

INSTITUTO FEDERAL DE SANTA CATARINA

CRISTIAN DE SOUZA

OTIMIZAÇÃO EM PROBLEMAS DE CORTES
UNIDIMENSIONAIS PARA ELETRODUTOS – UMA SIMULAÇÃO INDUSTRIAL

Jaraguá do Sul - SC

2016

CRISTIAN DE SOUZA

OTIMIZAÇÃO EM PROBLEMAS DE CORTES UNIDIMENSIONAIS PARA
ELETRODUTOS– UMA SIMULAÇÃO INDUSTRIAL

Trabalho de Conclusão de Curso apresentado ao curso Superior de Tecnologia em Fabricação Mecânica do Câmpus Geraldo Werninghaus do Instituto Federal de Santa Catarina para obtenção do diploma de Tecnólogo em Fabricação Mecânica.

Orientador: Prof. Dr. Gerson Ulbricht.

Prof. Ma. Miriam Hennig

Jaraguá do Sul – SC

2016

S729o Souza, Cristian de

Otimização em problemas de cortes unidimensionais para eletrodutos : uma simulação industrial / Cristian de Souza ; orientadores Gerson Ulbricht, Miriam Hennig . – Jaraguá do Sul, SC, 2016.

78 f.

Monografia (Graduação) – Instituto Federal de Santa Catarina.
Curso Superior de Tecnologia em Fabricação Mecânica

Inclui bibliografia

1. Cortes unidimensionais. 2. Algoritmos. 3. Geração de padrões para eletrodutos . I. Ulbricht, Gerson. II. Henning, Miriam. III. Instituto Federal de Santa Catarina. Curso Superior de Tecnologia em Fabricação Mecânica. IV. Título.

CDD 620.1

Catalogado por: Karla Viviane Garcia Moraes – CRB14/1002

OTIMIZAÇÃO EM PROBLEMAS DE CORTES UNIDIMENSIONAIS PARA
ELETRODUTOS – UMA SIMULAÇÃO INDUSTRIAL

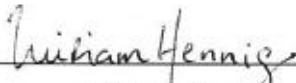
CRISTIAN DE SOUZA

Este trabalho foi julgado adequado para obtenção do Título de Tecnólogo em fabricação mecânica e aprovado na sua forma final pela banca examinadora do Curso superior em tecnologia em fabricação mecânica do Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina.

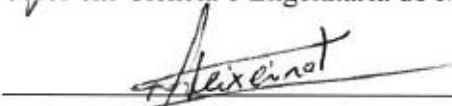
Jaraguá do Sul, 30 de agosto de 2016.

Banca Examinadora:


Gerson Ulbricht
Doutor em Métodos Numéricos em Engenharia


Miriam Hennig
Mestra em Letras


Joel Stryhalski
Doutor em Ciência e Engenharia de Materiais


Edson Sidnei Maciel Teixeira
Mestre em Engenharia de Produção/Doutorando em Design

Dedico a todos que de alguma forma
contribuíram para minha vida pessoal,
profissional e acadêmica.

AGRADECIMENTOS

À minha família, pelo apoio e incentivo durante o curso.

À minha namorada, pela paciência.

Ao orientador, pelo incentivo e apoio.

Aos meus amigos do IFSC, pelos momentos de solidariedade.

Ao IFSC – Campus Geraldo Werninghaus e a todos que participaram de modo geral nesse processo de formação.

"Não basta ensinar ao homem uma especialidade, porque se tornará assim uma máquina utilizável e não uma personalidade. É necessário que adquira um sentimento, senso prático daquilo que vale a pena ser empreendido, daquilo que é belo, do que é moralmente correto." (Albert Einstein)

RESUMO

Este trabalho tem como objetivo estudar um problema de cortes unidimensionais e apresentar um algoritmo que auxilie na geração de padrões de cortes que compõe o modelo de otimização. Após estudos em trabalhos já existentes, foi proposto um modelo matemático em que para as entradas e saídas de dados foi utilizada uma planilha de cálculo e para resolução do modelo a ferramenta Solver. Devido à quantidade elevada de padrões de corte existentes foi desenvolvido um algoritmo responsável pela geração destes padrões, o qual foi implementado em Scilab. O modelo matemático e computacional desenvolvido foi aplicado em um problema industrial envolvendo corte de eletrodutos, em que resultados apontaram sua fácil implementação no auxílio da tomada de decisão de cortes, de modo a buscar a redução de desperdícios de matéria prima. Resultados obtidos na indústria, mostraram redução nos desperdícios no processo de corte, indicando uma área promissora para pesquisas futuras.

Palavras-Chave: Algoritmo, Cortes unidimensionais, Geração de padrões para eletrodutos.

ABSTRACT

This work aims to study a problem of one-dimensional cuts and present an algorithm that helps in cutting patterns. After studies in existing models it was proposed an optimization model in which the data inputs and outputs in are a spreadsheet and for resolution of the model Solver tool was used. Due to the large amount of existing cutting patterns we developed an algorithm responsible for generating these patterns, which is implemented in Scilab. The mathematical and computational model developed was applied in an industrial problem involving cutting conduit, which obtained results showed its easy practical implementation of the aid decision to seek to reduce waste of raw materials. Results in the industry, showed a reduction of waste in the cutting process, indicating a promising area for future research.

Keywords: Algorithm, One-dimensional cut, Pattern generation for rigid conduits.

LISTA DE FIGURAS

Figura 1 – Requisitos da pesquisa operacional.....	21
Figura 2 – Fases para pesquisa operacional	22
Figura 3 - Região de Soluções ou Região Factível.....	30
Figura 4- Região de Soluções ou Região Factível.....	33
Figura 5- Região de Soluções Inteiras	34
Figura 6 – Pontos Referentes a Soluções Inteiras.....	36
Figura 7 - Esquema de um hidrogerador vertical	38
Figura 8 - Hidrogerador típico.....	39
Figura 9 - Exemplo de encaminhamento de cabos até painéis	39
Figura 10 - Instalação típica.	41
Figura 11 – Tipo de cortes no processo	42
Figura 12 – Fluxograma Algoritmo de busca ordenada	45
Figura 13 – Fluxograma Algoritmo de busca aleatória	49
Figura 14 – Planilha de padrões Pi	51
Figura 15 – Interface da planilha.....	52
Figura 16 – Variáveis de decisão e restrições.....	52
Figura 17 – Ilustração Parcial da Janela do Código Scilab	54
Figura 18 – Entrada de dados Scilab	55
Figura 19 – Parcial dos valores encontrados (considerando possíveis repetidos).....	56
Figura 20 – Parcial dos padrões encontrados (desconsiderando repetidos)	57
Figura 21 – Saída de dados do Scilab gravada em arquivo .txt.....	57
Figura 22 – Entrada dados no Excel.....	58
Figura 23 – Interface planilha – entrada de dados.....	59
Figura 24 – Planilha de dados para a ferramenta Solver	60
Figura 25 – Parâmetros do Solver	61
Figura 26 – Resultado Solver	62
Figura 27 – Resposta padrão	63

LISTA DE TABELAS

Tabela 1 – Dados gerais do problema	28
Tabela 2 – Valor da função objetivo para os pontos $P(x, y)$ correspondentes ao vértice.....	31
Tabela 3 – Valor da função objetivo para os pontos $P(x, y)$ correspondentes ao vértice.....	35
Tabela 4 – Testes para cada ponto factível pertencente à Fronteira de Soluções.....	37
Tabela 5- Comprimentos das peças	41
Tabela 6 – Exemplos de Padrões de corte (P_i)	43

LISTA DE ABREVIATURAS E SIGLAS

PCH - Pequenas centrais hidroelétricas

P.O. - Pesquisa operacional

Xcut1D - Algoritmo para otimização de corte unidimensional

PPL – Problemas de Programação Linear

M.O. – Mão de obra

H. M. – Hora máquina

F.O. – Função objetivo

SUMÁRIO

1 INTRODUÇÃO	15
1.1 Problema.....	16
1.2 Justificativa.....	16
1.3 Objetivos.....	17
1.3.1 Objetivo Geral	17
1.3.2 Objetivos Específicos	17
2 FUNDAMENTAÇÃO TEÓRICA.....	18
2.1 PCHs e hidrogeradores	18
2.2 Eletroduto.....	19
2.3 Conceito de Algoritmo.....	19
2.4 Pesquisa Operacional	20
2.4.1 Programação Linear.....	22
2.4.2 Programação Inteira.....	23
2.4.3 Método Simplex	23
2.5 Função Solver para Resolução de Problemas de Programação Linear	25
2.6 Scilab.....	25
2.7 Problemas de cortes.....	26
2.8 Tipo de pesquisa.....	27
3 METODOLOGIA DE RESOLUÇÃO.....	28
3.1 Resolução de um Problema de Programação Linear Contínua	28
3.2 Exemplo de Problema de Programação Inteira.....	32
4 MODELAGEM MATEMÁTICA E COMPUTACIONAL PROPOSTA PARA RESOLUÇÃO EM UM PROBLEMA DE CORTES NA INDÚSTRIA	38
4.1 Descrição do Problema Aplicado	38
4.1.1 Características do componente	40

4.1.2 Padrões de cortes	42
4.2 Algoritmos Propostos para Geração de Padrões de Corte	43
4.2.1 Algoritmo de busca ordenada: baseado em busca exaustiva por todos os padrões de corte	44
4.2.2 Algoritmo de busca aleatória: baseado em busca aleatória de padrões de corte	46
4.3 Modelo Matemático	50
4.4 Planilha para Resolução do Modelo Matemático por meio da Ferramenta Solver	51
5 RESOLUÇÃO DE UM PROBLEMA APLICADO	53
5.1 Resolução em uma Abordagem Real na Indústria	53
6 CONCLUSÃO.....	64
REFERÊNCIAS	67
APÊNDICE	69
APÊNDICE A - ALGORITMO DE BUSCA ORDENADA: CÓDIGO EM SCILAB (FIXADO EM 13 PEÇAS)	70
APÊNDICE B - ALGORITMO DE BUSCA ALEATÓRIA: CÓDIGO EM SCILAB (PARA N PEÇAS)	75

1 INTRODUÇÃO

Atualmente no mercado competitivo, as empresas vêm buscando aprimoramento para minimizar os custos dos produtos dentro da sua gama de mercado, tentando reduzi-los sem perder a qualidade do produto e do processo. Neste caso, um dos pontos fundamentais para a saúde financeira empresarial é a compra da matéria prima na quantidade correta evitando grandes estoques imobilizados, o que pode ocasionar desvalorização de mercado, desperdício, além da ocupação de espaço físico.

Sabendo que as condições socioeconômicas em que o país atualmente se encontra e o poder de barganha de preço no mercado está acirrado, além de não poder recusar pedidos, mesmo que sejam especiais (fora do padrão da empresa), o que não satisfaz a necessidade de baixo custo, buscam-se meios para reduzir ou eliminar o desperdício dentro dos processos executados.

Considerando problemas envolvendo o corte de matéria prima (assunto que será tratado neste trabalho) é possível estudar as particularidades e limitações no processo por meio da utilização de ferramentas para auxiliar na tomada de decisões, utilizando, por exemplo, métodos matemáticos e algoritmos para a redução de desperdício em corte de eletrodutos. Pode-se dizer que a operação de corte é simples (ato de cortar peças), no entanto, desmembrando as possibilidades em pontos que podem gerar desperdício, pode-se ver que o problema é complexo, à medida que são adicionadas mais variáveis (comprimentos ou padrões de corte diferentes) a serem utilizadas.

Para a resolução de um problema de corte unidimensional, é necessário que se tenha uma função objetivo traçada, dentro das limitantes conhecidos na empresa e suas restrições delineadas, para que partir daí seja estabelecido um modelo de otimização que possa auxiliar no processo de decisão.

Neste trabalho foi realizado um estudo sobre o processo de corte de eletrodutos, para a partir daí se propondo um método de auxílio à decisão minimizando o desperdício de matéria prima utilizando o Scilab para gerações de padrões de corte e a ferramenta Solver do Excel, contribuindo assim, com uma melhor utilização de recursos.

1.1 Problema

Este trabalho foi elaborado a partir de uma situação problema pertencente a uma empresa do ramo de equipamentos elétricos do Estado de Santa Catarina, a qual tem em seu portfólio a fabricação, bem como o projeto de geradores síncronos verticais aplicados à pequena central hidroelétrica.

Nesta empresa, no departamento de engenharia, existe a operação de geração de documento de orientação de montagem, dimensões e quantidades para eletrodutos rígidos, usados para o encaminhamento de cabos e também para a compra do material, quando o fornecedor qualificado pela empresa entrega o eletroduto com o comprimento de três metros, sendo que atualmente o cálculo para a definição da compra do material é realizado manualmente, sem auxílio de métodos científicos. Nota-se que esse processo está suscetível a erros, pois não é informada a orientação de corte ao executor da operação em campo, o que pode ocasionar falta ou desperdício de material. Também não é considerado um valor de largura da serra no projeto.

A partir desses problemas, nota-se a necessidade de implementar um estudo para analisar qual é a melhor combinação de cortes, de modo a minimizar as perdas. Assim pode-se incorporar essa informação no documento para orientar os padrões de corte a serem utilizados para cada eletroduto, por meio do desenvolvimento de um modelo matemático bem como um algoritmo de geração de padrões de corte.

Nesse sentido, surge o problema: em uma aplicação envolvendo a demanda de eletrodutos, como implementar métodos que possam auxiliar na redução do desperdício de matéria prima ocasionado no processo de corte?

1.2 Justificativa

Processos que envolvem cortes geralmente são caracterizados pela grande ocorrência de perdas ocasionadas pela sobra de material. E a busca pela minimização dessas perdas é uma tarefa constante, e exige um complexo arranjo de combinações possíveis de padrões de corte, geralmente realizados de forma arbitrária, sem utilização de métodos científicos.

Busca-se assim, por meio desse trabalho, desenvolver métodos que permitam encontrar a maior quantidade possível de padrões de cortes considerando como entrada o

comprimento do eletroduto comercial, largura da serra e comprimentos das peças cortadas. A partir desses padrões, busca-se informar qual é a melhor combinação de cortes de eletrodutos, de modo a minimizar as perdas, incorporando essa informação no documento a fim de orientar os padrões de cortes a serem utilizados para cada demanda a ser cumprida.

1.3 Objetivos

1.3.1 Objetivo Geral

Estruturar um método de auxílio à decisão no processo de cortes unidimensionais para aplicá-lo em cortes de eletrodutos na operação e especificação de corte, buscando a minimização do desperdício de matéria prima.

1.3.2 Objetivos Específicos

- Compreender o funcionamento de problemas de cortes unidimensionais envolvendo a programação linear inteira;
- Traçar um método para geração de padrões em problemas de cortes unidimensionais;
- Estruturar um algoritmo para geração de padrões para cortes unidimensionais;
- Aplicar o método desenvolvido no problema de corte de eletrodutos em uma simulação industrial.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo é apresentada uma revisão de literatura abordando tópicos das áreas envolvidas na pesquisa, as quais foram distribuídas em tópicos conforme mostrado a seguir.

2.1 PCHs e hidrogeradores

A energia elétrica é de suma importância para as atividades humanas executadas rotineiramente, seja em casa, em uma atividade doméstica ou na indústria, bem como no processamento da matéria prima. Também pode ser combinada com outros bens ou serviços que podem satisfazer as necessidades do ser humano no local onde vive. É importante frisar que o desenvolvimento de modo geral está atrelado também à energia elétrica e que nos tempos atuais existem diferenças em níveis de consumo de energia por habitante. Para atender ao consumo brasileiro, a hidroelétrica cumpre esta importante função (BORGES NETO; CARVALHO, 2013).

A geração hidroelétrica é considerada como um privilégio para poucos países, pois a disponibilidade de extensão de territórios, condições geográficas e mananciais são fatores que são de grande relevância para sua concretização. O Brasil está em uma condição favorável para esse tipo de geração de energia, uma vez que é o segundo maior produtor de energia elétrica do mundo. Para essa geração são aplicadas as centrais hidroelétricas, sejam elas grandes ou pequenas. No entanto, as grandes centrais hidroelétricas têm sido evitadas devido aos impactos ambientais de grande escala gerados por elas, o que favorece a criação de pequenas centrais hidroelétricas (PCHs) (BORGES NETO; CARVALHO, 2013).

As PCHs têm como seus elementos principais o leito do rio, seguido de uma barragem de contenção da água, com um conduto por onde passa água do rio. Esse conduto direciona a água para a turbina que está acoplada a um gerador, localizado na casa de máquinas. A água sai pelo canal de restituição, daí seguindo para o leito normal do rio (BORGES NETO; CARVALHO, 2013).

Segundo Simone e Creppe (2013), o gerador elétrico, basicamente, é um conversor que recebe energia mecânica e a converte em energia elétrica. Dentro desse grupo destacam-se os geradores elétricos síncronos que, na sua maioria, são empregados na geração de energia

no Brasil. Quando o acionamento mecânico desse gerador síncrono é feito através de uma turbina hidráulica, o gerador ganha a denominação de hidrogerador.

2.2 Eletroduto

O eletroduto é um tubo que pode ser fabricado em materiais como, por exemplo, aço, plástico, polipropileno. Esse componente é amplamente utilizado em aplicações para instalações elétricas expostas, sejam elas em ambientes industriais ou comerciais. O eletroduto tem como função propiciar a proteção mecânica dos cabos que nele são alocados, protegendo-os também das intempéries do ambiente externo (por exemplo, corrosão, ações da atmosfera, agentes agressivos como sais e ácidos). A ele é possível atribuir a função de proteção contra incêndios provenientes de superaquecimento dos condutores internos (LIMA FILHO, 2011).

Já para Cavalin e Ceverlin (2011), os eletrodutos são considerados tubos de metal (magnéticos ou não) ou de PVC (policloreto de vinila), podendo ser rígidos ou flexíveis, com ou sem costura, sendo pintados internamente e externamente ou com tratamento galvanizado (aplicados a eletrodutos de aço), com a função de proteger os condutores elétricos contra possíveis danos mecânicos, corrosão e incêndio.

