



UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CAMPUS PONTA GROSSA
GERÊNCIA DE PESQUISA E PÓS-GRADUAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO
PPGEP

MAURICIO TOMELIN

**METODOLOGIA BASEADA EM DADOS HISTÓRICOS
PARA DEFINIÇÃO DE LAYOUTS EM SISTEMAS JOB-
SHOP**

PONTA GROSSA

AGOSTO - 2008

Livros Grátis

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

Milhares de livros grátis para download.

MAURICIO TOMELIN

**METODOLOGIA BASEADA EM DADOS HISTÓRICOS
PARA DEFINIÇÃO DE LAYOUTS EM SISTEMAS JOB-
SHOP**

Dissertação apresentada como requisito parcial à obtenção do título de Mestre em Engenharia de Produção, do Programa de Pós-Graduação em Engenharia de Produção, Área de Concentração: Gestão Industrial, da Gerência de Pesquisa e Pós-Graduação, do Campus Ponta Grossa, da UTFPR.

Orientador: Prof. João Carlos Colmenero, Dr.

PONTA GROSSA

AGOSTO - 2008



Universidade Tecnológica Federal do Paraná
Campus de Ponta Grossa
Gerência de Pesquisa e Pós-Graduação
**PROGRAMA DE PÓS-GRADUAÇÃO EM
ENGENHARIA DE PRODUÇÃO**



TERMO DE APROVAÇÃO

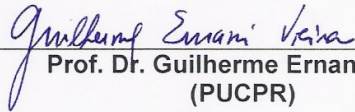
Título de Dissertação Nº 84

**METODOLOGIA BASEADA EM DADOS HISTÓRICOS PARA DEFINIÇÃO DE LAYOUTS
EM SISTEMAS JOB-SHOP**

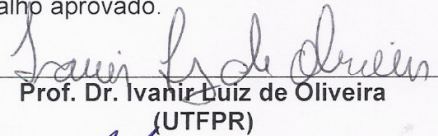
por

Mauricio Tomelim

Esta dissertação foi apresentada às **10 horas de 11 de agosto de 2008** como requisito parcial para a obtenção do título de MESTRE EM ENGENHARIA DE PRODUÇÃO, com área de concentração em Gestão Industrial, linha de pesquisa em **Gestão da Produção e Manutenção**, Programa de Pós-Graduação em Engenharia de Produção. O candidato foi argüido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

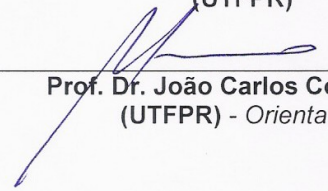


Prof. Dr. Guilherme Ernani Vieira
(PUCPR)



Prof. Dr. Ivanir Luiz de Oliveira
(UTFPR)

Prof. Dr. Kazuo Hatakeyama (UTFPR)



Prof. Dr. João Carlos Colmenero
(UTFPR) - Orientador

Visto do Coordenador:

Kazuo Hatakeyama (UTFPR)
Coordenador do PPGE

Dedico esta dissertação aos meus pais que com muito amor e compreensão souberam criar em seus filhos a garra para acreditarem que podem ser melhores e podem melhorar o ambiente que vivem. À minha esposa que sempre me apóia nos meus projetos, mesmo que às vezes parecem sem perspectivas de serem realizados, transformando os meus projetos em nossos projetos.

AGRADECIMENTOS

Agradeço a Deus pelo seu amor por nós, pela sua bondade e pela sua destra estendida sobre as nossas vidas, nos guiando e muitas vezes nos guardando.

A minha esposa Janete pelo companheirismo e apoio nos momentos de dificuldade. Por ter acreditado comigo neste sonho.

Aos meus pais Valdir e Dirce e meus sogros Nelson e Ellen pelo apoio, mesmo que distantes, sempre incentivaram e acreditaram que iríamos vencer.

Ao meu orientador por ter acreditado na possibilidade de vencermos este desafio, pela sua objetividade e determinação na busca de soluções para problemas reais.

Aos meus colegas do mestrado e da empresa, pelo grande apoio incentivo e participação nesta caminhada.

À instituição Universidade Tecnológica Federal do Paraná que se dedica à melhoria contínua na formação e crescimento profissional dos membros desta sociedade.

Ao Grupo Hübner por permitir a utilização dos dados acreditando que podemos sempre estar melhorando os processos produtivos.

RESUMO

O presente trabalho propõe uma metodologia para definição de *layout* de um *job-shop*. Comumente, os problemas de *layout* de *job-shop* são resolvidos aplicando-se o agrupamento dos recursos de produção através da estrutura celular. Esta estrutura consiste em agrupar os recursos para processar um determinado produto, com o cuidado de não agrupar recursos com a mesma função na mesma célula de manufatura. Ainda, pode-se agrupar os recursos por similaridade criando setores de recursos com a mesma função e os produtos visitam os setores para receberem o processamento necessário. Em ambos os casos acontece a distorção do objetivo do processamento dos produtos e o objetivo passa ser a utilização da combinação de recursos disponíveis. A metodologia proposta foi aplicada aos dados referentes ao período de três anos, coletados em uma empresa de usinagem. Os parâmetros utilizados foram a quantidade total de vendas, a frequência de vendas, a lucratividade, o tempo total de processamento e o tempo de espera de cada produto, gerando a classificação dos produtos. Os roteiros de produção foram comparados e em função dos parâmetros determinou-se o grau de importância de cada ligação apresentada entre os recursos baseados no grau de importância dos produtos. A comparação dos roteiros consistiu em quantificar os roteiros iguais e quantificar a diferença dos roteiros gerando roteiros parecidos. As ligações foram submetidas a um modelo matemático que determinou a posição de cada recurso no *layout*. A metodologia desenvolvida foi aplicada aos dados de uma empresa de usinagem que apresenta a característica de um *job-shop*. Os dados da empresa submetidos à metodologia permitiram a elaboração de uma sugestão de *layout* que tem como objetivo principal minimizar a distância dos recursos de processamento dos produtos.

Palavras-chave: Otimização, *layout*, modelo matemático, *job-shop*.

ABSTRACT

This paper proposes a methodology for a job-shop layout definition. Commonly, the job-shop layout problems are solved by applying the pooling of the production resources through a cellular structure. This structure is to pool the resources to process a particular product, been careful for not combine resources with the same function in the same manufacturing cell. Resources can be pooled to create sectors similarity with the same function and products visit these sectors to receive the necessary processing. In both cases the true product processing purpose is distorted and the goal becomes to use a combination of available resources. The proposed methodology was applied to a three years data period collected in a machining company. The parameters used were the total sales, the sales frequency, profitability, the total processing time and stand by time from each product, generating the products classification. The production path was compared and according to the parameters the importance degree of each link made between resources was determined based on the products importance degree. The comparison of the paths was done in ways to quantify the same routes and quantify the route differences, generating similar routes. The links were subjected to a mathematical model that determined each feature position in the layout. The developed methodology was applied to a machining company data that presents a job-shop character. The company's data submitted to the methodology enabled the development of a layout suggestion which main goal is to minimize the distance of products processing resources.

Keywords: Optimization, layout, mathematic model, job-shop.

LISTA DE FIGURAS

Figura 1 – Histograma da distribuição das classes.....	34
Figura 2 – Pseudocódigo para implementação do algoritmo de Levenshtein.....	38
Figura 3 – Gráfico dos custos apurados.	40
Figura 4 – Pseudocódigo para implementação do algoritmo de geração de roteiros classificados.	50
Figura 5 – Gráfico da distribuição de Pareto dos roteiros similares.	51
Figura 6 – Implementação do modelo no LINGO (Continuação).....	62
Figura 7 – Resultado do modelo resolvido.....	64
Figura 8 – Código fonte da implementação do algoritmo de Levenshtein em <i>Visual Basic for Applications</i>	72
Figura 9 – Código fonte da implementação do algoritmo para análise de Pareto em <i>Visual Basic for Applications</i>	74
Figura 10 – Código fonte da implementação do algoritmo para cálculo do tempo de espera.....	77

LISTA DE TABELAS

Tabela 1 – Distribuição das classes de produtos.....	35
Tabela 2 – Quantidade de ligações entre os recursos por classe.....	36
Tabela 3 – Custos Levenshtein agrupados.....	39
Tabela 4 – Produtos com custo igual a zero e com a média do valor ponderado maior do que cinco.	40
Tabela 5 – Produtos com custo igual a um e com a média do valor ponderado maior do que cinco.	43
Tabela 6 – Produtos com custo igual a dois e com a média do valor ponderado maior do que cinco.	44
Tabela 7 – Produtos com custo igual a três e com a média do valor ponderado maior do que cinco.	45
Tabela 8 – Produtos com custo igual a cinco e com a média do valor ponderado maior do que cinco.	46
Tabela 9 – Produtos com custo igual a seis e com a média do valor ponderado maior do que cinco.	46
Tabela 10 – Produtos com custo igual a sete e com a média do valor ponderado maior do que cinco.	47
Tabela 11 – Produtos com custo igual a oito e com a média do valor ponderado maior do que cinco.	48
Tabela 12 – Produtos com custo menor ou igual a cinco e com a média do valor ponderado maior ou igual a cinco.	49
Tabela 13 – Classificação dos roteiros gerados com o valor ponderado dos produtos analisados.	52
Tabela 14 – Roteiros gerados e suas seqüências.	53
Tabela 15 – Classificação dos roteiros gerados com o valor ponderado dos produtos analisados.	54

Tabela 16 – Roteiros selecionados e suas seqüências.....	54
Tabela 17 – Produtos com seu valor ponderado e o roteiro geral gerado através do algoritmo de classificação.....	55
Tabela 18 – Valores para o custo multiplicado pelo fluxo da combinação de cada equipamento.....	62
Tabela 19 – Resultados do modelo resolvido no LINGO.....	63
Tabela 20 – Posição dos recursos no <i>layout</i> original da empresa.....	65
Tabela 21 – Custo de cada ligação multiplicado pela distância na posição original.	66
Tabela 22 – Custo de cada ligação multiplicado pela distância na posição original.	66
Tabela 23 – Custo de cada ligação multiplicado pela distância no novo <i>layout</i> proposto.....	67
Tabela 24 – Custo de cada ligação multiplicado pela distância no novo <i>layout</i> proposto.....	67

LISTA DE ABREVIATURAS E SIGLAS

ABSModel	Nome do modelo que utiliza valores absolutos para otimizar o <i>layout</i>
APLP	<i>Apple's Plant Layout Procedure</i> (procedimentos para <i>layout</i> de planta de Apple)
COMSOAL	<i>Computer Method for Sequencing Operations for Assembly Lines</i> (método para seqüenciamento de operações de linhas de montagem)
CRAFT	<i>Computerized Relative Allocation of Facilities Technique</i> (técnica computacional de alocação relativa de <i>layout</i>)
LMIP	<i>Linear Mixed-Integer Programming</i> (programação linear inteira mista)
LOGIC	<i>Layout Optimization with Guillohtine Induced Cuts</i> (otimização de <i>layout</i> através de cortes induzidos)
MIP	<i>Mixed Integer Programming</i> (programação linear inteira)
MULTIPLE	<i>MULTI-floor Plant Layout Evaluation</i> (evolução de <i>layout</i> em múltiplos pisos)
QAP	<i>Quadratic Assigment Problem</i> (problema de designação quadrática)
RPLP	<i>Reed's Plant Layout Procedure</i> (procedimentos para <i>layout</i> de planta de Reed)
SFC	<i>spacefilling curves</i> (curvas de preenchimento de espaços)
SLP	<i>Muther's Systematic Layout Planning Procedure</i> (procedimento sistemático de planejamento de <i>layout</i>)

LISTA DE SÍMBOLOS

z	Valor a ser minimizado
m	Número de equipamentos ou setores
i e j	Indexadores dos setores analisados
f	Fluxo de transportes de um setor para outro
c	Custo unitário do transporte entre os setores
d	Distância entre os setores analisados
L_i^l	Limite inferior da variação do comprimento do setor
L_i^u	Limite superior da variação do comprimento do setor
W_i^l	Limite inferior da variação da largura do setor
W_i^u	Limite superior da variação da largura do setor
P_i^l	Limite inferior da variação do perímetro do setor
P_i^u	Limite superior da variação do perímetro do setor
M	Um número de alto valor
α_i	Coordenada x do centro do setor
$\alpha_{ij}^+, \alpha_{ij}^-, \beta_{ij}^+, \beta_{ij}^-$	Variáveis não negativas
β_i	Coordenada y do centro do setor
x_i^l	Coordenada x do lado esquerdo do setor
x_i^r	Coordenada x do lado direito do setor
y_i^b	Coordenada y do lado inferior do setor
y_i^t	Coordenada y do lado superior do setor
z_{ij}^x	Recebe 1 se o setor i estiver exatamente a direita do setor j e 0 caso contrário
z_{ij}^y	Recebe 1 se o setor i estiver exatamente acima do setor j e 0 caso contrário
n	Número de setores ou recursos analisados
l	Horizontal do setor ou recurso analisado
H	Dimensão horizontal da planta
x_{ij}^+, x_{ij}^-	Variáveis não negativas
V	Limite da planta para o eixo y
x_i	Coordenada x do recurso
y_i	Coordenada y do recurso

b_i	Tamanho do lado vertical do recurso
dh_i	Distância horizontal mínima necessária entre os recursos
dv_i	Distância vertical mínima necessária entre os recursos
z_{ij}	Variável utilizada para validar equações
P	Valor ponderado de classificação do produto i
Q	Quantidade total de unidades vendidas do produto i no período j de um total de n períodos
F	Quantidade de períodos (meses) que o produto i apresenta vendas
L	Lucratividade apresentada em unidades monetárias pelo produto i no período j de um total de n períodos
E	Tempo de espera total do produto em minutos
T	Tempo total de processamento do produto em minutos
α	Coeficiente aplicado sobre Q
β	Coeficiente aplicado sobre F
δ	Coeficiente aplicado sobre L
ε	Coeficiente aplicado sobre T_i
ϕ	Coeficiente aplicado sobre E_i
C_i	Capacidade produtiva simultânea da operação i em peças processadas pelo tempo da operação
R_i	Quantidade de recursos utilizados na operação i determinada pela rotina de balanceamento respeitando a quantidade de recursos disponíveis
L_i	É a quantidade de ligações de cada classe

SUMÁRIO

AGRADECIMENTOS

RESUMO

ABSTRACT

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

LISTA DE SÍMBOLOS

SUMÁRIO

1	INTRODUÇÃO	1
1.1	Objetivos do Trabalho	2
1.2	Justificativa	2
1.3	Estrutura do Trabalho	2
2	FUNDAMENTOS TEÓRICOS E REVISÃO BIBLIOGRÁFICA	4
2.1	Sistemas de Manufatura	4
2.2	<i>Layout</i>	5
2.2.1	Classificação de <i>Layout</i>	7
2.3	Otimização	9
2.3.1	Métodos	9
2.3.2	Modelos Matemáticos	11
2.4	Balanceamento de Linha	19
2.5	Matriz de Dados	20
3	METODOLOGIA	22
3.1	Coleta e Análise de Dados	22
3.2	Classificação dos dados	30
3.3	Montagem da Matriz	32
3.4	Ferramentas Computacionais	33
4	RESULTADOS E DISCUSSÃO	34
4.1	Distância de Levenshtein	36
4.2	Pareto	49
4.3	Roteiro Selecionado e Roteiros Combinados	53
4.4	Resolvendo o Modelo	56
4.5	Validação	64
5	CONCLUSÕES E RECOMENDAÇÕES	68
5.1	CONCLUSÕES	68

5.2	SUGESTÕES PARA TRABALHOS FUTUROS	69
	REFERÊNCIAS	70
	ANEXO A – CÓDIGOS FONTES	72

1 INTRODUÇÃO

O *layout* é a disposição de objetos e sua interação em uma área delimitada de modo funcional. A disposição leva em consideração a dimensão dos objetos, as dimensões necessárias para operação, manutenção, abastecimento de produtos a processar e escoamento de produtos processados. A interação considera a passagem de produtos ou informações de um objeto para outro, as alternativas de transporte, a prioridade ou importância das interações e os custos de cada interação. Os objetos a serem processados podem ser produtos ou informações. O *layout* pode ser apresentado através de desenhos com duas ou três dimensões confeccionados em softwares gráficos ou manualmente, ou ainda através de maquetes.

Os objetivos do *layout* podem variar conforme sua aplicação. Para um armazém a prioridade do *layout* é o menor percurso a ser percorrido para armazenar ou retirar uma mercadoria. Em uma empresa de serviços a prioridade é o fluxo dos clientes ou das informações. Para a indústria a prioridade é a melhor distribuição dos recursos para processamento do mix de produtos.

Durante a definição e desenvolvimento do *layout* fazem-se os ajustes necessários para atender os objetivos. A avaliação do atendimento aos objetivos propostos pode ser feita através de avaliações qualitativas ou quantitativas. A utilização de modelos matemáticos para análise de *layout* promove a precisão do *layout* em relação aos objetivos propostos.

Na definição do *layout* de produção são observados a distância dos recursos e o fluxo, que pode evitar a formação de gargalos devido a arranjos que bloqueiem ou tornem difícil o fluxo dos produtos entre os recursos. A definição do *layout* ideal torna-se difícil com o aumento da variedade dos produtos processados pela linha de produção. Quanto maior a diferença entre os produtos processados, maior será a variedade de recursos necessários para processá-los. Vários recursos diferentes na mesma linha tornam grande a variedade de produtos possíveis de produção, o que dificulta a definição do *layout* ideal ficando o dilema de otimizar o fluxo para alguns produtos em detrimento de outros, ou agrupar os recursos por similaridade não importando a seqüência dos produtos processados.

A otimização procura melhorar a utilização dos recursos disponíveis através de possibilidades de combinações que diminuam, por exemplo, o custo de movimentação. Askin (1993), Benjaafar (2002), Daita (1999), Francis (1974), Heragu (1997) e Tompkins (2003), propõem diversos métodos de otimização de *layout* por meio da qualificação das ligações dos recursos determinada pelo fluxo dos produtos propostos para a linha de produção analisada. Esta tarefa está relacionada à qualidade dos dados de entrada coletados.

Neste trabalho, foi utilizado o algoritmo de Levenshtein para classificar os roteiros de produção. O algoritmo permite definir quantitativamente a diferença de um roteiro para outro, possibilitando determinar o grau de semelhança dos roteiros de dois produtos. Através da teoria de Pareto foram determinados quais produtos e quantos produtos possuem o mesmo roteiro. Através do valor ponderado calculou-se a relevância de cada produto considerando o volume de vendas, a lucratividade, a frequência de vendas, o tempo total de processamento e o tempo total de espera. Como os produtos estão relacionados a roteiros e os roteiros determinam a seqüência de operação de cada produto, foi obtida a importância da ligação de cada par de recursos.

1.1 Objetivos do Trabalho

O principal objetivo do trabalho é a definição de metodologia baseada em dados históricos para definição do melhor *layout* para uma linha de produção que possui a característica de ser um *job-shop*. Em função das vendas apresentadas, especificar os dados relevantes para definição do *layout* e como combiná-los para determinar quantitativamente o melhor *layout*.

1.2 Justificativa

A justificativa é que o levantamento dos dados por si só não pode definir o melhor *layout*, necessitando de um modelo matemático que determine a combinação dos recursos de modo quantitativo, respeitando critérios como a não sobreposição de recursos, a alocação dos recursos dentro da área disponível, entre outros.

1.3 Estrutura do Trabalho

O presente trabalho está estruturado em cinco capítulos:

O primeiro capítulo faz a contextualização do problema de definição de *layout* em ambiente *job-shop*. Mostra a necessidade de aplicar o processo de otimização sobre a definição do *layout* e métodos sugeridos.

O segundo capítulo apresenta o referencial teórico sobre sistemas de manufatura, *layout* e sua classificação, métodos de otimização, modelos matemáticos para definição de *layout*, balanceamento de linha e organização de dados.

O terceiro capítulo apresenta a metodologia para análise do banco dos dados e as informações necessárias a sua formação. Delimita a classificação dos dados, a montagem da matriz que serve de base para resolução do modelo matemático selecionado. Mostra também as ferramentas computacionais utilizadas.

O quarto capítulo apresenta a aplicação dos métodos e os resultados dos modelos aplicados. Apresenta também os resultados da combinação dos métodos e os roteiros selecionados para serem submetidos ao modelo. Ao final é apresentada a resolução do modelo e a validação do novo *layout* com relação ao *layout* original.

O quinto capítulo apresenta as conclusões do trabalho e considerações sobre possíveis trabalhos acadêmicos futuros.

2 FUNDAMENTOS TEÓRICOS E REVISÃO BIBLIOGRÁFICA

2.1 Sistemas de Manufatura

O objetivo da manufatura é enriquecer a sociedade através da produção de produtos com funcionalidades desejáveis, esteticamente satisfatórios, não agressivos ao meio ambiente, economicamente viáveis, altamente confiáveis e de boa qualidade (ASKIN, 1993). Kalpakjian (1997) define um sistema de manufatura como o processo de transformação de materiais brutos em produtos, abrangendo desde o desenho de projeto e processamento de bens através de vários métodos e técnicas de produção. Groover (2000) define a manufatura como a aplicação de processos químicos e físicos para alterar a geometria, as propriedades e ou a aparência de um dado material para fazer partes de produtos ou produtos completos, incluindo a junção de partes na construção de produtos. O processo envolve a combinação de máquinas, ferramentas, energia e operações manuais. A manufatura é composta por uma seqüência de operações sucessivas que, em cada operação, torna o material mais próximo do seu estado final.

A manufatura é estudada sob dois aspectos relevantes, o econômico e o tecnológico. O aspecto econômico mostra a importância da manufatura na geração de riquezas através da agregação de valor aos produtos processados, ou seja, a matéria prima tem um valor, porém quando manufaturada passa a ter o valor da matéria prima mais o valor do serviço para transformá-la em outro produto que satisfaz a necessidade ou desejo de clientes. O aspecto tecnológico diz respeito ao processo de transformação dos materiais em bens ou partes de bens. A tecnologia abrange os equipamentos utilizados, a mão de obra necessária, os materiais, os produtos, eventos relativos à produção, a organização da produção, a estrutura das fábricas, a cadeia de fornecedores, entre outros detalhes. Considerando o trabalho de transformação de materiais, a palavra manufatura pode ser substituída pela palavra produção (GROOVER, 1996). A tecnologia de um sistema de produção é composta por equipamentos capazes de processar matérias primas e transformá-las em produtos que vão atender uma necessidade ou desejo. Os produtos processados visitam os setores nos quais são transformados.

