponiveis/55/55134/tde-07072015-155114/pt-br.php> Acesso em 28 abr 2016.

MORAIS, Rafael Diogo Vasconcellos. Utilização de um Algoritmo com Representação Gráfica para o Problema de Corte Bidimensional: estudo de caso para uma marcenaria, 2011. Dissertação de Mestrado, Universidade de São Paulo – USP.

MORABITO, Reinaldo. et al. Uma abordagem para o problema de corte de chapas de fibra de madeira reconstituída, 2005.

MORAES, Rafael Júnior de. Desenvolvimento de Modelo de Otimização para Problema de Corte em Tubos, 2014. Monografia, Centro Universitário Eurípides de Marília – UNIVEM. Acessado em 02 mar 2016.

PINTO, Maria. Algumas Contribuições à Resolução do Problema de Corte Integrado ao Problema de Sequenciamento dos Padrões, 2004. Tese de doutorado, INPE. Disponível em:<http://150.163.34.248/col/sid.inpe.br/jeferson/2004/07.02.13.58/doc/publicacao.pdf>. Acesso em 26 fev. 2016.

PINTO, Thiago de Souza. Uma proposta para resolver o Problema de Corte de Estoque Unidimensional com reaproveitamento de sobras por meio de dois objetos, 2008. Dissertação de mestrado, Universidade Estadual de Londrina – UEL. Disponível em: <http://www.bibliotecadigital.uel.br/document/?code=vtls000147646> Acesso em 05 mai 2016.

POLDI, Kelly C. e ARENALES, Marcos N. Heurísticas para o problema de corte de estoque unidimensional inteiro, 2006. Pesquisa Operacional vol 26, Instituto de Ciências Matemáticas e de Computação – ICMC – USP.

QUEIROZ, Maciel Manoel de. Métodos heurísticos aplicados ao Problema de Programação da frota de navios PLVs, 2011. Dissertação de mestrado, Escola Politécnica da Universidade de São Paulo.

SILVA, Sarah. Um estudo do problema integrado de dimensionamento de lotes e corte de estoque para uma fábrica de móveis de pequeno porte, 2008. Tese de doutorado, Escola Politécnica Universidade de São Paulo. Disponível em: <http://pro.poli.usp.br/wp-content/uploads/2012/pubs/otimizacao-no-processo-de-corte-unidimensional-de-barras-de-aco.pdf>. Acesso em: 26 fev. 2016.

TEMPONI, Elias Carlos Correia. Uma Proposta de Resolução do Problema de Corte Bidimensional via Abordagem Metaheurística, 2007. Dissertação de mestrado, Centro Federal de Educação Tecnológica de Minas Gerais. Disponível em: <http://www.mmc.cefetmg.br/info/downloads/D031-EliasCarlosCorreaTemponi.pdf>. Acesso em: 02 mar 2016.

WÄSCHER, G. et al. An Improved Typology Cutting and Packing Problems”. European Journal of Operational Research, (2006).

APÊNDICE A

Serão demonstrados nesta seção, alguns dos principais trechos de códigos para o desenvolvimento da solução final do projeto.

A.1. Algoritmo Genético

```
#Execução Principal
for ifora in range(1,201):
#Criando a cadeia de cromossomos inteira
    populacao = [
        "".join(    #Quantidade de itens utilizados no tamanho do gene
            ("".join(choice(['0','1']) for j in range(tam_gene-1)) + '1')
            for k in range(len(itens)))
        for i in range(P)
    ]
    for i in range(1000):
        pop_selecionada = torneio(populacao)
        shuffle(pop_selecionada)    #embaralha tudo
        for i in range(0,len(pop_selecionada),2): #Começa em 0, lê o pop_selecionada inteiro, pegando de 2 em 2 (seria um casal)
            papai,mamae = i,(i+1) % (len(pop_selecionada)) #índices pai e mãe
            if uniform(0,1) > CR:
                continue
            pop_selecionada[papai],pop_selecionada[mamae]
            = crossover(pop_selecionada[papai],pop_selecionada[mamae])
        for i in range(0,len(pop_selecionada)):
            pop_selecionada[i] = mutacao(pop_selecionada[i]) #Faz a mutação
        populacao = pop_selecionada #Atualiza a população no final
                                    #e recomeça tudo novamente
```

A.2. Arquivo .bat

```
@echo off
cls
Echo Batch Executar arquivo Python que gera um arquivo .ltx
Echo .
pause
@echo off
gera_arquivo_lindo_ltx.py %* #Nome do arquivo.py que será executado
@echo off
:ops
"C:\LINDO61\Lindow32.exe" #Caminho de execução do Software LINDO
:fim
echo concluido
pause
```

A.3. Gerador de arquivos .ltx

```
#Execução Principal / Entrada para a F.O.
funObj = raw_input("Digite a Funcao Objetivo: ")  # Exemplo MIN X1 + X2
my_list = list() #Criando lista
qtde = int(input("Digite a quantidade de restricoes: "))
n = 0
while (n < qtde):
    restricao = raw_input("\nDigite o Padrao de Corte %d): " % (n + 1))
    print ("\nRestricao digitada  " + restricao + "\n\n")
    arquivo.write("\t\t" + restricao + "\t\t\n")
    n += 1
    my_list.append(restricao)  # Add todas os padroes na lista: my_list
```