2.3 Conceito de Algoritmo

O algoritmo pode ser definido fazendo uma analogia à culinária, comparando-a por exemplo, com uma receita de um bolo na qual são descritos todos os ingredientes utilizados na preparação do bolo. Na sequência é informado o procedimento a ser seguido para a utilização de cada ingrediente e o correto momento de incluí-lo na receita. Assim, chega-se ao resultado final de um bolo. Transferindo essa analogia para o problema de cálculo, pode-se dizer que os ingredientes são as variáveis de entrada, e o procedimento é o cálculo com restrições e métodos a serem executados, e bolo é o resultado do cálculo ou solução proposta. Veja que para Boratti e Oliveira (2007, p.19) pode-se definir algoritmo como “uma sequência finita e lógica de instruções executáveis, especificadas em uma determinada linguagem, que mostram como resolver determinado problema”. Já para Goldbarg e Luna (2005, p. 499),

algoritmo é a “descrição do cálculo ou avaliação sistemática dessa função”.

Ele também pode ser definido como um passo a passo do que deve ser executado para chegar ao objetivo final. Esse passo a passo deve seguir um raciocínio lógico, ou seja, pensar em ordem, para que possamos evitar retrabalhos e processos longos até chegar ao objetivo final. Também se pode destacar que as informações descritas nesse passo a passo devem estar claras e precisas.

O algoritmo deve conter a estrutura básica contendo um início, um tempo de processo definido, um estado final e o resultado bem definido. Isso garante que ele execute um padrão ou procedimento para todos os problemas, sob as mesmas regras impostas, produzindo um resultado final em várias situações, dentro do formato do algoritmo, podendo ser executado quantas vezes forem necessárias. A importância do algoritmo é que se pode observar vários detalhes para incluir implementações futuras, melhorando seu desempenho, além de servir como orientação no processo de codificação, executado para criar interfaces em qualquer software de programação (FORBELLONE; EBERSPÄCHER, 2005).

Já para Aguilar (2008), o algoritmo é simplesmente um método para solucionar problemas, e frisa que para a solução do problema é necessário que haja três passos: o projeto de um algoritmo, sua codificação em uma linguagem de programação e a execução e validação do programa construído. Também destaca, assim como Forbellone e Eberspächer (2005), que o algoritmo deve ser preciso e bem definido, de modo a executar o mesmo algoritmo com os mesmos valores e chegar ao mesmo resultado, e que também deve ser finito, ou seja, ter um tempo de execução. E frisa que o algoritmo deve conter três partes fundamentais: a entrada, processamento e saída.

No caso de algoritmos aplicados a problemas de cortes unidimensionais, um trabalho foi desenvolvido por Cunha e Dias (1999), no qual foi desenvolvido um algoritmo para otimização de corte unidimensional chamado “Xcut1D”, o qual foi aplicado na resolução de um problema já publicado em outros artigos a fim de verificar sua eficácia. Concluiu-se que os resultados encontrados foram bons e que resultaram na redução de perdas, sendo o “Xcut1D” uma ferramenta de auxílio na tomada de decisões pelo analista de produção.

2.4 Pesquisa Operacional

Segundo Andrade (1998), a Pesquisa Operacional (PO) surgiu na Segunda Guerra

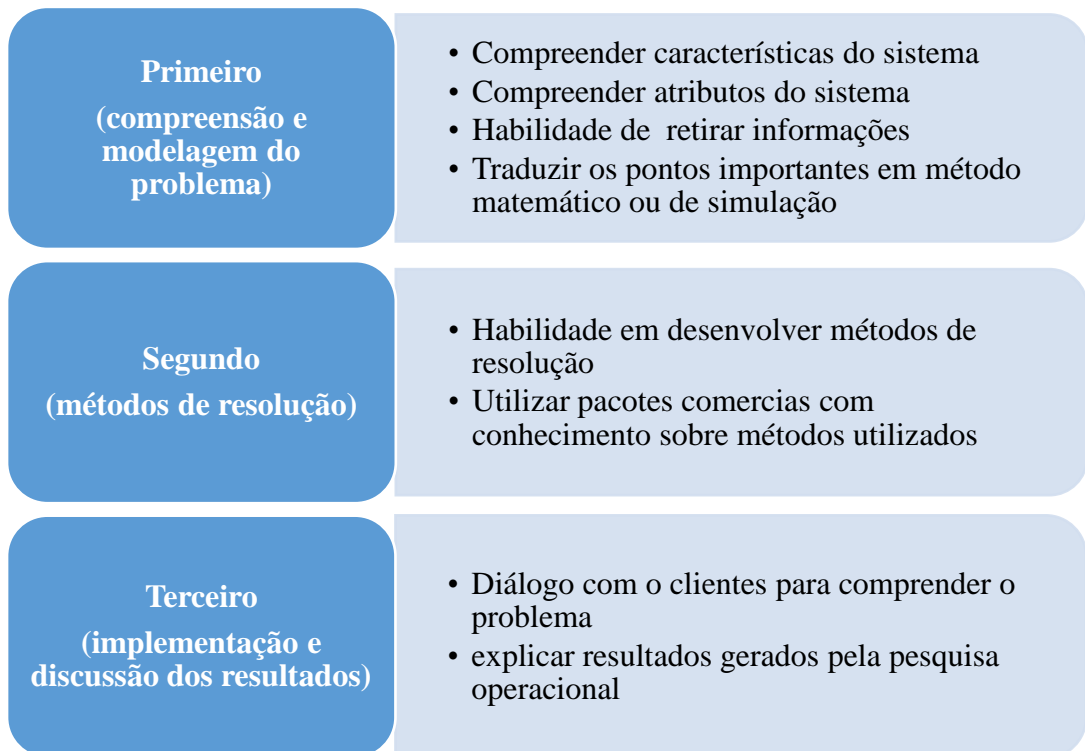
Mundial, com o propósito de auxiliar na solução de problemas em operações militares. Com sua aplicação essa ferramenta tornou-se atrativa para o mundo empresarial e acadêmico, geralmente focado em problemas de administração.

A PO destaca-se pela sua utilização em análises de decisão, aplicadas por grupos interdisciplinares, visando melhorar a utilização de matéria prima limitada na empresa, ou em uma otimização de processo. Ela é uma ferramenta flexível que se adequa a diversas situações do cotidiano da empresa, sendo baseada em modelos matemáticos, criando um ambiente de testes antes de implantar a solução, sendo possível analisar o impacto gerado e suas vantagens.

Já para Arenales *et al.* (2007), a PO é considerada uma aplicação de métodos científicos para encontrar soluções em problemas complexos, de modo a auxiliar em processos decisórios como, por exemplo, no planejamento e operacionalização de sistemas, focando na utilização eficiente de recursos escassos.

Arenales *et al.* (2007) citam ainda, na apresentação do livro, nas palavras de Gabriel R. Bitran, que para a utilização da Pesquisa Operacional, é necessário que estejam claros três requisitos:

Figura 1 – Requisitos da pesquisa operacional



A figura 1 adaptada de Arenales *et al.* (2007) deixa claro os passos a ser seguido durante o processo de utilização da pesquisa operacional, no qual enfatiza três requisitos que é desmembrado em tópicos para facilitar o desenvolvimento. Pode-se subter que a compreensão e modelagem do problema é o passo fundamental para que se possa implementar e discutir o resultados corretamente.

Silva *et al* (1998, p. 11) citam a Pesquisa Operacional como um “método científico de tomada de decisões”. Os autores deixam claro que se trata de modo geral da descrição de um sistema organizado, utilizando-se como auxílio um modelo para encontrar a melhor solução para executar o processo na empresa. Os autores também frisam seis fases para que a pesquisa operacional possa ser utilizada:

Figura 2 – Fases para pesquisa operacional

1ª	• Formulação do problema
2ª	• Construção do modelo
3ª	• Cálculo da solução através do modelo
4ª	• Teste do modelo e da solução
5ª	• Estabelecimento de controles da solução
6ª	• Implantação e acompanhamento

Fonte: adaptado de Silva *et al* (1998, p. 11)

A figura 2 adaptada de Silva *et al* (1998, p. 11) demonstra claramente as fases para que seja utilizada a pesquisa operacional, no qual determinam de modo sucinto as descrições das mesmas. Veja que estas fases vão de encontro com a ideia de Arenales *et al.* (2007).

2.4.1 Programação Linear

Uma ferramenta utilizada na resolução de problemas da Pesquisa Operacional é a programação linear, a qual permite criar modelos com grande quantidade de variáveis e restrições, sendo possível ser implementada computacionalmente. Problemas não lineares

podem muitas vezes ser linearizados, o que torna menos exaustiva (computacionalmente falando) a busca de soluções (PRADO, 1999).

A programação linear envolvendo variáveis de decisão que podem assumir qualquer valor real, é chamada de programação linear contínua. Já quando há variáveis inteiras, passa a ser chamada de programação inteira, a qual envolve uma quantidade maior de testes na busca de soluções viáveis, conforme será exemplificado mais adiante. (SILVA *et al*, 1998).

2.4.2 Programação Inteira

A Programação Inteira é semelhante à programação linear contínua, porém, consiste em considerar uma parte ou todas as variáveis de decisão inteiras, ou seja, tais variáveis não podem assumir valores decimais. Se há variáveis inteiras, esse problema caracteriza-se por um problema de programação inteira (LOESCH; HEIN, 2009).

Para Lachtermacher (2007), trata-se de problemas que envolvem programação matemática na qual uma ou mais variáveis de decisão assumem valores inteiros. O autor destaca dois tipos básicos de programação inteira: a total e a mista. Tem-se uma programação inteira total quando todas as variáveis assumem valores inteiros e uma programação mista quando é considerada uma parte dos valores como inteiros e a outra parte contínua ($\in R$).

2.4.3 Método Simplex

O método Simplex é um procedimento iterativo utilizado para resolver grandes sistemas envolvendo equações e inequações, o qual baseia-se na inversão sucessiva de matrizes, sendo constituído por uma lista de critérios e também de um teste de otimalidade (SILVA *et al.*, 1998). Para efetuar a aplicação, o problema deve ter uma solução inicial, pois a partir dessa solução serão calculadas outras soluções, com a troca de variáveis básicas por não básicas, criando novos resultados (ANDRADE, 2000).

Para esse método existem dois tipos de descrição: um para maximização e outro para minimização. A maximização consiste em aumentar os valores encontrados, como, por

exemplo, lucros de empresa ou produção de uma linha de produtos. Já a minimização de perdas pode adotar o mesmo procedimento da maximização, porém multiplicando a função objetivo por (-1) (PRADO, 1999). De modo geral, a aplicação do método Simplex indica que é necessária a existência de três características: a função objetivo, as variáveis de decisão e a solução inicial para que, a partir dessa, sejam testadas e comparadas novas soluções que, quando melhores, passam a substituí-las, tornando-se soluções correntes (SILVA *et al.*,1998).

Nota-se que o método gráfico possui implementação relativamente simples quando se trabalha com problemas com duas variáveis, já que sua representação se dá no espaço bidimensional (R^2), quando a região de soluções é o polígono limitado por retas.

Quando se trabalha com problemas com três variáveis, a implementação do método gráfico deverá ocorrer no espaço tridimensional (R^3), onde a região de soluções é o sólido geométrico (poliedro) limitado por planos, o que torna bastante complexa sua análise.

Já para mais de três variáveis trabalha-se no espaço n-dimensional (R^n), o que torna impossível a implementação gráfica. De modo geral, para problemas de programação linear é mais usual se trabalhar algebricamente ao invés de graficamente, recorrendo-se assim ao método Simplex.

O método Simplex é um procedimento iterativo utilizado para resolver grandes sistemas envolvendo equações e inequações, o qual baseia-se na inversão sucessiva de matrizes, sendo constituído por uma lista de critérios e também de um teste de otimalidade (SILVA *et al.*, 1998). Para efetuar a aplicação, o problema deve ter uma solução inicial, pois a partir dessa solução serão calculadas outras soluções, com a troca de variáveis básicas por não básicas, criando novos resultados.

Para esse método existem dois tipos de descrição: um para maximização e outro para minimização. A maximização consiste em aumentar os valores encontrados, como, por exemplo, lucros de empresa ou produção de uma linha de produtos. Já a minimização de perdas pode adotar o mesmo procedimento da maximização, porém multiplicando a função objetivo por (-1). De modo geral, a aplicação do método Simplex indica que é necessária a existência de três características: a função objetivo, as variáveis de decisão e a solução inicial para que, a partir dela, sejam testadas e comparadas novas soluções que, quando melhores, passam a substituí-las, tornando-se soluções correntes (SILVA *et al.*,1998).

2.5 Função Solver para Resolução de Problemas de Programação Linear

A função Solver do Excel é um suplemento que possibilita a resolução de cálculos complexos, o qual usa algoritmo de otimização linear e não linear. A ferramenta é de fácil utilização e encontra-se disseminada na indústria moderna atualmente. Nas palavras de Bueno (2007, p. 49):

O solver é uma ferramenta incorporada ao Excel, da Microsoft, que resolve modelos matemáticos usando algoritmos de otimização linear e não linear. O uso dessa ferramenta evita que nos preocupemos com cálculos complexos, permitindo que concentremos nossos esforços apenas na modelagem do problema e na análise de seus resultados. Apesar de haver vários outros softwares de otimização, o Excel é uma boa escolha, já que as planilhas eletrônicas estão presentes em praticamente todas as empresas modernas. Além disso, é um software de fácil operação.

Verifica-se a importância que o autor dá à modelagem do problema, a qual é fundamental para a resolução dos cálculos; assim evita-se o erro em cálculos complexos, deixando o software com essa função, e pode-se focar na análise dos resultados obtidos como resposta. O autor menciona, ainda, que existem outros softwares que podem exercer a mesma função para a resolução dos modelos matemáticos.

2.6 Scilab

O Scilab é um software desenvolvido com funções matemáticas com aplicação no campo da engenharia e áreas científicas e fornece um ambiente de computação numérica de grande capacidade, além de sua interface ser de fácil entendimento para o usuário. E destaca-se por ser um software livre (gratuito), ou seja, sem a necessidade de pagar uma licença para sua instalação e utilização. Assim sendo, ele é aplicado em áreas acadêmicas, tanto como em áreas industriais. Pode-se utilizar ele em cálculos e simulação, aplicados a engenharia e ciências habituais e para funções gráficas bidimensionais e tridimensionais para visualizar e exportar dados. Também pode ser utilizado para resolução de problemas de otimização, em que são gerados algoritmos (no qual será abordado neste trabalho), e para estatística, em que se aplica a análise de dados e modelagem. Pode também ser aplicado para controle de projeto e análise de sistemas, processamento de sinais e desenvolvimento de aplicações. (S.A.S, 2015)

2.7 Problemas de cortes

Atualmente as indústrias cada vez mais buscam melhorias na manufatura dos seus produtos, de modo a minimizar as perdas.

Em processos que envolvem corte de matéria prima, no momento em que são compradas chapas e barras comerciais, e subdivididas em peças menores, a correta combinação dos cortes dessas peças menores é fundamental para reduzir as sobras de material. A partir desse ponto, surge um problema de otimização de corte para a redução de perdas. Os problemas de corte podem ser aplicados a várias indústrias como, por exemplo, corte de bobina de papel ou de aço ou cortes de barras metálicas (ARENALES *et al.*, 2007).

Nesse sentido, pode-se destacar alguns estudos que vêm sendo feitos nessa área, como o artigo sobre a otimização do problema de corte unidimensional (um estudo de caso na indústria papelreira descrito por Oliveira, Salvador e Silva em 2013), o qual identificou um ponto de melhoria no processo de corte de bobinas de papel jumbo para a minimização de perdas, utilizando um modelo de programação linear, com o auxílio da ferramenta LINGO para sua solução.

Outro artigo que pode ser citado é o de Wawrzynczak, Ulbricht e Teixeira (2016), os quais trabalharam em um modelo matemático para cortes de barras de aço no processo de fabricação de triângulos foi desenvolvido, esse trabalho foi elaborado baseado em dados de uma empresa fabricante de instrumentos musicais em Santa Catarina. No processo de corte das barras, notou-se uma significativa redução no desperdício de matéria prima. A partir da combinação de quatro tamanhos de corte foram gerados 160 padrões de cortes diferentes. Esse modelo matemático foi resolvido com o auxílio da ferramenta Solver do Microsoft Excel.

Outro trabalho foi publicado por Silveira Júnior *et al* (2004), os quais enfatizaram um problema de cortes de bobinas de aço em uma indústria metalmeccânica. Como referência para a elaboração do problema, foi considerado o corte de peças para produzir 100 fogões na empresa, com determinadas dimensões. Com base nisso, foram realizados experimentos computacionais, os quais utilizaram uma programação linear para a resolução do problema, com o auxílio da ferramenta LINDO, quando foi construída uma tabela para auxílio na tomada de decisão no corte e seleção de bobinas e estoque para minimização de perdas no processo.

2.8 Tipo de pesquisa

O foco desse trabalho está direcionado a um estudo de caso, ou seja, a situação problema do departamento de engenharia pertencente a uma empresa do Estado de Santa Catarina, a qual tem em seu portfólio o projeto e a fabricação de geradores síncronos verticais aplicados às pequenas centrais hidroelétricas.

Nas palavras de Severino (2007, p.121) o estudo de caso é uma “pesquisa que se concentra no estudo de um caso particular, considerado representativo de um conjunto de casos análogos, por ele significativamente representativo”. O autor destaca que este tipo de pesquisa é uma situação problema particular dentro de um contexto, em que pode ser aplicada a mesma solução, em outras situações similares, podendo assim contribuir para a referência e propostas de soluções factíveis.

Severino (2007, p.121), complementa que “o caso escolhido para a pesquisa deve ser significativo e bem representativo, de modo a ser apto a fundamentar uma generalização para situações análogas, autorizando inferências”. O autor também destaca que o estudo de caso pode ser aplicados em situações problemas similares, e também propondo a mesma solução do estudo de caso referenciado. Logo, podem-se incluir referências e melhorias à solução referenciada.