A classificação de um sistema de manufatura considera os parâmetros volume de produção e variedade de produtos. O volume e a variedade dos produtos influenciam no fluxo dos produtos dentro do sistema que classifica o arranjo fabril para maximizar o aproveitamento dos equipamentos, diminuindo-se a quantidade de produtos em processo esperando para serem processados, aumentando a produtividade.

O sistema de manufatura pode ser dividido em níveis hierárquicos, sendo o topo a empresa como um todo. Logo abaixo são apresentados os setores. Os setores, por sua vez, são compostos por máquinas ou recursos. A combinação dos recursos dentro dos departamentos ou setores é definida pelo tipo de *layout* aplicado ao sistema de manufatura (ASKIN, 1993).

2.2 *Layout*

Layout é a disposição física de equipamentos que leva em consideração as dimensões dos equipamentos e os espaços necessários para sua operação e manutenção, bem como espaços reservados para corredores utilizados no abastecimento de matérias primas e escoamento dos produtos processados. Além das necessidades de movimentação de produtos, deve ser previsto o acesso a planta e serviços de apoio para a produção e para os colaboradores que operam a fábrica (TOMPKINS, 2003).

Objetivando a definição de um *layout* fabril, Groover (2000) aponta como parâmetro importante a quantidade anual produzida de cada produto definidos para a planta. O autor sugere a classificação dos volumes de produção conforme abaixo podendo haver variações arbitrárias:

- ◆ *Baixa produção, de 1 a 100 unidades anuais:* estes *layouts* apresentam grande variedade de produtos.
- ◆ *Média produção, de 100 a 10.000 unidades anuais;*
- ◆ *Alta produção, acima de 10.000:* estes *layouts* apresentam pequena ou nenhuma variação de produtos.

Ainda conforme Groover (2000) existe uma correlação inversa entre o volume de produção e a variação de produtos, sendo quanto menor o volume de produção

maior será a variação dos produtos e quanto maior o volume de produção menor será a variação dos produtos, tornando o *layout* especializado em determinados produtos.

Baseado na programação e volumes de produção, combinação de produtos (suas variações) e processos, Francis (1974) define como principais objetivos para o estudo de otimização de *layout* de uma planta, os seguintes pontos:

- ◆ Minimizar investimentos em equipamentos;
- ◆ Minimizar o tempo total de produção;
- ◆ Melhorar o aproveitamento de espaço;
- ◆ Melhorar a segurança e conforto dos colaboradores;
- ◆ Manter a flexibilidade do arranjo dos equipamentos e operações;
- ◆ Minimizar o custo de manuseio de materiais;
- ◆ Minimizar a variação de tipos de equipamentos de manuseio de materiais;
- ◆ Otimizar o processo de manufatura;
- ◆ Otimizar a estrutura organizacional.

Tompkins (2003) aponta os objetivos de estudo de *layout* abrangendo a interface com o cliente e interação da cadeia de fornecedores como:

- ◆ Promover a satisfação dos clientes através da facilidade de negociação e atendendo as suas necessidades;
- ◆ Maximizar o retorno sobre investimentos em equipamentos maximizando aproveitamento, minimizando obsolescência, maximizando participação dos colaboradores e maximizando melhorias contínuas;
- ◆ Maximizar a velocidade para responder rapidamente ao cliente;
- ◆ Reduzir custos e aumentar a lucratividade da cadeia produtiva;
- ◆ Integrar a cadeia produtiva através de parcerias e comunicação;
- ◆ Atender a missão da organização através do manuseio, controle e armazenamento de materiais;

- ◆ Utilização efetiva de pessoas, equipamentos, espaço e energia;
- ◆ Maximizar o retorno sobre investimento;
- ◆ Ser adaptável e facilitar manutenção;
- ◆ Promover a segurança e satisfação dos colaboradores.

A definição de *layout* leva em consideração os produtos a serem produzidos, o processo de produção e os volumes programados de produção (FRANCIS, 1974, GROOVER, 2000 e TOMPKINS, 2003). Estas considerações fazem com que os objetivos do *layout* se alterem de acordo com:

- ◆ *Produto*: o *layout* desenhado para atender o produto leva em consideração os equipamentos e elementos necessários para atender a produção dos produtos especificados. Este tipo de *layout* é aplicado a produtos com grande volume de produção e a planta é dedicada a este produto permitindo pequenas ou nenhuma alteração no processo produtivo;
- ◆ *Processo*: o *layout* desenhado para atender o processo leva em consideração a estrutura necessária para produzir uma gama de produtos que permitem ser processados pelos recursos disponibilizados nos processos. Este tipo de *layout* atende a uma variedade moderada de produtos e pode ser utilizado com quantidades anuais pequenas, médias ou altas;

O volume e a frequência de entregas influenciam a definição do *layout* determinando quanto e quando cada produto será produzido na planta. As informações de volume de produção são baseadas nos relatórios de marketing e em combinações provenientes de cálculos de previsão de demanda (TOMPKINS, 2003). A quantidade e a frequência de produção determinam também a classificação do *layout* em *job-shop*, produção em lotes, manufatura celular, linha de produção com fluxo contínuo ou produção em massa (GROOVER, 2000).

2.2.1 Classificação de *Layout*

A aplicação da planta classifica o *layout* em prestação de serviços, industriais e armazéns (HERAGU, 1997). No desenvolvimento e definição do *layout*, Francis (1974) divide a classificação em *layouts* para novas instalações, rearranjos de instalações e ampliações de instalações. A ocupação de espaço classifica o *layout*

em unidimensional ou multidimensional, em espaço discreto ou espaço contínuo, e ainda limitado ou ilimitado. As medidas de distância dos recursos influenciam na determinação do *layout* e podem ser retilínea e euclidiana. Com relação aos objetivos, o autor classifica em qualitativo e quantitativo. Os objetivos qualitativos tratam de parâmetros que não podem ser quantificados, ao passo que dos objetivos quantitativos estão em minimizar o custo total, minimizar o maior custo, minimizar o tempo de espera, minimizar o tempo total de lotes, entre outros.

Considerando o fluxo de produtos na planta, Askin (1993), Gopalakrishnan (2003), Groover (2000) e Tompkins (2003) classificam *layout* do seguinte modo:

- ◆ *Posição Fixa*: Utilizado em processos de manufatura em que o produto é muito grande para transitar entre as diversas etapas do processo ou postos de transformação (recursos) para ser processado. Normalmente em processos como o de construção civil, fabricação de aeronaves e navios, os materiais, equipamentos e trabalhadores são deslocados até os produtos para proceder às operações necessárias para conclusão de cada etapa do processo.
- ◆ *Layout orientado ao produto ou flow-shop*: Sistemas de produção que utilizam o posicionamento orientado ao produto possuem a característica de produzirem produtos com pequenas variações, porém que apresentam o mesmo processo produtivo ou pelo menos parecido. Esta característica afeta a variedade de produtos que a planta pode produzir. Neste sistema os produtos visitam os postos de trabalho que normalmente são arranjados em linha de produção contínua.
- ◆ *Layout orientado ao processo ou job-shop*: Enquanto a orientação ao produto limita a variedade de produtos, o posicionamento orientado ao processo promove maior variedade de produtos que possam ser produzidos na planta. Na orientação ao produto, o projeto da planta fabril deve promover a maior flexibilidade possível. Diversos casos seguem a tendência de agrupar os recursos por similaridade fazendo com que os lotes visitem diversos setores para serem processados, podendo ser utilizado um único recurso do setor visitado. Devido ao fato da grande variedade de roteiros que os produtos apresentam nesta estrutura, existe o desafio de tornar o sistema produtivo diminuindo o tempo de transporte

entre os setores e o tempo de espera em filas para o processamento no recurso programado. Nestes sistemas é comum existirem altas quantidades de produtos em processo, aumentando o tempo total de produção dos lotes. Os lotes são divisões das quantidades programadas para serem produzidas com prazos definidos que normalmente espelham pedidos de clientes.

- ◆ *Layout orientado para agrupamento tecnológico ou células:* em um ambiente de produção de diversos produtos diferentes, existe a tendência de se criar células para atender ao posicionamento orientado ao processo e assim reunindo os equipamentos em setores com similaridade de recursos.

2.3 Otimização

O correto posicionamento dos recursos de produção é uma preocupação dos especialistas e as soluções apresentadas até então não atendem a flexibilidade desejada pelos administradores das empresas (BENJAAFAR, 2002). Buscando uma alternativa precisa para a definição do *layout* de um sistema de manufatura, utilizam-se modelos matemáticos que possam determinar o correto posicionamento dos recursos objetivando a redução de custos (TOPKINS, 2003 e HERAGU, 1997). Heragu (1997) divide o problema de otimização de *layout* em dois tipos: linha simples e múltiplas linhas. A linha simples considera a disposição dos recursos apenas em um eixo vertical ou horizontal, ou seja, uma única linha podendo ter curvaturas; e as linhas múltiplas são caracterizadas pela disposição dos recursos no eixo vertical e horizontal de um plano de duas dimensões.

2.3.1 Métodos

Os métodos para desenvolvimento de *layout* são classificados de acordo com o seu ponto de partida (TOMPKINS, 2003): construção ou melhoria. A construção consiste no desenvolvimento do *layout* novo a partir de um rascunho ou uma idéia ao passo que a melhoria trabalha com um *layout* existente e visa aumentar ou rearranjar a disposição de equipamentos ou o fluxo do produto. Ambos têm o mesmo objetivo, oferecer a melhor combinação de recursos baseado nas informações coletadas.

Existem conceitos básicos em métodos desenvolvidos há vários anos, como é o caso de *Apple's Plant Layout Procedure* (APLP), *Reed's Plant Layout Procedure*

(RPLP) e *Muther's Systematic Layout Planning (SLP) Procedure*, que são utilizados em várias metodologias nos dias atuais (TOMPKINS, 2003).

Basicamente, os métodos são divididos nas seguintes etapas (TOMPKINS, 2003 e FRANCIS, 1974):

- ◆ Coleta e análise dos dados: APLP, RPLP e SLP apontam este passo como a coleta e pré-classificação dos dados e o seu interesse se restringem à: obter e analisar os dados básicos, desenhar ou detalhar o processo produtivo, obter o plano do fluxo do material e levar em consideração o plano de manuseio de material (APLP); analisar os produtos ou produto a ser produzido, determinando o fluxo do processo incluindo operações, transporte, armazenamento e inspeções, e tempos padrões para cada operação (RPLP); e obter dados dos produtos e das atividades, como fluxo de materiais e relações de atividades e preencher o diagrama de relacionamento (SLP).
- ◆ Definição de equipamentos: APLP e RPLP apontam esta etapa como sendo a definição de equipamentos e necessidades de armazenamento, detalhado da seguinte forma: calcular os requisitos de equipamentos, planejar cada estação de trabalho, selecionar os equipamentos específicos para os produtos, coordenar os agrupamentos de operações relacionadas, desenhar as inter-relações das atividades, determinar os requisitos de armazenamento, planejar os serviços e atividades auxiliares (APLP); determinar a seleção de máquinas e colaboradores e seus respectivos balanceamentos, e necessidades de manuseio de materiais (RPLP); SLP condensa esta etapa com a anterior.
- ◆ Definição e escolha do *layout*: APLP, RPLP e SLP apontam esta etapa como o fechamento do método com a sugestão de *layout* baseada nas informações dos itens anteriores: determinar os requisitos de espaço, alocar as atividades ao espaço total disponível, considerar os tipos de construções, construir o *layout* mestre, e evoluir e ajustar o *layout* checando-o com as pessoas apropriadas (APLP); preencher a tabela de planejamento de *layout* e prover a sugestão de *layout* baseada na tabela (RPLP); prover o diagrama de espaço de relacionamento, modificar as considerações, verificar as limitações práticas e desenvolver e desenhar as alternativas de *layout* (SLP).

- ◆ Ao final a aplicação e evolução do *layout*: obter aprovação, implementar o *layout* e acompanhar a sua implementação (APLP); prover futuras ampliações (RPLP); verificar e acompanhar a evolução do *layout* (SLP).

Baseados nestes procedimentos ou métodos são criados modelos que representam a realidade e permitem o aprimoramento dos *layouts* através de combinações matemáticas.

2.3.2 Modelos Matemáticos

Askin (1993) classifica os modelos de manufatura em modelos físicos e modelos matemáticos. Os modelos físicos são usados há vários anos e mostram através de desenhos, bi ou tri dimensionais a forma e disposição de construções e equipamentos, permitindo melhor visualização do ambiente proposto e de modo mais objetivo e prático do que a simples descrição textual. Os modelos matemáticos são compostos de equações matemáticas ou instruções lógicas relacionadas que tem o objetivo de representar a realidade de um dado problema (HILLIER, 2001).

Quanto às funções objetivo, Tompkins (2003) classifica os modelos em minimização da somatória dos fluxos multiplicados pela distância percorrida, similar ao problema de designação quadrática (*Quadratic Assignment Problem (QAP)*), ou então maximizar algum indicador determinado. O objetivo de minimizar o custo de cada ligação entre recursos pode ser expresso pela Equação 1:

$$\text{Minimizar } z = \sum_{i=1}^m \sum_{j=1}^m f_{ij} c_{ij} d_{ij} \quad \text{Eq. 1}$$

Onde:

z – valor a ser minimizado;

m – número de equipamentos ou setores;

i e j – indexadores dos setores analisados;

f – fluxo de transportes de um setor para outro. É a quantidade de vezes que um produto é transportado do setor i para o setor j . Pode ser um valor total de interações ou o valor por alguma unidade de tempo;

c – custo unitário do transporte entre os setores;

d – distância entre os setores analisados, normalmente é a distância retilínea entre os centros dos setores comparados.

Com o objetivo de analisar os dados relativos aos problemas dos ambientes representados pelos modelos, Tompkins (2003) apresenta alguns algoritmos propostos que trabalham com setores e apresentam sugestões de *layout*:

- ◆ O algoritmo de troca de pares proporciona a troca de pares seqüenciais considerando o custo de transporte entre os setores multiplicado pelo montante do fluxo. Este algoritmo toma como base um *layout* existente para sua ampliação, retração ou simplesmente o rearranjo dos equipamentos para melhoria do fluxo do processo. Em cada etapa o algoritmo gera várias seqüências de pares e calcula o custo total de cada seqüência e a que apresentar o menor custo é utilizada para uma nova comparação. O algoritmo segue com as comparações até encontrar o menor custo possível para os pares combinados.
- ◆ O algoritmo do método baseado em gráfico tem suas raízes na teoria dos grafos e seu principal objetivo é a adjacência dos setores. Seus dados de entrada são a importância da ligação dos setores podendo ser representada pela tabela de relacionamentos encontrada no SLP. Cada relação de setores adjacentes recebe um peso e o algoritmo gera combinações diferentes que alteram o valor da somatória dos pesos considerando na somatória somente os pesos dos setores adjacentes. O peso pode representar o custo da ligação de cada setor.
- ◆ O algoritmo CRAFT (*Computerized Relative Allocation of Facilities Technique*) inicia o processamento baseado em um *layout* existente procedendo à troca somente dos setores que sejam adjacentes ou com a mesma área e não leva em consideração o formato do setor. Depois de determinado o *layout* inicial, o algoritmo calcula o centro de cada setor e armazena em uma matriz a distância dos centros multiplicada pelo custo unitário de cada par de setores. A cada iteração o algoritmo considera a possibilidade de trocar dois ou três setores ao mesmo tempo e verifica a melhor alternativa de alteração através da redução do custo total do *layout*. Após encontrar o melhor custo, o algoritmo salva o novo *layout* e computa os novos valores. O novo *layout* gerado serve de base para

nova análise de otimização e assim sucessivamente até que seja encontrado um *layout* que todas as possíveis trocas não reduzam seu valor de custo total.

- ◆ O algoritmo de BLOCPLAN é baseado em faixas ou blocos. Seus dados de entrada podem ser a tabela de relacionamento ou a tabela “de para” que apresenta a relação dos setores e os custos de cada ligação. Inicialmente o algoritmo divide a planta em faixas e acomoda os setores nas faixas observando uma das tabelas. Após colocar os setores nas faixas é feito o cálculo do custo de cada ligação considerando a distância a ser percorrida entre cada setor, não somente os adjacentes, e assim é obtido o custo total do *layout*. A cada iteração o algoritmo busca trocar setores com o objetivo de diminuir o custo total do *layout*. Quando o algoritmo não conseguir diminuir o valor total do *layout* foi encontrado o melhor *layout*.
- ◆ O algoritmo LOGIC (*Layout Optimization with Gillohtine Induced Cuts*) utiliza a tabela “de para” como base para determinação da disposição dos setores, considerando também as dimensões da planta, podendo ser utilizado para um *layout* novo, bem como para melhoria de um *layout* existente. As operações do algoritmo consistem em executar cortes horizontais e verticais na planta e alocando os setores para cada área resultante dos cortes. Os cortes podem ser horizontais e verticais e não são sobre postos. A cada corte, a posição x ou y de cada corte é armazenada e uma árvore de relacionamento dos setores é gerada. A Equação 1 é utilizada para medir a qualidade do *layout*. Este algoritmo não se limita a áreas retangulares. Após a disposição inicial dos setores o algoritmo propõe a troca de setores de mesma área, não necessariamente a mesma forma, com o objetivo de diminuir o custo total do *layout*. Caso não seja encontrada nenhuma troca que diminua o valor total do *layout*, o *layout* é considerado otimizado.
- ◆ O MULTIPLE (*MULTI-floor Plant Layout Evaluation*) trabalha com seqüenciamento da ocupação da divisão do espaço disponibilizado em espaços menores marcando cada divisão como ocupada por um determinado setor. A seqüência de ocupação é determinada pela tabela de seqüenciamento dos setores. Utilizando-se do SFCs (*spacefilling curves*) combinado com a curva de Hilbert, o algoritmo segue ocupando as divisões do *layout* de quatro em quatro

espaços. O simples preenchimento do *layout* não garante a otimização, para isto são calculadas as distâncias dos setores e multiplicando pelo custo das ligações, gera-se um novo seqüenciamento e o *layout* é preenchido novamente pelo algoritmo. Um novo cálculo do custo do *layout* é feito e assim sucessivamente até que não haja redução de custo. Neste ponto o *layout* estará otimizado.

O modelo MIP (*Mixed Integer Programming*) propõe a representação matemática do problema de otimização de *layout* (TOMPKINS, 2003). Assumindo que todos os departamentos são retangulares e suas dimensões conhecidas podendo sofrer alterações durante a resolução, porém respeitando limites de área, o algoritmo calcula a distância retilínea dos centros, e sua função objetivo está baseada na Equação 1. Além de trabalhar com um *layout* pré-estabelecido pode também trabalhar a partir de um *layout* novo e considera apenas duas dimensões (x e y). O modelo linear é apresentado na Equação 2:

$$\begin{aligned}
 \text{Minimizar} \quad & z = \sum_i \sum_j f_{ij} c_{ij} (\alpha_{ij}^+ + \alpha_{ij}^- + \beta_{ij}^+ + \beta_{ij}^-) & \text{Eq. 2} \\
 \text{Sujeito a:} \quad & L_i^l \leq (x_i'' - x_i') \leq L_i^u \\
 & W_i^l \leq (y_i'' - y_i') \leq W_i^u \\
 & P_i^l \leq 2(x_i'' - x_i' + y_i'' - y_i') \leq P_i^u \\
 & 0 \leq x_i' \leq -x_i'' \leq B_x \\
 & 0 \leq y_i' \leq -y_i'' \leq B_y \\
 & \alpha_i = 0,5 x_i' + 0,5 x_i'' \\
 & \beta_i = 0,5 y_i' + 0,5 y_i'' \\
 & \alpha_i - \alpha_j = \alpha_{ij}^+ - \alpha_{ij}^- \\
 & \beta_i - \beta_j = \beta_{ij}^+ - \beta_{ij}^- \\
 & x_j'' \leq x_i' + M (1 - z_{ij}^x) \\
 & y_j'' \leq y_i' + M (1 - z_{ij}^y) \\
 & z_{ij}^x + z_{ji}^x + z_{ij}^y + z_{ji}^y \geq 1 \\
 & \alpha_i, \beta_i \geq 0
 \end{aligned}$$

$$x_i', x_i'', y_i', y_i'' \geq 0$$

$$\alpha_{ij}^+, \alpha_{ij}^-, \beta_{ij}^+, \beta_{ij}^- \geq 0$$

$$z_{ij}^x, z_{ij}^y \in \{0, 1\} \text{ Inteiro}$$

Onde:

f – fluxo de transportes de um setor para outro. É a quantidade de vezes que um produto é transportado do setor i para o setor j . Pode ser o valor total de interações ou a quantidade de ocorrências por alguma unidade de tempo;

c – custo unitário do transporte entre os setores;

i e j – indexadores dos setores analisados;

L_i^l – limite inferior da variação do comprimento do setor;

L_i^u – limite superior da variação do comprimento do setor;

W_i^l – limite inferior da variação da largura do setor;

W_i^u – limite superior da variação da largura do setor;

P_i^l – limite inferior da variação do perímetro do setor;

P_i^u – limite superior da variação do perímetro do setor;

M – um número de alto valor;

α_i – coordenada x do centro do setor;

$\alpha_{ij}^+, \alpha_{ij}^-, \beta_{ij}^+, \beta_{ij}^-$ – variáveis não negativas;

β_i – coordenada y do centro do setor;

x_i' – coordenada x do lado esquerdo do setor;

x_i'' – coordenada x do lado direito do setor;

y_i' – coordenada y do lado inferior do setor;

y_i'' – coordenada y do lado superior do setor;

z_{ij}^x – recebe 1 se o setor i estiver exatamente a direita do setor j e 0 caso contrário;

z_{ij}^y – recebe 1 se o setor i estiver exatamente acima do setor j e 0 caso contrário;

