

Método para definição de *layout* em sistemas *job-shop* baseado em dados históricos

Maurício Tomelin^{a,*}, João Carlos Colmenero^b

^{a,*}mauricio.tomelin@hotmail.com, UTFPR-PG, Brasil

^bcolmenero@utfpr.edu.br, UTFPR-PG, Brasil

Resumo

O presente trabalho propõe um método para a solução do problema de definição de *layout* tipo *job-shop* baseado em dados históricos de produção, recursos disponíveis e características de produção. Os parâmetros utilizados foram a quantidade total e a frequência de vendas, a lucratividade, o tempo total de processamento e o tempo de espera de cada produto, gerando assim a classificação dos produtos. Os roteiros de produção foram comparados e, em função dos parâmetros, foi determinado o grau de importância de cada ligação entre os recursos. As ligações foram submetidas a um modelo matemático para a determinação da posição de cada recurso no *layout*. O método desenvolvido foi aplicado a uma empresa de usinagem com característica de *job-shop*. Os dados submetidos ao método permitiram uma sugestão de *layout* que tem como objetivo minimizar a distância dos recursos de processamento.

Palavras-chave

Otimização. *Layout*. Modelo matemático. *Job-shop*.

1. Introdução

A disposição de recursos de produção em uma instalação (*layout*) afeta diretamente os custos de produção e a produtividade. Uma alocação de recursos racional contribui significativamente para o aumento da eficiência das operações e reduções dos custos de movimentação (DRIRA; PIERREVAL; HAJRI-GABOUJ, 2007).

Os *layouts* de produção industrial são geralmente definidos de acordo com o fluxo do produto na linha de produção em quatro categorias: *layout* fixo, *layout* por processo, *layout* por produto e *layout* celular. Cada tipo de *layout* apresenta um conjunto de vantagens e limitações referentes a tempo de processamento, trabalho em processo, flexibilidade do produto e da demanda, utilização de maquinário e de operadores, especialização dos operadores e custos de manutenção (FRANCIS; MCGINNIS Jr; WHITE, 1992).

O *layout* por processo (*job-shop*) é o mais utilizado quando muitos produtos distintos são fabricados e a demanda não é estável. Contudo, as práticas de produção *just-in-time* (produção puxada e pequenos lotes) acentuam as desvantagens

apresentadas pelo *layout* por processo: aumento do custo de movimentação de materiais, maior estoque em processo e maior tempo total de produção (FRAMINAN, 2007).

A otimização de *layout* procura melhorar a utilização dos recursos disponíveis através de possibilidades de combinações que maximizem ou minimizem algum objetivo estabelecido (por exemplo, minimização do custo de movimentação). Askin e Standridge (1993), Benjaafar, Heragu e Irani (2002), Daita, Irani e Kotamraju (1999), Francis e White (1974), Heragu (1997) e Tompkins et al. (2003) propõem diversos métodos de otimização de *layout* por meio da qualificação das ligações dos recursos determinada pelo fluxo dos produtos da linha de produção analisada. Essa tarefa está relacionada à qualidade dos dados de entrada coletados.

Com o objetivo de minimizar o tempo total de produção e implementar as vantagens do *layout* por produto, Guinet e Legrand (1998) e Framinan (2007) propuseram modelos para definir um *layout* de produto para um *job-shop*. Contudo, tais modelos não permitem o retorno do produto para um determinado recurso (comum em *job-shops*).

O presente trabalho propõe um método para definição de *layout* de processo (*job-shop*) baseado na classificação do grau de importância de dados históricos de volume e frequência de vendas, lucratividade, tempos de operação e tempos de espera entre as operações de fabricação e modelagem matemática para a determinação do posicionamento dos recursos.

2. Revisão bibliográfica

2.1. *Layout*

O *layout* de instalações industriais é a definição da disposição de recursos de produção e sua interação em um espaço delimitado. A disposição considera as dimensões dos recursos e as dimensões necessárias para operação, manutenção, abastecimento de produtos a processar e escoamento de produtos processados. A interação leva em conta a passagem de produtos e/ou informações de um recurso para outro, as alternativas de transporte, a prioridade ou importância das interações e os custos. Na definição do *layout* de produção são observados as distâncias entre recursos e o fluxo de produção de modo a evitar a formação de gargalos devido a arranjos que bloqueiem ou dificultem o fluxo dos produtos entre os recursos.