3 METODOLOGIA DE RESOLUÇÃO

A seguir será abordada a metodologia para a resolução de um problema de programação linear contínua, em contraste com um problema de programação linear inteira.

3.1 Resolução de um Problema de Programação Linear Contínua

Neste tópico é citado um exemplo buscando auxiliar no entendimento da complexidade da resolução de Problemas de Programação Linear (PPL).

Para exemplificar, foi considerado um caso fictício de uma empresa que visa maximizar os lucros, com base em sua capacidade de horas-máquina e mão de obra disponível no momento, sendo os dados apresentados na Tabela 1.

Tabela 1 – Dados gerais do problema

Descrição	Produto 1 (x_1)	Produto 2 (x_2)	Disponibilidade (h)
Lucro (R\$)	20	30	-
Mão de obra (h)	3	2	60
Hora máquina (h)	2	4	80

Fonte: Elaborado pelo autor (2016)

As variáveis de decisão nesse caso fictício para maximizar lucros da empresa representam as quantidades a serem fabricadas de cada produto, sendo representadas por x_1 e x_2 onde:

- x_1 : é a quantidade a ser produzida do produto 1;
- x_2 : é a quantidade a ser produzida do produto 2.

A expressão matemática que representa o lucro, dadas as quantidades de x_1 e x_2 a serem produzidas, é chamada de Função Objetivo, a qual é mostrada na equação (1).

$$\text{Max } L = 20x_1 + 30x_2 \quad (1)$$

Considerando o objetivo de obter o máximo lucro, para satisfazer a função objetivo mostrada na expressão (1), busca-se obter o maior valor possível para a soma $20x_1 + 30x_2$, desde que não seja violada nenhuma das restrições técnicas do modelo.

As restrições técnicas consideram os recursos a serem utilizados no processo produtivo. Como na maior parte dos casos esses recursos são limitados, busca-se a combinação entre as quantidades x_1 e x_2 , dentro das limitações do modelo que melhor satisfaçam a essas restrições.

No exemplo aqui considerado, as restrições técnicas são: a quantidade de mão de obra (M.O.) e quantidade de horas máquina (H.M.) existentes, as quais são representadas respectivamente pelas expressões (2), (3) e (4).

$$M.O.) \quad 3x_1 + 2x_2 \leq 60 \quad (2)$$

$$H.M.) \quad 2x_1 + 4x_2 \leq 80 \quad (3)$$

$$x_1, x_2 \geq 0 \quad (4)$$

A expressão (2) indica a quantidade utilizada de mão de obra (em horas), na qual cada unidade produzida do produto 1 utiliza 3h de trabalho e cada unidade do produto 2 um montante de 2h, sendo que o total produzido não pode ultrapassar as 60 horas de mão de obra disponíveis.

A expressão (3) indica a quantidade utilizada de horas máquina na qual cada unidade produzida do produto 1 (x_1) utiliza 2h, e cada unidade do produto 2 (x_2) utiliza 4h de funcionamento de máquina, sendo que o total produzido não pode extrapolar as 80 horas de funcionamento.

A expressão (4) mostra o domínio das variáveis de decisão que, por serem quantidades de produtos, devem ser maiores ou iguais a zero.

O modelo completo de programação neste caso é mostrado no Quadro 1.

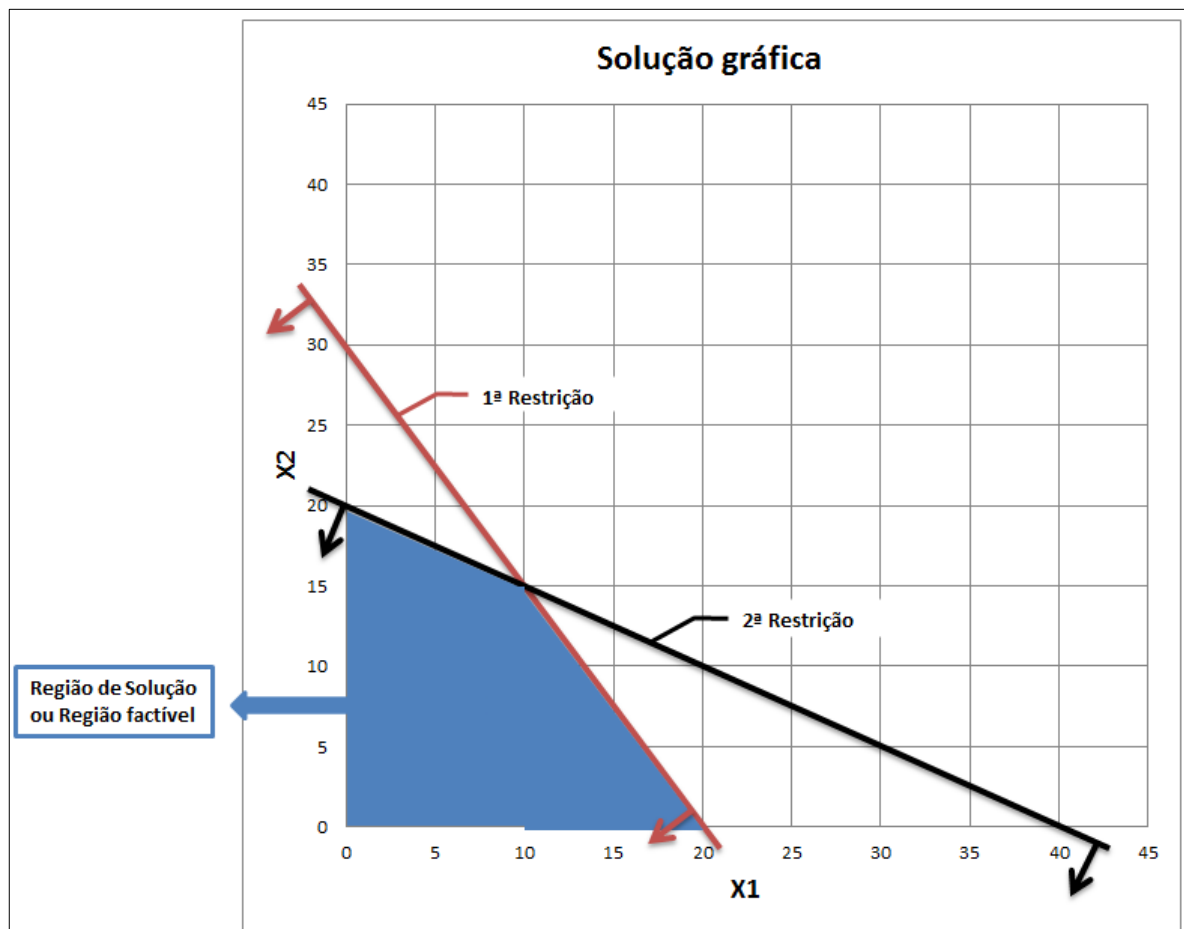
Quadro 1 – Dados gerais do problema

$$\begin{aligned} \text{Max } L &= 20x_1 + 30x_2 \\ \text{Sujeito a:} \\ \text{M.O.) } 3x_1 + 2x_2 &\leq 60 \\ \text{H.M.) } 2x_1 + 4x_2 &\leq 80 \\ x_1, x_2 &\geq 0 \end{aligned}$$

Fonte: Elaborado pelo autor (2016).

Desenhando as expressões graficamente mostradas em (2) e (3) em um mesmo sistema de eixos cartesianos, tem-se a Figura 1.

Figura 3 - Região de Soluções ou Região Factível



Fonte: Elaborado pelo autor (2016)

A área destacada na Figura 1 corresponde à região de soluções, ou seja, é a região

onde qualquer ponto $P(x, y)$ satisfaz as duas desigualdades mostradas nas expressões (2) e (3), as quais são as restrições técnicas do problema.

Resta, então, achar o melhor ponto ótimo $P^*(x, y)$, que é o ponto $P(x, y)$, contido na região de soluções, que resulta no melhor valor para a função objetivo mostrada na expressão (1).

Sendo assim, basta tomar para testes os pontos localizados nos extremos da figura formada pela região de soluções, ou seja, os vértices da figura.

Vértices formados pela intersecção das retas correspondentes às restrições, nesse caso mostrado nas expressões (2) e (3), podem ser facilmente encontrados pela resolução do sistema de equações (considera-se as desigualdades como igualdades, pois interessam os pontos pertencentes às retas) mostrado em (5).

$$\begin{cases} 3x_1 + 2x_2 = 60 \\ 2x_1 + 4x_2 = 80 \end{cases} \quad (5)$$

A Tabela 2 mostra os resultados obtidos quando os pontos $P(x, y)$, que compõem os vértices da Figura 3, são substituídos na função objetivo (F.O.), mostrada na expressão (1).

Tabela 2 – Valor da função objetivo para os pontos $P(x, y)$ correspondentes ao vértice

Ponto $P(x, y)$	Lucro (Valor da Função Objetivo)
$P_1(0, 20)$	600
$P_2(10, 15)$	650
$P_3(20, 0)$	400

Fonte: Elaborado pelo autor (2016)

Nesse caso, a solução que gera o maior valor da função objetivo é $P(10, 15)$, a qual corresponde à produção de 10 unidades do produto 1 e 15 unidades do produto 2, resultando num lucro de R\$ 650,00.

Problemas contendo mais de duas variáveis podem ser resolvidos por um método iterativo, o qual é baseado em inversões sucessivas de matrizes. Tal procedimento é conhecido como Método Simplex. Maiores detalhes desse método podem ser vistos em Silva

et al. (1998), Arenales *et al.* (2007) e Goldberg e Luna (2005).

3.2 Exemplo de Problema de Programação Inteira

As expressões (6) a (10) mostram um modelo de programação linear inteira, onde as variáveis de decisão x_1 e x_2 pertencem ao conjunto dos números inteiros positivos ($x_1, x_2 \in Z^+$).

$$\text{Max } L = 5x_1 + 8x_2 \quad (6)$$

$$\text{Sujeito a:} \quad (7)$$

$$x_1 + x_2 \leq 6 \quad (8)$$

$$5x_1 + 9x_2 \leq 45 \quad (9)$$

$$x_1, x_2 \in Z^+ \quad (10)$$

A seguir é mostrado o processo de resolução gráfica para esse tipo de problema, onde em um primeiro passo, é construído o gráfico correspondente às restrições (8) e (9).

Tomando-se a restrição (8), substitui-se a desigualdade (\leq), por uma igualdade, para plotar a reta correspondente à função:

$$x_1 + x_2 = 6$$

Atribuem-se valores convenientes de forma a construir o gráfico:

$$\text{Para } x_1 = 0 \rightarrow x_2 = 6$$

$$\text{Para } x_2 = 0 \rightarrow x_1 = 6$$

O mesmo procedimento é feito para a segunda restrição, mostrada na expressão (9).

$$5x_1 + 9x_2 = 45$$

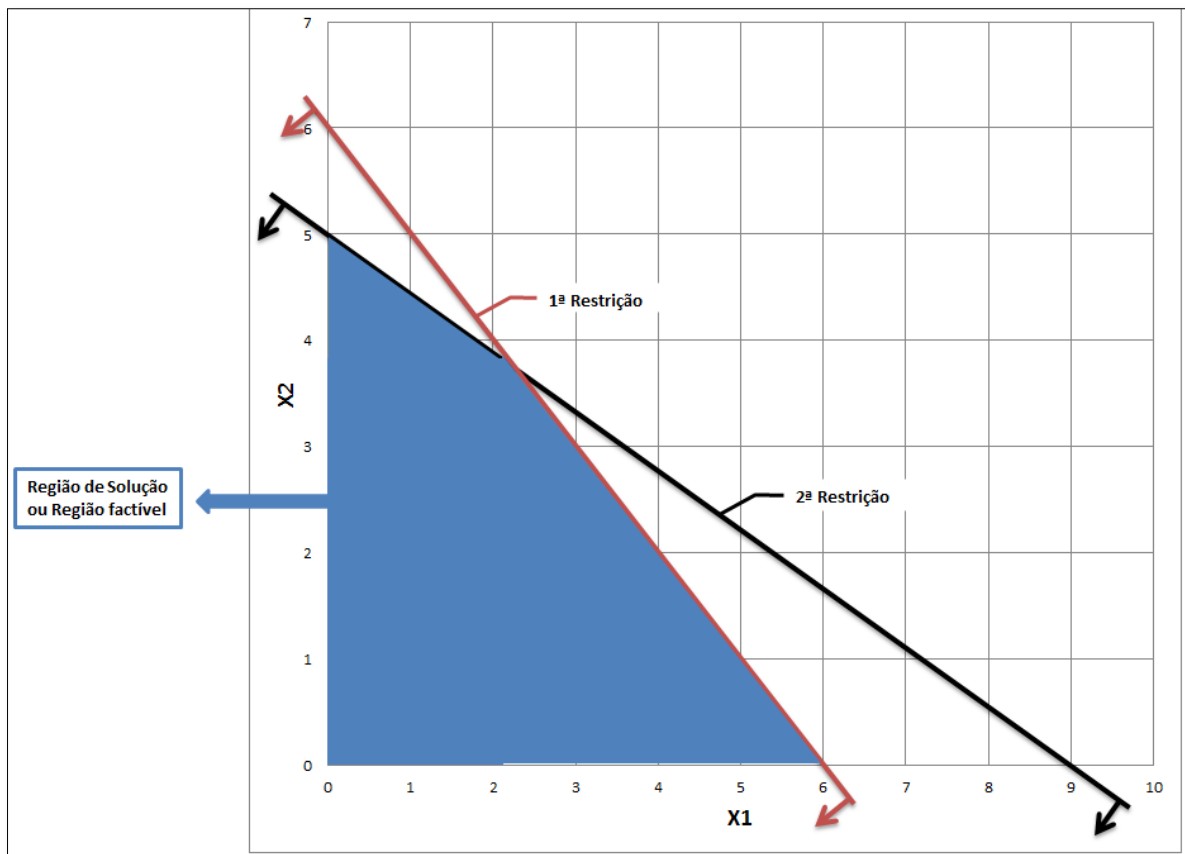
Atribuindo, do mesmo modo, valores convenientes têm-se:

$$\text{Para } x_1 = 0 \rightarrow x_2 = 5$$

Para $x_2 = 0 \rightarrow x_1 = 9$

A partir desses cálculos é possível traçar as duas retas em um mesmo plano cartesiano. Considerando então as desigualdades (\leq), tem-se como gráfico correspondente a região pertencente ao primeiro quadrante (já que $x_1, x_2 \in Z^+$, na expressão (10)), conforme mostrado na Figura 4.

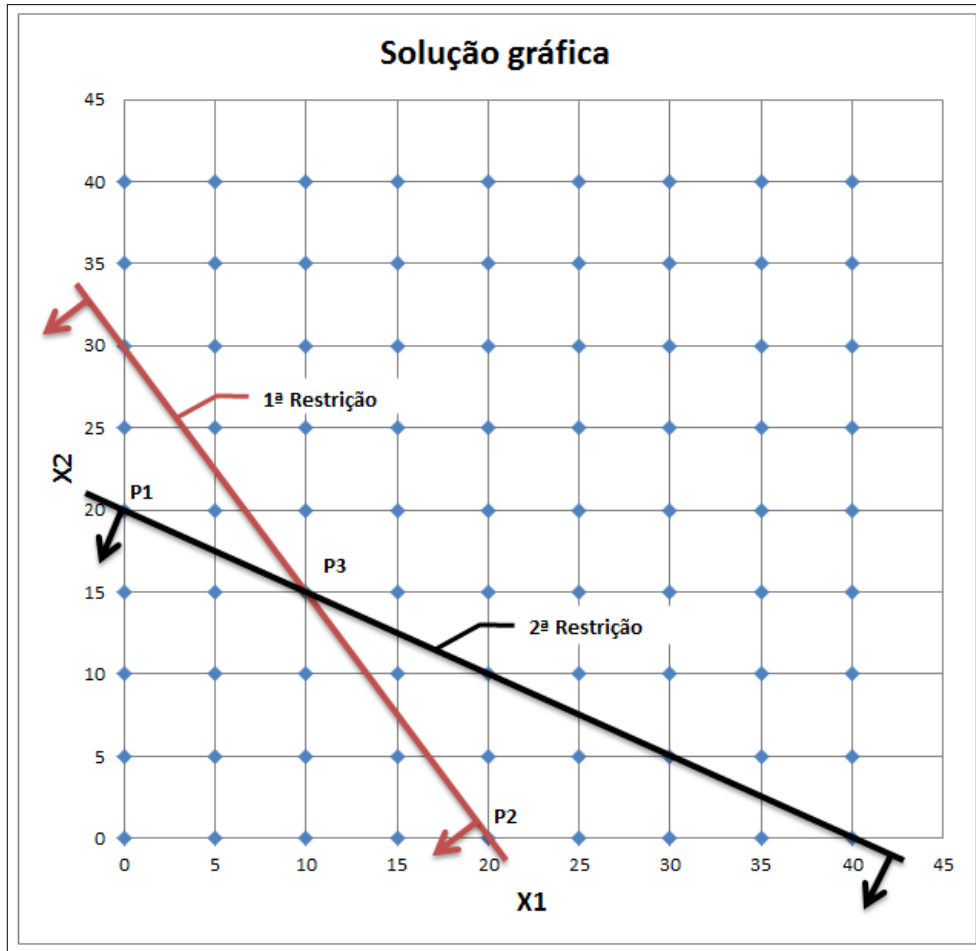
Figura 4- Região de Soluções ou Região Factível



Fonte: Elaborado pelo autor (2016)

O diferencial da programação inteira em relação à contínua é que, como solução, não podem ser escolhidos quaisquer valores dentro da região de soluções, mas sim apenas valores inteiros, que são representados por pontos na Figura 5.