Heragu (1997) propõe modelos matemáticos para resolver *layouts* de linhas simples que considera a disposição dos recursos apenas em um eixo vertical ou horizontal, ou seja, uma única linha podendo ter curvaturas chamado ABSModel 1; e linhas múltiplas, caracterizadas pela disposição dos recursos no eixo vertical e horizontal de um plano de duas dimensões, chamado ABSModel 2 e ABSModel 3:

- ◆ O ABSModel 1 possui este nome porque considera a diferença absoluta das coordenadas x dos pontos direito e esquerdo de cada setor em sua função objetivo. Os setores não têm a necessidade de possuir o mesmo tamanho e como este modelo considera somente uma linha as coordenadas y são desconsideradas. A Equação 3 apresenta o modelo proposto pelo autor:

$$\begin{aligned} \text{Minimizar} \quad & z = \sum_{i=1}^{n-1} \sum_{j=i+1}^n c_{ij} f_{ij} |x_i - x_j| & \text{Eq. 3} \\ \text{Sujeito à:} \quad & |x_i - x_j| \geq \frac{1}{2}(l_i + l_j) + d_{ij} \\ & H - \frac{1}{2}l_i \geq x_i \geq \frac{1}{2}l_i \end{aligned}$$

Onde:

n – número de setores ou recursos analisados;

c – custo unitário do transporte entre os setores;

f – fluxo de transportes de um setor para outro. É a quantidade de vezes que um produto é transportado do setor i para o setor j ;

l – tamanho horizontal do setor ou recurso analisado;

d – distância mínima necessária entre os setores i e j ;

H – dimensão horizontal da planta;

i e j – indexadores dos setores analisados;

O ABSModel 1 é um modelo de programação não linear, o que torna sua solução difícil com o aumento da quantidade de recursos, por esse motivo é

apresentado o modelo linear LMIP 1 (*Linear Mixed-Integer Programming Model*) na Equação 4:

$$\begin{aligned} \text{Minimizar} \quad & z = \sum_{i=1}^{n-1} \sum_{j=i+1}^n c_{ij} f_{ij} (x_{ij}^+ - x_{ij}^-) && \text{Eq. 4} \\ \text{Sujeito à:} \quad & x_i - x_j + Mz_{ij} \geq \frac{1}{2}(l_i + l_j) + d_{ij} \\ & -(x_i - x_j) + M(1 - z_{ij}) \geq \frac{1}{2}(l_i + l_j) + d_{ij} \\ & x_{ij}^+ - x_{ij}^- = x_i - x_j \\ & x_{ij}^+, x_{ij}^- \geq 0 \\ & z_{ij} = 0 \text{ ou } 1 \\ & z_i > 0 \end{aligned}$$

Onde:

n – número de setores ou recursos analisados;

c – custo unitário do transporte entre os setores;

f – fluxo de transportes de um setor para outro. É a quantidade de vezes que um produto é transportado do setor i para o setor j ;

l – tamanho horizontal do setor ou recurso analisado;

d – distância mínima necessária entre os setores i e j ;

i e j – indexadores dos setores analisados;

x_{ij}^+, x_{ij}^- – variáveis não negativas;

M – número de valor muito alto;

- ◆ O ABSModel 2 tem a mesma base do ABSModel 1 porém este segundo considera também o eixo y para localização dos equipamentos, baseado no problema de designação quadrática (*QAP – Quadratic Assignment Problem*). Este modelo considera que as dimensões de cada setor são as mesmas, ou seja, todos os setores têm a mesma largura e o mesmo comprimento. Geralmente é utilizado para otimização de postos de trabalho de setores como de escritórios. Na Equação 5 é apresentado o modelo:

$$\begin{aligned} \text{Minimizar} \quad z &= \sum_{i=1}^{n-1} \sum_{j=i+1}^n c_{ij} f_{ij} (|x_i - x_j| + |y_i - y_j|) & \text{Eq. 5} \\ \text{Sujeito à:} \quad & |x_i - x_j| + |y_i - y_j| \geq 1 \\ & x_i, y_i = \text{inteiro} \\ & H \geq x_i \geq 1 \\ & V \geq y_i \geq 1 \end{aligned}$$

Onde:

n – número de setores ou recursos analisados;

c – custo unitário do transporte entre os setores;

f – fluxo de transportes de um setor para outro. É a quantidade de vezes que um produto é transportado do setor i para o setor j ;

i e j – indexadores dos setores analisados;

x_i – coordenada x do recurso;

y_i – coordenada y do recurso;

H – limite da planta para o eixo x ;

V – limite da planta para o eixo y ;

- ◆ O ABSModel 3 tem a mesma base dos outros dois modelos, porém este último além de considerar os eixos x e y , e considera setores de dimensões ou de áreas diferentes. A função objetivo do ABSModel 3 é idêntica a função objetivo do ABSModel 2, sendo que os modelos diferem nas restrições. Na Equação 6 é apresentado o modelo:

$$\begin{aligned} \text{Minimizar} \quad z &= \sum_{i=1}^{n-1} \sum_{j=i+1}^n c_{ij} f_{ij} (|x_i - x_j| + |y_i - y_j|) & \text{Eq. 6} \\ \text{Sujeito à:} \quad & |x_i - x_j| + Mz_{ij} \geq \frac{1}{2}(l_i + l_j) + dh_{ij} \\ & |y_i - y_j| + M(1 - z_{ij}) \geq \frac{1}{2}(b_i + b_j) + dv_{ij} \\ & z_{ij}(1 - z_{ij}) = 0 \end{aligned}$$

Onde:

n – número de setores ou recursos analisados;

c – custo unitário do transporte entre os setores;

f – fluxo de transportes de um setor para outro. É a quantidade de vezes que um produto é transportado do setor i para o setor j ;

i e j – indexadores dos setores analisados;

x_i – coordenada x do centro do recurso;

y_i – coordenada y do centro do recurso;

l_i – tamanho do lado horizontal do recurso;

b_i – tamanho do lado vertical do recurso;

dh_i – distância horizontal mínima necessária entre os recursos;

dv_i – distância vertical mínima necessária entre os recursos;

z_{ij} – variável utilizada para fazer com que as restrições sejam válidas;

M – número de valor muito alto;

2.4 Balanceamento de Linha

O balanceamento atua sobre a linha de produção, modificando a seqüência das estações, desde que seja possível esta alteração, alterando o número de equipamentos associados a cada estação, ou colocando operações em paralelo. Os principais objetivos do balanceamento segundo Scholl (2006) são: aumentar a quantidade de produtos produzidos ao mesmo tempo no final da linha diminuindo o tempo de ciclo de produção; diminuir o tempo total de produção; diminuir o tempo de espera dos produtos nas estações; diminuir o tempo de espera das estações para processar os produtos da linha; diminuir a quantidade de recursos necessários para o volume de produção, combinando operações e colaboradores.

O balanceamento de uma linha de produção é um problema combinatório (TONGE, 1960 e BAYBARS, 1986). Através de combinações e alteração de parâmetros definidos, o balanceamento tenta nivelar o processo de produção evitando paradas e

aumentando o aproveitamento dos recursos disponíveis. O balanceamento interfere no *layout* da disposição dos equipamentos de produção, bem como na disposição de estoque de matérias primas e materiais acabados, prontos para serem entregues aos clientes.

O correto balanceamento da linha de produção faz parte da base das decisões gerenciais de produção e pode ser aplicado no processo de construção de uma linha de produção ou no processo de ampliação de uma linha existente (SCHOLL, 2006).

O balanceamento de um ambiente de produção depende de dados do processamento dos produtos. Os dados são: tempo de cada operação, tempo de processamento, seqüência das operações, tempo total do produto na linha (somatório dos tempos de processamento e espera) e recursos disponíveis para processamento. A seqüência de processamento dos produtos é predeterminada e os dados de processamento do balanceamento são previamente conhecidos (BAYBARS, 1986 e SCHOLL, 2006).

O objetivo, segundo Boysen (2006) e Baybars (1986), classifica os problemas de balanceamento da seguinte forma: (i) objetivando minimizar o tempo de espera, minimizando o número de estações alocadas quando o tempo de ciclo é determinado; (ii) minimizar o tempo de ciclo diminuindo o tempo de espera, quando o número de estações é determinado; (iii) maximizar a eficiência do arranjo que é dado por tempo total das operações (tarefas) dividido pelo número de estações multiplicado pelo tempo de ciclo, minimizando o tempo de espera podendo alterar a quantidade de estações ou tempo de ciclo; (iv) dados o tempo de ciclo e a quantidade de estações um balanceamento praticável é encontrado.

2.5 Matriz de Dados

A definição do *layout* depende do relacionamento entre os recursos ou setores disponíveis, determinados pelos produtos que serão processados. Técnicas como o SLP (*Systematic Layout Planning*) são mostradas por Francis (1974), Askin (1993) e Tompkins (2003) para a obtenção e organização dos dados referentes ao grau de importância de cada relação entre setores e equipamentos. Para poder determinar as relações, o primeiro levantamento consiste nos objetivos da planta ou linha de produção. Os objetivos compreendem os produtos a serem processados, seus

volumes e sua frequência, e para atender os objetivos existem recursos combinados, os recursos são equipamentos e pessoas que processam operações manuais. A combinação dos recursos depende de dados como os limites da planta, as dimensões dos equipamentos e postos de trabalho, fluxo dos materiais, o grau de importância da relação dos equipamentos e recursos, combinados para atender os objetivos propostos. A combinação dos recursos é organizada em uma tabela de relacionamento com a informação do seu respectivo grau de importância. O diagrama de espaço é construído utilizando os dados das ligações dos recursos e suas dimensões, e então são geradas sugestões de *layout* que serão analisadas e, sob a avaliação de especialistas, uma das opções é selecionada e implantada.

Outra opção de análise é proposta por Daita (1999) em algoritmos de análise de fluxo de produtos. Os algoritmos têm como base os dados dos roteiros dos produtos a serem produzidos, que determinam a seqüência da utilização dos recursos, frequência de visitas dos produtos aos recursos entre outros dados. O principal objetivo dos algoritmos é determinar uma seqüência de recursos que pretende atender o maior número possível de roteiros de produtos minimizando o tempo de deslocamento entre os recursos. Após a definição do fluxo padrão, define-se a disposição dos recursos através de algum método de disposição dos equipamentos no espaço disponibilizado.

Os autores mostram que os principais dados para definição do *layout* são: dimensões da planta, produtos a serem produzidos, recursos disponíveis para processamento dos produtos, seqüência de utilização dos recursos por cada produto, volume e frequência de produção de cada produto e custo de produção unitário de cada produto ou outra unidade de medida monetária que classifique o produto. Benjaafar (2002) discute os objetivos dos estudos de *layout* e propõem uma re-adequação para atender a sistemas de manufatura que apresentam grande variedade de produtos e volumes de produção. O autor afirma que os dados acima mencionados devem ser acrescidos dos custos da alteração do *layout* provocada pela alteração no mix de produtos processados.

3 METODOLOGIA

Os códigos dos produtos, dos recursos e os valores monetários foram normalizados para descaracterizar qualquer produto ou processo. Os dados coletados são referentes aos produtos que apresentaram vendas durante o período. Com relação às vendas considerou-se o volume total de vendas, a frequência mensal e a lucratividade apresentada. Os produtos considerados possuem roteiros de produção que determinam sua seqüência de operações e quais recursos são utilizados. Através dos roteiros foi possível apurar os tempos de operação que foram utilizados para determinar o tempo total de utilização dos recursos e os tempos de espera que surgem durante o processo devido aos processos gargalos. Estes dados foram utilizados para determinar a relevância dos produtos dentro do conjunto de produtos considerados. Após a categorização dos produtos agrupou-se os roteiros determinando a importância de cada ligação que foi encontrada nos roteiros. Outros dados como dimensão dos recursos também foi coletada e fez parte dos cálculos para determinação do *layout*.

Os dados utilizados neste trabalho para validação da metodologia são referentes às vendas e engenharia de processos de uma empresa usinagem da região de Curitiba, Paraná, compreendendo o período de agosto de 2003 a agosto de 2006.

3.1 Coleta e Análise de Dados

A coleta de dados foi realizada sobre os pedidos de vendas que compreendeu um período relevante. O período estudado abrangeu diversas fases sazonais com o objetivo de determinar a demanda geral dos produtos. Foram consideradas as demandas de vendas de produtos que apresentaram vendas durante o período disponível de dados, com isso foram analisadas as quantidades de vendas e seus prazos de entrega. Para a análise seguiu-se os seguintes procedimentos:

- ◆ Os dados utilizados foram: a quantidade, o preço unitário, a data de entrega e o código do produto;
- ◆ A cada produto foi atribuído um valor hipotético ao preço de venda e a margem de lucro;

- ◆ Para o cálculo do lucro de cada produto definiu-se um percentual de lucratividade geral o qual foi multiplicado pelo preço de cada produto em cada período;
- ◆ As quantidades de produtos programadas para entrega em cada mês foram totalizadas por produto e por mês, gerando o volume mensal de produtos vendidos;
- ◆ Não foi considerado se os pedidos foram atendidos com atraso ou em dia, utilizando-se a data de entrega desejada pelo cliente registrada em cada item de pedido para totalizar as quantidades de vendas mensais;
- ◆ Cada produto estava associado a um roteiro de produção e os dados dos roteiros fizeram parte da coleta.

A seleção dos dados baseou-se nas informações obtidas na empresa, sendo relativos ao cadastro de produto, pedidos de vendas, roteiros de produção, e recursos de produção. As informações das vendas de cada produto foram compiladas com o objetivo de obter a relevância individual dos produtos de modo a considerar os parâmetros:

- ◆ Quantidade total vendida foi considerada para análise do montante total de vendas e pode levar a considerar um produto melhor do que outro, quanto maior o montante melhor foi a venda, e pode justificar a construção de uma linha de produção exclusiva;
- ◆ Frequência de vendas foi considerada para análise da continuidade das vendas, podendo mostrar a continuidade ou o fim de vida útil do produto, além de mostrar fases sazonais quando associada a períodos de tempo específicos;
- ◆ Lucratividade total apresentada foi um dos parâmetros principais, e mostra o retorno financeiro apresentado pelo produto;
- ◆ Tempo total de operação baseado em cada operação determinada no processo de produção. Associado aos outros parâmetros pode mostrar a ocupação fabril versus a lucratividade e frequência;
- ◆ Tempo total de espera mostra problemas de gargalos no processo produtivo. Os gargalos podem ser provocados por ineficiência ou maquinário obsoleto. Combinado com o tempo total de operação pode mostrar que um produto,

mesmo apresentando pouco tempo de operação pode ser problemático por causa da espera apresentada.

Com base nestas informações calculou-se o valor ponderado para classificar cada produto. O valor ponderado é resultado de uma equação que permite a alteração da relevância dos parâmetros que foram considerados para este cálculo. A classificação da relevância dos produtos para a análise foi obtida através da Equação 7:

$$P_i = \alpha \sum_j^n Q + \beta F + \delta \sum_j^n L + \varepsilon T - \phi E \quad \text{Eq. 7}$$

Onde:

P – valor ponderado de classificação do produto i ;

Q – quantidade total de unidades vendidas do produto i no período j de um total de n períodos;

F – quantidade de períodos (meses) que o produto i apresenta vendas;

L – lucratividade apresentada em unidades monetárias pelo produto i no período j de um total de n períodos;

E – tempo de espera total do produto em minutos;

T – tempo total de processamento do produto em minutos;

α – coeficiente aplicado sobre Q ;

β – coeficiente aplicado sobre F ;

δ – coeficiente aplicado sobre L ;

ε – coeficiente aplicado sobre T_i ;

ϕ – coeficiente aplicado sobre E_i .

Os coeficientes (α , β , δ , ε , ϕ) aplicados na Equação 7, os quais são constantes, determinaram como os dados apurados influenciaram na classificação dos produtos para análise, alterando a interferência da quantidade total (Q), da freqüência de períodos (F), da lucratividade (L), do tempo total (T) e do tempo total de espera (E). Desta forma puderam-se analisar os produtos em relação a sua freqüência de vendas, lucratividade, quantidade total de vendas, tempo total de operação ou tempo

total de espera. A Equação 8 apresenta a restrição com relação aos coeficientes em que sua somatória deve ser igual a um.

$$\alpha + \beta + \delta + \varepsilon + \phi = 1 \quad \text{Eq. 8}$$

Alterando-se o valor de cada coeficiente pôde-se fazer com que um dos valores encontrados tivesse um maior peso sobre a classificação geral dos produtos fazendo com que o seu grau de importância fosse modificado.

Para o balanceamento da produção, por se tratar de um *job-shop*, buscou-se algum método que pudesse resolver este problema, porém os métodos estudados atendem em parte o problema proposto. O algoritmo de Johnson para duas máquinas (ASKIN, 1993), não pode ser aplicado por possuir a restrição de não retorno do produto a um recurso pelo qual o produto já tenha passado. A metodologia COMSOAL (ASKIN, 1993 e CHUTIMA, 2004), é utilizada para balanceamento de linhas de produção, determinando a melhor seqüência das operações para o processamento de um determinado produto. Esta metodologia estudada foi implementada, porém não pode ser aplicada aos dados deste estudo porque a seqüência de produção é definida pela área de engenharia sendo que as operações são interdependentes e referenciais entre si, ou seja, a operação 1 é referência para a operação 2, sendo necessário a troca não só de maquinário, mas também de dispositivos de fixação para poder proceder a alteração de seqüência. Esta é uma das principais características do produto usinado, sendo que a usinagem depende das referências estabelecidas em cada etapa do processo, esta seqüência não pode ser alterada. Diante destas condições balanceou-se a linha de produção através da alocação de mais recursos para cada operação, quando estes estavam disponíveis, ou diminuição destes em suas operações. Esta forma de balanceamento buscou tornar o fluxo de produção mais contínuo possível, diminuindo as esperas de produtos ou recursos durante a produção dos itens. Para o algoritmo proposto é necessário destacar as seguintes premissas:

- ◆ O tempo de espera poderia ser do recurso esperando por produtos para processar, ou o produto esperando para ser processado;
- ◆ Cada operação foi responsável pela sua fila de espera. A espera dependia do processamento da operação predecessora, quando existir, porém no momento

da análise da operação era disparada a diminuição ou aumento de recursos da operação antecessora, quando necessário;

- ◆ Não foram considerados os tempos de *set-up*, somente os tempos de operação foram considerados como tempo normal de operação, devido ao fato do tempo de *set-up* ser aplicado no cálculo de carga máquina e não durante o tempo normal de operação;
- ◆ O balanceamento dependeu da quantidade de recursos disponíveis para as operações de fabricação analisadas pelo algoritmo;
- ◆ Os recursos cadastrados na estrutura da produção da empresa foram considerados como disponíveis a todo e qualquer momento;
- ◆ Não foi considerada carga de utilização dos recursos, pois o objetivo do algoritmo foi tornar o fluxo de produção o mais contínuo possível e a disponibilidade do recurso, devido ao excesso de carga, ou eventual manutenção, não foi considerada no escopo do algoritmo;
- ◆ No desenvolvimento não se considerou o volume de pedidos ou demanda de produção, apenas os tempos de produção e a quantidade dos recursos ditos como necessários para cada operação dos produtos selecionados.

A tabela de balanceamento foi composta pelas seguintes colunas:

- ◆ *CodEmp*: código da empresa a qual pertence o processo produtivo analisado;
- ◆ *CodRot*: código do roteiro de produção ligado ao produto analisado;
- ◆ *CodEtg*: código do estágio de produção. É um agrupamento de operações dentro do processo produtivo;
- ◆ *SfxEtr*: sufixo do estágio para roteiros alternativos. Utilizado para organização de opções possíveis de estágios de produção;
- ◆ *SeqRot*: seqüência operacional do roteiro de produção. Mantendo a seqüência da operação dentro do estágio;
- ◆ *SfxSeq*: sufixo da operação para roteiros alternativos. Utilizado para seqüências opcionais de operações dentro de cada estágio;

- ◆ *CodAlt*: código alternativo do produto em análise. Estas colunas, até este campo, compõem a chave de acesso aos registros da tabela. A chave é utilizada para acesso ao registro, ou relacionamento com outras tabelas;
- ◆ *TipEsp*: tipo de espera. Pode conter os valores: “Produto” para quando o produto esperar na operação para ser processado; “Recurso” para quando o recurso esperar para processar algum produto; e “Ninguém” quando nem o recurso nem o produto esperarem;
- ◆ *EspTmp*: Tempo de espera; se for positivo quem espera é o produto, se for negativo quem espera é o recurso e se for zero, ninguém espera;
- ◆ *EspAbs*: Valor absoluto do tempo de espera. Campo a ser totalizado para análise do produto;
- ◆ *QtdCre*: Quantidade de recursos utilizados na operação, determinado pelo balanceamento;
- ◆ *QtdTmp*: Quantidade de produtos processados relativos a capacidade produtiva simultânea e a quantidade de recursos disponíveis. Informação calculada durante o processo de balanceamento;
- ◆ *CapSmt*: Capacidade produtiva simultânea. Quantos produtos podem ser processados no tempo determinado. Esta informação está predefinida na base de dados;
- ◆ *TmpPrp*: Tempo proporcional necessário para realizar a operação. É relativo à capacidade produtiva simultânea. Esta informação está predefinida na base de dados;
- ◆ *CodCre*: Código do centro de recurso. O centro de recurso pode ser um equipamento específico, um grupo de equipamentos que compõem uma célula ou um agrupamento de colaboradores que apresentam similaridade na sua função. Esta informação é determinada pela administração da produção;
- ◆ *LimCap*: Limitante de capacidade (pessoas ou máquinas). Esta informação é determinada pela administração da produção, determinando se a operação é limitada pelos equipamentos utilizados ou por pessoas;

- ◆ *CreDsp*: Quantidade de centros de recursos disponíveis (utilizados apenas para máquinas). Informação determinada pela administração da produção e registrada no cadastro de centro de recursos;
- ◆ *BalCnf*: Identifica se a seqüência está balanceada ou não.