Baseado na programação e volumes de produção, combinação de produtos (suas variações) e processos, Francis e White (1974) definem os seguintes objetivos principais para o estudo da otimização de *layout* de uma instalação:

- minimizar o tempo total de produção;
- melhorar o aproveitamento de espaço físico;
- reduzir o custo de manuseio de materiais;
- diminuir a variação de tipos de equipamentos de manuseio de materiais;
- minimizar os investimentos em equipamentos;
- melhorar a segurança e o conforto dos colaboradores;
- preservar a flexibilidade do arranjo dos equipamentos e operações;
- otimizar o processo de manufatura;
- otimizar a estrutura organizacional.

Uma vez que a definição de *layout* considera os produtos a serem produzidos, o processo de produção e os volumes programados de produção (FRANCIS; WHITE, 1974; GROOVER, 2000; TOMPKINS et al., 2003), os objetivos do *layout* variam conforme o produto e/ou processo em questão.

O *layout* pode ser classificado de acordo com os seguintes critérios:

- fluxo do(s) produto(s) na instalação: (i) posição fixa, (ii) *layout* orientado ao produto ou *flow-shop*, (iii) *layout* orientado ao processo ou *job-shop*, e (iv) *layout* orientado para agrupamento tecnológico ou celular (ASKIN; STANDRIDGE, 1993; GOPALAKRISHNAN; WENG; GUPTA, 2003; GROOVER, 2000; TOMPKINS et al., 2003);
- quantidade e frequência de produção: (i) *layout* em *job-shop*, (ii) produção em lotes, (iii) manufatura celular, (iv) linha de produção com fluxo contínuo e (v) produção em massa (GROOVER, 2000; TOMPKINS et al., 2003);
- finalidade da instalação: (i) prestação de serviços, (ii) industriais e (iii) armazéns (HERAGU, 1997);
- natureza da instalação: (i) novas instalações, (ii) rearranjos de instalações e (iii) ampliações de instalações (FRANCIS; WHITE, 1974).

Os métodos para o desenvolvimento de *layouts* são classificados de acordo com seu ponto de partida (TOMPKINS et al., 2003): construção ou melhoria. A construção consiste no desenvolvimento do *layout* novo, ao passo que a melhoria trabalha com um *layout* existente e visa aumentar ou rearranjar a disposição de equipamentos ou o fluxo do(s) produto(s).

Conceitos baseados em métodos desenvolvidos há vários anos, tais como o *Apple's Plant Layout Procedure* (APLP), o *Reed's Plant Layout Procedure* (RPLP) e o *Muther's Systematic Layout Planning Procedure* (SLP), são utilizados em várias metodologias atualmente. Os métodos são divididos em etapas de coleta e análise dos dados, definição de equipamentos, definição e escolha do *layout*, e aplicação e evolução do *layout* (TOMPKINS et al., 2003; FRANCIS; WHITE, 1974; SULE, 2009). Contudo, as soluções apresentadas por esses métodos em geral não atendem a flexibilidade desejada pelos administradores das empresas (BENJAFFAR; HERAGU; IRANI, 2002).

Askin e Standridge (1993) classificam os modelos de manufatura em modelos físicos e modelos matemáticos. Os modelos físicos são usados há vários anos e mostram, por meio de desenhos bi ou tridimensionais, a forma e a disposição de construções e equipamentos, permitindo assim a visualização do ambiente proposto de modo mais objetivo e prático do que uma simples descrição textual. Os modelos matemáticos são compostos de expressões matemáticas (função objetivo e restrições) que representam a realidade de um dado problema.

Tompkins et al. (2003) classificam os modelos, conforme as funções objetivo utilizadas no modelo matemático, em (i) minimização da somatória dos fluxos multiplicados pela distância percorrida, similar ao problema de designação quadrática (*Quadratic Assignment Problem* – QAP), ou (ii) maximização de algum indicador determinado. A minimização do custo de cada ligação entre recursos pode ser expressa pela Equação 1:

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

onde:

- z = custo de movimentação entre equipamentos ou setores;
- m = número de setores;
- i e j = indexadores dos setores analisados;
- f_{ij} = fluxo de transportes do setor i para o setor j . É a quantidade de vezes que um produto é transportado do setor i para o setor j ;
- c_{ij} = custo unitário do transporte entre os setores;
- d_{ij} = distância entre os setores i e j .