Figura 5- Região de Soluções Inteiras



Fonte: Elaborado pelo autor (2016)

A solução ótima do problema contínuo (sem considerar variáveis inteiras) é um dos pontos $P_1(0, 6)$, $P_2(6, 0)$ e $P_3(x, y)$, o qual é encontrado pela resolução do sistema de equações mostrado em (11).

$$\begin{cases} x_1 + x_2 = 6 \\ 5x_1 + 9x_2 = 45 \end{cases} \quad (11)$$

Resolvendo o sistema de equações mostrado (11), obtêm-se $x_1 = 2,25$ e $x_2 = 3,75$. A Tabela 3 mostra os resultados para a função objetivo (expressão (6)), para os pontos P_1 , P_2 e P_3 .

Tabela 3 – Valor da função objetivo para os pontos $P(x, y)$ correspondentes ao vértice

Ponto $P(x, y)$	Lucro (Valor da Função Objetivo) $5x_1 + 8x_2$
$P_1(0, 6)$	$5 \cdot 6 + 8 \cdot 0 = 30$
$P_2(6, 0)$	$5 \cdot 0 + 8 \cdot 6 = 48$
$P_3(2,25, 3,75)$	$5 \cdot 2,25 + 8 \cdot 3,75 = 41,25$

Fonte: Elaborado pelo autor (2016)

Como o problema é de maximização, a solução ótima é dada pelo maior valor possível da função objetivo, ou seja, $x_1 = 2,25$ e $x_2 = 3,75$ totalizando um lucro de \$ 41,25.

A solução obtida nesse caso representa a solução do problema contínuo, mas não do problema de programação inteira, pois as variáveis x_1 e x_2 não contêm valores inteiros. Nesse caso, poderia ser considerado o fato de arredondar os valores de x_1 e x_2 para o inteiro mais próximo, ou para o inteiro imediatamente inferior, o que é um procedimento errôneo.

Procedendo com arredondamentos para as variáveis de decisão têm-se:

$$\text{Para } x_1 = 2,25 \rightarrow x_1 \cong 2$$

$$\text{Para } x_2 = 3,75 \rightarrow x_2 \cong 4$$

Verifica-se, nesse caso, que essa é uma solução infactível, pois não pertence à região de soluções, conforme mostrado no Quadro 2.

Quadro 2 – Factibilidade da solução arredondada

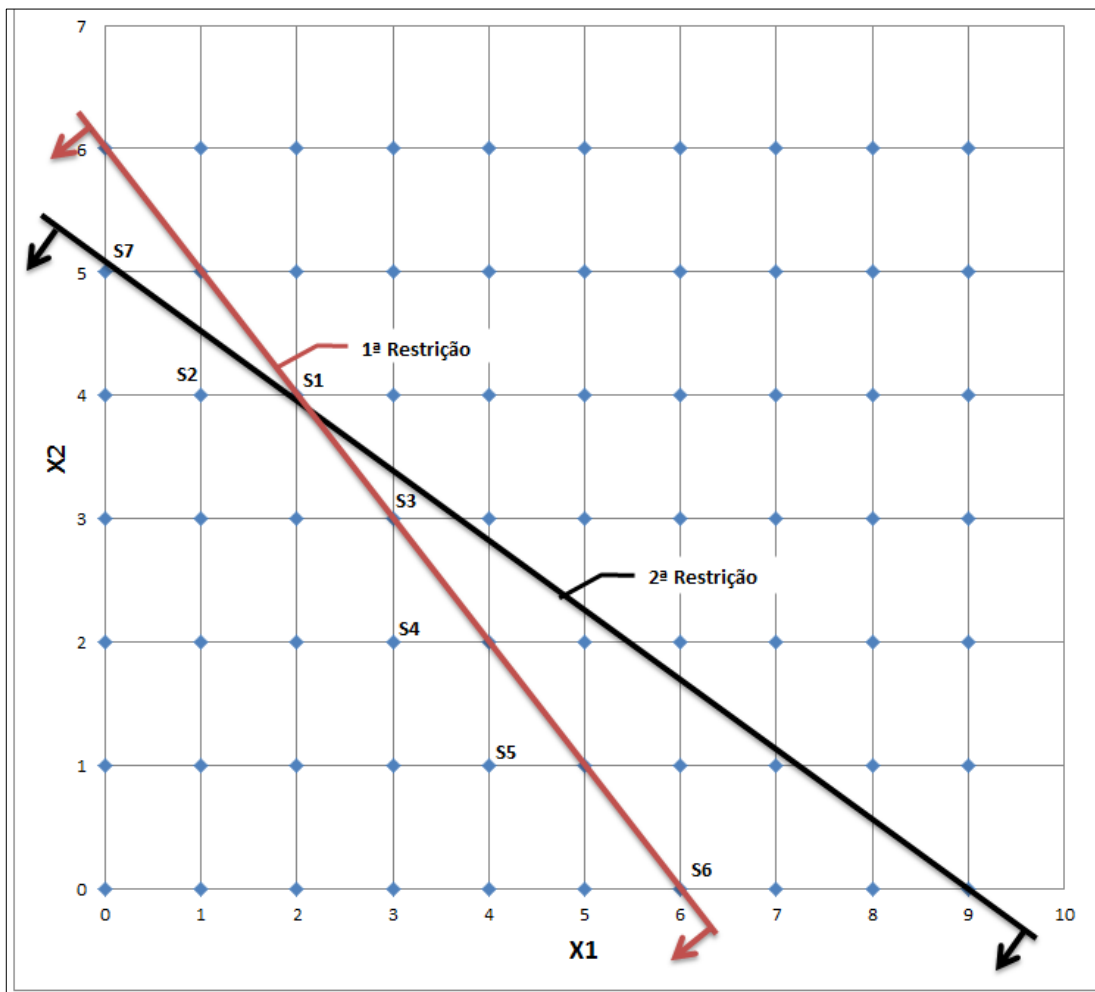
Teste			Factibilidade		É uma solução?	
Ponto	x_1	x_2	$5x_1 + 8x_2$	$x_1 + x_2 \leq 6$		$5x_1 + 9x_2 \leq 45$
*	2	4	$5 \cdot 2 + 8 \cdot 4 = 42$	Sim	Não	Não

Nota-se que os arredondamentos não são indicados para resolver problemas de programação linear inteira (PLI).

Sendo assim, é necessário pesquisar pontos próximos à fronteira da região de soluções. A Figura 6 mostra os pontos S_1 a S_7 , pertencentes a essa fronteira, os quais são possíveis candidatos à solução ótima do PLI.

Testando soluções inteiras factíveis têm-se:

Figura 6 – Pontos Referentes a Soluções Inteiras



Fonte: Elaborado pelo autor (2016)

A Tabela 4 mostra os testes realizados para cada um dos pontos factíveis (S_1 a S_7).

Tabela 4 – Testes para cada ponto factível pertencente à Fronteira de Soluções

Ponto	Teste		Factibilidade (restrições)		Lucro
	x_1	x_2	$x_1 + x_2 \leq 6$	$5x_1 + 9x_2 \leq 45$	(Valor da Função Objetivo) $5x_1 + 8x_2$
S1	4	2	Sim	Sim	36
S2	4	1	Sim	Sim	28
S3	3	3	Sim	Sim	39
S4	2	3	Sim	Sim	34
S5	1	4	Sim	Sim	37
S6	6	0	Sim	Sim	30
S7	0	5	Sim	Sim	40

Fonte: Elaborado pelo autor (2016)

Analisando a Tabela 4, nota-se que a solução inteira que maximiza o lucro é a solução S7, a qual não possui relacionamento com o arredondamento dos valores obtidos com a solução contínua do mesmo problema.

Sendo assim, para o PLI tem-se a solução ótima: $x_1 = 0$ e $x_2 = 5$, ocasionando um lucro $L = 40$.

4 MODELAGEM MATEMÁTICA E COMPUTACIONAL PROPOSTA PARA RESOLUÇÃO EM UM PROBLEMA DE CORTES NA INDÚSTRIA

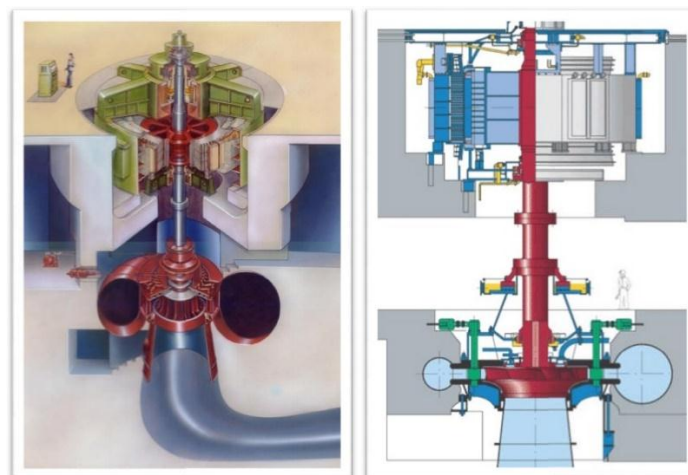
Neste capítulo é apresentada a descrição do problema prático na indústria de eletrodutos, bem como do modelo matemático e dos algoritmos de geração de padrões propostos nesse trabalho.

4.1 Descrição do Problema Aplicado

A aplicação do projeto no cenário industrial ocorreu em uma empresa em Jaraguá do Sul, em Santa Catarina, que projeta e fabrica geradores síncronos verticais para pequenas centrais hidrelétricas no estado de Santa Catarina. Ela tem no seu escopo de fornecimento o projeto e montagem de todos eletrodutos para o encaminhamento dos cabos de força e iluminação montados dentro de um fosso de concreto, onde está montado o gerador.

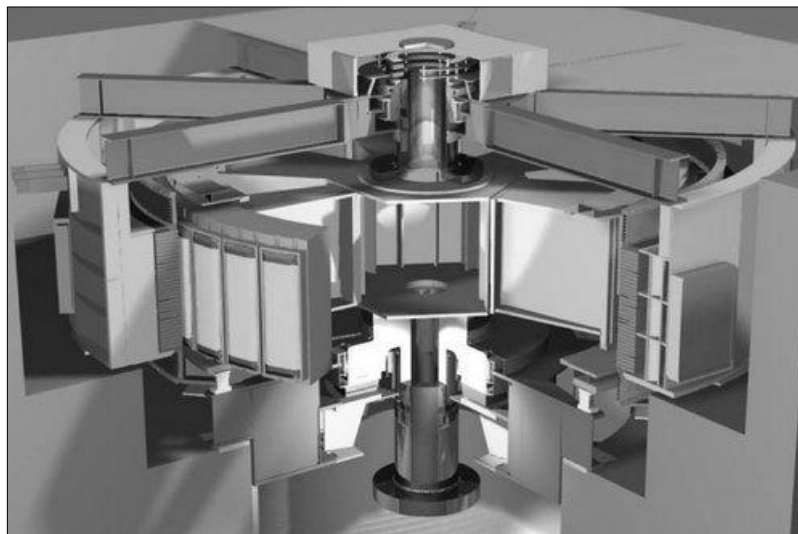
As figuras 7 e 8 ilustram essa aplicação de hidrogeradores. Maiores detalhes construtivos podem ser vistos em Simone (2000), na página 34 e características construtivas em Borges Neto e Carvalho (2013), na página 33.

Figura 7 - Esquema de um hidrogerador vertical



Fonte: <http://www.cchpe.net/ProductInfo.aspx?get=29>

Figura 8 - Hidrogerador típico



Fonte: <http://www.taiheelectric.com/imageRepository/c6241518-d0f1-4451-aafe-a636a59f01e3.jpg>

Nos hidrogeradores mostrados nas figuras 7 e 8, os cabos de força e iluminação são protegidos por um eletroduto rígido, a fim de evitar danos, protegendo as conexões elétricas de poeira e umidade, bem como o operador de uma possível conexão elétrica exposta, além de organizar o encaminhamento dos cabos até o painel de comando e proteção do circuito, conforme ilustrado na Figura 9. Maiores detalhes construtivos podem ser vistos em Walenia (2010), p. 12.

Figura 9 - Exemplo de encaminhamento de cabos até painéis



Fonte: http://www.blue-sol.com/energia-solar/wp-content/uploads/2012/09/IMG_0884.jpg

Para a orientação da montagem desses componentes dentro do fosso de concreto, usado para o encaminhamento dos eletrodutos, é emitido um documento de engenharia para

definir o posicionamento, bem como os materiais utilizados e suas quantidades. Também é indicado o comprimento de corte de cada segmento de eletroduto a ser utilizado no projeto.

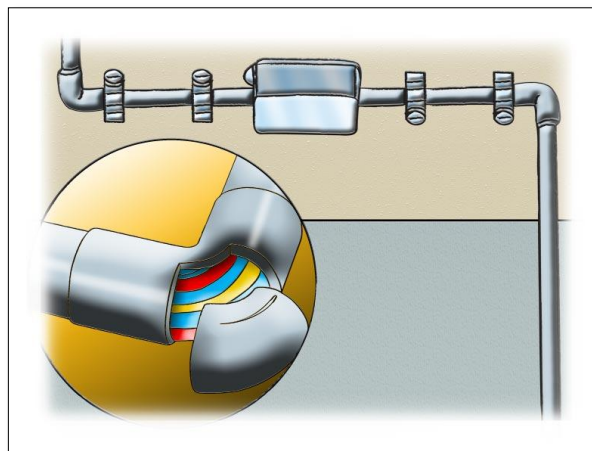
Para a compra desses materiais é gerada uma necessidade de pedido, na qual é informada a quantidade total de cada material necessário para a execução do projeto. Porém, para eletrodutos, existe uma particularidade de compra, pois comercialmente está disponível na empresa a compra de um fornecedor qualificado de eletrodutos, que tem como produto final o eletroduto de comprimento de 3 metros.

Atualmente, a engenharia especifica a soma total dos cortes de eletrodutos para a compra, considerando mais 20% de coeficiente de segurança. No entanto, ela não especifica o sequenciamento de corte dos eletrodutos para o operador em obra e também não considera a largura da serra em seu projeto, o que pode gerar falta ou desperdício de material na obra.

4.1.1 Características do componente

O eletroduto utilizado pela empresa e utilizado como padrão nesse trabalho possui 3 metros como comprimento padrão comercial, podendo ser das bitolas de $\frac{1}{2}$ " , $\frac{3}{4}$ " e 1". Ele é utilizado para encaminhamento de cabos, de acordo com o projeto. Nesse projeto constam as medidas necessárias em que esse eletroduto deverá ser cortado, em comprimentos de acordo com a necessidade. Basicamente, o comprimento de 3 metros será seccionado em comprimentos menores para montagem do encaminhamento dos eletrodutos, conforme ilustrado na Figura 10.

Figura 10 - Instalação típica.



Fonte: <http://www.cec.com.br/dicas-seguranca-recomendacoes-gerais-para-instalacoes-eletricas?id=149>

O corte do material é executado pelo operador no momento da montagem em campo, de acordo com as quantidades e comprimentos especificados. Essa quantidade e comprimento varia de acordo com o projeto, tornando específica a aplicação. Nesse caso serão considerados 13 comprimentos, conforme mostrado na Tabela 5.

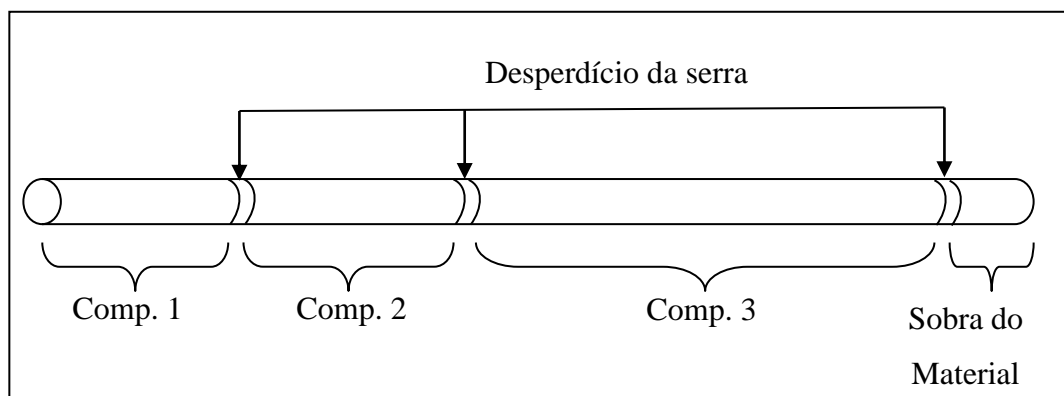
Descrição	Medida (mm)
Comprimento 1	255
Comprimento 2	1960
Comprimento 3	1130
Comprimento 4	1465
Comprimento 5	100
Comprimento 6	405
Comprimento 7	1000
Comprimento 8	1500
Comprimento 9	1455
Comprimento 10	735
Comprimento 11	705
Comprimento 12	515
Comprimento 13	510

Fonte: Elaborado pelo autor (2016)

Com o objetivo de suprir a necessidade solicitada, notou-se que o processo de especificação e corte pode gerar perdas ou falta de matéria prima em campo, devido à falta de informação da ordem de corte. O procedimento atual da engenharia não considera também a

largura da serra no corte do material. A Figura 11 destaca o processo de corte representando três comprimentos como ilustrativos e a largura da serra.

Figura 11 – Tipo de cortes no processo



Fonte: Elaborado pelo autor (2016)

4.1.2 Padrões de cortes

De acordo com a Tabela 5, nota-se que existem 13 medidas diferentes existentes no projeto específico, no qual o eletroduto padrão é cortado com uma serra, podendo gerar sobra de material. A largura da serra é de aproximadamente 2,5 mm, portanto, será considerada essa perda para cada padrão de corte a ser serrado no eletroduto de 3 metros.

Analisando esse caso, é possível encontrar vários padrões (combinações de diversas medidas) de corte possíveis para a mesma barra, gerando diferentes sobras de material para cada uma. O intuito é gerar a maior quantidade possível de padrões de corte e identificar quais padrões serão utilizados de modo a atender às demandas e buscar ao mesmo tempo o menor desperdício possível de matéria prima. Na Tabela 6, constam alguns padrões de cortes possíveis, elaborados de forma manual com o auxílio de uma planilha eletrônica, com os comprimentos e a sobra do material.