Para o balanceamento foi desenvolvido o seguinte algoritmo:

- ◆ Apagar todos os registros da tabela de balanceamento;
- ◆ Carregar a tabela de balanceamento com os dados de cada operação de cada um dos produtos selecionados e colocando em cada linha a quantidade máxima de recursos disponíveis para a operação e o cálculo da espera referente a cada operação;
- ◆ Para cada operação de cada produto que está na tabela de balanceamento, diminuir ou aumentar o número de recursos utilizados (dependendo da disponibilidade dos recursos) com o objetivo de diminuir o tempo de espera e calcular o tempo de espera em cada alteração;

O tempo de espera foi calculado do seguinte modo (os passos abaixo apresentados são mutuamente exclusivos):

- ◆ Se a capacidade produtiva simultânea, a quantidade de recursos, a quantidade de tempo necessário para a operação atual são todos iguais a operação anterior, nem o produto nem o recurso espera;
- ◆ Se a quantidade de recursos da operação atual for diferente da quantidade de recursos da operação anterior utilizar a Equação 9 para calcular o tempo de espera. Se o resultado for negativo (menor que zero) o recurso espera, se o resultado for positivo o produto espera, se o resultado for igual a zero ninguém espera;

$$E_i = (T_i - ((C_i * T_{i-1}) / (C_{i-1} / R_{i-1}))) * (R_i - R_{i-1}) \quad \text{Eq. 9}$$

- ◆ Se a capacidade produtiva simultânea atual for maior do que capacidade produtiva simultânea da operação anterior e a quantidade de recursos da operação atual forem iguais à quantidade de recursos da operação anterior, utilizar a Equação 10 para calcular o tempo de espera; se o resultado for

negativo, o recurso espera; se for positivo, o produto espera; caso seja igual à zero, ninguém espera;

$$E_i = T_i - T_{i-1} \quad \text{Eq. 10}$$

- ◆ Se a capacidade produtiva simultânea atual for igual à capacidade produtiva simultânea anterior, utilizar a Equação 11 para calcular o tempo de espera. Para um resultado negativo, o recurso espera, para um resultado positivo o produto espera, sendo igual à zero, ninguém espera;

$$E_i = T_i - ((C_i * T_{i-1}) / (C_{i-1} / R_{i-1})) \quad \text{Eq. 11}$$

- ◆ Se a capacidade produtiva simultânea atual for menor do que a capacidade produtiva anterior, e o tempo de operação atual for menor que o tempo de operação anterior, utilizar a Equação 12 para calcular o tempo de espera. Neste caso em especial, tanto o recurso como o produto esperam, porém somente considerar o que esperar mais tempo, o produto ou o recurso;

$$E_i = T_{i-1} - (T_i * C_{i-1}) \quad \text{Eq. 12}$$

- ◆ Em última exceção às outras condições, calcular o tempo de espera através da Equação 13. Se o resultado for negativo o recurso espera, se for positivo, o produto espera e se for igual à zero, ninguém espera. A informação de quem espera, e o tempo de espera é armazenado na tabela de balanceamento.

$$E_i = T_i - ((C_i * T_{i-1}) / (C_{i-1} / R_{i-1})) \quad \text{Eq. 13}$$

Para todas as equações utiliza-se a seguinte notação:

E_i – tempo de espera em minutos da operação i ;

T_i – tempo de operação em minutos da operação i ;

C_i – capacidade produtiva simultânea da operação i em peças processadas no tempo da operação;

R_i – quantidade de recursos utilizados na operação i determinada pela rotina de balanceamento respeitando a quantidade de recursos disponíveis.

Após ter determinado o tempo de espera para cada operação de cada produto armazenado da na tabela de balanceamento, foi analisada novamente cada

operação com o objetivo de minimizar o tempo de espera em cada uma das operações.

3.2 Classificação dos dados

Após o cálculo do valor ponderado foram definidas dez classes (conforme fórmula de Struges) que categorizaram cada um dos produtos. Quanto maior o valor ponderado maior é a importância do produto na análise e maior é a classe, tornando a classe dez a mais importante e a classe um a menos importante. Com a categorização dos produtos em suas classes, foi encontrada a semelhança dos roteiros apresentados por cada produto. Desta forma, foram encontrados os produtos que utilizam os mesmos recursos para serem produzidos, possibilitando a geração de roteiros que atendam mais de um produto, mesmo que no cadastro do produto esteja cadastrado um roteiro individualizado.

Primeiramente foram consideradas as classes com menor número de produtos, foi desenvolvida a análise e então se estendeu para os demais intervalos observando os resultados para proceder possíveis ajustes no desenvolvimento do estudo. O gráfico do histograma foi gerado, e uma tabela que mostra os dados de origem do gráfico, que foram os intervalos de classes definidos para os dados obtidos. A primeira coluna mostra a descrição dos intervalos e a segunda coluna mostra a quantidade de produtos que foram agrupados naquele intervalo.

Os roteiros agrupados foram classificados de acordo com a quantidade de produtos em cada valor ponderado. Esta classificação permitiu agrupar os roteiros de acordo com o valor ponderado definido para os produtos. Seguindo a classificação dos roteiros com o valor ponderado, foi determinado qual é o roteiro mais importante dentre os roteiros encontrados. Cada roteiro possui a informação de quais recursos foram utilizados e qual é a seqüência em que foram utilizados. A partir dos roteiros, foram analisadas as ligações dos recursos para determinar a sua relevância. Durante a classificação, foi observado que não ocorresse repetição das combinações, ou seja, as ligações consideradas que possuíssem recursos iguais com cadastros diferentes, foram agrupados e seus valores somados. Assim foi gerada a classificação de cada ligação encontrada, gerando a matriz com a informação de peso das ligações dos recursos.

Ao combinar as ligações organizadamente e com o objetivo de facilitar a posterior utilização dos dados, foram desenvolvidos alguns algoritmos que atuam na combinação dos recursos. Os algoritmos estão descritos a seguir.

O primeiro algoritmo desenvolvido foi o algoritmo de Levenshtein que calculou a distância, ou quanto uma palavra é diferente da outra. Desenvolvido por Vladimir Levenshtein em 1965, foi criado para medir a diferença entre duas palavras. Atualmente, o algoritmo é comumente utilizado em corretores ortográficos. O algoritmo consiste em criar uma matriz com m linhas equivalentes à quantidade de caracteres da primeira palavra e n colunas equivalentes à quantidade de caracteres da segunda palavra. A primeira linha com coluna zero recebe o número relativo à sua posição m . E a primeira coluna com linha zero recebe o número relativo à sua posição n . Após a carga inicial da matriz inicia a comparação de cada caractere das duas palavras. A comparação acontece com cada caractere da primeira palavra sendo comparado com cada caractere da segunda. Quando acontece uma inserção, o posicionamento da matriz desloca-se na horizontal, ao passo que quando acontece uma exclusão, o deslocamento acontece na vertical. O algoritmo procede à análise até chegar à última posição da matriz ($m \times n$). A última posição da matriz armazena o custo da comparação entre as duas palavras (LEVENSHTEIN, 2007). O custo pode ser multiplicado por algum fator a ser definido de acordo com a necessidade da comparação. O algoritmo de Levenshtein foi utilizado para classificar os roteiros de produção dos produtos, as operações dos roteiros substituíram os caracteres das palavras permitindo calcular a diferença dos roteiros agrupando-se assim os produtos em roteiros similares. Neste agrupamento não foram considerados os tempos, apenas a seqüência de operação. O tempo não foi relevante porque o objetivo não era balancear a linha e sim aproximar os recursos de acordo com a relevância da ligação.

Os produtos foram classificados de acordo com os seus roteiros para encontrar a particularidade dos roteiros que, baseado na lei de Pareto para a produção, mostra que oitenta por cento do volume de produção corresponde a quinze por cento da linha de produtos. Em contra partida, quinze por cento do volume de produção equivale aos itens com maior volume de produção e oitenta e cinco por cento equivale a *job-shop* (TOMPKINS, 2003). Esta análise foi importante para verificar a

existência da característica de *job-shop* na produção estudada, pois podem existir padrões de seqüenciamento que não são percebidos no cotidiano da rotina de definição do processo para usinagem dos produtos.

3.3 Montagem da Matriz

O objetivo final foi a utilização de um modelo matemático que determinasse a posição de cada recurso dentro do espaço da planta da fábrica. O modelo considerou o custo de cada ligação, o volume de fluxo entre cada par de recursos e as dimensões dos recursos. Para o custo da ligação foi considerada a quantidade de ligações em cada uma das classes definidas no valor ponderado dos produtos multiplicados por um coeficiente definido. Assim um par de recursos que possua ligação com classe dez tem um custo maior do que um par de recursos que possua ligação com classe um. Para o fluxo, por sua vez, foi considerado o valor total das ligações de cada par de recursos analisado. As dimensões dos recursos foram coletadas da planta da empresa e consideraram as distâncias de segurança e operação necessárias, individualmente para cada recurso. O comprimento de cada recurso foi dividido por dois, encontrando-se a posição x do centro do recurso que serviu de entrada ao modelo. Na resolução do modelo, não foram consideradas as ligações de um recurso para ele mesmo e a matriz foi organizada de forma a considerar a ligação do recurso de menor código para o maior, e em caso de ligação contrária os valores encontrados foram somados a ligação original formando uma matriz diagonal superior. As ligações eram relativas a dois recursos que participam de um roteiro para um produto. Dentro do roteiro foi especificado qual é a seqüência da passagem do produto pelos recursos, formando as ligações. Para um produto que pertencesse à classe 10 e apresentou uma ligação de um recurso i com um recurso j , a ligação 10 recebeu uma unidade. O custo de cada ligação é determinado pela Equação 14:

$$c = \sum_{i=1}^{i=10} L_i * i \quad \text{Eq. 14}$$

Onde:

L_i – é a quantidade de ligações de cada classe i sendo $i = (1,2,3,4,5,6,7,8,9,10)$.

A Equação 14 determinou que a quantidade de ligações da classe 10 teve um peso para custo maior do que a quantidade de ligações da classe 1. Ou seja, quanto maior a classe, maior foi o seu peso em relação às classes menores. O modelo priorizou a aproximação das ligações das classes maiores que apresentaram maior custo.

3.4 Ferramentas Computacionais

Os dados foram extraídos de uma instância de banco de dados *Oracle* e armazenados em arquivo do *Microsoft Access*, os algoritmos foram implementados em *Visual Basic for Applications* no *Microsoft Access*. Os gráficos foram construídos utilizando o *Microsoft Excel*. A ferramenta utilizada para resolver o modelo foi o *LINGO* da empresa *Lindo Systems Inc* na versão 11.0.

4 RESULTADOS E DISCUSSÃO

Os dados coletados foram calculados e o valor ponderado foi associado para cada produto. Foi Utilizada a metodologia apresentada na coleta de dados, os produtos foram classificados em intervalos de 21425 no valor ponderado encontrado, gerando o gráfico apresentado na Figura 1, o histograma da distribuição das classes do valor ponderado.

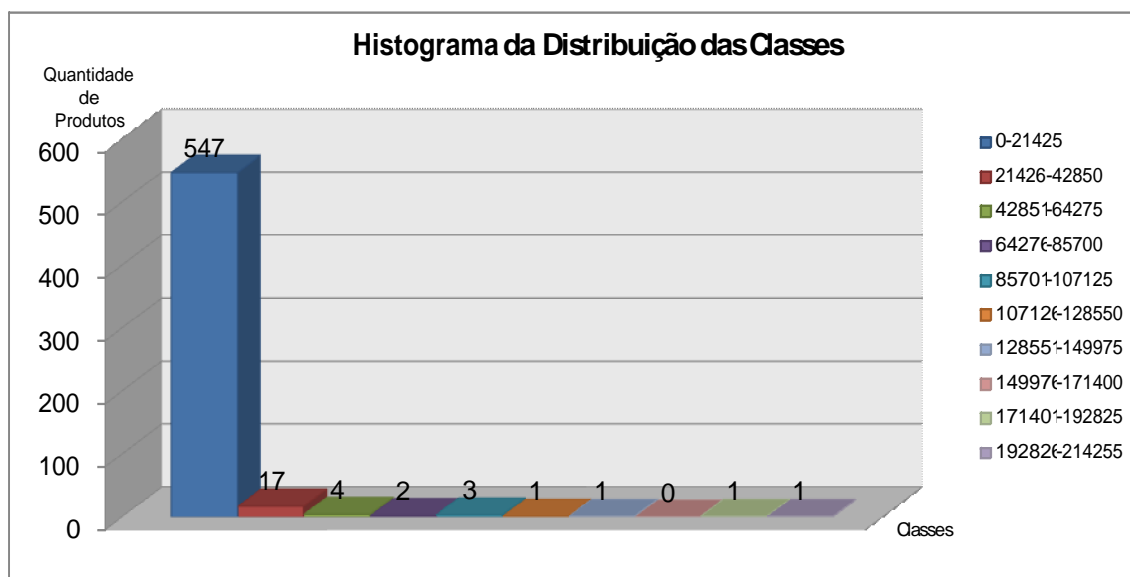


Figura 1 – Histograma da distribuição das classes.

Com a classificação dos produtos através de um valor ponderado, aplicou-se a determinação do intervalo das classes que foram apresentadas no histograma. O intervalo de classes é o conjunto de observações apresentadas de forma contínua sem sobreposição de intervalos de tal forma que cada valor do conjunto possa ser alocado em apenas um dos intervalos. Uma forma de determinação do intervalo de classes é a regra empírica, que consiste em uma tabela que determina qual é o intervalo a ser utilizado. No presente trabalho utilizou-se a fórmula de Struges na determinação dos intervalos de classe que apresentou o resultado de 21425 para cada classe.

Os dados de origem do gráfico estão apresentados na Tabela 1. A primeira coluna mostra a classe de cada intervalo, a segunda coluna mostra a descrição dos

intervalos e a terceira coluna mostra a quantidade de produtos que se encontra no intervalo.

Tabela 1 – Distribuição das classes de produtos.

Classe	Intervalo	Quantidade
1	0-21425	547
2	21426-42850	17
3	42851-64275	4
4	64276-85700	2
5	85701-107125	3
6	107126-128550	1
7	128551-149975	1
8	149976-171400	0
9	171401-192825	1
10	192826-214255	1

A classificação dos produtos mostra a concentração de 94% na classe 1 que é a de menor valor ponderado. Esta classificação foi utilizada para a análise das ligações dos recursos, encontrada nos roteiros dos produtos. Não foi possível estabelecer um padrão a partir desta tabela isolada, porque seriam considerados como relevantes diversos produtos que apresentaram o volume de produção, frequência e lucratividade baixas.

Analisando o fluxo dos produtos entre os recursos cadastrados, para cada operação de cada produto, foi desenvolvida uma forma de medir a relevância de cada ligação. Fluxos de produtos, dados ou pessoas podem ser medidos qualitativamente e quantitativamente (TOMPKINS, 2003), e para o presente estudo foram utilizadas as classes combinadas com a quantidade de ligações como forma de medição da relevância do fluxo de entre os recursos. A cada produto foi associado um valor ponderado e cada produto possui um roteiro que especificou a seqüência dos recursos visitados, o que permitiu a classificação das ligações dos recursos com o valor ponderado do produto. Na Tabela 2 foram dispostas as relevâncias do fluxo de produtos entre os recursos. As colunas contêm o recurso de origem e o de destino, o total de ligações para cada combinação de recursos e a quantidade de ligações para

cada classe. Os registros contidos nesta tabela são referentes às primeiras dez ligações encontradas de um total de duzentas e sessenta e quatro ligações. Para classificar as ligações foi construído um algoritmo que buscou o recurso de origem e destino especificado nos roteiros de produção e computou em cada coluna a quantidade de ligações total e referente a cada classe definida na coleta de dados.

Tabela 2 – Número de ligações entre os recursos por classe.

Recursos		Classe										
Orig.	Dest.	Total	1	2	3	4	5	6	7	8	9	10
3	37	1	1	0	0	0	0	0	0	0	0	0
3	66	5	5	0	0	0	0	0	0	0	0	0
4	4	1	0	0	1	0	0	0	0	0	0	0
4	37	2	2	0	0	0	0	0	0	0	0	0
4	66	42	42	1	0	0	0	0	0	0	0	0
4	81	6	6	0	0	0	0	0	0	0	0	0
4	118	18	15	2	0	0	1	0	0	0	0	0
4	129	8	8	0	0	0	0	0	0	0	0	0
5	5	26	22	4	0	0	0	0	0	0	0	0
5	13	13	11	2	0	0	0	0	0	0	0	0

Ficou notório que, devido à produção estudada ser um *job-shop*, os recursos não apresentaram ligações padrões. O recurso 4, por exemplo, apresentou ligação para ele mesmo, para o recurso 37, para o recurso 66, para o recurso 81, para o recurso 118 e para o recurso 129. A seqüência a ser utilizada foi determinada pelo roteiro do produto desenvolvido pela engenharia de processo. Através desta análise foi possível determinar a relevância de cada combinação de seqüência de recurso e pôde estipular um peso de acordo com a classe de produtos de interesse, um dos parâmetros utilizados no modelo matemático utilizado na formulação do *layout* para a fábrica.

4.1 Distância de Levenshtein

Baseado no trabalho de Daita (1999) que apresentou a utilização do algoritmo de Levenshtein para agrupar produtos com semelhança de operações, foi implementado o algoritmo para verificar a similaridade entre as seqüências

produtivas dos produtos analisados. O primeiro objetivo da implementação foi constatar a existência de similaridade e estabeleceu uma métrica sobre a similaridade entre os roteiros de produção. Todos os produtos que compõem o conjunto selecionado foram comparados entre si. Somente os produtos com roteiro de produção cadastrado foram selecionados. Conforme a análise procedida pelo algoritmo proposto, o custo da diferença do produto A em relação ao produto B é o mesmo que o custo do produto B em relação ao produto A.

O algoritmo consiste em comparar cada operação de um par de produtos e armazenar em células de uma matriz o custo de cada interação aplicada sobre os roteiros. As interações consistem em inclusão, exclusão de seqüências. Sempre que há uma interação o custo é aumentado em uma unidade ao passo que quando as operações são iguais o custo não se altera.

A implementação do algoritmo foi feita em *Visual Basic for Applications* disponível no *Microsoft Access*. O algoritmo percorreu uma lista de registros que continham os produtos selecionados (*ProOri*); a cada registro, foi carregada outra tabela (*ProCmp*) que continha os demais produtos que se pretendia comparar com o produto carregado a partir do conjunto de registros principal. A cada combinação de pares de produtos analisados foi aplicado o algoritmo de Levenshtein e o resultado foi armazenado em uma tabela (*CmpLev*) que continha o produto original, o produto comparado e o custo da comparação. A matriz utilizada recebe o nome de *MatLev* no algoritmo desenvolvido. Os produtos que não apresentaram roteiros de produção cadastrados foram desconsiderados na análise. A Figura 2 mostra o pseudocódigo da implementação.

```

Carregar os produtos selecionados em ProOri;
Abrir a tabela CmpLev que vai receber os resultados das comparações e apagar todos os
registros existentes;
Enquanto não terminar os registros de ProOri faça:
  Carregar as operações do produto atual de RotOri em ordem de execução;
  Contar as operações do produto em armazenar em nQtdOri;
  Carregar os produtos a serem comparados em ProCmp, excluindo-se o produto
  original;
  Enquanto não terminar os registros em ProCmp faça:
    Carregar as operações do produto comparado e armazenar em RotCmp;
    Contas as operações do produto comparado e armazenar em nQtdCmp;
    Inicializar a matriz MatLev de comparação com nQtdOri linhas e nQtdCmp colunas;
    Zerar a matriz MatLev;
    Para i = 0 até nQtdOri faça:
      MatLev (i, 0) = i;
    Próximo i;
    Para j = 0 até nQtdCmp faça:
      MatLev (0, j) = j;
    Próximo j;
    Posicionar no primeiro registro de RotOri;
    Para i = 1 até nQtdOri faça:
      Posicionar no primeiro registro de RotCmp;
      Para j = 1 até nQtdCmp faça:
        Se (Recurso de RotOri) = (Recurso do RotCmp) então:
          Atribuir Custo = 0;
        Senão
          Atribuir Custo = 1;
        Fim Se;
        Atribuir MatLev (i, j) = Mínimo (MatLev(i - 1, j) + 1, MatLev(i, j - 1) +
        1, MatLev(i - 1, j - 1) + Custo)
        Vá para o próximo registro de RotCmp;
      Próximo j;
    Vá para o próximo registro de RotOri;
  Próximo i;
  Se a comparação não existir na tabela CmpLev então
    Inserir um novo registro em CmpLev, armazenando o código do produto
    original, o código do produto comparado e o custo da comparação
    MatLev (nQtdOri, nQtdCmp);
  Fim Se;
  Ir para o próximo registro de ProCmp;
Fim Enquanto;
Ir para o próximo registro de ProOri;
Fim Enquanto;

```

Figura 2 – Pseudocódigo para implementação do algoritmo de Levenshtein.

Ao analisar a o resultado da execução do algoritmo, o seu resumo foi apresentado na Tabela 3. Os custos das comparações tinham seu valor de zero a dezoito devido ao fato de que cada produto apresentou no máximo dezoito operações. Foi atribuído o custo de uma unidade monetária para cada interação, o custo era equivalente ao número de operações diferentes de cada comparação, ou seja, é correto afirmar que cada unidade corresponde ao número de operações diferentes encontradas entre os dois produtos de cada análise.

Tabela 3 – Custos Levenshtein agrupados.

Custo da Alteração (número de operações diferentes)	Quantidade de Combinações	Percentual
0	8.398	7,038
1	854	0,716
2	2.686	2,251
3	6.314	5,292
4	6.551	5,490
5	15.726	13,180
6	13.912	11,660
7	4.836	4,053
8	3.244	2,719
9	3.217	2,696
10	7.932	6,648
11	12.258	10,274
12	10.595	8,880
13	10.034	8,410
14	7.681	6,438
15	3.814	3,197
16	1.148	0,962
17	111	0,093
18	5	0,004
Total	119.316	

A primeira coluna corresponde ao número de operações diferentes encontrado na comparação de cada produto. A segunda coluna apresenta a quantidade de combinações que obtiveram o número de operações diferentes. A terceira coluna mostra o percentual das combinações relativo a somatória de combinações. Mostrando os dados da Tabela 3 em um gráfico, foi gerado o gráfico apresentado na Figura 3. O gráfico mostrou que a maior concentração de quantidade de combinações está nos grupos de cinco, seis, onze, doze e treze operações diferentes. Isto mostrou que a existência de roteiros similares é pequena, praticamente nula em relação ao montante das combinações.