Modelos matemáticos para a determinação do melhor posicionamento dos recursos para a redução de custos são utilizados na definição do *layout* de sistemas de manufatura (HERAGU, 1997; TOPKINS et al., 2003). Segundo Heragu (1997), o problema de otimização de *layout* pode ser classificado em dois tipos: linha simples e linhas múltiplas. O problema do tipo linha simples considera a disposição dos recursos apenas em um eixo vertical ou horizontal; o problema de linhas múltiplas é caracterizado pela disposição dos recursos nos eixos vertical e horizontal de um plano bidimensional. Heragu (1997) propõe modelos matemáticos para resolver *layouts* de linhas simples que consideram a disposição dos recursos apenas em um eixo vertical ou horizontal, ou seja, uma única linha podendo ter curvaturas (modelo ABSModel 1), e modelos de linhas múltiplas, caracterizadas pela disposição dos recursos no eixo vertical e horizontal de um plano de duas dimensões (modelos ABSModel 2 e ABSModel 3). Os modelos ABSModel são não lineares (em geral, ótima solução local) e, por este motivo, foi desenvolvido o modelo LMIP1 (modelo de programação linear mista), como mostra a Equação 2:

$$\text{Minimizar } \sum_{i=1}^{n-1} \sum_{j=i+1}^n c_{ij} f_{ij} (x_{ij}^+ - x_{ij}^-) \quad (2)$$

$$x_i - x_j + M z_{ij} \geq \frac{1}{2} (l_i + l_j) + d_{ij} \quad \text{restrição 1}$$

$$-(x_i - x_j) + M(1 - z_{ij}) \geq \frac{1}{2} (l_i + l_j) + d_{ij} \quad \text{restrição 2}$$

$$x_{ij}^+ - x_{ij}^- = x_i - x_j \quad \text{restrição 3}$$

$$x_{ij}^+, x_{ij}^- \geq 0 \quad \text{restrição 4}$$

$$z_{ij} = 0 \text{ ou } 1 \quad \text{restrição 5}$$

$$x_i > 0 \quad \text{restrição 6}$$

onde:

- n = número de setores ou recursos;
- c_{ij} = custo unitário do transporte entre os setores i e j ;
- f_{ij} = fluxo de transportes de um setor para outro (representa a quantidade de vezes que um produto é transportado do setor i para o setor j);
- l = tamanho horizontal do setor ou recurso;
- d_{ij} = distância mínima necessária entre os setores i e j ;
- i e j = indexadores dos setores;
- x_{ij}^+, x_{ij}^- = variáveis não negativas;
- M = número positivo de valor alto.

O objetivo do modelo é minimizar o valor que corresponde à somatória do custo multiplicado pelo fluxo e pela distância dos pares de recursos analisados. As restrições 1 e 2 garantem a não sobreposição dos recursos. A restrição 3 garante a equivalência entre as duas partes da igualdade, enquanto a restrição 4 garante a não negatividade das variáveis. A restrição 5 permite que somente uma das restrições 1 e 2 seja considerada, e a restrição 6 faz com que os valores utilizados garantam o posicionamento dos recursos dentro da instalação.

2.2. Balanceamento de linha

Segundo Scholl e Becker (2006), os principais objetivos do balanceamento são (i) aumentar a quantidade de produtos produzidos ao mesmo tempo no final da linha, diminuindo o tempo de ciclo de produção; (ii) reduzir o tempo total de produção; (iii) minimizar o tempo de espera dos produtos; (iv) diminuir o tempo de espera das estações; (v) reduzir a quantidade de recursos, combinando operações e colaboradores.

O balanceamento de uma linha de produção é um problema combinatório (TONGE, 1960; 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. O balanceamento interfere no *layout* dos equipamentos

de produção, bem como na disposição de estoque de matérias-primas e materiais acabados.

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

O balanceamento de um ambiente de produção depende de dados do processamento dos produtos: tempo de cada operação, sequência das operações, tempo total do produto na linha e recursos disponíveis. A sequência de processamento é predeterminada e os dados do balanceamento são conhecidos (BAYBARS, 1986; SCHOLL; BECKER, 2006).