Tabela 6 – Exemplos de Padrões de corte (P_i)

Comprimento (mm)	Padrões de Corte (P_i)								
	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	...
255	11	10	8	0	2	5	0	1	...
1960	0	0	0	1	1	0	0	1	...
1130	0	0	0	0	0	0	1	0	...
1465	0	0	0	0	0	0	1	0	...
100	1	4	2	1	0	2	3	3	...
405	0	0	0	1	0	1	0	1	...
1000	0	0	0	0	0	0	0	0	...
1500	0	0	0	0	0	0	0	0	...
1455	0	0	0	0	0	0	0	0	...
735	0	0	0	0	0	0	0	0	...
705	0	0	1	0	0	0	0	0	...
515	0	0	0	1	1	1	0	0	...
510	0	0	0	0	0	1	0	0	...
Comprimento total cortado (mm)	2905	2950	2945	2980	2985	2905	2895	2920	...
Desperdício da largura da serra (mm)	30	35	27,5	10	10	25	12,5	15	...
Sobra de material (mm)	66	15,00	27,5	10	5	70	92,5	65	...

Fonte: Elaborado pelo autor (2016)

4.2 Algoritmos Propostos para Geração de Padrões de Corte

Foram elaborados 2 diferentes algoritmos para geração de padrões de corte. Para o desenvolvimento dos dois algoritmos, foram seguidas três etapas: a primeira foi a elaboração de um projeto de algoritmo básico, inicialmente em manuscrito. A segunda etapa consistiu na codificação do algoritmo por meio de uma linguagem de programação, momento no qual optou-se pelo software Scilab, por se tratar de um aplicativo de licença livre e amplamente utilizado. A terceira etapa foi a execução dos algoritmos e validação do funcionamento.

Para os dois algoritmos foram considerados os pilares básicos para a estruturação de um algoritmo, o qual tem como partes fundamentais o início, o tempo de processo, o estado final do processo e o resultado bem definido (BORATTI; OLIVEIRA, 2007).

Nos tópicos a seguir são apresentados detalhamentos sobre o funcionamento dos dois algoritmos propostos.

4.2.1 Algoritmo de busca ordenada: baseado em busca exaustiva por todos os padrões de corte

O Algoritmo de busca ordenada, foi baseado na premissa de buscar todos os possíveis padrões de corte para que a partir destes, o modelo de otimização buscasse pela melhor combinação possível de padrões a serem utilizados.

Para a entrada de dados foi informado o comprimento da barra padrão, bem como de cada medida a ser cortada com suas respectivas quantidades.

Na próxima etapa foi elaborado o código para gerar todas as combinações possíveis conforme os passos descritos a seguir:

Passo 1: Entrada: comprimento da barra padrão, comprimento de cada medida a ser cortada, quantidades.

Passo 2: Verificar o maior número possível da peça $p(i)$ a ser cortada em cada barra: número possível da peça $p(i) = (\text{Comprimento da barra}) / \text{Medida da peça } p(i)$.

Passo 3: Arredondar para baixo (comando *floor*): $\text{floor } p(i) = \text{número possível da peça } p(i)$.

Passo 4: Gerar os padrões com laços de repetição com a maior quantidade possível de cada peça a compor um padrão, reduzindo sucessivamente as quantidades e aumentando as quantidades das peças de medida menor.

Passo 5: Armazenar todos os padrões encontrados na matriz “matriz_padrao” sendo que cada coluna representa um padrão encontrado.

Passo 6: calcular o desperdício de cada coluna padrão gerada, considerando todos os valores possíveis encontrados, sejam positivos ou negativos.

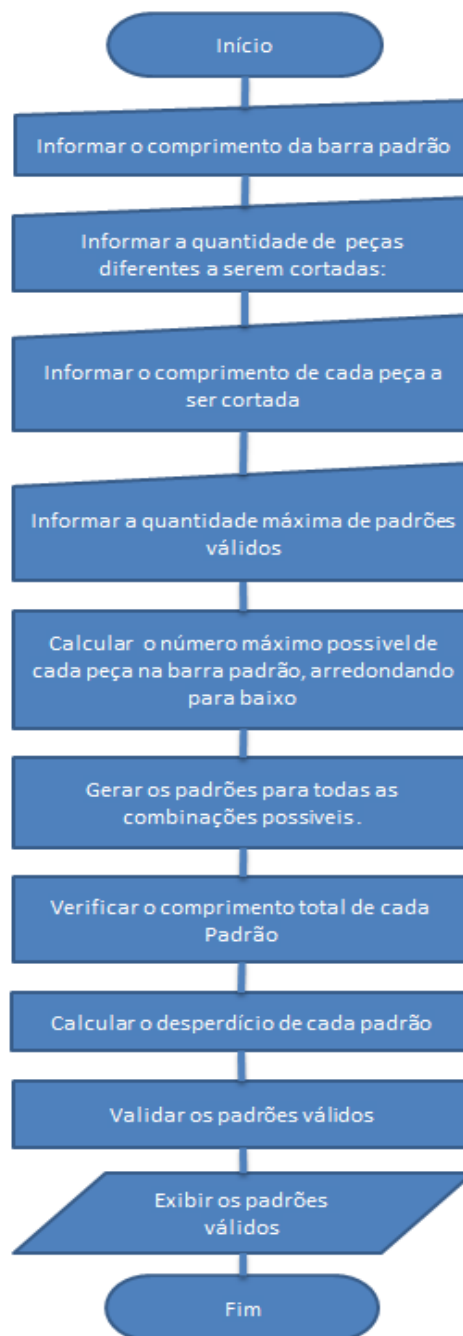
Passo 7: executar um teste para cada coluna da “matriz_padrao” de forma a validar (ou não) o padrão, eliminando padrões que tenham desperdícios negativos que são valores desconsiderados no problema e armazenar os padrões validados na “matriz_padrao_valido”.

Passo 8: executar a saída de dados, no qual todos os valores encontrados são indicados na tela do software Scilab e em um arquivo tipo “.txt”, que são todas as combinações possíveis, as dimensões dos padrões gerados, o desperdício do padrão e, por

fim, os padrões válidos.

Com o intuito de facilitar a visualização do fluxo do processo descrito acima, elaborou-se o fluxograma mostrado na Figura 12. O código computacional construído para esse algoritmo (programado em Scilab) pode ser encontrado no apêndice A deste trabalho.

Figura 12 – Fluxograma Algoritmo de busca ordenada



Fonte: Elaborado pelo autor (2016)

O Algoritmo de busca ordenada foi testado inicialmente para a produção de peças de 3 tamanhos distintos obtendo êxito na geração de padrões dentro de um tempo pequeno (menor que um minuto).

Aplicando o algoritmo no problema real, envolvendo peças de 13 tamanhos distintos, não foi obtida uma resposta do algoritmo mesmo após 3 horas de execução. Isso se deve ao grande número de combinações geradas, pois trata-se de um problema de explosão combinatorial (FLORIAN *et al*, 1980).

Buscando obter resultados mesmo parciais para o algoritmo de busca ordenada para problemas maiores (como o envolvendo 13 medidas), foi incluída na etapa de entrada de dados a quantidade máxima para padrões válidos encontrados. Nota-se, porém, que o algoritmo não é capaz de percorrer todo o espaço de busca, pois fica concentrado nas primeiras peças, executando as combinações, não atingindo as últimas peças mesmo em quinze horas de execução.

Sendo assim, foi proposto um segundo algoritmo capaz de percorrer todo o espaço, mesmo que aleatoriamente, conforme mostrado no tópico a seguir.

4.2.2 Algoritmo de busca aleatória: baseado em busca aleatória de padrões de corte

O Algoritmo de busca aleatória possui uma estruturação diferenciada, pois teve a convenção de buscar por padrões de corte de forma aleatória.

Neste algoritmo é iniciada a execução com a entrada de dados semelhante ao primeiro algoritmo, porém com algumas melhorias, dentre elas, a inclusão do campo para largura da serra e também do número máximo de peças a serem cortadas em um padrão.

O código para gerar padrões válidos segue os passos descritos a seguir:

Passo 1: Entrada: comprimento da barra padrão, comprimento de cada medida a ser cortada, quantidades, largura da serra, máximo de peças distintas a serem cortadas em um padrão.

Passo 2: Verificar o maior número possível da peça $p(i)$ a ser cortada em cada barra: número possível da peça $p(i) = (\text{Comprimento da barra}) / \text{Medida da peça } p(i)$.

Passo 3: Arredondar para baixo (comando *floor*): $\text{floor } p(i)$ = número possível da peça $p(i)$. Guardar esses resultados no vetor “QtdeMáx” que representa a quantidade máxima possível de ser cortada da peça do tipo “ i ” em uma barra padrão.

Passo 4: Gerar os padrões:

Passo 4.1: Gerar um vetor binário do tipo: [1 0 0 1 0 ...] onde “1” significa fazer e “0” significa não fazer, a peça do tipo “ i ”.

Passo 4.2: Gerar um vetor aleatório no intervalo [0, 1] da mesma dimensão do vetor gerado no passo 4.1, por exemplo: [0,76 0,09 0,85 0,5 0,08 ...]

Passo 4.3: Multiplicar elemento a elemento o vetor do passo 4.1 pelo do passo 4.2, obtendo um vetor que significa produzir ou não um certo percentual da peça do tipo “ i ”.

Nesse exemplo tem-se como resultado: [0,76 0 0 0,5 0 ...]

Passo 4.4: Multiplicar o vetor obtido no passo 4.3 pelo vetor de quantidades máximas (QtdeMáx) obtido no passo 2. Assim é obtido o percentual da quantidade máxima a ser cortado em um padrão de corte. Nesse exemplo, seria cortado 76% da quantidade máxima da peça 1 e 50% da quantidade máxima da peça 4.

Passo 4.5: Arredondar cada elemento do vetor para o inteiro mais próximo, obtendo-se assim um padrão de corte.

Passo 5: Validação do padrão de corte obtido:

Passo 5.1: Multiplicar cada quantidade obtida pelo comprimento de cada peça, obtendo-se o comprimento total a ser cortado. Por exemplo:

Padrão gerado:	5	0	0	1	0
Medidas das peças :	50	60	100	200	210

Nesse caso: têm-se:

$$(5 \times 50) + (0 \times 60) + (0 \times 100) + (1 \times 200) + (0 \times 210) = 450$$

Passo 5.2: calcular o desperdício de cada coluna padrão gerada

$$(\text{Desperdício do padrão gerado}) = (\text{Comprimento da barra}) - (\text{Comprimento calculado no passo 5})$$

Nesse caso, se a barra padrão medisse 300 cm por exemplo, seria obtido:

$$(\text{Desperdício do padrão gerado}) = 300 - 450$$

$$(\text{Desperdício do padrão gerado}) = -150$$

Passo 5.3: Teste de validação do padrão de corte

Executar o teste:

Se $[(\text{Desperdício do padrão gerado}) \leq 0]$ E $[(\text{Desperdício do padrão gerado}) \leq (\text{número máximo de peças a serem cortadas em um padrão})]$

Então:

Aproveita o padrão, ou seja, esse padrão será válido e será guardado na coluna i da matriz: Padrão_Válido $(i \times j)$

Caso contrário:

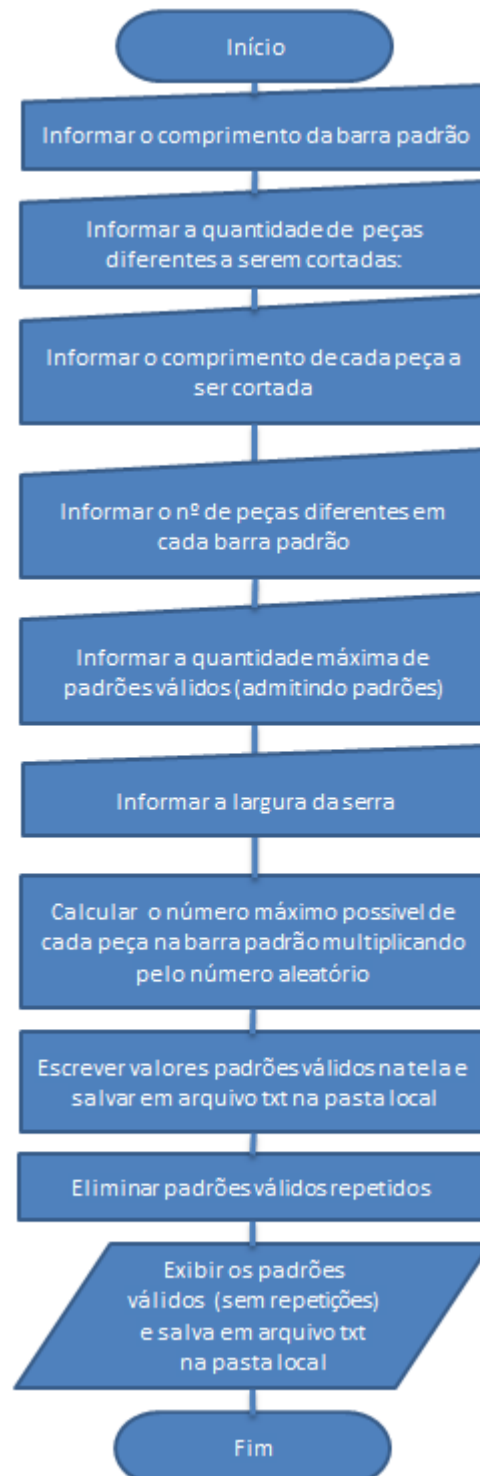
Volta ao passo 4.

Passo 6: Critério de parada: obtenção de uma quantidade pré-definida pelo usuário, de padrões válidos.

Passo 7: Saída de dados: Matriz Padrão_Válido $(i \times j)$, gravada em um arquivo em formato “.txt”.

Com o intuito de facilitar a visualização desse outro fluxo do processo descrito acima, elaborou-se o fluxograma mostrado na Figura 13. O código computacional construído deste segundo algoritmo (programado em Scilab) pode ser encontrado no apêndice B deste trabalho.

Figura 13 – Fluxograma Algoritmo de busca aleatória



Fonte: Elaborado pelo autor (2016)

4.3 Modelo Matemático

Na formulação do modelo foi considerada a existência de estoques pequenos, utilizadas em algumas empresas, de modo a facilitar o atendimento a pedidos extras.

A seguir é apresentado o objetivo do modelo, parâmetros, variáveis de decisão e modelo matemático.

Objetivo do modelo: minimizar o desperdício de matéria prima (sobras de eletrodutos).

Parâmetros:

n : número de padrões de corte pré-definido pelo usuário;

D_i : Desperdício de matéria prima quando for utilizado o padrão de corte i ($i = 1, \dots, n$);

E_{ki} : Quantidade a ser cortada do eletroduto k no padrão i

$Q_{\min(k)}$: Demanda ou quantidade mínima a ser produzida do eletroduto k .

$Q_{\max(k)}$: Demanda ou quantidade máxima a ser produzida do eletroduto k .

Variáveis de decisão:

P_i : Quantidade a ser produzida do padrão de corte i ($i = 1, \dots, n$). (n : número de padrões de corte pré-definido pelo usuário)

Modelo Matemático:

Minimizar $D_1P_1 + D_2P_2 + D_3P_3 + \dots + D_nP_n$

Sujeito às restrições:

Demanda_Min_Eletroduto_1) $E_{11}P_1 + E_{12}P_2 + E_{13}P_3 + \dots + E_{1n}P_n \geq Q_{\min(1)}$

Demanda_Máx_Eletroduto_1) $E_{11}P_1 + E_{12}P_2 + E_{13}P_3 + \dots + E_{1n}P_n \leq Q_{\max(1)}$

Demanda_Min_Eletroduto_2) $E_{21}P_1 + E_{22}P_2 + E_{23}P_3 + \dots + E_{2n}P_n \geq Q_{\min(2)}$

Demanda_Máx_Eletroduto_2) $E_{21}P_1 + E_{22}P_2 + E_{23}P_3 + \dots + E_{2n}P_n \leq Q_{\max(2)}$

$$\text{Demanda_Min_Eletroduto_3)} E_{31} P_1 + E_{32} P_2 + E_{33} P_3 + \dots + E_{2n} P_n \geq Q_{\min(3)}$$

$$\text{Demanda_Máx_Eletroduto_3)} E_{31} P_1 + E_{32} P_2 + E_{33} P_3 + \dots + E_{2n} P_n \leq Q_{\max(3)}$$

$$\vdots$$

$$\text{Demanda_Min_Eletroduto_k)} E_{k1} P_1 + E_{k2} P_2 + E_{k3} P_3 + \dots + E_{kn} P_n \geq Q_{\min(k)}$$

$$\text{Demanda_Máx_Eletroduto_k)} E_{k1} P_1 + E_{k2} P_2 + E_{k3} P_3 + \dots + E_{kn} P_n \leq Q_{\max(k)}$$

$$P_1, \dots, P_n \in Z^+$$

4.4 Planilha para Resolução do Modelo Matemático por meio da Ferramenta Solver

Após gerados os padrões de cortes por meio do Algoritmo de Busca Aleatória, os resultados devem ser transferidos para uma planilha de cálculo, conforme mostrado na Figura 14, a qual foi desenvolvida a partir das pesquisas realizadas nesse trabalho.