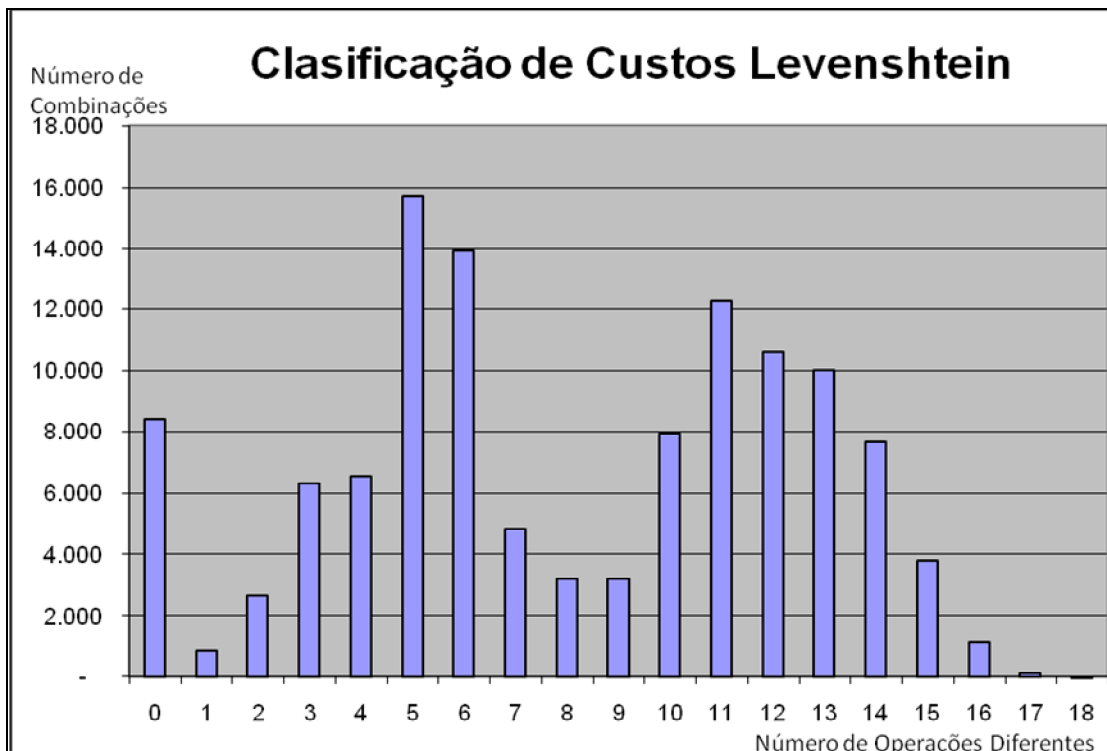


Figura 3 – Gráfico dos custos apurados.

Os dados de custos das ligações e de média de valor ponderado dos produtos de cada combinação foram cruzados. A Tabela 4 mostra os dados relativos à combinação dos custos com as médias dos valores ponderados. Para esta tabela foram selecionados os produtos que apresentaram um custo zero, ou seja, que possuísem a mesma seqüência de produção. O custo corresponde ao número de operações diferentes entre os produtos comparados.

Tabela 4 – Produtos com custo igual a zero e média do valor ponderado maior do que cinco.

Produtos Comparados			20	496	26	492
Valor Ponderado			9	10	5	7
122	1	Custo	0			
		Média	5			
128	1	Custo	0			
		Média	5			
129	1	Custo	0			
		Média	5			

Tabela 4 – Produtos com custo igual a zero e média do valor ponderado maior do que cinco (continuação).

130	1	Custo	0			
		Média	5			
138	2	Custo	0			
		Média	5,5			
145	1	Custo	0			
		Média	5			
17	1	Custo	0			
		Média	5			
26	5	Custo	0			
		Média	7			
54	2	Custo	0			
		Média	5,5			
74	1	Custo	0			
		Média	5			
77	1	Custo	0			
		Média	5			
8	8	Custo	0			
		Média	5			
484	1	Custo		0		
		Média		5,5		
488	1	Custo		0		
		Média		5,5		
490	1	Custo		0		
		Média		5,5		
492	7	Custo		0		
		Média		8,5		
493	1	Custo		0		
		Média		5,5		
494	1	Custo		0		
		Média		5,5		
502	1	Custo		0		
		Média		5,5		

Tabela 4 – Produtos com custo igual a zero e média do valor ponderado maior do que cinco (continuação).

507	1	Custo		0		
		Média		5,5		
509	1	Custo		0		
		Média		5,5		
513	2	Custo		0		
		Média		6		
514	1	Custo		0		
		Média		5,5		
570	1	Custo		0		
		Média		5,5		
579	1	Custo		0		
		Média		5,5		
29	5	Custo			0	
		Média			5	
576	4	Custo				0
		Média				5,5

Foram considerados os produtos que a média dos valores ponderados fosse igual ou maior que cinco. A Tabela 4 é composta por vinte e sete combinações, ou seja, são quatro roteiros que atendem aos produtos que compõem a tabela.

A Tabela 5 mostra os produtos que apresentaram uma operação de diferença e que a média dos valores ponderados foi maior ou igual a cinco. O custo corresponde ao número de operações diferentes entre os produtos comparados. Os produtos que apresentaram média de valor ponderado menor que cinco não foram considerados devido a pouca relevância para o negócio. A Tabela 5 é composta por seis combinações e apenas uma operação difere entre o roteiro do produto 20 e os demais produtos apresentados.

Tabela 5 – Produtos com custo igual a um e média do valor ponderado maior do que cinco.

Produtos Comparados		20	
		Valor Ponderado	
106	1	Custo	1
		Média	5
109	1	Custo	1
		Média	5
123	1	Custo	1
		Média	5
51	1	Custo	1
		Média	5
82	1	Custo	1
		Média	5
85	1	Custo	1
		Média	5

Os produtos que apresentaram duas operações de diferença e que a média dos valores ponderados foi maior ou igual a cinco são mostrados na Tabela 6. O custo corresponde ao número de operações diferentes entre os produtos comparados. Os produtos que apresentaram média de valor ponderado menor que cinco foram desprezados devido a pouca relevância para o negócio. Dezesesseis combinações compõem a Tabela 6. O produto 20 e o produto 496 apresentam o roteiro com menor diferença para outros produtos com média do valor ponderado maior do que cinco.

Tabela 6 – Produtos com custo igual a dois e média do valor ponderado maior do que cinco.

Produtos Comparados		20	496
Valor Ponderado		9	10
10	1	Custo	2
		Média	5
13	1	Custo	2
		Média	5
140	1	Custo	2
		Média	5
19	1	Custo	2
		Média	5
22	2	Custo	2
		Média	5,5
28	1	Custo	2
		Média	5
31	1	Custo	2
		Média	5
39	1	Custo	2
		Média	5
56	1	Custo	2
		Média	5
76	1	Custo	2
		Média	5
84	1	Custo	2
		Média	5
86	1	Custo	2
		Média	5
495	1	Custo	2
		Média	5,5
498	1	Custo	2
		Média	5,5
508	1	Custo	2
		Média	5,5
577	1	Custo	2
		Média	5,5

A Tabela 7 mostra os produtos que apresentaram três operações de diferença e que a média dos valores ponderados foi maior ou igual a cinco. O custo corresponde ao número de operações diferentes entre os produtos comparados. As combinações que apresentaram média de valor ponderado menor que cinco foram desprezados devido a pouca relevância para o negócio. A Tabela 5 é composta por cinco produtos e na pior das hipóteses seus roteiros são compostos por trinta recursos diferentes. As combinações com quatro operações diferentes não estão sendo tabuladas, pois nenhuma delas apresentou média dos valores ponderados maior ou iguais a cinco.

Tabela 7 – Produtos com custo igual a três e média do valor ponderado maior do que cinco.

Produtos Comparados		496	
Valor Ponderado		10	
499	2	Custo	3
		Média	6
500	2	Custo	3
		Média	6
501	1	Custo	3
		Média	1
515	1	Custo	3
		Média	1

Na Tabela 8 são mostrados os produtos que apresentaram cinco operações de diferença e que a média dos valores ponderados foi maior ou igual a cinco. O custo corresponde ao número de operações diferentes entre os produtos comparados. As combinações que apresentaram média de valor ponderado menor que cinco foram desprezados devido a pouca relevância para o negócio. A Tabela 8 é composta por sete produtos.

Tabela 8 – Produtos com custo igual a cinco e média do valor ponderado maior do que cinco.

Produtos Comparados			20
Valor Ponderado			9
465	1	Custo	5
		Média	5
471	1	Custo	5
		Média	5
475	1	Custo	5
		Média	5
476	1	Custo	5
		Média	5
480	1	Custo	5
		Média	5
481	1	Custo	5
		Média	5

Os produtos que apresentaram seis operações de diferença e que a média dos valores ponderados foi maior ou igual a cinco são mostrados na Tabela 9. O custo corresponde ao número de operações diferentes entre os produtos comparados. As combinações que apresentaram média de valor ponderado menor que cinco foram desprezadas devido a pouca relevância para o negócio. A Tabela 9 é composta por três produtos.

Tabela 9 – Produtos com custo igual a seis e média do valor ponderado maior do que cinco.

Produtos Comparados			20
Valor Ponderado			9
468	1	Custo	6
		Média	5
477	1	Custo	6
		Média	5

A Tabela 10 mostra os produtos que apresentaram sete operações de diferença e que a média dos valores ponderados foi maior ou igual a cinco. O custo corresponde ao número de operações diferentes entre os produtos comparados. As combinações que apresentaram média de valor ponderado menor que cinco foram desprezados

devido a pouca relevância para o negócio. A Tabela 10 é composta por dois produtos.

Tabela 10 – Produtos com custo igual a sete e média do valor ponderado maior do que cinco.

Produtos Comparados		708	
Valor Ponderado		6	
719	4	Custo	7
		Média	5

A combinação dos produtos que apresentaram oito operações de diferença e que a média dos valores ponderados foi maior ou igual a cinco são mostrados na Tabela 11. O custo corresponde ao número de operações diferentes entre os produtos comparados. As combinações que apresentaram média de valor ponderado menor que cinco foram desprezados devido a pouca relevância para o negócio. A Tabela 11 é composta por dezenove produtos e é a primeira combinação de produtos que apresenta custo para os três produtos principais em pelo menos um produto comparado.

Tabela 11 – Produtos com custo igual a oito e média do valor ponderado maior do que cinco.

Produtos Comparados			20	496	29
Valor Ponderado			9	10	7
136	1	Custo	8		
		Média	5		
137	1	Custo	8		
		Média	5		
153	1	Custo	8		
		Média	5		
154	2	Custo	8		
		Média	5,5		
156	1	Custo	8		
		Média	5		
157	1	Custo	8		
		Média	5		
159	1	Custo	8		
		Média	5		
482	1	Custo	8		
		Média	5		
483	1	Custo	8		
		Média	5		
59	2	Custo	8		
		Média	5,5		
62	2	Custo	8		
		Média	5,5		
662	1	Custo	8		
		Média	5		
666	2	Custo	8		
		Média	5,5		
668	1	Custo	8		
		Média	5		
671	1	Custo	8		
		Média	5		
151	5	Custo	8	8	8
		Média	7	5	6

A Tabela 12 mostra os produtos que apresentaram ambos, valor ponderado maior ou igual a cinco e custo menor do que nove. O custo corresponde ao número de operações diferentes entre os produtos comparados.

Tabela 12 – Produtos com custo menor ou igual a cinco e média do valor ponderado maior ou igual a cinco.

Produtos Comparados		26	29	20	496
Valor Ponderado		5	5	9	10
151	5	Custo	8	8	8
		Média	5	5	7
29	5	Custo	0		0
		Média	5		7
26	5	Custo			0
		Média			7
492	7	Custo			0
		Média			8,5

Os produtos apresentados na Tabela 12 mostram que poucas combinações são relevantes diante do montante total das combinações, são sete combinações relevantes diante de mais de cento e dezenove mil combinações.

4.2 Pareto

A semelhança entre os processos dos produtos de uma empresa pode ser uma forma de buscar a uniformidade do *layout* de produção (DAITA, 1999). Baseado nesta afirmação, foi desenvolvido o algoritmo que definiu novos roteiros padrões a partir dos roteiros dos produtos analisados, classificando-os de acordo com o seu valor ponderado. Os produtos que possuem roteiro foram carregados de uma tabela (*e075pro*) e analisados um a um. As seqüências de operações foram lidas na tabela *e710sqr* que possui relacionamento com a tabela de produtos. Os roteiros foram comparados com os roteiros armazenados na tabela *FluPar*, se existir a seqüência das operações, foi adicionado um no contador, caso contrário a seqüência gerou um novo registro nesta tabela. A Figura 4 mostra o algoritmo desenvolvido.

```
Carregar os produtos selecionados em e075pro que possuem roteiro;  
Abrir a tabela FluPar que vai receber os roteiros e apagar todos os registros  
existentes;  
Enquanto não terminar os registros de e075pro faça:  
  Carregar as operações do produto atual em e710sqr em ordem de execução;  
  Carregar no vetor CodCre as operações de e710sqr;  
  Se existe o roteiro do vetor CodCre em FluPar então:  
    Alterar o registro em FluPar para acrescentar uma unidade no total de  
    utilização;  
    Acrescentar uma unidade no total de utilização de acordo com a classificação  
    ponderada do produto;  
  Senão  
    Criar um novo registro em FluPar gerando um novo código de roteiro;  
    Armazenar o roteiro do vetor CodCre em FluPar e atribuir valores as colunas de  
    classificação ponderada e quantidade total de utilização;  
  Fim se;  
  Ir para o próximo registro de e075pro;  
Fim Enquanto;
```

Figura 4 – Pseudocódigo para implementação do algoritmo de geração de roteiros classificados.

Na determinação do *layout* a ser definido para a produção é necessário o conhecimento da sua composição (TOMPKINS, 2003 e DAITA, 1999). Para esta análise foi desenvolvido um algoritmo que montou roteiros com as seqüências comuns entre os produtos analisados. Além da construção do roteiro, o algoritmo também classificou os produtos que utilizaram cada um dos roteiros de acordo com o valor ponderado, contando o número de produtos total de produtos por roteiro e por valor ponderado. A Figura 5 mostra o gráfico de variação de quantidade de produtos por roteiro.

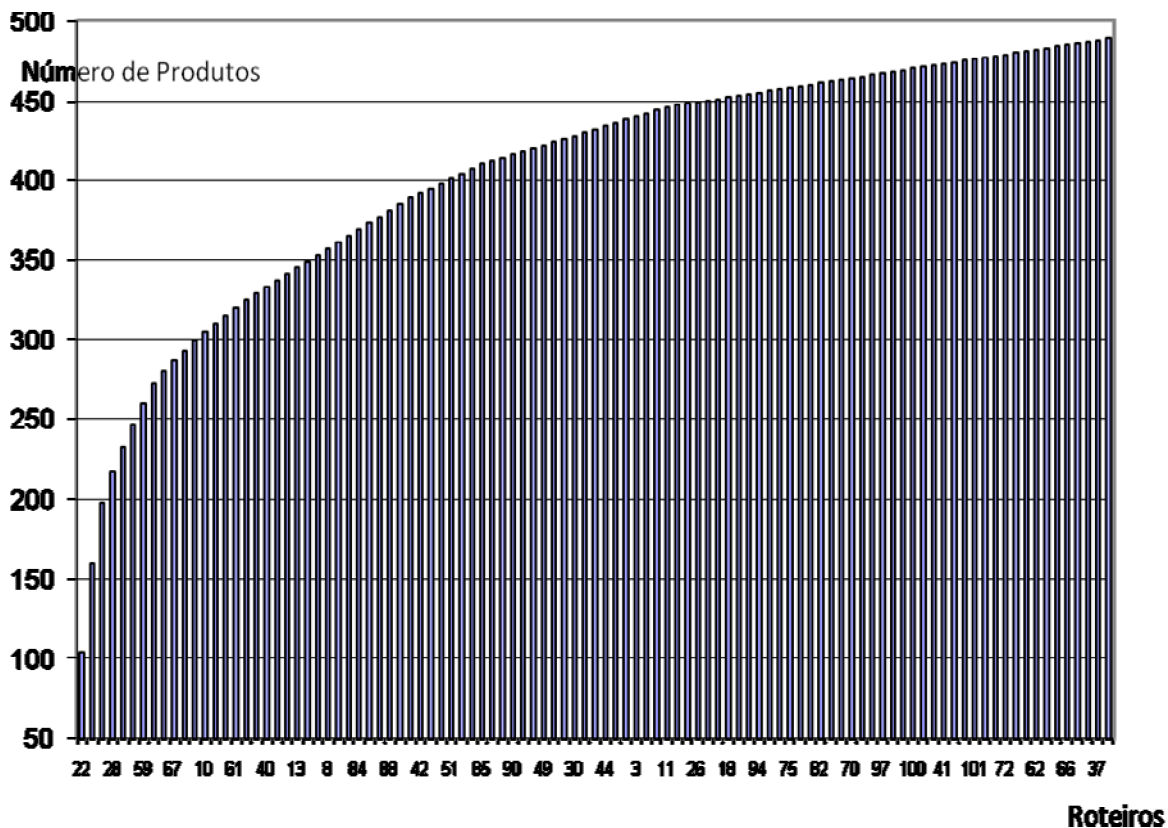


Figura 5 – Gráfico da distribuição de Pareto dos roteiros similares.

Buscando a melhor compilação dos dados provenientes do algoritmo para analisar minuciosamente os dados obtidos, foram apresentados na Tabela 13 os roteiros com suas quantidades de produtos que os utilizam, juntamente com sua classificação de acordo com o valor. Os dados foram classificados para trazer primeiramente os valores em ordem do valor ponderado, ou seja, a maior quantidade de combinações para o valor ponderado 10 foi trazido antes da maior quantidade de combinações para o valor ponderado 9 e assim sucessivamente até o valor ponderado 1. Somente os roteiros que apresentaram quantidade total de produtos maior que cinco foram considerados. Nesta tabela foram apresentados os roteiros que são utilizados por trezentos e quarenta e três produtos, o que corresponde a 70,14 por cento dos produtos que foram classificados pelo algoritmo criado.

Tabela 13 – Classificação dos roteiros gerados com o valor ponderado dos produtos analisados.

Roteiro	Valor Ponderado									Total
	10	9	7	6	5	4	3	2	1	
34	1		1			1		1	11	15
4		1			2			2	9	14
101				1						1
15					1			1	6	8
67						1	1	3	2	7
38							1		37	38
30							1		1	2
100							1			1
59								2	11	13
13								2	2	4
36								2	1	3
5								1	11	12
61								1	4	5
56								1	3	4
57								1	2	3
22									104	104
21									56	56
28									20	20
32									6	6
10									6	6
2									6	6
83									5	5
78									5	5
20									5	5
Total	1	1	1	1	3	2	4	17	313	343

Os roteiros apresentados na Tabela 13 são detalhados na Tabela 14, com suas seqüências de operações e o recurso utilizado em cada seqüência.

Tabela 14 – Roteiros gerados e suas seqüências.

Roteiro	Operações																
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
34	65	24	33	24	63	33	33	73	87	55	93	75	77	66	37		
4	65	44	62	60	34	24	70	54	120	81	42	91	70	66	37		
101	65	124	124	32	48	70	82	66	37								
15	65	44	27	70	4	118	81	42	91	70	66	36					
67	65	129	122	31	66	37	110										
38	65	25	62	24	79	25	24	74	17	78	83	66	37				
30	65	44	27	70	118	42	70	66	36								
100	4	4															
59	5	5	5	13	66	37											
13	65	44	25	62	70	118	81	46	91	70	66	37					
36	65	24	33	24	63	33	33	73	75	77	66	37					
5	65	44	62	60	34	24	70	54	120	81	42	53	91	70	4	66	37
61	28	58	66	37													
56	65	32	128	88	66	37											
57	65	44	61	28	31	61	70	4	118	81	44	91	70	66	37		
22	82	14	14	14	7	66	37										
21	14	37															
28	65	64	66	37													
32	65	44	25	62	70	54	120	81	42	91	70	66	37				
10	65	44	62	60	34	24	70	54	120	81	42	53	91	70	66	37	
2	10	11	66	37													
83	65	24	66	37													
78	65	25	66	74													
20	65	114	65	30	48	67	112	65	66	37							

Os dados apresentados não mostraram uniformidade completa e o melhor *layout* não pôde ser conseguido através destas combinações. Por esta razão analisou-se a relação das operações em relação ao valor ponderado adiante neste trabalho.

4.3 Roteiro Selecionado e Roteiros Combinados

Os produtos foram classificados de acordo com o valor ponderado, gerando a classificação para os seus respectivos roteiros com o valor ponderado. Na base de

dados, cada produto possui um roteiro especificamente cadastrado, porém os roteiros possuem semelhanças. Selecionado o roteiro que possui produtos com o valor ponderado 10 foi obtido o roteiro 34. O roteiro selecionado tem semelhança com outros roteiros apresentados na Tabela 15.

Tabela 15 – Classificação dos roteiros gerados com o valor ponderado dos produtos analisados.

Roteiro	Valor Ponderado					Total
	10	7	4	2	1	
8					4	4
34	1	1	1	1	11	15
36				2	1	3
37					1	1
83					5	5
Total	1	1	1	3	22	28

A semelhança dos roteiros apresentados aconteceu em sua seqüência produtiva. Observando que o objetivo desta combinação é a definição de um *layout*, foram encontrados os roteiros que utilizem os mesmos recursos que foram utilizados no roteiro 34. Os roteiros apresentados na Tabela 15 foram detalhados na Tabela 16, com suas seqüências de operações e o recurso utilizado em cada seqüência.

Tabela 16 – Roteiros selecionados e suas seqüências.