Segundo Boysen, Fliedner e Scholl (2006) e Baybars (1986), os modelos de programação de problemas de balanceamento têm por funções objetivo comuns (i) minimizar o tempo de espera, (ii) minimizar o tempo de ciclo e (iii) maximizar a eficiência do arranjo.

3. Modelagem

O desenvolvimento do modelo de *layout* para *job-shop* foi realizado por meio dos seguintes procedimentos:

- Coleta de dados dos produtos comercializados, durante determinado período, referentes à quantidade total vendida, quantidade por período, períodos que apresentaram vendas, lucratividade, tempo total de produção, tempo de espera, roteiro de produção (com tempos e recursos utilizados) e coleta das dimensões retangulares dos recursos de produção (considerando as distâncias de operação, manutenção e segurança necessárias);
- Balanceamento dos roteiros por meio da alocação da quantidade de recursos disponíveis considerando o tempo de espera de cada operação;
- Classificação da importância de cada produto e agrupamento dos produtos em classes;
- Definição e classificação dos novos roteiros conforme a classificação dos produtos;
- Definição do custo de cada ligação por meio do cálculo da quantidade de ligações de cada classe;
- Seleção dos roteiros relevantes;
- Determinação do *layout*.

3.1. Coleta e análise de dados

O período estudado abrangeu diversas fases sazonais com o objetivo de determinar a demanda geral dos produtos. Para a análise adotaram-se os seguintes procedimentos:

- Os dados utilizados na elaboração do modelo foram a quantidade, o preço unitário, a data de entrega e o código do produto. As quantidades de produtos programadas para entrega em cada mês foram totalizadas por produto e por mês. A cada produto foi atribuído um valor hipotético ao preço de venda e a margem de lucro. Para o cálculo do lucro, definiu-se um percentual de lucratividade geral, o qual foi multiplicado pelo preço em cada período;
- Como prazo de entrega utilizou-se a data de entrega desejada pelo cliente. A frequência de vendas foi considerada para análise da continuidade das vendas;
- Cada produto possui um roteiro de produção com tempo de operação baseado em cada etapa do processo;
- O tempo total de espera representa problemas de gargalo no processo, que podem ser provocados por ineficiência ou maquinário obsoleto.

3.2. Balanceamento dos roteiros

Para o balanceamento da produção, por se tratar de um *job-shop*, buscou-se um método que pudesse resolver o problema; contudo, os métodos estudados atendem o problema proposto só em parte. O algoritmo de Johnson para duas máquinas (ASKIN; STANDRIDGE, 1993) não pôde ser aplicado por possuir a restrição de não retorno do produto a um recurso que já tenha sido utilizado. A metodologia COMSOAL (ASKIN; STANDRIDGE, 1993; CHUTIMA; SUPHAPRUKSAPONGSE, 2004) é utilizada para balanceamento de linhas de produção, determinando a melhor sequência das operações. Essa metodologia não pode ser aplicada aos dados deste trabalho porque a sequência de produção é definida e 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 proceder a alteração de sequência. Esta é uma das principais características do produto usinado, sendo que a usinagem depende das referências estabelecidas em cada etapa do processo. Diante dessas condições, balanceou-se a linha de produção através da alocação de mais recursos para cada operação, quando estes estavam disponíveis, ou diminuição destes. Essa forma de balanceamento buscou tornar o fluxo de produção o mais contínuo possível, diminuindo as esperas de produtos ou recursos. 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 era disparada a diminuição ou aumento de recursos da operação antecessora;
- Não foram considerados tempos de setup, mas somente tempos de operação, uma vez que o tempo de setup é aplicado no cálculo de carga da máquina e não durante o tempo normal;
- O balanceamento dependeu da quantidade de recursos disponíveis para as operações;
- Os recursos cadastrados na estrutura da produção foram considerados como disponíveis;
- Não foi considerada carga de utilização dos recursos, porque o objetivo do algoritmo foi tornar o fluxo de produção o mais contínuo possível, e a indisponibilidade do recurso, devido ao excesso de carga, não foi considerada no escopo;
- No desenvolvimento não se considerou o volume de pedidos ou demanda de produção, apenas os tempos de produção e a quantidade dos recursos.