Figura 14 – Planilha de padrões Pi

Objeto		P1	P2	P3	P4	P5	P6	P7	P8
Comprimento Eletroduto 1	255	0	0	0	0	0	0	0	0
Comprimento Eletroduto 2	1960	0	1	0	1	0	0	1	0
Comprimento Eletroduto 3	1130	0	0	0	0	0	0	0	0
Comprimento Eletroduto 4	1465	1	0	0	0	0	0	0	0
Comprimento Eletroduto 5	100	0	0	0	0	7	0	10	0
Comprimento Eletroduto 6	405	0	0	0	0	0	0	0	0
Comprimento Eletroduto 7	1000	0	1	0	0	0	0	0	0
Comprimento Eletroduto 8	1500	0	0	1	0	0	1	0	0
Comprimento Eletroduto 9	1455	0	0	0	0	0	1	0	1
Comprimento Eletroduto 10	735	1	0	2	0	2	0	0	2
Comprimento Eletroduto 11	705	1	0	0	0	1	0	0	0
Comprimento Eletroduto 12	515	0	0	0	0	0	0	0	0
Comprimento Eletroduto 13	510	0	0	0	2	0	0	0	0
	Soma	2905	2960	2970	2980	2875	2955	2960	2925
Largura da Serra (mm)	Desperdício cortes	95	40	30	20	125	45	40	75
2,5	Desperdício Serra	7,5	5	7,5	7,5	25	5	27,5	7,5
----->	Sobra de material	87,50	35,00	22,50	12,50	100,00	40,00	12,50	67,50
Comprimento eletroduto(mm)									
3000									
Função Objetivo:	Desperdício (mm):	<input type="text" value="975"/>							

Fonte: Elaborado pelo autor (2016)

Os dados (Parâmetros do modelo matemático): comprimento do eletroduto, largura da serra e comprimento de cada medida a ser cortada (eletroduto i), e a quantidade de cada eletroduto são informadas na planilha, conforme mostrado na Figura 15.

Figura 15 – Interface da planilha

Entrada de dados			
Comprimento eletroduto (mm)	3000		
Largura da Serra considerada (mm)	2,5		
Comprimento Eletroduto 1 (mm)	255	Quantidade eletroduto 1	6
Comprimento Eletroduto 2 (mm)	1960	Quantidade eletroduto 2	3
Comprimento Eletroduto 3 (mm)	1130	Quantidade eletroduto 3	1
Comprimento Eletroduto 4 (mm)	1465	Quantidade eletroduto 4	2
Comprimento Eletroduto 5 (mm)	100	Quantidade eletroduto 5	1
Comprimento Eletroduto 6 (mm)	405	Quantidade eletroduto 6	3
Comprimento Eletroduto 7 (mm)	1000	Quantidade eletroduto 7	18
Comprimento Eletroduto 8 (mm)	1500	Quantidade eletroduto 8	4
Comprimento Eletroduto 9 (mm)	1455	Quantidade eletroduto 9	1
Comprimento Eletroduto 10 (mm)	735	Quantidade eletroduto 10	2
Comprimento Eletroduto 11 (mm)	705	Quantidade eletroduto 11	4
Comprimento Eletroduto 12 (mm)	515	Quantidade eletroduto 12	1
Comprimento Eletroduto 13 (mm)	510	Quantidade eletroduto 13	1

Fonte: Elaborado pelo autor (2016)

As variáveis a serem calculadas pelo Solver são mostradas na Figura 16.

Figura 16 – Variáveis de decisão e restrições

Variáveis de decisão	P1	P2	P3	P4	P5	P6	P7	P8	
	0	0	0	3	0	0	0	0	
Restrições:		Qtde produzida	Demanda						
Qtde Mínima de eletroduto 1	6	6							
Qtde Máxima de eletroduto 1	6	6							
Qtde Mínima de Eletroduto 2	3	3							
Qtde Máxima de eletroduto 2	3	3							
Qtde Mínima de eletroduto 3	1	1							
Qtde Máxima de eletroduto 3	1	1							
Qtde Mínima de eletroduto 4	2	2							
Qtde Máxima de eletroduto 4	2	2							
Qtde Mínima de eletroduto 5	0	1							
Qtde Máxima de eletroduto 5	0	1							

Fonte: Elaborado pelo autor (2016)

5 RESOLUÇÃO DE UM PROBLEMA APLICADO

Este capítulo mostra uma aplicação da metodologia de auxílio à decisão proposta nesse trabalho em um caso prático na indústria de eletrodutos.

5.1 Resolução em uma Abordagem Real na Indústria

Para resolução do problema foi utilizada a entrada de dados mostrada na Tabela 5, apresentada no Capítulo 4, a qual se refere a 13 tipos de peças, descrevendo seus comprimentos e as respectivas quantidades. Para facilitar a leitura do trabalho, esses dados são novamente apresentados na Tabela 7.

Tabela 7- Comprimentos e quantidades de peças para implementação prática.

Descrição	Medida (mm)	Quantidade de peças
Comprimento 1	255	6
Comprimento 2	1960	3
Comprimento 3	1130	1
Comprimento 4	1465	2
Comprimento 5	100	1
Comprimento 6	405	3
Comprimento 7	1000	18
Comprimento 8	1500	4
Comprimento 9	1455	1
Comprimento 10	735	2
Comprimento 11	705	4
Comprimento 12	515	1
Comprimento 13	510	1

Fonte: Elaborado pelo autor (2016)

Como primeiro passo, com utilização do software Scilab, foi acessado o algoritmo 2. Esse acesso pode ser realizado clicando em:

MENU: Arquivo > Abrir > Algoritmo_ver_final.sce.

Deve-se clicar no ícone “Executar” para visualizar a execução do algoritmo no Scilab.

Figura 17 – Ilustração Parcial da Janela do Código Scilab

```

1 //Programa Geração de padrões de cortes unidimensionais
2 //Versão com geração aleatória
3
4 clear
5clc
6
7 //-----
8 //Dados:
9 //C=300; //Comprimento da barra (cm)
10 //peca=[50 60 100 200 210]; //Vetor Comprimento das peças
11 //p=3; //Número máximo de peças diferentes em cada padrão
12
13 ... printf('\n')
14 ... printf('PROGRAMA PARA GERAÇÃO DE PADRÕES PARA CORTES UNIDIMENSIONAIS')
15 ... printf('\n\n')
16
17 //INÍCIO: ENTRADA DE DADOS-----
18 C= input ( 'Comprimento da barra (mm): ');
19 Qtdepeças= input ( 'Qtde de tipos distintos de peças: ');
20 printf('\n\n')
21 for i=1:Qtdepeças
22 ... printf('Peça %g de %g:', i, Qtdepeças)
23 pecaSemSerra(i)= input ( 'Medida da peça (mm): ');
24 end
25 printf('\n\n')
26 p= input ( 'Número máximo de peças diferentes em cada padrão (<= à qtde de peças): ');
27 QtdeMaxPadraoValido= input ( 'Qtde máxima de Padrões Válidos (Admitindo repetições): ');
28 LargSerra= input ( 'Largura da Serra (mm): ');
29
30 //Incrementar a largura da serra em cada peça:
31 for i=1:Qtdepeças
32 peca(i)= pecaSemSerra(i)+LargSerra;
33 end
34 //FIM: ENTRADA DE DADOS-----
35
36 peca=peca';
37
38
39 //INÍCIO: Código para gerar uma matriz 0, 1 com no máximo k números 1 por coluna
40 jj=1;
41 //quantidades máximas de cada peça cortada:
42 [ncolunasPeca nPecas]=size(peca); //Lê o número de elementos do vetor Peça
43 for i =1:nPecas
44 Qmax_peca(i)=floor(C/peca(i)); //Quantidade máxima que cabe de peças (arredondar p baixo)
45 end
46
47
48 //Sortear p entre as n peças:
49
50 k=1;
51
52 while k <=QtdeMaxPadraoValido //loop para criar a qtde máxima de vetores binários
53 SomaElemVetorBinario=1000000; //número grande

```

Fonte: Elaborado pelo autor (2016)

Na Interface de utilização do Scilab, foi iniciada a entrada de dados por meio da digitação das quantidades e comprimentos informados na Tabela 7.

Para esse caso específico, considerou-se como número máximo de peças diferentes em cada padrão aquele igual a cinco.

Também foi considerada como quantidade máxima de padrões válidos, admitindo repetições, o valor de 230 padrões. Considerou-se esta quantidade, pois adicionou-se um coeficiente de segurança trinta padrões que poderiam se repetir. Após execução do algoritmo, foram escolhidos aleatoriamente 200 padrões de corte diferentes. Essa quantidade de padrões foi definida levando em conta a limitação do número máximo de variáveis inteiras da ferramenta Solver do Microsoft Excel. Essa informação está disponível em: <https://support.office.com/pt-br/article/Especificaca%C3%A7%C3%B5es-e-limites-do-Microsoft-Excel-1672b34d-7043-467e-8e27-269d656771c3>).

A Interface criada no algoritmo, onde é digitada a entrada de dados, é mostrada na Figura 18.

Figura 18 – Entrada de dados Scilab

```

Scilab 5.5.2 Console

PROGRAMA PARA GERAÇÃO DE PADRÕES PARA CORTES UNIDIMENSIONAIS

Comprimento da barra (mm): 3000
Qtde de tipos distintos de peças: 13

Peça 1 de 13:
Medida da peça (mm): 255
Peça 2 de 13:
Medida da peça (mm): 1960
Peça 3 de 13:
Medida da peça (mm): 1130
Peça 4 de 13:
Medida da peça (mm): 1465
Peça 5 de 13:
Medida da peça (mm): 100
Peça 6 de 13:
Medida da peça (mm): 405
Peça 7 de 13:
Medida da peça (mm): 1000
Peça 8 de 13:
Medida da peça (mm): 1500
Peça 9 de 13:
Medida da peça (mm): 1455
Peça 10 de 13:
Medida da peça (mm): 735
Peça 11 de 13:
Medida da peça (mm): 705
Peça 12 de 13:
Medida da peça (mm): 515
Peça 13 de 13:
Medida da peça (mm): 510

Número máximo de peças diferentes em cada padrão (<= à qtde de peças): 5
Qtde máxima de Padrões Válidos (Admitindo repetições): 230
Largura da Serra (mm): 2.5

```

Fonte: Elaborado pelo autor (2016)

Em seguida, o Algoritmo de busca aleatória gerou 230 padrões possíveis, dentro dos parâmetros indicados na entrada de dados. Esses padrões possíveis admitem ainda repetições. Uma parte desses 230 padrões gerados é apresentada na Figura 19.

Figura 19 – Parcial dos valores encontrados (considerando possíveis repetidos)

column 217 to 230													
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	6.	0.
0.	0.	0.	0.	0.	0.	0.	1.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	1.	0.	0.	0.	1.	0.	0.	0.
15.	15.	0.	0.	12.	0.	0.	0.	4.	0.	0.	22.	0.	0.
0.	0.	6.	0.	0.	1.	0.	0.	0.	6.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	1.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	1.	0.	0.	0.	0.	0.	0.	1.	0.	0.	0.
1.	1.	0.	0.	0.	0.	0.	0.	1.	0.	0.	0.	1.	0.
0.	0.	0.	0.	1.	0.	0.	0.	0.	0.	0.	1.	0.	1.
0.	0.	0.	2.	0.	0.	0.	0.	0.	0.	0.	0.	0.	1.
0.	0.	0.	0.	0.	5.	1.	0.	2.	0.	0.	0.	0.	0.
0.	0.	1.	0.	2.	0.	0.	2.	0.	1.	0.	0.	0.	3.

FILTRANDO DADOS: APRESENTAR SOMENTE PADRÕES NAO REPETIDOS

Fonte: Elaborado pelo autor (2016)

O próximo passo do algoritmo consistiu em encontrar os valores padrões válidos sem repetições (esse procedimento é feito pelo próprio algoritmo), no qual foram encontrados 222 padrões do universo de 230 padrões encontrados anteriormente. A Figura 20 mostra uma parte desses padrões não repetidos.

Figura 20 – Parcial dos padrões encontrados (desconsiderando repetidos)

```

FILTRANDO DADOS: APRESENTAR SOMENTE PADRÕES NAO REPETIDOS

PadraoValidoNaoRepetido
.

                                column 221 to 222

0.    0.
0.    0.
0.    0.
1.    0.
0.    0.
0.    0.
0.    0.
0.    0.
0.    0.
0.    2.
2.    0.
0.    0.
0.    0.
0.    0.

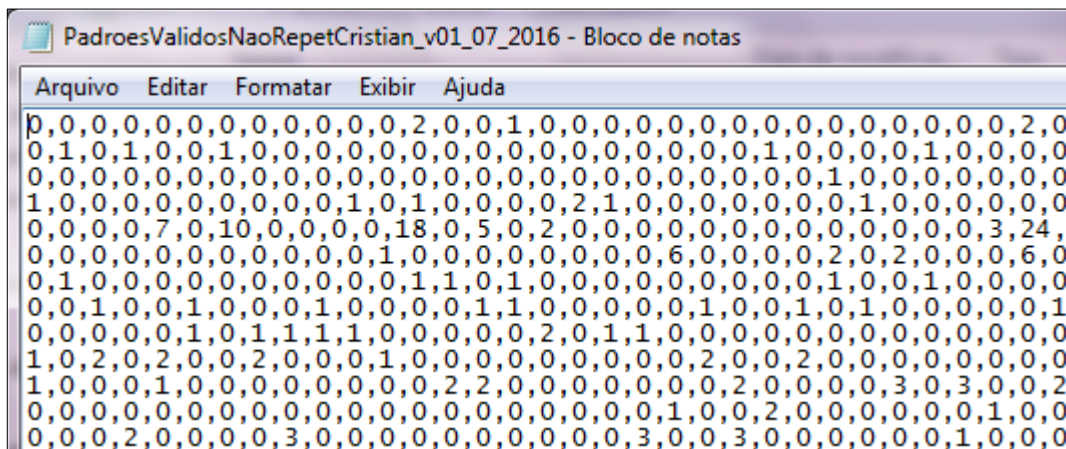
->pwd
ans =

I:\TCC_2016
    
```

Fonte: Elaborado pelo autor (2016)

O algoritmo gerado grava os resultados em um arquivo de texto (.txt). Sendo assim, buscou-se o arquivo salvo com extensão “.txt” na pasta local do programa Scilab. Uma parte desse arquivo é mostrada na Figura 21.

Figura 21 – Saída de dados do Scilab gravada em arquivo .txt

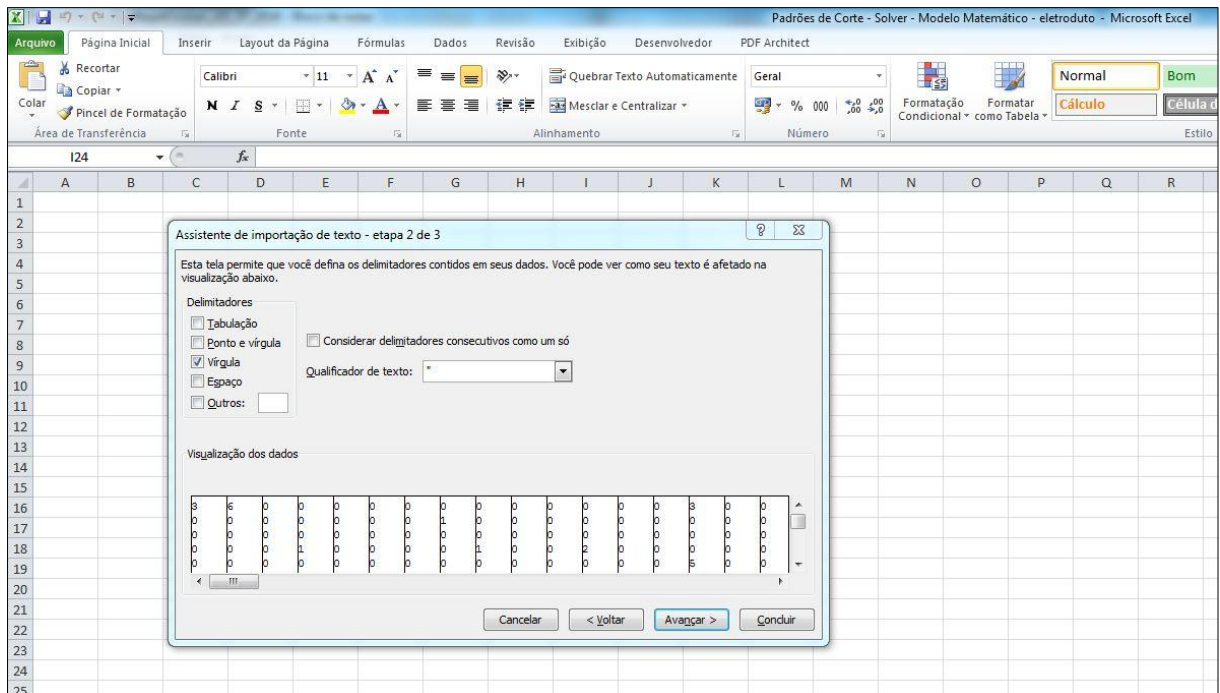


Fonte: Elaborado pelo autor (2016)

Indica-se copiar e colar, em especial, os valores do arquivo “.txt”, em uma planilha em

branco do Microsoft Excel para formatar os dados em células e colunas, conforme mostrado na Figura 22.

Figura 22 – Entrada dados no Excel



Fonte: Elaborado pelo autor (2016)

Na seqüência deve-se preencher todos os campos da interface da planilha com os dados do problema em questão, conforme mostrado na Figura 23.

Figura 23 – Interface planilha – entrada de dados

Entrada de dados			
Comprimento eletroduto (mm)	3000		
Largura da Serra considerada (mm)	2,5		
Comprimento Eletroduto 1 (mm)	255	Quantidade eletroduto 1	6
Comprimento Eletroduto 2 (mm)	1960	Quantidade eletroduto 2	3
Comprimento Eletroduto 3 (mm)	1130	Quantidade eletroduto 3	1
Comprimento Eletroduto 4 (mm)	1465	Quantidade eletroduto 4	2
Comprimento Eletroduto 5 (mm)	100	Quantidade eletroduto 5	1
Comprimento Eletroduto 6 (mm)	405	Quantidade eletroduto 6	3
Comprimento Eletroduto 7 (mm)	1000	Quantidade eletroduto 7	18
Comprimento Eletroduto 8 (mm)	1500	Quantidade eletroduto 8	4
Comprimento Eletroduto 9 (mm)	1455	Quantidade eletroduto 9	1
Comprimento Eletroduto 10 (mm)	735	Quantidade eletroduto 10	2
Comprimento Eletroduto 11 (mm)	705	Quantidade eletroduto 11	4
Comprimento Eletroduto 12 (mm)	515	Quantidade eletroduto 12	1
Comprimento Eletroduto 13 (mm)	510	Quantidade eletroduto 13	1

Fonte: Elaborado pelo autor (2016)

O próximo passo consiste em colar os padrões de corte obtidos do arquivo txt (conforme figuras 21 e 22) nos campos de padrões da interface da planilha para a execução do Solver mostrada na Figura 21.