Roteiro	Operação																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
8	65																	
34	65	24	33	24	63	33	33	73	87	55	93	75	77	66	37			
37	65	24	33	24	73	87	24	63	33	33	73	87	55	93	75	77	66	37
36	65	24	33	24	63	33	33	73	75	77	66	37						
83	65	24	66	37														

As tabelas apresentadas estão em ordem do código de roteiros. Os produtos que possuem os roteiros apresentados são apresentados na Tabela 17. A tabela mostra os produtos com valor ponderado e o roteiro que utilizam na produção.

Tabela 17 – Produtos com seu valor ponderado e o roteiro geral gerado através do algoritmo de classificação.

Produto	Valor Ponderado	Roteiro
25	1	8
133	1	8
174	1	8
23	1	8
493	1	34
496	10	34
492	7	34
490	1	34
570	1	34
484	1	34
488	1	34
507	1	34
502	1	34
494	1	34
513	2	34
509	1	34
576	4	34
579	1	34
514	1	34
501	1	36
499	2	36
500	2	36
515	1	37
779	1	83
785	1	83
773	1	83
786	1	83
778	1	83

4.4 Resolvendo o Modelo

No caso específico deste estudo, não foram utilizados valores para as distâncias mínimas entre os recursos devido ao fato de serem considerados os espaços para operação e manutenção durante a definição das dimensões dos recursos.

A ferramenta utilizada para resolver o modelo foi o LINGO na versão 11.0. A implementação do modelo em linguagem matemática para a ferramenta passou pela adequação do modelo à ferramenta. Na Equação 15 é rerepresentado o modelo matemático para comparar o ajuste das variáveis para a ferramenta. Os valores de i e j podem variar de 1 até 17 que correspondem a quantidade de recursos diferentes (n) encontrados no roteiro selecionado. O custo e o fluxo foram multiplicados e o resultado foi colocado na frente de cada combinação x_p para os x^+ e x^- . A variável M recebeu o valor de 99999 que correspondeu ao número máximo representado com a mesma quantidade de algarismos da maior dimensão encontrada para restrição de distância mínima de centros (21565). As variáveis z receberam um ou zero de acordo com a restrição que estará sendo utilizada. O valor da distância mínima entre os pontos centrais de cada comparação de recursos foi calculado previamente e o resultado foi inserido no modelo.

$$\begin{aligned}
 \text{Minimizar} \quad & z = \sum_{i=1}^{n-1} \sum_{j=i+1}^n c_{ij} f_{ij} (x_{ij}^+ - x_{ij}^-) && \text{Eq. 15} \\
 \text{Sujeito à:} \quad & x_i - x_j + Mz_{ij} \geq \frac{1}{2}(l_i + l_j) + d_{ij} \\
 & -(x_i - x_j) + M(1 - z_{ij}) \geq \frac{1}{2}(l_i + l_j) + d_{ij} \\
 & x_{ij}^+ - x_{ij}^- = x_i - x_j \\
 & x_{ij}^+, x_{ij}^- \geq 0 \\
 & z_{ij} = 0 \text{ ou } 1 \\
 & z_i > 0
 \end{aligned}$$


```

model:
min =
!início da função objetivo;
  1 * xp12 + 98040 * xp17 +
851 * xp23 + 16 * xp25 + 851 * xp26 + 1632 * xp27 + 36 * xp28 + 1 * xp29
+ 16 * xp210 + 1 * xp212 +
851 * xp36 + 4 * xp37 + 851 * xp39 +
  4 * xp47 + 147084 * xp48 + 193488 * xp414 +
640 * xp512 + 448 * xp513 +
144 * xp78 + 16 * xp714 +
627 * xp811 + 576 * xp814 +
  15 * xp910 + 693 * xp912 +
851 * xp1011 + 448 * xp1013 +
  1 * xn12 + 98040 * xn17 +
851 * xn23 + 16 * xn25 + 851 * xn26 + 1632 * xn27 + 36 * xn28 + 1 * xn29
+ 16 * xn210 + 1 * xn212 +
851 * xn36 + 4 * xn37 + 851 * xn39 +
  4 * xn47 + 147084 * xn48 + 193488 * xn414 +
640 * xn512 + 448 * xn513 +
144 * xn78 + 16 * xn714 +
627 * xn811 + 576 * xn814 +
  15 * xn910 + 693 * xn912 +
851 * xn1011 + 448 * xn1013;
!fim da função objetivo;
!início das restrições referentes a não sobreposição de equipamentos;
99999 * z12 + x1 - x2 > 11250;
99999 * z13 + x1 - x3 > 9365;
99999 * z14 + x1 - x4 > 9805;
99999 * z15 + x1 - x5 > 9125;
99999 * z16 + x1 - x6 > 8500;
99999 * z17 + x1 - x7 > 11250;
99999 * z18 + x1 - x8 > 11450;
99999 * z19 + x1 - x9 > 8520;
99999 * z110 + x1 - x10 > 8750;
99999 * z111 + x1 - x11 > 9135;
99999 * z112 + x1 - x12 > 8700;
99999 * z113 + x1 - x13 > 9060;
99999 * z114 + x1 - x14 > 21565;
99999 * z23 + x2 - x3 > 4615;
99999 * z24 + x2 - x4 > 5055;
99999 * z25 + x2 - x5 > 4375;
99999 * z26 + x2 - x6 > 3750;
99999 * z27 + x2 - x7 > 6500;
99999 * z28 + x2 - x8 > 6700;
99999 * z29 + x2 - x9 > 3770;
99999 * z210 + x2 - x10 > 4000;
99999 * z211 + x2 - x11 > 4385;
99999 * z212 + x2 - x12 > 3950;
99999 * z213 + x2 - x13 > 4310;
99999 * z214 + x2 - x14 > 16815;
99999 * z34 + x3 - x4 > 3170;
99999 * z35 + x3 - x5 > 2490;
99999 * z36 + x3 - x6 > 1865;
99999 * z37 + x3 - x7 > 4615;
99999 * z38 + x3 - x8 > 4815;
99999 * z39 + x3 - x9 > 1885;
99999 * z310 + x3 - x10 > 2115;
99999 * z311 + x3 - x11 > 2500;
99999 * z312 + x3 - x12 > 2065;
99999 * z313 + x3 - x13 > 2425;
99999 * z314 + x3 - x14 > 14930;
99999 * z45 + x4 - x5 > 2930;
99999 * z46 + x4 - x6 > 2305;
99999 * z47 + x4 - x7 > 5055;
99999 * z48 + x4 - x8 > 5255;
99999 * z49 + x4 - x9 > 2325;
99999 * z410 + x4 - x10 > 2555;
99999 * z411 + x4 - x11 > 2940;

```

Figura 6 – Implementação do modelo no LINGO.

99999	*	z412	+	x4	-	x12	>	2505;
99999	*	z413	+	x4	-	x13	>	2865;
99999	*	z414	+	x4	-	x14	>	15370;
99999	*	z56	+	x5	-	x6	>	1625;
99999	*	z57	+	x5	-	x7	>	4375;
99999	*	z58	+	x5	-	x8	>	4575;
99999	*	z59	+	x5	-	x9	>	1645;
99999	*	z510	+	x5	-	x10	>	1875;
99999	*	z511	+	x5	-	x11	>	2260;
99999	*	z512	+	x5	-	x12	>	1825;
99999	*	z513	+	x5	-	x13	>	2185;
99999	*	z514	+	x5	-	x14	>	14690;
99999	*	z67	+	x6	-	x7	>	3750;
99999	*	z68	+	x6	-	x8	>	3950;
99999	*	z69	+	x6	-	x9	>	1020;
99999	*	z610	+	x6	-	x10	>	1250;
99999	*	z611	+	x6	-	x11	>	1635;
99999	*	z612	+	x6	-	x12	>	1200;
99999	*	z613	+	x6	-	x13	>	1560;
99999	*	z614	+	x6	-	x14	>	14065;
99999	*	z78	+	x7	-	x8	>	6700;
99999	*	z79	+	x7	-	x9	>	3770;
99999	*	z710	+	x7	-	x10	>	4000;
99999	*	z711	+	x7	-	x11	>	4835;
99999	*	z712	+	x7	-	x12	>	3950;
99999	*	z713	+	x7	-	x13	>	4310;
99999	*	z714	+	x7	-	x14	>	16815;
99999	*	z89	+	x8	-	x9	>	3970;
99999	*	z810	+	x8	-	x10	>	4200;
99999	*	z811	+	x8	-	x11	>	4585;
99999	*	z812	+	x8	-	x12	>	4150;
99999	*	z813	+	x8	-	x13	>	4510;
99999	*	z814	+	x8	-	x14	>	17015;
99999	*	z910	+	x9	-	x10	>	1270;
99999	*	z911	+	x9	-	x11	>	1655;
99999	*	z912	+	x9	-	x12	>	1220;
99999	*	z913	+	x9	-	x13	>	1580;
99999	*	z914	+	x9	-	x14	>	14085;
99999	*	z1011	+	x10	-	x11	>	1885;
99999	*	z1012	+	x10	-	x12	>	1450;
99999	*	z1013	+	x10	-	x13	>	1810;
99999	*	z1014	+	x10	-	x14	>	14315;
99999	*	z1112	+	x11	-	x12	>	1835;
99999	*	z1113	+	x11	-	x13	>	2195;
99999	*	z1114	+	x11	-	x14	>	14700;
99999	*	z1213	+	x12	-	x13	>	1760;
99999	*	z1214	+	x12	-	x14	>	14265;
99999	*	z1314	+	x13	-	x14	>	14625;
99999	*	z12	+	x1	-	x2	<	88749;
99999	*	z13	+	x1	-	x3	<	90364;
99999	*	z14	+	x1	-	x4	<	90194;
99999	*	z15	+	x1	-	x5	<	90874;
99999	*	z16	+	x1	-	x6	<	91499;
99999	*	z17	+	x1	-	x7	<	88749;
99999	*	z18	+	x1	-	x8	<	88549;
99999	*	z19	+	x1	-	x9	<	91479;
99999	*	z110	+	x1	-	x10	<	91249;
99999	*	z111	+	x1	-	x11	<	90864;
99999	*	z112	+	x1	-	x12	<	91299;
99999	*	z113	+	x1	-	x13	<	90939;
99999	*	z114	+	x1	-	x14	<	79434;
99999	*	z23	+	x2	-	x3	<	95384;
99999	*	z24	+	x2	-	x4	<	94944;
99999	*	z25	+	x2	-	x5	<	95624;
99999	*	z26	+	x2	-	x6	<	96249;
99999	*	z27	+	x2	-	x7	<	93499;
99999	*	z28	+	x2	-	x8	<	93299;
99999	*	z29	+	x2	-	x9	<	96229;
99999	*	z210	+	x2	-	x10	<	95999;

Figura 6 – Implementação do modelo no LINGO (Continuação).

99999	*	z211	+	x2	-	x11	<	95614;
99999	*	z212	+	x2	-	x12	<	96049;
99999	*	z213	+	x2	-	x13	<	95689;
99999	*	z214	+	x2	-	x14	<	83184;
99999	*	z34	+	x3	-	x4	<	96829;
99999	*	z35	+	x3	-	x5	<	97509;
99999	*	z36	+	x3	-	x6	<	98134;
99999	*	z37	+	x3	-	x7	<	95384;
99999	*	z38	+	x3	-	x8	<	95184;
99999	*	z39	+	x3	-	x9	<	98114;
99999	*	z310	+	x3	-	x10	<	97884;
99999	*	z311	+	x3	-	x11	<	97499;
99999	*	z312	+	x3	-	x12	<	97934;
99999	*	z313	+	x3	-	x13	<	97574;
99999	*	z314	+	x3	-	x14	<	85069;
99999	*	z45	+	x4	-	x5	<	97069;
99999	*	z46	+	x4	-	x6	<	97694;
99999	*	z47	+	x4	-	x7	<	94944;
99999	*	z48	+	x4	-	x8	<	94744;
99999	*	z49	+	x4	-	x9	<	97674;
99999	*	z410	+	x4	-	x10	<	97444;
99999	*	z411	+	x4	-	x11	<	97059;
99999	*	z412	+	x4	-	x12	<	97494;
99999	*	z413	+	x4	-	x13	<	97134;
99999	*	z414	+	x4	-	x14	<	84629;
99999	*	z56	+	x5	-	x6	<	98374;
99999	*	z57	+	x5	-	x7	<	95624;
99999	*	z58	+	x5	-	x8	<	95424;
99999	*	z59	+	x5	-	x9	<	98354;
99999	*	z510	+	x5	-	x10	<	98124;
99999	*	z511	+	x5	-	x11	<	97739;
99999	*	z512	+	x5	-	x12	<	98174;
99999	*	z513	+	x5	-	x13	<	97814;
99999	*	z514	+	x5	-	x14	<	85309;
99999	*	z67	+	x6	-	x7	<	96249;
99999	*	z68	+	x6	-	x8	<	96049;
99999	*	z69	+	x6	-	x9	<	98979;
99999	*	z610	+	x6	-	x10	<	98749;
99999	*	z611	+	x6	-	x11	<	98364;
99999	*	z612	+	x6	-	x12	<	98799;
99999	*	z613	+	x6	-	x13	<	98439;
99999	*	z614	+	x6	-	x14	<	85934;
99999	*	z78	+	x7	-	x8	<	93299;
99999	*	z79	+	x7	-	x9	<	96229;
99999	*	z710	+	x7	-	x10	<	95999;
99999	*	z711	+	x7	-	x11	<	95164;
99999	*	z712	+	x7	-	x12	<	96049;
99999	*	z713	+	x7	-	x13	<	95689;
99999	*	z714	+	x7	-	x14	<	83184;
99999	*	z89	+	x8	-	x9	<	96029;
99999	*	z810	+	x8	-	x10	<	95799;
99999	*	z811	+	x8	-	x11	<	95414;
99999	*	z812	+	x8	-	x12	<	95849;
99999	*	z813	+	x8	-	x13	<	95489;
99999	*	z814	+	x8	-	x14	<	82984;
99999	*	z910	+	x9	-	x10	<	98729;
99999	*	z911	+	x9	-	x11	<	98344;
99999	*	z912	+	x9	-	x12	<	98779;
99999	*	z913	+	x9	-	x13	<	98419;
99999	*	z914	+	x9	-	x14	<	85914;
99999	*	z1011	+	x10	-	x11	<	98114;
99999	*	z1012	+	x10	-	x12	<	98549;
99999	*	z1013	+	x10	-	x13	<	98189;
99999	*	z1014	+	x10	-	x14	<	85684;
99999	*	z1112	+	x11	-	x12	<	98164;
99999	*	z1113	+	x11	-	x13	<	97804;
99999	*	z1114	+	x11	-	x14	<	85299;
99999	*	z1213	+	x12	-	x13	<	98239;
99999	*	z1214	+	x12	-	x14	<	85734;

Figura 6 – Implementação do modelo no LINGO (Continuação).

```

99999 * z1314 + x13 - x14 < 85374;
!fim das restrições referentes a não sobreposição de equipamentos;
!início das restrições que garantem que xpij e xnij sejam iguais a zero ou um;
- xp12 + xn12 + x1 - x2 = 0;
- xp17 + xn17 + x1 - x7 = 0;
- xp23 + xn23 + x2 - x3 = 0;
- xp24 + xn25 + x2 - x5 = 0;
- xp26 + xn26 + x2 - x6 = 0;
- xp27 + xn27 + x2 - x7 = 0;
- xp28 + xn28 + x2 - x8 = 0;
- xp29 + xn29 + x2 - x9 = 0;
- xp210 + xn210 + x2 - x10 = 0;
- xp212 + xn212 + x2 - x12 = 0;
- xp36 + xn36 + x3 - x6 = 0;
- xp37 + xn37 + x3 - x7 = 0;
- xp39 + xn39 + x3 - x9 = 0;
- xp47 + xn47 + x4 - x7 = 0;
- xp48 + xn48 + x4 - x8 = 0;
- xp414 + xn414 + x4 - x14 = 0;
- xp512 + xn512 + x5 - x12 = 0;
- xp513 + xn513 + x5 - x13 = 0;
- xp78 + xn78 + x7 - x8 = 0;
- xp714 + xn714 + x7 - x14 = 0;
- xp811 + xn811 + x8 - x11 = 0;
- xp814 + xn814 + x8 - x14 = 0;
- xp910 + xn910 + x9 - x10 = 0;
- xp912 + xn912 + x9 - x12 = 0;
- xp1011 + xn1011 + x10 - x11 = 0;
- xp1013 + xn1013 + x10 - x13 = 0;
!fim das restrições que garantem que xpij e xnij sejam iguais a zero ou um;
!início das restrições que garantem que os zij somente recebam somente valores
inteiros;
@BIN(z12);
@BIN(z13);
@BIN(z14);
@BIN(z15);
@BIN(z16);
@BIN(z17);
@BIN(z18);
@BIN(z19);
@BIN(z110);
@BIN(z111);
@BIN(z112);
@BIN(z113);
@BIN(z114);
@BIN(z23);
@BIN(z24);
@BIN(z25);
@BIN(z26);
@BIN(z27);
@BIN(z28);
@BIN(z29);
@BIN(z210);
@BIN(z211);
@BIN(z212);
@BIN(z213);
@BIN(z214);
@BIN(z34);
@BIN(z35);
@BIN(z36);
@BIN(z37);
@BIN(z38);
@BIN(z39);
@BIN(z310);
@BIN(z311);
@BIN(z312);
@BIN(z313);
@BIN(z314);
@BIN(z45);

```

Figura 6 – Implementação do modelo no LINGO (Continuação).

```
@BIN(z46);
@BIN(z47);
@BIN(z48);
@BIN(z49);
@BIN(z410);
@BIN(z411);
@BIN(z412);
@BIN(z413);
@BIN(z414);
@BIN(z56);
@BIN(z57);
@BIN(z58);
@BIN(z59);
@BIN(z510);
@BIN(z511);
@BIN(z512);
@BIN(z513);
@BIN(z514);
@BIN(z67);
@BIN(z68);
@BIN(z69);
@BIN(z610);
@BIN(z611);
@BIN(z612);
@BIN(z613);
@BIN(z614);
@BIN(z78);
@BIN(z79);
@BIN(z710);
@BIN(z711);
@BIN(z712);
@BIN(z713);
@BIN(z714);
@BIN(z89);
@BIN(z810);
@BIN(z811);
@BIN(z812);
@BIN(z813);
@BIN(z814);
@BIN(z910);
@BIN(z911);
@BIN(z912);
@BIN(z913);
@BIN(z914);
@BIN(z1011);
@BIN(z1012);
@BIN(z1013);
@BIN(z1014);
@BIN(z1112);
@BIN(z1113);
@BIN(z1114);
@BIN(z1213);
@BIN(z1214);
@BIN(z1314);
!fim das restrições que garantem que os zij somente recebam somente valores inteiros;
!início da restrição que os x recebam valores não negativos e que os equipamentos
estejam inteiramente dentro da área da fábrica;
x1 - 8000 >= 0;
x2 - 3250 >= 0;
x3 - 1365 >= 0;
x4 - 1805 >= 0;
x5 - 1125 >= 0;
x6 - 500 >= 0;
x7 - 3250 >= 0;
x8 - 3450 >= 0;
x9 - 520 >= 0;
x10 - 750 >= 0;
x11 - 1135 >= 0;
x12 - 700 >= 0;
```

Figura 6 – Implementação do modelo no LINGO (Continuação).

```
x13 - 1060 >= 0;
x14 - 13565 >= 0;
!fim da restrição que os x recebam valores não negativos e que os equipamentos
estejam inteiramente dentro da área da fábrica;
```

Figura 6 – Implementação do modelo no LINGO (Continuação).

O resultado do modelo depende diretamente dos valores do custo e do fluxo. Para o modelo calculado montou-se a Tabela 18 que contém os valores do custo multiplicado pelo fluxo da combinação de cada recurso.

Tabela 18 – Valores para o custo multiplicado pelo fluxo da combinação de cada equipamento.

Origem	Destino											
	2	3	5	6	7	8	9	10	11	12	13	14
1	1				98040							
2		851	16	851	1632	36	1	16		1		
3				851	4		851					
4					4	147084						193488
5										640	448	
6						144						16
7									627			576
8								15		693		
9									851		448	
10												

Os dados foram dispostos de forma a considerar o fluxo em um único sentido, por exemplo, do recurso 2 para o recurso 3 e em caso de retorno, os valores são somados a combinação do recurso 2 para o recurso 3, sendo assim, somente na diagonal superior existiram valores diferentes de zero. O fluxo de um recurso para ele mesmo é desconsiderado para o cálculo do *layout*. As combinações que não apresentaram ligação estão zeradas.

O resultado do modelo são valores para x que são o ponto central de cada um dos recursos submetidos ao modelo. Na Tabela 19 são mostrados os resultados do cálculo do modelo.

Tabela 19 – Resultados do modelo resolvido no LINGO.

Recurso	Índice	Dimensão Horizontal	$\frac{l}{2}$	Solução		
				Posição Central	Posição Inferior	Posição Superior
				x	x	x
0	1	16000	8000	72.950	64.950	80.950
24	2	6500	3250	55.200	51.950	58.450
33	3	2730	1365	49.585	48.220	50.950
37	4	3610	1805	28.935	27.130	30.740
55	5	2250	1125	44.655	43.530	45.780
63	6	1000	500	51.450	50.950	51.950
65	7	6500	3250	61.700	58.450	64.950
66	8	6900	3450	34.190	30.740	37.640
73	9	1040	520	47.700	47.180	48.220
75	10	1500	750	40.660	39.910	41.410
77	11	2270	1135	38.775	37.640	39.910
87	12	1400	700	46.480	45.780	47.180
93	13	2120	1060	42.470	41.410	43.530
99999	14	27130	13565	13.565	0	27.130

Os resultados mostrados na coluna de solução, posição central, são calculados para encontrar a posição inferior e superior de x para cada recurso. Para resolução do modelo, cada recurso recebe um índice, mostrado na segunda coluna da Tabela 19. Foram acrescentados dois recursos aos roteiros, o recurso 0 que corresponde ao estoque de matéria prima e representa o espaço necessário para este setor da fábrica e o recurso 99999 que corresponde ao estoque de material acabado. A inclusão destes dois departamentos garantiu a consideração das relações dos recursos, quando existirem, com os mesmos.