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 produto selecionado, colocando em cada linha a quantidade máxima de recursos disponíveis para a operação e o cálculo da espera referente à 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 reduzir o tempo de espera, calculando o tempo de espera em cada alteração.

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

- Se a capacidade produtiva simultânea, a quantidade de recursos e o tempo para a operação atual são iguais aos da 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 3 para calcular o tempo de espera. Se o resultado for negativo o recurso espera, se for positivo o produto espera, se o resultado for igual a zero não há espera;

$$E_i = (T_i - ((C_i T_{i-1}) / (C_{i-1} / R_{i-1}))) (R_i - R_{i-1}) \quad (3)$$

- Se a capacidade produtiva simultânea atual for maior que a capacidade produtiva simultânea da operação anterior, e a quantidade de recursos da operação atual for igual à quantidade de recursos da operação anterior, utilizar a Equação 4 para calcular o tempo de espera; se o resultado for negativo, o recurso espera; se for positivo, o produto espera; caso seja igual a zero, não há espera;

$$E_i = T_i - T_{i-1} \quad (4)$$

- Se a capacidade produtiva simultânea atual for igual à capacidade produtiva simultânea anterior, utilizar a Equação 5 para calcular o tempo de espera. Para resultado negativo, o recurso espera, para resultado positivo o produto espera, sendo igual a zero, não há espera;

$$E_i = T_i - ((C_i T_{i-1}) / (C_{i-1} / R_{i-1})) \quad (5)$$

- Se a capacidade produtiva simultânea atual for menor que a capacidade produtiva anterior, e o tempo de operação atual for menor que o tempo da operação anterior, utilizar a Equação 6 para calcular o tempo de espera. Nesse caso, tanto o recurso como o produto esperam, porém somente considerar o que esperar mais tempo;

$$E_i = T_{i-1} - (T_i C_{i-1}) \quad (6)$$

- Em última exceção às outras condições, calcular o tempo de espera através da Equação 7. Se o resultado for negativo, o recurso espera, se for positivo, o produto espera, e se for igual a zero, não há espera. A informação de quem espera e o tempo de espera são armazenadas na tabela de balanceamento.

$$E_i = T_i - ((C_i T_{i-1}) / (C_{i-1} / R_{i-1})) \quad (7)$$

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 e quantidade de recursos disponíveis.

3.3. Classificação dos produtos e roteiros

Por meio da Equação 8 foram calculados os valores ponderados para a classificação de cada produto; assim, determinou-se a relevância de cada item no *mix* de produtos estudados. A equação permite a alteração da relevância dos parâmetros considerados, podendo gerar classificações diferentes de acordo com o objetivo da análise.

$$P_i = \alpha \sum_j^n Q + \beta F + \delta \sum_j^n L + \varepsilon T - \phi E \quad (8)$$

onde:

- P = valor ponderado de classificação do produto i ;
- Q = quantidade total de unidades vendidas do produto i no período j de n períodos;
- F = quantidade de períodos (meses) que o produto i apresenta vendas;
- L = lucratividade apresentada pelo produto i no período j em unidades monetárias;
- 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 ;
- ϕ = coeficiente aplicado sobre E ;

As constantes α , β , δ , ε , ϕ determinaram como os dados apurados influenciaram na classificação dos produtos para análise, alterando a interferência da quantidade total (Q), da frequência de períodos (F), da lucratividade (L), do tempo total (T) e do tempo total de espera (E). Desta forma, pôde-se analisar os produtos em relação a sua frequência de vendas, lucratividade, quantidade total de vendas, tempo total de operação e tempo total de espera. A Equação 9 apresenta a restrição aplicada aos coeficientes.

$$\alpha + \beta + \delta + \varepsilon + \phi = 1 \quad (9)$$

O grau de importância de cada parâmetro pode ser estabelecido pela modificação do peso do respectivo coeficiente na classificação geral dos produtos.

Os produtos foram categorizados por meio do cálculo do valor e subsequentemente distribuídos em classes definidas, conforme fórmula de Struges. Assim, quanto maior o valor ponderado maior é a

importância do produto na análise e maior a classe (classe dez a mais importante, classe um a menos). Com a categorização dos produtos foi encontrada a semelhança dos roteiros: os produtos que utilizam a mesma sequência de recursos foram organizados, possibilitando assim a geração de roteiros que atendam a mais de um produto.