Figura 24 – Planilha de dados para a ferramenta Solver

Objeto	Comprimento (mm)	P1	P2	P3	P4	P5	P6	P7	P8	P9
Comprimento Eletroduto 1	255	0	0	0	0	8	0	0	0	1
Comprimento Eletroduto 2	1960	0	0	0	0	0	0	0	0	0
Comprimento Eletroduto 3	1130	0	0	0	0	0	0	0	0	1
Comprimento Eletroduto 4	1465	0	0	0	0	0	2	0	0	0
Comprimento Eletroduto 5	100	16	0	0	0	2	0	0	0	0
Comprimento Eletroduto 6	405	2	0	3	0	0	0	0	0	0
Comprimento Eletroduto 7	1000	0	0	1	0	0	0	0	0	0
Comprimento Eletroduto 8	1500	0	0	0	0	0	0	0	0	0
Comprimento Eletroduto 9	1455	0	2	0	0	0	0	2	1	0
Comprimento Eletroduto 10	735	0	0	1	4	0	0	0	2	0
Comprimento Eletroduto 11	705	0	0	0	0	1	0	0	0	0
Comprimento Eletroduto 12	515	1	0	0	0	0	0	0	0	0
Comprimento Eletroduto 13	510	0	0	0	0	0	0	0	0	3
Soma		2925	2910	2950	2940	2945	2930	2910	2925	2915
Largura da Serra (mm)	Desperdício cortes	75	90	50	60	55	70	90	75	85
2,5	Desperdício Serra	47,5	5	12,5	10	27,5	5	5	7,5	12,5
----->	Sobra de material	27,50	85,00	37,50	50,00	27,50	65,00	85,00	67,50	72,50
Comprimento eletroduto(mm)										
3000										
Função Objetivo:	Desperdício (mm):	792,5								
Variáveis de decisão		P1	P2	P3	P4	P5	P6	P7	P8	P9
		0	0	0	0	0	0	0	0	0
Restrições:		Qtde produzida	Demanda							
Qtde Mínima de eletroduto 1		8	6							
Qtde Máxima de eletroduto 1		8	9							
Qtde Mínima de Eletroduto 2		5	3							
Qtde Máxima de eletroduto 2		5	5							
Qtde Mínima de eletroduto 3		2	1							
Qtde Máxima de eletroduto 3		2	2							
Qtde Mínima de eletroduto 4		3	2							
Qtde Máxima de eletroduto 4		3	3							

Fonte: Elaborado pelo autor (2016)

Na sequência, após salvar a planilha, deve-se clicar no menu Dados no Microsoft Excel, no ícone do Solver¹, onde é aberta uma janela para configuração. Deve-se conferir se a opção “tornar variáveis irrestritas não negativas” está ativada.

Também foi adicionada no ícone de opções do Solver, uma limitação de tempo de execução máxima para parada de 900 segundos (15 min). Deve-se clicar em “Ok” para confirmar e, após isso, clicar em resolver para que o Solver inicie a resolução, conforme mostrado na Figura 25.

¹ Se não encontrar o ícone Solver no Menu Dados, você deverá antes habilitar a disponibilidade acessando o menu Arquivo > Opções > Suplementos do Excel.

Figura 25 – Parâmetros do Solver

Parâmetros do Solver

Definir Objetivo:

Para: Máx. Mín. Valor de:

Alterando Células Variáveis:

Sujeito às Restrições:

\$D\$28:\$GQ\$28 = número inteiro
 \$D\$32 >= \$E\$32
 \$D\$33 <= \$E\$33
 \$D\$34 >= \$E\$34
 \$D\$35 <= \$E\$35
 \$D\$36 >= \$E\$36
 \$D\$37 <= \$E\$37
 \$D\$38 >= \$E\$38
 \$D\$39 <= \$E\$39
 \$D\$40 >= \$E\$40
 \$D\$41 <= \$E\$41
 \$D\$42 >= \$E\$42
 \$D\$43 <= \$E\$43

Tornar Variáveis Irrestritas Não Negativas

Selecionar um Método de Solução:

Método de Solução
 Selecione o mecanismo GRG Não Linear para Problemas do Solver suaves e não lineares. Selecione o mecanismo LP Simplex para Problemas do Solver lineares. Selecione o mecanismo Evolutionary para problemas do Solver não suaves.

Ajuda Resolver Fechar

P4	P5	P6	P7
0	0	0	0
0	0	0	0
0	0	0	0
1	0	0	0
0	0	0	0
0	3	2	0
0	0	0	0
0	0	0	0
1	0	1	2
0	0	0	0
0	1	1	0
0	2	0	0
0	0	0	0
2920,9	2945	2966,68	291
79,07	55,05	33,32	90,1
5	15	10	5
74,07	40,05	23,32	85,1

Fonte: Elaborado pelo autor (2016)

Após 900 segundos de execução, o Solver encontrou um desperdício total de 792,5 mm, gerando a quantidade produzida mostrada na Figura 26.

Figura 26 – Resultado Solver

Opções do Solver
Tempo Máx. 900 s, Iterações 10000000, Precisão 0,000001
Subproblemas Máx. Ilimitado, Soluç. Máx. Núm. Inteiro Ilimitado, Tolerância de Número Inteiro 1%, Assumir Não Negativo

Célula do Objetivo (Min.)

Célula	Nome	Valor Original	Valor Final
\$D\$25	Desperdício (mm): P1	895	792,5

Restrições

Célula	Nome	Valor da Célula	Fórmula	Status	Margem de Atraso
\$D\$32	Qtde Mínima de eletroduto 1 Qtde produzida	8	\$D\$32>=\$E\$32	Não-associação	2
\$D\$33	Qtde Máxima de eletroduto 1 Qtde produzida	8	\$D\$33<=\$E\$33	Não-associação	1
\$D\$34	Qtde Mínima de Eletroduto 2 Qtde produzida	5	\$D\$34>=\$E\$34	Não-associação	2
\$D\$35	Qtde Máxima de eletroduto 2 Qtde produzida	5	\$D\$35<=\$E\$35	Associação	0
\$D\$36	Qtde Mínima de eletroduto 3 Qtde produzida	2	\$D\$36>=\$E\$36	Não-associação	1
\$D\$37	Qtde Máxima de eletroduto 3 Qtde produzida	2	\$D\$37<=\$E\$37	Associação	0
\$D\$38	Qtde Mínima de eletroduto 4 Qtde produzida	3	\$D\$38>=\$E\$38	Não-associação	1
\$D\$39	Qtde Máxima de eletroduto 4 Qtde produzida	3	\$D\$39<=\$E\$39	Associação	0
\$D\$40	Qtde Mínima de eletroduto 5 Qtde produzida	1	\$D\$40>=\$E\$40	Associação	0
\$D\$41	Qtde Máxima de eletroduto 5 Qtde produzida	1	\$D\$41<=\$E\$41	Não-associação	1
\$D\$42	Qtde Mínima de eletroduto 6 Qtde produzida	5	\$D\$42>=\$E\$42	Não-associação	2
\$D\$43	Qtde Máxima de eletroduto 6 Qtde produzida	5	\$D\$43<=\$E\$43	Associação	0
\$D\$44	Qtde Mínima de eletroduto 7 Qtde produzida	18	\$D\$44>=\$E\$44	Associação	0
\$D\$45	Qtde Máxima de eletroduto 7 Qtde produzida	18	\$D\$45<=\$E\$45	Não-associação	9
\$D\$46	Qtde Mínima de eletroduto 8 Qtde produzida	5	\$D\$46>=\$E\$46	Não-associação	1
\$D\$47	Qtde Máxima de eletroduto 8 Qtde produzida	5	\$D\$47<=\$E\$47	Não-associação	1
\$D\$48	Qtde Mínima de eletroduto 9 Qtde produzida	2	\$D\$48>=\$E\$48	Não-associação	1
\$D\$49	Qtde Máxima de eletroduto 9 Qtde produzida	2	\$D\$49<=\$E\$49	Associação	0
\$D\$50	Qtde Mínima de eletroduto 10 Qtde produzida	3	\$D\$50>=\$E\$50	Não-associação	1
\$D\$51	Qtde Máxima de eletroduto 10 Qtde produzida	3	\$D\$51<=\$E\$51	Associação	0
\$D\$52	Qtde Mínima de eletroduto 11 Qtde produzida	6	\$D\$52>=\$E\$52	Não-associação	2
\$D\$53	Qtde Máxima de eletroduto 11 Qtde produzida	6	\$D\$53<=\$E\$53	Associação	0
\$D\$54	Qtde Mínima de eletroduto 12 Qtde produzida	2	\$D\$54>=\$E\$54	Não-associação	1
\$D\$55	Qtde Máxima de eletroduto 12 Qtde produzida	2	\$D\$55<=\$E\$55	Associação	0
\$D\$56	Qtde Mínima de eletroduto 13 Qtde produzida	5	\$D\$56>=\$E\$56	Não-associação	4
\$D\$57	Qtde Máxima de eletroduto 13 Qtde produzida	2	\$D\$57<=\$E\$57	Associação	0

\$D\$28:\$G\$28=Número Inteiro

Fonte: Elaborado pelo autor (2016)

Com os padrões encontrados pelo Solver, montou-se a tabela mostrada na Figura 27, denominada como “Reposta Padrão”, para inclusão do documento de orientação em montagem afim de informar qual é a sequência de corte para cada barra de eletroduto.

Figura 27 – Resposta padrão

		Padrão 22	Padrão 62	Padrão 84	Padrão 100	Padrão 115	Padrão 116	Padrão 124	Padrão 170	Padrão 182	Padrão 185
Comprimento Eletroduto 1	255			3	2						
Comprimento Eletroduto 2	1960	1									
Comprimento Eletroduto 3	1130		2								
Comprimento Eletroduto 4	1465								1		
Comprimento Eletroduto 5	100										1
Comprimento Eletroduto 6	405					1		1		1	
Comprimento Eletroduto 7	1000	1			1	2	1	1	1	1	
Comprimento Eletroduto 8	1500			1				1		1	
Comprimento Eletroduto 9	1455						1				
Comprimento Eletroduto 10	735				2						1
Comprimento Eletroduto 11	705		1	1							3
Comprimento Eletroduto 12	515						1				
Comprimento Eletroduto 13	510					1			1		
Perda considerada (largura da serra)	2,5	2960	2965	2970	2980	2915	2970	2905	2975	2905	2950
Comprimento eletroduto (mm)	3000	3000	3000	3000	3000	3000	3000	3000	3000	3000	3000
Sobra de material	792,5	35	27,5	17,5	7,5	75	22,5	87,5	17,5	87,5	37,5
Qtde a ser executado cada padrão	-	5	1	2	1	2	2	2	3	1	1

Fonte: Elaborado pelo autor (2016)

A figura 27 demonstra as informações necessárias para o processo de corte, no qual podemos ver qual comprimento será cortado em cada padrão. Nela consta a quantidade e comprimento de cada peça necessária para o projeto. Por exemplo, pode-se ver o padrão 22, que indica ao operador que o eletroduto comercial será cortado em 2 peças, no qual será uma peça de 1960 mm e outra peça de 1000 mm. Também consta na figura 27, o valor da perda considerada (largura da serra), bem como o comprimento do eletroduto considerado no projeto. E também indica o número de vezes que este padrão será executado e a sobra do material após cada padrão cortado e a soma total das sobras. Assim o operador poderá executar os cortes de acordo com necessidade, de modo a minimizar as perdas de materiais em campo.

6 CONCLUSÃO

Geralmente processos que envolvem cortes são caracterizados pela grande ocorrência de perdas ocasionadas pela sobra de material. A busca pela minimização dessas perdas é uma tarefa constante, e exige um complexo arranjo de combinações possíveis de padrões de corte, o que muitas vezes, na prática, é realizado de forma arbitrária, sem utilização de métodos científicos.

Um aspecto importante deste trabalho foi mostrar a possibilidade de executar um modelo matemático computacional de baixo custo, pois utiliza um software de programação com licença livre (Scilab) e outro software geralmente já presente nas empresas (Microsoft Excel) podendo assim alcançar benefícios significativos em relação ao desperdício, contribuindo com as questões ambientais e, conseqüentemente, aumentando a lucratividade da empresa.

O modelo de otimização aqui proposto pode ser utilizado para diversas categorias de problemas que envolvem cortes unidimensionais como, por exemplo, no corte de barras, bobinas, perfis, tubos, entre outros, o que justifica a importância do desenvolvimento desta pesquisa. O algoritmo final executado neste trabalho foi específico, mas pode ser uma referência para outros estudos de caso.

Quando em um projeto industrial que envolve processos de cortes existe informação documental clara, evita-se transtornos por falta ou desperdício de material. Além disso, o uso de tal documentação auxilia o operador de produção em relação ao procedimento correto, eliminando assim a responsabilidade de seleção da ordem de corte dos materiais em obra.

No procedimento atual de especificação de material para cada projeto, esse pode estar suscetível a erros no processo de cortes do eletroduto executado na obra, podendo gerar perdas desnecessárias de material, ou até mesmo a falta de material devido à sequência de cortes inapropriada. Além disso, o procedimento não considera a largura da serra como desperdício do corte, apesar de o procedimento atual utilizar 20% de coeficiente de segurança, o que não evita a eliminação de erro nos cortes.

Atualmente, no documento disponibilizado para o operador na empresa estudada neste trabalho, não consta a informação precisa de corte, ficando na responsabilidade do

operador definir o plano de corte, o que indica um procedimento empírico, muitas vezes o operador não está instruído sobre esta responsabilidade e informação, além de não ter o tempo hábil para a avaliação correta.

Em relação aos algoritmos desenvolvidos, com o primeiro foi possível encontrar resultados nos testes quando utilizados comprimentos de grande valor numérico (comprimentos grandes), pois o número de possibilidades é menor, resultando em problema de baixa complexidade para o Scilab encontrar todas as possibilidades e selecionar os padrões. Porém, aplicando o algoritmo no cenário real na empresa, ele não apresentou resultados mesmo após 15 horas de execução. Foi possível perceber que esse algoritmo embora formulado corretamente, buscava por todos os padrões possíveis mas não fornecia resultados em tempo hábil e, ainda, o número de padrões distintos que poderia ser gerado certamente ultrapassaria os limites do Microsoft Excel no que se refere ao número de variáveis no problema.

Já o segundo algoritmo mostrou-se bastante rápido em relação ao primeiro, além de delimitar critérios de paradas e seleção de alguns dos melhores padrões, tornando assim uma execução mais rápida do algoritmo, sendo possível sua aplicação no cotidiano da empresa.

Fazendo uma comparação entre o procedimento atual utilizado na empresa e o procedimento proposto com o algoritmo, foi identificada uma redução possível de matéria prima de aproximadamente 1,7%. Foram analisados outros casos que puderam chegar a mais de 5%, dependendo dos padrões encontrados. O que delimitou os resultados foi o tempo limite, mas aumentando para 30 minutos ou 1 hora, poderiam ser obtidos resultados ainda melhores.

Essa implementação vem a agregar na qualidade da informação repassada ao operador em obra, trazendo mais confiabilidade, buscando assim um processo de corte robusto, evitando perdas desnecessárias.

Outro aspecto do trabalho foi a resolução de um problema prático na indústria. Neste quesito, o presente trabalho conseguiu atingir o objetivo geral proposto, de desenvolver uma metodologia para geração de padrões de corte (o que é feito pelo algoritmo em Scilab) e escolha dos padrões a serem utilizados (obtido pela ferramenta Solver do Excel), de modo a aplicá-lo em um problema de cortes de eletrodutos em uma empresa do ramo de equipamentos elétricos da cidade de Jaraguá do Sul, no estado de Santa Catarina, na operação de especificação e orientação de corte. Como resultado do modelo de otimização, foi gerada

uma tabela denominada resposta padrão a ser utilizada no processo de corte. Resultados apontaram possíveis reduções no desperdício de até 5%, conforme já explanado neste trabalho.

Para trabalhos futuros, poderá ser feita a aplicação deste algoritmo em outros problemas de cortes unidimensionais aplicados na indústria, como, por exemplo, eletrocalhas ou tubulações hidráulicas.

Pode-se também propor melhorias no algoritmo apresentado na forma da escolha dos padrões, através da aplicação de outras heurísticas de busca que possam refinar os tipos de padrões encontrados, para que esses sejam utilizados no modelo de otimização.

Pode-se ainda realizar um estudo na empresa com a finalidade de identificar outras necessidades, sendo que uma delas, por exemplo, poderia ser a limitação da quantidade de padrões a serem obtidos como resposta do modelo matemático, para evitar muitas programações diferentes da máquina de corte.

Outra sugestão para futuros trabalhos é a utilização de diferentes tipos de *Solvers* como *Cplex*, *Lingo* e *LINDO (Linear Designer Optimizer)*, pois esses permitem a utilização de um maior número de variáveis inteiras, o que permite uma maior entrada de padrões de corte no modelo de otimização, porém, possuem a desvantagem referente ao custo das licenças de utilização.

Outras sugestões para trabalhos futuros poderão ser identificadas durante a implementação prática do modelo no ambiente industrial, ou com a utilização de novos métodos de modelagem e resolução computacional, o que indica um caminho promissor para novas pesquisas.