Utilizando os resultados apresentados na Tabela 19, foi gerado o desenho do *layout* conforme segue na Figura 7. Os códigos atribuídos a cada recurso correspondem ao índice da matriz de recursos mostrado na Tabela 19. Conforme discutido por diversos autores (TOMPKINS, 2003, DAITA, 1999 e HERAGU, 1997), após a definição do *layout* através de algum modelo matemático, deve-se analisar o resultado e propor adequações que não são possíveis de serem implementadas no

modelo. O *layout* definido é uma linha, por este motivo não é necessário alterar o posicionamento dos recursos. As formas retangulares serão substituídas pelo desenho real dos recursos, e no momento da substituição, pode acontecer a rotação fazendo com que o sentido das formas se altere ligeiramente.

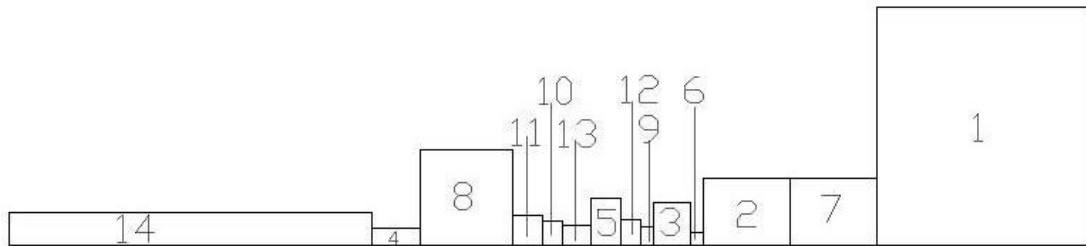


Figura 7 – Resultado do modelo resolvido.

4.5 Validação

A posição original dos recursos na planta está apresentada na Tabela 20 com os valores de x e y da posição inferior e superior de cada recurso. A posição do ponto central de cada um dos recursos foi calculada e a partir desta foi possível calcular a distância entre os recursos.

Tabela 20 – Posição dos recursos no *layout* original da empresa.

Recurso	Dimensão Horizontal	$\frac{l}{2}$	Dimensão Vertical	$\frac{l}{2}$	Posição					
					Centro		Inferior		Superior	
					x	y	x	y	x	y
0	16000	8000	17850	8925	37714	23301	29714	14376	45714	32226
24	4680	2340	5300	2650	53117	49512	50777	46862	55457	52162
33	2730	1365	3150	1575	36440	69349	35075	67774	37805	70924
37	3609	1805	1230	615	78923	44250	77118	43635	80727	44865
55	2250	1125	3500	1750	60528	34939	59403	33189	61653	36689
63	1000	500	1000	500	86593	57100	86093	56600	87093	57600
65	6500	3250	5000	2500	37483	11885	34233	9385	40733	14385
66	6900	3450	7200	3600	19605	61333	16155	57733	23055	64933
73	1040	520	1370	685	88018	48533	87498	47848	88538	49218
75	1500	750	1800	900	86180	26924	85430	26024	86930	27824
77	2270	1135	2190	1095	57738	54243	56603	53148	58873	55338
87	1400	700	1902	951	87684	45613	86984	44662	88384	46564
93	2120	1060	1530	765	85777	44341	84717	43576	86837	45106
99999	27130	13565	2420	1210	60590	18703	47025	17493	74155	19913

A distância encontrada em milímetros foi multiplicada pelo valor ponderado e pela quantidade de produtos que transita na ligação de cada par de recursos. Os valores do cálculo de cada ligação estão apresentados na Tabela 21 e 22.

Os valores de custos apresentados nas tabelas 21 até 24 são referenciais para comparativo entre o *layout* inicial e final. A relação percentual entre os dois mostra a economia proporcionada pela aplicação da metodologia para melhoria do *layout*.

Tabela 21 – Custo de cada ligação multiplicado pela distância na posição original.

	24	33	37	55	63	65	66
0	41.614	-	-	-	8.105.751.120	-	-
24		31.073.414	97.080	18.708.384	228.623.616	1.917.396	45.333
33				49.781.798	249.608	-	211.48.201
37					82.082	10.855.461.078	-
55						-	-
63						13.582.800	-
65							-

Procedendo-se a somatória dos valores das tabelas 21 e 22, e dividindo estes valores por mil, obtêm-se o valor total de 25.786.051 unidades de custo, que é o valor de referência apresentando pelo modelo matemático utilizado, o qual deverá ser reduzido pelo novo *layout* sugerido. O valor encontrado foi utilizado para comparar o total do *layout* antes e depois da aplicação do cálculo do modelo.

Tabela 22 – Custo de cada ligação multiplicado pela distância na posição original.

	73	75	77	87	93	99999
0	-	-	-	-	-	-
24	574.080	-	9.352	-	-	-
33	-	-	-	-	-	-
37	-	-	-	-	1.343.870.904	-
55	-	-	14.140.160	16.947.840	-	-
63	-	-	-	-	217.200	-
65	-	39.962.472	-	-	46.512.000	-
66	12.18.195	-	31.339.539	-	-	-
73		19.953.397	-	1.457.792	-	-
75			-	-	-	-
77				3.356.112	3.528.513	3.839.161.608
87					41.327	756.056
93						1.096.041.125

Após o cálculo da posição dos recursos utilizando o modelo matemático, calculou-se o valor de custo de cada ligação relativo à posição de cada par de recursos no novo *layout*. Os valores das ligações estão apresentados nas tabelas 23 e 24.

Tabela 23 – Custo de cada ligação multiplicado pela distância no novo *layout* proposto.

	24	33	37	55	63	65	66
0	17.750	-	-	-	2.107.860.000	-	-
24		4.778.365	420.240	8.973.795	6.120.000	234.000	21.010
33				4.195.430	7.460	-	13.101.145
37					90.060	4.819.207.260	-
55						-	-
63						1.476.000	-
65							-

Procedeu-se o cálculo do custo das ligações da nova sugestão de *layout* com o mesmo procedimento do cálculo do custo do *layout* original. O custo total encontrado para o novo *layout* foi de 12.769.843 unidades de custo.

Tabela 24 – Custo de cada ligação multiplicado pela distância no novo *layout* proposto.

	73	75	77	87	93	99999
0	-	-	-	-	-	-
24	120.000	-	16.425	-	-	-
33	-	-	-	-	-	-
37	-	-	-	-	2.618.860.080	-
55	-	-	3.763.200	817.600	-	-
63	-	-	-	-	143.680	-
65	-	13.192.080	-	-	11.076.480	-
66	202.650	-	3.177.405	-	-	-
73		5.991.040	-	546.560	-	-
75			-	-	-	-
77				670.335	343.635	2.520.974.790
87					52.130	56.140
93						623.336.325

A alteração no *layout* mostrou que o novo custo corresponde a 49,52% do valor do *layout* original.

5 CONCLUSÕES E RECOMENDAÇÕES

5.1 CONCLUSÕES

A determinação do *layout* de uma linha de produção passa pelo processo de definição e quantidade dos produtos a serem produzidos, quais e quantos equipamentos estão disponíveis para processá-los e a área disponível para instalá-los. A consideração dos fatores e como devem ser analisados é de extrema importância.

Diversos autores apontam para o agrupamento celular como solução para a estrutura de produção que caracteriza um *job-shop* como forma de otimização de *layout* utilizando o agrupamento de equipamentos parecidos analisando o fluxo entre os setores definidos e distribuindo os setores em função da combinação encontrada. Este trabalho propôs a metodologia de análise de dados para qualificar as ligações entre cada recurso encontrado em um *job-shop* e aplicação de um modelo matemático para definir a posição dos recursos.

Mesmo em uma linha de produção que possui a característica de *job-shop* que não apresenta uniformidade de seqüências de operações de processamento de produtos, é possível definir um *layout* adequado através da análise das ligações dos recursos apresentadas pelos roteiros de produção.

O estudo considerou os volumes de produtos manufaturados, a quantidade distinta de itens, a lucratividade de cada item, a freqüência de produção, o roteiro que cada produto segue e os equipamentos disponíveis para o cálculo do valor ponderado que classificou os produtos de acordo com estes parâmetros.

Foi utilizado o algoritmo de Levenshtein para classificar a diferença dos roteiros quantitativamente. O algoritmo mostrou que pode haver roteiros utilizando os recursos em comum, porém isto não significa que estes roteiros são iguais. A análise da igualdade entre os roteiros foi desenvolvida utilizando um algoritmo para gerar os dados do diagrama de Pareto que gerou novos roteiros padrões e quais produtos utilizam cada roteiro. A análise dos roteiros identificou os recursos principais para a produção, considerando o valor ponderado dos produtos e a diferença dos roteiros de produção associados a cada produto. As ligações destes recursos foram

submetidas ao modelo matemático LMIP1 que determinou a posição linear dos recursos no *layout*.

Os resultados obtidos utilizando-se os dados da empresa mostraram que a combinação dos recursos no *layout* definido apresenta o menor custo.

Através do estudo foi possível relacionar o volume de vendas, lucratividade, tempos de produção, tempos de espera e fluxo entre recursos na definição do *layout* através de uma análise quantitativa obtida através da aplicação de um modelo matemático. Para validação do modelo pode-se utilizar dados de outros períodos e comparar os novos resultados aos resultados do período estudado.

5.2 SUGESTÕES PARA TRABALHOS FUTUROS

As análises aqui apresentadas podem ser refeitas modificando os períodos analisados para definir possíveis alterações no *layout* ou determinar possíveis sazonalidades na utilização do *layout*. O *layout* proposto pode ser considerado como referencial para análises em períodos menores determinando o ganho em unidades monetárias com a utilização do *layout*. Através do algoritmo de Levenshtein pode-se custear a alteração do *layout* referencial para o novo *layout* proveniente de análises periódicas apontando qual o momento mais adequado para mudar o *layout* que consiste na comparação da diferença do custo do *layout* atual e o ganho do novo *layout*, e o custo da alteração do *layout*.

Pode-se ainda desenvolver um algoritmo que resolva este modelo considerando os limites da planta. Assim o algoritmo utilizaria o custo de cada ligação posicionando primeiramente as ligações de maior valor até a de menor valor, observando para não repetir a utilização da ligação, posicionando os recursos em x e y sendo o y calculado como o centro do maior recurso alocado. Quando o x atingisse o limite da planta, inicia-se um novo x como se fosse uma nova linha com a mesma consideração para o y . Esta resolução pode gerar espaços vazios no *layout* que seriam eliminados no momento da confecção do *layout* final por um especialista que substitui os retângulos pelos recursos.

Outra possibilidade é o aprimoramento do modelo matemático linear utilizado para determinação do eixo y , além do x , otimizando o posicionamento de todos os recursos da planta.

REFERÊNCIAS

- ASKIN, Ronald G. **Modeling and analysis of manufacturing systems**. New Jersey: John Wiley & Sons, 1993.
- BAYBARS, Ilker. **A Survey of Exact Algorithms for the Simple Assembly Line Balancing Problem**. Management Science 32, page 909-932, 1986.
- BENJAAFAR, Saif; Heragu Sunderesh S.; Irani, Shahrukh A. **Next generation factory layouts: Research challenges and recent progress**. Interfaces 32, page 58-78, 2002.
- BOYSEN, Nils, Fliedner, Malte, Scholl, Armin. **A Classification of Assembly Line Balancing Problems**. Arbeits und Diskussionspapiere der Wirtschaftswissenschaftlichen Fakultät der Friedrich-Schiller-Universität Jena, 2006.
- CHUTIMA, Parames; Suphapruksapongse, Haruetai. **Practical Assembly Line Balancing in a Monitor Manufacturing Company**. Bangkok: Tharnmasalnt t. J. Sc. Tech., Vol. 9, No. 2, Abril-Junho de 2004.
- DAITA, S.T., Irani, S.A. & Kotamraju, S. **Algorithms for Production Flow Analysis**. International Journal of Production Research, 1999.
- FRANCIS, Richard L. **Facility Layout and Location**. New Jersey: Prentice-Hall, 1974.
- GOLDBARG, Marco César. **Otimização Combinatória e Programação Linear**. Rio de Janeiro: Elsevier, 2005.
- GOPALAKRISHNAN, B; Weng, Li; Gupta, D. P. **Facilities design using a split departmental layout configuration**. Facilities 21, volume 3/4, páginas 66-73, 2003.
- GROOVER, Mikell P. **Fundamentals of Modern Manufacturing**. New Jersey: John Wiley & Sons, 1996.
- GROOVER, Mikell P. **Automation, Production Systems, and Computer Integrated Manufacturing**. New Jersey: Prentice-Hall, 2000.

HERAGU, S. **Facilities Design**. Boston, PWS Publishing Company, 1997.

HILLIER, Frederick S.; Lieberman, Gerald J. **Introduction to Operations Research**. McGraw-Hill Higher Education, 2001. Sétima edição.

KALPAKJIAN, Serope. **Manufacturing Processes for Engineering Materials**. Menlo Park: Prentice Hall, 1997. Terceira edição.

LEVENSHTEIN, Vladimir. **Efficient Implementation of the Levenshtein-Algorithm, Fault-tolerant Search Technology, Error-tolerant Search Technologies**. <http://www.levenshtein.net/>. Acessado em outubro de 2007.

SCHOLL, Armin; Becker, Christian. **State of-the art exact heuristic solution procedures for simple assembly line balancing**. European Journal of Operations Research 168, pages 666-693, 2006.

TONGE, Fred M. **Summary of a Heuristic Line Balancing Procedure**. Management Science 7, page 21-42, 1960.

TOMPKINS, James A.; White, John A.; Bozer, Yavuz A.; Tanchoco, J. M. A. **Facilities Planning**. Massachusetts: John Willey & Sons Inc, 2003.

ANEXO A – CÓDIGOS FONTES

```

Private Sub Levenshtein_Click()
On Error GoTo Err_Levenshtein_Click
    Dim intI As Integer, intJ As Integer, nQtdOri As Integer, nQtdCmp As Integer,
nVlrCst As Integer
    Dim MatCmp(0 To 20, 0 To 20) As Single
    Dim ProOri As Recordset
    Dim ProCmp As Recordset
    Dim RotOri As Recordset
    Dim RotCmp As Recordset
    Dim CmpLev As Recordset
    Dim ResLev As Recordset
    Dim DB As Database
    Set DB = CurrentDb
    Set ProOri = DB.OpenRecordset("SELECT ClaPon.CodAlt, e075pro.CODROT FROM ClaPon
INNER JOIN e075pro ON ClaPon.CodAlt = e075pro.CodAlt " +
"ORDER BY ClaPon.VlrPon DESC")
    Set CmpLev = DB.OpenRecordset("CmpLev")
    While Not CmpLev.EOF
        CmpLev.Delete
        CmpLev.MoveFirst
    Wend
    While Not ProOri.EOF
        Set RotOri = DB.OpenRecordset("select SeqRot, CodCre from e710sqr where
e710sqr.CodRot = ' " + ProOri![CODROT] + "' order by SeqRot")
        While Not RotOri.EOF
            RotOri.MoveNext
        Wend
        nQtdOri = RotOri.RecordCount
        If nQtdOri > 0 Then
            RotOri.MoveFirst
            Set ProCmp = DB.OpenRecordset("SELECT ClaPon.CodAlt, e075pro.CODROT FROM
ClaPon INNER JOIN e075pro ON ClaPon.CodAlt = e075pro.CodAlt " +
"where ClaPon.CodAlt <> ' " + ProOri![CodAlt] + "' ORDER BY ClaPon.VlrPon DESC")
            While Not ProCmp.EOF
                Set ResLev = DB.OpenRecordset("select AltOri from CmpLev where AltOri
= ' " + ProCmp![CodAlt] + "' and AltCmp = ' " + ProOri![CodAlt] + "'")
                If ResLev.EOF Then
                    Set RotCmp = DB.OpenRecordset("select SeqRot, CodCre from e710sqr
where e710sqr.CodRot = ' " + ProCmp![CODROT] + "' order by SeqRot")
                    While Not RotCmp.EOF
                        RotCmp.MoveNext
                    Wend
                    nQtdCmp = RotCmp.RecordCount
                    If nQtdCmp > 0 Then
                        'inicializa a matriz original
                        RotCmp.MoveFirst
                        For i = 0 To 20
                            For j = 0 To 20
                                MatCmp(i, j) = 0
                            Next j
                        Next i
                        For i = 0 To nQtdOri
                            MatCmp(i, 0) = i
                        Next i
                        For j = 0 To nQtdCmp
                            MatCmp(0, j) = j
                        Next j
                        RotOri.MoveFirst
                        For i = 1 To nQtdOri

```

Figura 8 – Código fonte da implementação do algoritmo de Levenshtein em *Visual Basic for Applications*.


```

        RotCmp.MoveFirst
        For j = 1 To nQtdCmp
            If RotOri![CodCre] = RotCmp![CodCre] Then
                nVlrCst = 0
            Else
                nVlrCst = 1
            End If
            MatCmp(i, j) = Minimo((MatCmp(i - 1, j) + 1),
MatCmp(i, j - 1) + 1, MatCmp(i - 1, j - 1) + nVlrCst)
            RotCmp.MoveNext
        Next j
        RotOri.MoveNext
    Next i
    CmpLev.AddNew
    CmpLev![AltOri] = ProOri![CodAlt]
    CmpLev![AltCmp] = ProCmp![CodAlt]
    CmpLev![VlrCst] = MatCmp(nQtdOri, nQtdCmp)
    CmpLev.Update
End If
End If
ProCmp.MoveNext
Wend
End If
ProOri.MoveNext
Wend

On Error Resume Next

Exit_Levenshtein_Click:
Exit Sub

Err_Levenshtein_Click:
MsgBox Err.Description
Resume Exit_Levenshtein_Click

End Sub

Private Function Minimo(ByVal VlrPri As Integer, ByVal VlrSec As Integer, ByVal
VlrTer As Integer) As Single

    Dim MinVal As Single

    MinVal = VlrPri
    If VlrSec < MinVal Then
        MinVal = VlrSec
    End If
    If VlrTer < MinVal Then
        MinVal = VlrTer
    End If
    Minimo = MinVal

End Function

```

Figura 8 – Código fonte da implementação do algoritmo de Levenshtein em *Visual Basic for Applications* (Continuação).

```

Private Sub Pareto_Click()
On Error GoTo Err_Pareto_Click
    Dim iQtdOpe As Integer, iUltRot As Integer, i As Integer, nQtdCmp As Integer,
nVlrCst As Integer
    Dim CodCre(1 To 20) As Integer
    Dim e075pro As Recordset
    Dim e710sqr As Recordset
    Dim FluPar As Recordset
    Dim DB As Database
    Set DB = CurrentDb
    Set e075pro = DB.OpenRecordset("SELECT ClaPon.VlrPon, e075pro.CodRot,
e075pro.AltRot " + _
                                "FROM ClaPon INNER JOIN e075pro ON ClaPon.CodAlt =
e075pro.CodAlt " + _
                                "WHERE (((e075pro.CODROT)>' ') order by
ClaPon.VlrPon, e075pro.CodRot")
    Set FluPar = DB.OpenRecordset("SELECT AltRot FROM FluPar")
    While Not FluPar.EOF
        FluPar.Delete
        FluPar.MoveFirst
    Wend
    iUltRot = 1
    While Not e075pro.EOF
        Set e710sqr = DB.OpenRecordset("select e725cre.CreAlt from e710sqr inner join
e725cre on (e710sqr.CodCre = e725cre.CodCre) " + _
                                        "and (e710sqr.CodEmp = e725cre.CodEmp) where
CodRot = '" + e075pro![CODROT] + _
                                        "' order by e710sqr.CodEtg, e710sqr.SeqRot")

        iQtdOpe = 0
        For i = 1 To 20
            CodCre(i) = 0
        Next i
        While Not e710sqr.EOF
            iQtdOpe = iQtdOpe + 1
            CodCre(iQtdOpe) = e710sqr![CreAlt]
            e710sqr.MoveNext
        Wend
        Set FluPar = DB.OpenRecordset("select * from FluPar where Cre001 = " +
Format(CodCre(1), "###0") + " and Cre002 = " + Format(CodCre(2), "###0") + _
                                " and Cre003 = " + Format(CodCre(3), "###0") +
" and Cre004 = " + Format(CodCre(4), "###0") + _
                                " and Cre005 = " + Format(CodCre(5), "###0") +
" and Cre006 = " + Format(CodCre(6), "###0") + _
                                " and Cre007 = " + Format(CodCre(7), "###0") +
" and Cre008 = " + Format(CodCre(8), "###0") + _
                                " and Cre009 = " + Format(CodCre(9), "###0") +
" and Cre010 = " + Format(CodCre(10), "###0") + _
                                " and Cre011 = " + Format(CodCre(11), "###0") +
" and Cre012 = " + Format(CodCre(12), "###0") + _
                                " and Cre013 = " + Format(CodCre(13), "###0") +
" and Cre014 = " + Format(CodCre(14), "###0") + _
                                " and Cre015 = " + Format(CodCre(15), "###0") +
" and Cre016 = " + Format(CodCre(16), "###0") + _
                                " and Cre017 = " + Format(CodCre(17), "###0") +
" and Cre018 = " + Format(CodCre(18), "###0") + _
                                " and Cre019 = " + Format(CodCre(19), "###0") +
" and Cre020 = " + Format(CodCre(20), "###0"))
        If FluPar.EOF Then
            FluPar.AddNew
            FluPar![AltRot] = iUltRot
            FluPar![Cre001] = CodCre(1)
            FluPar![Cre002] = CodCre(2)
            FluPar![Cre003] = CodCre(3)
            FluPar![Cre004] = CodCre(4)
            FluPar![Cre005] = CodCre(5)
            FluPar![Cre006] = CodCre(6)
            FluPar![Cre007] = CodCre(7)
            FluPar![Cre008] = CodCre(8)
        End If
    End While
End Sub