REFERÊNCIAS

- AGUILAR, Luis Joyanes. **Fundamentos de programação:** Algoritmos, Estruturas de dados e objetos. São Paulo: Mcgraw-hill, 2008. 690 p.
- ANDRADE, E. L., **Introdução à Pesquisa Operacional:** Métodos e Modelos Para a Análise De Decisão. 2. ed. Rio de Janeiro: LTC, 2000.
- ARENALES, M. *et al.* **Pesquisa Operacional.** 4ª reimpressão. Rio de Janeiro: Elsevier, 2007.
- BORATTI, Isaias Camilo; OLIVEIRA, Álvaro Borges de. **Introdução à Programação Algoritmos.** 3. ed. Florianópolis: Visual Books, 2007. 158 p.
- BORGES NETO, Manuel Rangel; CARVALHO, Paulo. **Geração de energia elétrica:** Fundamentos. São Paulo: Érica Ltda., 2013. 158 p.
- BUENO, Fabrício. **Otimização gerencial com EXCEL.** Florianópolis: Visual Books, 2007. 136 p.
- CAVALIN, Geraldo; CERVELIN, Severino. **Instalações elétricas prediais.** 21. ed. São Paulo: Érica, 2011. 421 p.
- CUNHA, R.R. M.; DIAS, A. **Algoritmo de otimização de corte unidimensional análise de dados,** Anais, XV Congresso brasileiro de engenharia mecânica, 10 p., 1999.
- FORBELLONE, André Luiz Villar; EBERSPÄCHER, Henri Frederico. **Lógica de programação:** A construção de algoritmos e estruturas de dados. 3. ed. São Paulo: Prentice Hall, 2005. 218 p.
- FLORIAN, M.; LENSTRA, J. K.; KAN, A. H. G. R. **Deterministic production planning: Algorithms and complexity.** Management Science, v. 26, n. 7, p. 669-679, 1980.
- GOLDBARG, M. C.; LUNA, H. C. L. **Otimização Combinatória e Programação Linear: Modelos e Algoritmos.** 2ª. ed. Rio de Janeiro: Elsevier, 2005
- LACHTERMACHER, Gerson. **Pesquisa Operacional na tomada de decisões:** modelagem em excel. 3. ed. Rio de Janeiro: Elsevier, 2007. 213 p.
- LIMA FILHO, Domingos Leite. **Projetos de instalações elétricas prediais.** 12. ed. São Paulo: Érica, 2011. 271 p.
- LOESCH, Cláudio; HEIN, Nelson. **Pesquisa Operacional:** Fundamentos e modelos. São Paulo: Saraiva, 2009.
- OLIVEIRA, B.; SALVADOR, R.; SILVA, V. M. D. **Otimização do Problema de Corte Unidimensional: Um Estudo de Caso na Indústria Papeleira.** Anais: EEPA, Encontro de Engenharia de Produção Agroindustrial, 7, Campo Mourão- PR, 2013.

PRADO, Darci. **Programação linear**. Belo Horizonte : DG, 1999

S.A.S, Scilab Enterprises. **Scilab: About Scilab**. 2015. Disponível em: <<http://www.scilab.org/scilab/about>>. Acesso em: 07 set. 2016.

SEVERINO, Antônio Joaquim. **Metodologia de trabalho científico**. 23. ed. São Paulo: Cortez, 2007.

SILVA, E. M. *et al.* **Pesquisa Operacional**. 3ª. ed. São Paulo: Atlas, 1998.

SILVEIRA JÚNIOR, J. A.; PINHEIRO, P. R.; THOMAZ, A. C. F. **Otimização das Perdas em Cortes Guilhotinados para Bobinas de Aço na Indústria Metalmeccânica**, Anais, SBPO: Simpósio Brasileiro de Pesquisa Operacional, 36 p. 1759 – 1770, 2004.

SIMONE, Gilio Alusio; CREPPE, Renato Crivellari. **Conversão eletromecânica de energia: Uma introdução ao estudo**. São Paulo: Érica Ltda., 2013. 324 p.

SIMONE, Gilio Alusio. **Centrais e aproveitamentos hidrelétricos: Uma introdução ao estudo**. São Paulo: Érica, 2000. 246 p.

WALENIA, Paulo Sérgio. **Projetos Elétricos Industriais**. Curitiba: Base Editorial, 2010. 288 p.

WAVRZYNCZAK, Hione Cleder; ULBRICHT, Gerson; TEIXEIRA, Edson Sidnei Maciel. **Modelo matemático para cortes de barras de aço no processo de fabricação de triângulos**. Holos, [s.l.], v. 8, p.252-261, 13 jan. 2016. Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte (IFRN). <http://dx.doi.org/10.15628/holos.2015.3605>.

APÊNDICE

**APÊNDICE A - ALGORITMO DE BUSCA ORDENADA: CÓDIGO EM SCILAB
(FIXADO EM 13 PEÇAS)**

```
//Programa Geração de padrões de cortes unidimensionais
clear
clc
kk=1;
printf('---Programa Geração de padrões de cortes unidimensionais---');
printf ('\n');// linha em branco
printf ('\n');// linha em branco
//-----
printf('Dados:');
//Dados:
//MaximoPadroes=3;
C= input ( 'Informe o comprimento da barra padrão (mm): ');
//Comprimento da barra (mm)
printf ('\n');// linha em branco
QtdeMaxPadraoValido= input ( 'Informe a quantidade máxima de Padrões Válidos
desejados: ');

printf ('\n');// linha em branco
printf('ATENÇÃO: COLOCAR A PEÇA DE MENOR COMPRIMENTO COMO PEÇA 1!');
printf ('\n');// linha em branco
printf ('\n');// linha em branco
peca1=input ('Informe o comprimento da peça 1 (mm): ');
//Comprimento da peça 1 (mm)
peca2=input ('Informe o comprimento da peça 2 (mm): ');
//Comprimento da peça 2 (mm)
peca3=input ('Informe o comprimento da peça 3 (mm): ');
//Comprimento da peça 3 (mm)
peca4=input ('Informe o comprimento da peça 4 (mm): ');
//Comprimento da peça 4 (mm)
peca5=input ('Informe o comprimento da peça 5 (mm): ');
//Comprimento da peça 5 (mm)
peca6=input ('Informe o comprimento da peça 6 (mm): ');
//Comprimento da peça 6 (mm)
peca7=input ('Informe o comprimento da peça 7 (mm): ');
//Comprimento da peça 7 (mm)
peca8=input ('Informe o comprimento da peça 8 (mm): ');
//Comprimento da peça 8 (mm)
peca9=input ('Informe o comprimento da peça 9 (mm): ');
//Comprimento da peça 9 (mm)
```

```

peca10=input ('Informe o comprimento da peça 10 (mm): ');
//Comprimento da peça 10 (mm)
peca11=input ('Informe o comprimento da peça 11 (mm): ');
//Comprimento da peça 11 (mm)
peca12=input ('Informe o comprimento da peça 12 (mm): ');
//Comprimento da peça 12 (mm)
peca13=input ('Informe o comprimento da peça 13 (mm): ');
//Comprimento da peça 13 (mm)
//-----

//-----
//INÍCIO: Código para gerar todas as combinações (Matriz Padrao)
//quantidades máximas de cada peça cortada:
Qmax_peca1=floor(C/peca1); //Quantidade máxima que cabe de peca1 (arredondar p baixo)
Qmax_peca2=floor(C/peca2); //Quantidade máxima que cabe de peca2 (arredondar p baixo)
Qmax_peca3=floor(C/peca3); //Quantidade máxima que cabe de peca3 (arredondar p baixo)
Qmax_peca4=floor(C/peca4); //Quantidade máxima que cabe de peca4 (arredondar p baixo)
Qmax_peca5=floor(C/peca5); //Quantidade máxima que cabe de peca5 (arredondar p baixo)
Qmax_peca6=floor(C/peca6); //Quantidade máxima que cabe de peca6 (arredondar p baixo)
Qmax_peca7=floor(C/peca7); //Quantidade máxima que cabe de peca7 (arredondar p baixo)
Qmax_peca8=floor(C/peca8); //Quantidade máxima que cabe de peca8 (arredondar p baixo)
Qmax_peca9=floor(C/peca9); //Quantidade máxima que cabe de peca9 (arredondar p baixo)
Qmax_peca10=floor(C/peca10); //Quantidade máxima que cabe de peca10 (arredondar p
baixo)
Qmax_peca11=floor(C/peca11); //Quantidade máxima que cabe de peca11 (arredondar p
baixo)
Qmax_peca12=floor(C/peca12); //Quantidade máxima que cabe de peca12 (arredondar p
baixo)
Qmax_peca13=floor(C/peca13); //Quantidade máxima que cabe de peca13 (arredondar p
baixo)
//Gera padrões
//Padrao(no. do padrão,tamanho da peça cortada)
jj=1;
qtde=Qmax_peca1;

for k1=Qmax_peca1:-1:0 do
Padrao(1,jj)= k1
    for k2=Qmax_peca2:-1:0 do
        Padrao(2,jj)= k2
            for k3=Qmax_peca3:-1:0 do
                Padrao(3,jj)= k3

```

```

for k4=Qmax_peca4:-1:0 do
Padrao(4,jj)= k4
  for k5=Qmax_peca5:-1:0 do
Padrao(5,jj)= k5
    for k6=Qmax_peca6:-1:0 do
Padrao(6,jj)= k6
      for k7=Qmax_peca7:-1:0 do
Padrao(7,jj)= k7
        for k8=Qmax_peca8:-1:0 do
Padrao(8,jj)= k8
          for k9=Qmax_peca9:-1:0 do
Padrao(9,jj)= k9
            for k10=Qmax_peca10:-1:0 do
Padrao(10,jj)= k10
              for k11=Qmax_pecal1:-1:0 do
Padrao(11,jj)= k11
                for k12=Qmax_pecal2:-1:0 do
Padrao(12,jj)= k12
                  for k13=Qmax_pecal3:-1:0 do
Padrao(1,jj)= k1
Padrao(2,jj)= k2
Padrao(3,jj)= k3
Padrao(4,jj)= k4
Padrao(5,jj)= k5
Padrao(6,jj)= k6
Padrao(7,jj)= k7
Padrao(8,jj)= k8
Padrao(9,jj)= k9
Padrao(10,jj)= k10
Padrao(11,jj)= k11
Padrao(12,jj)= k12
Padrao(13,jj)= k13

//=====
//Calcula comprimento:
dimensao=Padrao(1,jj)*peca1+Padrao(2,jj)*peca2+Padrao(3,jj)*peca3+Padrao(4,jj)*peca4+P
adrao(5,jj)*peca5+Padrao(6,jj)*peca6+Padrao(7,jj)*peca7+Padrao(8,jj)*peca8+Padrao(9,jj)*p
eca9+Padrao(10,jj)*peca10+Padrao(11,jj)*peca11+Padrao(12,jj)*peca12+Padrao(13,jj)*peca
13;

//Calcula desperdício

```



```
desperdicioPadrao=C-dimensao;
```

```
//INÍCIO: Teste para ver se valida o padrão
```

```
if desperdicioPadrao>=0 & desperdicioPadrao<= peca1 then //peca1: Menor peça
```

```
PadraoValido(1,kk)=Padrao(1,jj)
```

```
PadraoValido(2,kk)=Padrao(2,jj)
```

```
PadraoValido(3,kk)=Padrao(3,jj)
```

```
PadraoValido(4,kk)=Padrao(4,jj)
```

```
PadraoValido(5,kk)=Padrao(5,jj)
```

```
PadraoValido(6,kk)=Padrao(6,jj)
```

```
PadraoValido(7,kk)=Padrao(7,jj)
```

```
PadraoValido(8,kk)=Padrao(8,jj)
```

```
PadraoValido(9,kk)=Padrao(9,jj)
```

```
PadraoValido(10,kk)=Padrao(10,jj)
```

```
PadraoValido(11,kk)=Padrao(11,jj)
```

```
PadraoValido(12,kk)=Padrao(12,jj)
```

```
PadraoValido(13,kk)=Padrao(13,jj)
```

```
kk=kk+1;
```

```
end
```

```
if kk > QtdeMaxPadraoValido
```

```
printf('Padrões Válidos): \n')
```

```
disp(PadraoValido) //Escreve os padrões válidos na tela
```

```
write_csv(PadraoValido, 'saidaTeste2') //Escreve os padrões válidos em um arquivo
```

```
return //Finaliza o programa se executar a Qtde máxima permitida de padrões válidos
```

```
end
```

```
//FIM: Teste para validaR o padrão
```

```
//-----
```

```
//=====
```

```
jj=jj+1; //Este incremento deve ficar dentro do loop mais interno
```

```
end
```

```
end
```

```
end
```

```
end
```

```
end
```

```
end
```

```
end
```

```
                end
            end
        end
    end

    end
end
end
end
end
//FIM: Código para gerar todas as combinações (Matriz Padrao)
//-----

//-----

//printf('Resultados: \n \n')
//printf('Todas as combinações (padrão): \n')
//printf('\n')

//printf('desperdicioPadrao): \n')
//disp(desperdicioPadrao')
//printf('\n')

//printf('Padrões Válidos): \n')
//disp(PadraoValido)

//write_csv(PadraoValido, 'saidaTeste2') //ESCREVE EM UM ARQUIVO DE SAIDA DE
DADOS EM COLUNAS

//FIM: SAÍDA DE DADOS
//-----
```

**APÊNDICE B - ALGORITMO DE BUSCA ALEATÓRIA: CÓDIGO EM SCILAB
(PARA N PEÇAS)**

Código em Scilab do Algoritmo 2

```
//Programa Geração de padrões de cortes unidimensionais
//Versão com geração aleatória

clear
clc

//-----
//Dados:
//C=300; //Comprimento da barra (cm)
//peca=[50 60 100 200 210]; //Vetor Comprimento das peças
//p=3; //Número máximo de peças diferentes em cada padrão

    printf('\n')
    printf('PROGRAMA PARA GERAÇÃO DE PADRÕES PARA CORTES
UNIDIMENSIONAIS')
    printf('\n \n')

//INÍCIO: ENTRADA DE DADOS -----
C= input ( 'Comprimento da barra (mm): ');
Qtdepeças= input ( 'Qtde de tipos distintos de peças: ');
printf('\n \n')
for i=1:Qtdepeças
    printf('Peça %g de %g:',i,Qtdepeças)
    pecaSemSerra(i)= input ( 'Medida da peça (mm): ');
end
printf('\n \n')
p= input ( 'Número máximo de peças diferentes em cada padrão (<= à qtde de peças): ');
QtdeMaxPadraoValido= input ( 'Qtde máxima de Padrões Válidos (Admitindo repetições): ');
LargSerra= input ( 'Largura da Serra (mm): ');

//Incrementar a largura da serra em cada peça:
for i=1:Qtdepeças
    peca(i)= pecaSemSerra(i)+LargSerra;
end
//FIM: ENTRADA DE DADOS -----

peca=peca';
```

```

//INÍCIO: Código para gerar uma matriz 0, 1 com no máximo k números 1 por coluna
jj=1;
//quantidades máximas de cada peça cortada:
[ncolunasPeca nPecas]=size(peca);//Lê o numero de elementos do vetor Peça
for i =1:nPecas
Qmax_peca(i)=floor(C/peca(i)); //Quantidade máxima que cabe de pecas (arredondar p baixo)
end

//Sortear p entre as n peças:

k=1;

while k <=QtdeMaxPadraoValido//loop para criar a qtde máxima de vetores binários
    SomaElemVetorBinario=1000000;//número grande
    while SomaElemVetorBinario > p //loop para criar um vetor binario com no máx p
    elementos
        VetorBinario=round(rand(1,nPecas));
        SomaElemVetorBinario=sum(VetorBinario);
        //Armazenar os vetores binarios em uma matriz
    end

    //Multiplicar o vetor binário pelo rand:
    VetorRandomicoEsparso=VetorBinario.*(rand(1,nPecas));
    //Isso gerou um vetor com o percentual [0,1] da Qtde a ser produzida:
    //Multiplicar o vetor com o percentual pela qtde máxima de cada peça:
    VetorPercentQtdeMax=VetorRandomicoEsparso.*Qmax_peca';
    //Arredondar:
    VetorArredPercentQtdeMax=round(VetorPercentQtdeMax);

    //Calcula comprimento:
    Comprimento=0;
    for i = 1:nPecas
        Comprimento=Comprimento+VetorArredPercentQtdeMax(i)*peca(i);
    end
    //Fim Calcula comprimento

    //Calcula desperdício
    desperdicioPadrao=C-Comprimento;
    //FIM Calcula desperdício

//Verifica se padrão é aproveitável, e armazena na Matriz Padrão Válido

```

```

if desperdicioPadrao>=0 & desperdicioPadrao<= min(peca) then //Menor peça
    for ii=1:nPecas
        PadraoValido(ii,jj)=VetorArredPercentQtdeMax(ii);

        //COLOCAR AQUI O CÓDIGO PARA EVITAR PADRÕES REPETIDOS

    end
    jj=jj+1;
    k=k+1;
end
end

disp(PadraoValido)//Escreve os padrões válidos na tela

write_csv(PadraoValido, 'PadroesValidosCristian_v05_06_2016') //Escreve os padrões
válidos em um arquivo
//Digitar: pwd para saber em que pasta está gravando.

//FIM:

//CÓDIGO PARA EVITAR PADRÕES REPETIDOS -----
    printf('\n')
    printf('\n')
    printf('FILTRANDO DADOS: APRESENTAR SOMENTE PADRÕES NAO
REPETIDOS \n \n')

[nlinhasPadraoValido ncolunasPadraoValido]=size(PadraoValido);//Lê dimensão
PadraoValido

//for i=1:nlinhasPadraoValido
kkk=1;
for colunatestada=2:ncolunasPadraoValido

HaRepeticoes=0;
for j=2:ncolunasPadraoValido
    if isequal(PadraoValido(:,colunatestada),PadraoValido(:,colunatestada-1)) then // comando
isequal(a,b) compara dois vetores. Se são igual retorna %T senão %F
        HaRepeticoes=1; //Se houver algum repetido HaRepeticoes=1
    end
end
end

if HaRepeticoes==0 //Se nao há repetições, guarda o padrão:

```

```
PadraoValidoNaoRepet(:,kkk)=PadraoValido(:,colunatestada);  
kkk=kkk+1;  
end  
end  
printf('PadraoValidoNaoRepetido \n:')  
disp(PadraoValidoNaoRepet)//Escreve os padrões válidos na tela
```