```

Figura 9 – Código fonte da implementação do algoritmo para análise de Pareto em *Visual Basic for Applications*.

```
FluPar![Cre009] = CodCre(9)
FluPar![Cre010] = CodCre(10)
FluPar![Cre011] = CodCre(11)
FluPar![Cre012] = CodCre(12)
FluPar![Cre013] = CodCre(13)
FluPar![Cre014] = CodCre(14)
FluPar![Cre015] = CodCre(15)
FluPar![Cre016] = CodCre(16)
FluPar![Cre017] = CodCre(17)
FluPar![Cre018] = CodCre(18)
FluPar![Cre019] = CodCre(19)
FluPar![Cre020] = CodCre(20)
FluPar![LigTot] = 1
Select Case e075pro![VlrPon]
Case 1
    FluPar![Lig001] = 1
Case 2
    FluPar![Lig002] = 1
Case 3
    FluPar![Lig003] = 1
Case 4
    FluPar![Lig004] = 1
Case 5
    FluPar![Lig005] = 1
Case 6
    FluPar![Lig006] = 1
Case 7
    FluPar![Lig007] = 1
Case 8
    FluPar![Lig008] = 1
Case 9
    FluPar![Lig009] = 1
Case 10
    FluPar![Lig010] = 1
Case 11
    FluPar![Lig011] = 1
Case 12
    FluPar![Lig012] = 1
Case 13
    FluPar![Lig013] = 1
Case 14
    FluPar![Lig014] = 1
Case 15
    FluPar![Lig015] = 1
Case 16
    FluPar![Lig016] = 1
Case 17
    FluPar![Lig017] = 1
Case 18
    FluPar![Lig018] = 1
Case 19
    FluPar![Lig019] = 1
Case 20
    FluPar![Lig020] = 1
End Select
e075pro.Edit
e075pro![AltRot] = FluPar![AltRot]
e075pro.Update
FluPar.Update
iUltRot = iUltRot + 1
Else
    FluPar.Edit
    FluPar![LigTot] = FluPar![LigTot] + 1
    Select Case e075pro![VlrPon]
    Case 1
        FluPar![Lig001] = FluPar![Lig001] + 1
```

Figura 9 – Código fonte da implementação do algoritmo para análise de Pareto em *Visual Basic for Applications* (Continuação).

```
Case 2
    FluPar![Lig002] = FluPar![Lig002] + 1
Case 3
    FluPar![Lig003] = FluPar![Lig003] + 1
Case 4
    FluPar![Lig004] = FluPar![Lig004] + 1
Case 5
    FluPar![Lig005] = FluPar![Lig005] + 1
Case 6
    FluPar![Lig006] = FluPar![Lig006] + 1
Case 7
    FluPar![Lig007] = FluPar![Lig007] + 1
Case 8
    FluPar![Lig008] = FluPar![Lig008] + 1
Case 9
    FluPar![Lig009] = FluPar![Lig009] + 1
Case 10
    FluPar![Lig010] = FluPar![Lig010] + 1
Case 11
    FluPar![Lig011] = FluPar![Lig011] + 1
Case 12
    FluPar![Lig012] = FluPar![Lig012] + 1
Case 13
    FluPar![Lig013] = FluPar![Lig013] + 1
Case 14
    FluPar![Lig014] = FluPar![Lig014] + 1
Case 15
    FluPar![Lig015] = FluPar![Lig015] + 1
Case 16
    FluPar![Lig016] = FluPar![Lig016] + 1
Case 17
    FluPar![Lig017] = FluPar![Lig017] + 1
Case 18
    FluPar![Lig018] = FluPar![Lig018] + 1
Case 19
    FluPar![Lig019] = FluPar![Lig019] + 1
Case 20
    FluPar![Lig020] = FluPar![Lig020] + 1
End Select
e075pro.Edit
e075pro![AltRot] = FluPar![AltRot]
e075pro.Update
FluPar.Update
End If
e075pro.MoveNext
Wend

Exit_Pareto_Click:
    Exit Sub

Err_Pareto_Click:
    MsgBox Err.Description
    Resume Exit_Pareto_Click

End Sub
```

Figura 9 – Código fonte da implementação do algoritmo para análise de Pareto em *Visual Basic for Applications* (Continuação).

```

Private Sub Levenshtein_Click()
On Error GoTo Err_Levenshtein_Click
Private Sub Balanceamento_Click()
On Error GoTo Err_Balanceamento_Click
    Dim Cns_RotBal As Recordset
    Dim Tab_RotBal As Recordset
    Dim RotBal2 As Recordset
    Dim e720opr As Recordset
    Dim e725cre As Recordset
    Dim nQtdCreAnt As Integer
    Dim nCapSmtAnt As Integer
    Dim nQtdTmpAnt As Integer
    Dim nTmpPrpAnt As Double
    Dim nEspTmpAnt As Double
    Dim nEspAbsAnt As Double
    Dim nEspTmpRet As Double
    Dim nEspAbsRet As Double
    Dim iCreAlt As Integer
    Dim nQtdCre As Integer
    Dim nQtdTmp As Integer
    Dim nTmpCre As Double
    Dim nTmpPro As Double
    Dim aTipEspRet As String
    Dim aTipExp As String
    Dim aRotAnt As String
    Dim DB As Database
    Dim sRotAnt As String
    Set DB = CurrentDb
    Set Tab_RotBal = DB.OpenRecordset("Balanceamento_Roteiro")
    Set Cns_RotBal = DB.OpenRecordset("Consulta_Roteiro_Balanceamento")

    'Zerar tabela de balanceamento e posicionar na primeira posição da tabela de
roteiros
    While Not Tab_RotBal.EOF
        Tab_RotBal.Delete
        Tab_RotBal.MoveFirst
    Wend
    Cns_RotBal.MoveFirst

Passo_1:
'Carregar o roteiro do produto a ser analisado acrescentando informações de
quantos recursos
'podem ser utilizados para cada operação, e a quantidade de produtos que fica
pronta
'com a quantidade de recursos e a quantidade pelo tempo necessário.
If Not Cns_RotBal.EOF Then
    sRotAnt = Cns_RotBal![CODROT]
End If
While Not Cns_RotBal.EOF
    Tab_RotBal.AddNew
    Tab_RotBal![CodEmp] = Cns_RotBal![CodEmp]
    Tab_RotBal![CODROT] = Cns_RotBal![CODROT]
    Tab_RotBal![CodEtg] = Cns_RotBal![CodEtg]
    Tab_RotBal![SfxEtr] = Cns_RotBal![SfxEtr]
    Tab_RotBal![SeqRot] = Cns_RotBal![SeqRot]
    Tab_RotBal![SfxSeq] = Cns_RotBal![SfxSeq]
    Tab_RotBal![CodAlt] = Cns_RotBal![CodAlt]
    Tab_RotBal![CapSmt] = Cns_RotBal![CapSmt]
    Tab_RotBal![TmpPrp] = Cns_RotBal![TmpPrp]
    Tab_RotBal![BalCnf] = True
    If Cns_RotBal![CodCre] > " " Then
        Tab_RotBal![CodCre] = Cns_RotBal![CodCre]
    Else
        Set e720opr = DB.OpenRecordset("select CodCre from e720opr where CodEmp =
" + Format(Cns_RotBal![CodEmp], "####") + _
" and CodOpr = " + Cns_RotBal![CodOpr] + "'")
        If Not e720opr.EOF Then
            Tab_RotBal![CodCre] = e720opr![CodCre]

```

Figura 10 – Código fonte da implementação do algoritmo para cálculo do tempo de espera.

```

        End If
        e720opr.Close
    End If
    Set e725cre = DB.OpenRecordset("select CreAlt, LimCap from e725cre where
CodEmp = " + Format(Cns_RotBal![CodEmp], "###") + _
" and CodCre = '" + Tab_RotBal![CodCre] + "'")
    If Not e725cre.EOF Then
        iCreAlt = e725cre![CreAlt]
        Tab_RotBal![LimCap] = e725cre![LimCap]
    End If
    e725cre.Close
    If Tab_RotBal![LimCap] = "M" Then
        Set e725cre = DB.OpenRecordset("select * from e725cre where CodEmp = " +
Format(Cns_RotBal![CodEmp], "###") + _
" and CreAlt = " + Format(iCreAlt, "###"))
        While Not e725cre.EOF
            e725cre.MoveNext
        Wend
        Tab_RotBal![QtdCre] = e725cre.RecordCount
        Tab_RotBal![CreDsp] = e725cre.RecordCount
    Else
        Tab_RotBal![CreDsp] = 1
        Tab_RotBal![QtdCre] = 1
    End If
    Tab_RotBal![QtdTmp] = Tab_RotBal![QtdCre] * Tab_RotBal![CapSmt]
    Tab_RotBal!.Update
    Cns_RotBal.MoveNext
Wend
Passo_2:
'Para cada operação é calculado o tempo de espera que consiste em subtrair o
tempo de operação
'do recurso atual pela multiplicação da quantidade de produtos que a operação
atual pelo tempo de
'processamento da operação anterior (coluna K da planilha).
'Passar cada operação verificando se o tempo de espera é positivo, quem espera é
o produto;
'se o tempo de espera é negativo, quem espera é o recurso; por último, se o tempo
de espera é zero, ninguém espera
Set RotBal2 = DB.OpenRecordset("select * from Balanceamento_Roteiro order by
CodAlt, CodRot, CodEtg, SeqRot")
RotBal2.MoveFirst
If Not RotBal2.EOF Then
    RotBal2.Edit
    RotBal2![EspTmp] = 0
    RotBal2![EspAbs] = 0
    RotBal2![TipEsp] = "Ninguém"
    RotBal2.Update
    nQtdCreAnt = RotBal2![QtdCre]
    nCapSmtAnt = RotBal2![CapSmt]
    nTmpPrpAnt = RotBal2![TmpPrp]
    nQtdTmpAnt = RotBal2![QtdTmp]
    aRotAnt = RotBal2![CODROT]
    RotBal2.MoveNext
End If
While Not RotBal2.EOF
    RotBal2.Edit
    If RotBal2![CODROT] <> aRotAnt Then
        RotBal2![EspTmp] = 0
        RotBal2![EspAbs] = 0
        RotBal2![TipEsp] = 0
        aRotAnt = RotBal2![CODROT]
    Else
        Call CalEsp(RotBal2![QtdCre], RotBal2![CapSmt], RotBal2![TmpPrp],
RotBal2![QtdTmp], nQtdCreAnt, nCapSmtAnt, nTmpPrpAnt, nQtdTmpAnt, nEspTmpRet,
nEspAbsRet, aTipEspRet)
        RotBal2![EspTmp] = nEspTmpRet
        RotBal2![EspAbs] = nEspAbsRet
        RotBal2![TipEsp] = aTipEspRet
    End If
End While

```

Figura 10 – Código fonte da implementação do algoritmo para cálculo do tempo de espera (continuação).

```

End If
RotBal2.Update
nQtdCreAnt = RotBal2![QtdCre]
nCapSmtAnt = RotBal2![CapSmt]
nTmpPrpAnt = RotBal2![TmpPrp]
nQtdTmpAnt = RotBal2![QtdTmp]
RotBal2.MoveNext
Wend

Passo_3:
'Para o produto esperando pode-se aumentar o número de recursos da operação,
'caso existam recursos disponíveis, ou pode-se diminuir a produção da operação
anterior.
RotBal2.Close
Set RotBal2 = DB.OpenRecordset("select * from Balanceamento_Roteiro order by
CodAlt, CodRot, CodEtg, SeqRot")
RotBal2.MoveFirst
While Not RotBal2.EOF
If RotBal2![CODROT] <> aRotAnt Then
RotBal2.Edit
RotBal2![EspTmp] = 0
RotBal2![EspAbs] = 0
RotBal2![TipEsp] = "Ninguém"
aRotAnt = RotBal2![CODROT]
RotBal2.Update
Else
If RotBal2![EspAbs] = 0 Then
RotBal2.Edit
RotBal2![BalCnf] = True
RotBal2.Update
RotBal2.MoveNext
End If
If Not RotBal2.EOF Then
If RotBal2![TipEsp] = "Recurso" Then
If RotBal2![QtdCre] > 1 Then
RotBal2.Edit
RotBal2![QtdCre] = RotBal2![QtdCre] - 1
RotBal2.Update
RotBal2.MovePrevious
nQtdCreAnt = RotBal2![QtdCre]
nCapSmtAnt = RotBal2![CapSmt]
nTmpPrpAnt = RotBal2![TmpPrp]
nQtdTmpAnt = RotBal2![QtdTmp]
RotBal2.MoveNext
RotBal2.Edit
RotBal2![QtdTmp] = RotBal2![QtdCre] * RotBal2![CapSmt]
Call CalEsp(RotBal2![QtdCre], RotBal2![CapSmt],
RotBal2![TmpPrp], RotBal2![QtdTmp], nQtdCreAnt, nCapSmtAnt, nTmpPrpAnt, nQtdTmpAnt,
nEspTmpRet, nEspAbsRet, aTipEspRet)
RotBal2![EspTmp] = nEspTmpRet
RotBal2![EspAbs] = nEspAbsRet
RotBal2![TipEsp] = aTipEspRet
RotBal2.Update
Else
'verifica a operação anterior, a possibilidade de aumentar os
recursos utilizados para diminuir a espera da última operação analisada.
RotBal2.MovePrevious
If RotBal2.BOF Then
RotBal2.MoveNext
End If
If RotBal2![CreDsp] > RotBal2![QtdCre] Then
RotBal2.Edit
RotBal2![QtdCre] = RotBal2![QtdCre] + 1
RotBal2.Update
RotBal2.MovePrevious
nQtdCreAnt = RotBal2![QtdCre]
nCapSmtAnt = RotBal2![CapSmt]
nTmpPrpAnt = RotBal2![TmpPrp]

```

Figura 10 – Código fonte da implementação do algoritmo para cálculo do tempo de espera (continuação).

```

        nQtdTmpAnt = RotBal2![QtdTmp]
        RotBal2.MoveNext
        RotBal2.Edit
        Call CalEsp(RotBal2![QtdCre], RotBal2![CapSmt],
RotBal2![TmpPrp], RotBal2![QtdTmp], nQtdCreAnt, nCapSmtAnt, nTmpPrpAnt, nQtdTmpAnt,
nEspTmpRet, nEspAbsRet, aTipEspRet)
        RotBal2![EspTmp] = nEspTmpRet
        RotBal2![EspAbs] = nEspAbsRet
        RotBal2![TipEsp] = aTipEspRet
        RotBal2.Update
        'salva dados para análise da próxima operação.
        nQtdCreAnt = RotBal2![QtdCre]
        nCapSmtAnt = RotBal2![CapSmt]
        nTmpPrpAnt = RotBal2![TmpPrp]
        nQtdTmpAnt = RotBal2![QtdTmp]
        RotBal2.MoveNext
        'verifica espera da operação atualmente analisada (última
operação analisada).
        RotBal2.Edit
        Call CalEsp(RotBal2![QtdCre], RotBal2![CapSmt],
RotBal2![TmpPrp], RotBal2![QtdTmp], nQtdCreAnt, nCapSmtAnt, nTmpPrpAnt, nQtdTmpAnt,
nEspTmpRet, nEspAbsRet, aTipEspRet)
        RotBal2![EspTmp] = nEspTmpRet
        RotBal2![EspAbs] = nEspAbsRet
        RotBal2![TipEsp] = aTipEspRet
        RotBal2.Update
    Else
        RotBal2.Edit
        RotBal2![BalCnf] = True
        RotBal2.Update
        RotBal2.MoveNext
    End If
    RotBal2.MoveNext
End If
Else
    If RotBal2![TipEsp] = "Produto" Then
        If RotBal2![QtdCre] < RotBal2![CreDsp] Then
            RotBal2.Edit
            RotBal2![QtdCre] = RotBal2![QtdCre] + 1
            RotBal2.Update
            RotBal2.MovePrevious
            'salva dados para análise da próxima operação.
            nQtdCreAnt = RotBal2![QtdCre]
            nCapSmtAnt = RotBal2![CapSmt]
            nTmpPrpAnt = RotBal2![TmpPrp]
            nQtdTmpAnt = RotBal2![QtdTmp]
            RotBal2.MoveNext
            'verifica espera da operação atualmente analisada (última
operação analisada).
            RotBal2.Edit
            Call CalEsp(RotBal2![QtdCre], RotBal2![CapSmt],
RotBal2![TmpPrp], RotBal2![QtdTmp], nQtdCreAnt, nCapSmtAnt, nTmpPrpAnt, nQtdTmpAnt,
nEspTmpRet, nEspAbsRet, aTipEspRet)
            RotBal2![EspTmp] = nEspTmpRet
            RotBal2![EspAbs] = nEspAbsRet
            RotBal2![TipEsp] = aTipEspRet
            RotBal2.Update
        Else
            RotBal2.MovePrevious
            If RotBal2![QtdCre] > 1 Then
                RotBal2.MovePrevious
                nQtdCreAnt = RotBal2![QtdCre]
                nCapSmtAnt = RotBal2![CapSmt]
                nTmpPrpAnt = RotBal2![TmpPrp]
                nQtdTmpAnt = RotBal2![QtdTmp]
                RotBal2.MoveNext
                nQtdCre = RotBal2![QtdCre]
                nEspAbsAnt = RotBal2![EspAbs]
            End If
        End If
    End If
End If

```

Figura 10 – Código fonte da implementação do algoritmo para cálculo do tempo de espera (continuação).


```

        nEspTmpAnt = RotBal2![EspTmp]
        While nEspAbsAnt >= RotBal2![EspAbs] And nQtdCre > 1
            nQtdCre = nQtdCre - 1
            nQtdTmp = nQtdCre * RotBal2![CapSmt]
            Call CalEsp(nQtdCre, RotBal2![CapSmt],
RotBal2![TmpPrp], nQtdTmp, nQtdCreAnt, nCapSmtAnt, nTmpPrpAnt, nQtdTmpAnt,
nEspTmpRet, nEspAbsRet, aTipEspRet)
            If nEspAbsAnt > nEspAbsRet Then
                nEspAbsAnt = nEspAbsRet
                nEspTmpAnt = nEspTmpRet
            End If
        Wend
        RotBal2.Edit
        RotBal2![EspAbs] = nEspAbsAnt
        RotBal2![EspTmp] = nEspTmpAnt
        RotBal2![QtdCre] = nQtdCre
        RotBal2![QtdTmp] = RotBal2![QtdCre] *
RotBal2![CapSmt]
        RotBal2.Update
    End If
    'salva dados para análise da próxima operação.
    nQtdCreAnt = RotBal2![QtdCre]
    nCapSmtAnt = RotBal2![CapSmt]
    nTmpPrpAnt = RotBal2![TmpPrp]
    nQtdTmpAnt = RotBal2![QtdTmp]
    RotBal2.MoveNext
    'verifica espera da operação atualmente analisada (última
operação analisada).
    RotBal2.Edit
    Call CalEsp(RotBal2![QtdCre], RotBal2![CapSmt],
RotBal2![TmpPrp], RotBal2![QtdTmp], nQtdCreAnt, nCapSmtAnt, nTmpPrpAnt, nQtdTmpAnt,
nEspTmpRet, nEspAbsRet, aTipEspRet)
    RotBal2![EspTmp] = nEspTmpRet
    RotBal2![EspAbs] = nEspAbsRet
    RotBal2![TipEsp] = aTipEspRet
    RotBal2.Update
    End If
    RotBal2.MoveNext
    End If
    End If
    End If
    End If
    Wend

Exit_Balanceamento_Click:
    Exit Sub

Err_Balanceamento_Click:
    MsgBox Err.Description
    Resume
'    Exit_Balanceamento_Click

End Sub
Sub CalEsp(nQtdCreAtu As Integer, nCapSmtAtu As Integer, nTmpPrpAtu As Double,
nQtdTmpAtu As Integer, nQtdCreAnt As Integer, _
nCapSmtAnt As Integer, nTmpPrpAnt As Double, nQtdTmpAnt As Integer, nEspTmpRet As
Double, nEspAbsRet As Double, aTipEspRet As String)
    Dim nTmpCre As Double
    Dim nTmpPro As Double

    If nQtdCreAtu = nQtdCreAnt And nCapSmtAtu = nCapSmtAnt And nTmpPrpAtu =
nTmpPrpAnt And nQtdTmpAtu = nQtdTmpAnt Then
        nEspTmpRet = 0
        nEspAbsRet = 0
        aTipEspRet = 0
    Else
        If nQtdCreAtu <> nQtdCreAnt Then

```

Figura 10 – Código fonte da implementação do algoritmo para cálculo do tempo de espera (continuação).

```

        nEspTmpRet = (nTmpPrpAtu - ((nCapSmtAtu * nTmpPrpAnt) / (nCapSmtAnt /
nQtdCreAnt))) * (nQtdCreAtu - nQtdCreAnt)
    Else
        If nQtdCreAtu = nQtdCreAnt And nCapSmtAtu > nCapSmtAnt Then
            nEspTmpRet = nTmpPrpAtu - nTmpPrpAnt
        Else
            If nCapSmtAtu > nCapSmtAnt Then
                nEspTmpRet = nTmpPrpAtu - ((nCapSmtAtu * nTmpPrpAnt) /
(nCapSmtAnt / nQtdCreAnt))
            Else
                If nCapSmtAtu < nTmpPrpAtu And nTmpPrpAtu < nTmpPrpAnt Then
                    nTmpCre = nTmpPrpAnt - (nTmpPrpAtu * nCapSmtAnt)
                    nTmpPro = nTmpPrpAtu
                    If nTmpCre > nTmpPro Then
                        nEspTmpRet = nTmpCre * -1
                    Else
                        nEspTmpRet = nTmpPro
                    End If
                Else
                    nEspTmpRet = nTmpPrpAtu - ((nCapSmtAtu * nTmpPrpAnt) /
(nCapSmtAnt / nQtdCreAnt))
                End If
            End If
        End If
    End If
End If
If nEspTmpRet < 0 Then
    nEspAbsRet = nEspTmpRet * -1
    aTipEspRet = "Recurso"
Else
    nEspAbsRet = nEspTmpRet
    If nEspTmpRet = 0 Then
        aTipEspRet = "Ninguém"
    Else
        aTipEspRet = "Produto"
    End If
End If
End Sub

```

Figura 10 – Código fonte da implementação do algoritmo para cálculo do tempo de espera (continuação).

This document was created with Win2PDF available at <http://www.win2pdf.com>.
The unregistered version of Win2PDF is for evaluation or non-commercial use only.
This page will not be added after purchasing Win2PDF.

Livros Grátis

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

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

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

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

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