Os roteiros foram classificados de acordo com a quantidade de produtos em cada classe de valor ponderado. Durante a classificação foi observado que não ocorresse a repetição das combinações, gerando desta forma a classificação de cada ligação e a matriz com a importância das ligações dos recursos.

Os produtos foram classificados de acordo com os seus roteiros para a determinação de sua particularidade que, baseado na lei de Pareto para a produção, mostra que 80% do volume de produção correspondem a 15% da linha de produtos. Em contrapartida, 15% do volume de produção equivalem aos itens com maior volume, e 85% a um *job-shop* (TOMPKINS et al., 2003). Essa análise foi importante para verificar a existência da característica de *job-shop* na produção estudada, pois podem existir padrões de sequenciamento que não são percebidos no cotidiano da rotina de definição do processo.

Foi implementado o algoritmo desenvolvido por Vladimir Levenshtein para medir a diferença entre duas palavras (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 ; a primeira coluna com linha zero recebe o número relativo à sua posição n . Após a carga da matriz inicia-se a comparação entre as duas palavras: cada caractere da primeira palavra é comparado com cada caractere da segunda. Quando ocorre uma inserção, o posicionamento da matriz desloca-se na horizontal, ao passo que quando ocorre uma exclusão o deslocamento acontece na vertical. O algoritmo procede à análise até chegar à última posição ($m \times n$). A última posição da matriz armazena o custo da comparação entre as duas palavras (LEVENSHTEIN, 2007), o qual pode ser multiplicado por um fator. O algoritmo de Levenshtein foi utilizado para classificar os roteiros de produção, com as operações dos roteiros substituindo os caracteres das palavras; desse modo, calculou-se a diferença dos roteiros e agruparam-se os produtos em roteiros similares. Nesse agrupamento não foram considerados os tempos, mas somente a sequência de operação.

Baseado no trabalho de Daita, Irani e Kotamraju (1999), que utiliza o algoritmo de Levenshtein para agrupar produtos com semelhança de operações, fez-se a implementação do algoritmo para verificação da similaridade entre as sequências produtivas. O primeiro objetivo da implementação foi estabelecer a métrica da similaridade entre os roteiros. Todos os produtos que compõem o conjunto selecionado foram comparados entre si e somente aqueles com roteiro cadastrado foram selecionados. 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.

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 produtos que se pretendia

comparar com o produto carregado a partir do conjunto de registros principal. O resultado foi armazenado em uma tabela (*CmpLev*) que continha o produto original, o produto comparado e o custo da comparação. A Figura 1 mostra o pseudocódigo da implementação.

O resultado do algoritmo permitiu a montagem das tabelas dos produtos com a quantidade de operações diferentes e com a média dos respectivos valores ponderados.

A semelhança entre os processos dos produtos de uma empresa pode ser uma forma de buscar a uniformidade do *layout* dos recursos de produção (DAITA; IRANI; KOTAMRAJU, 1999). Baseado nessa afirmação foi desenvolvido o algoritmo que definiu os novos roteiros padrões a partir dos roteiros dos produtos analisados e permitiu classificá-los de acordo com seu valor ponderado. Os produtos que possuem roteiro foram carregados na tabela *e075pro*

```

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 = 0 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 1. Pseudocódigo do algoritmo de Levenshtein.

e analisados. As sequências 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 existisse a sequência das operações, seria adicionado um ao contador, caso contrário a sequência gerou um novo registro nessa tabela. A Figura 2 mostra o pseudocódigo de algoritmo desenvolvido.

Além da construção do roteiro, o algoritmo também classificou, por meio da determinação do número total de produtos por roteiro e por valor ponderado, os produtos que utilizaram cada um dos roteiros.

Finalmente, foi construída uma matriz para inserção dos dados no modelo matemático.

4. Resultados e discussão

4.1. Coleta de dados e construção da matriz de dados

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

Os códigos dos produtos e dos recursos e os valores monetários foram normalizados para descaracterizar qualquer produto ou processo. Com relação às vendas foram considerados o volume total, a frequência mensal e a lucratividade. Os produtos possuem roteiros de produção que determinam sua sequência de operações e recursos utilizados. Os roteiros contêm os tempos de operação que foram utilizados para determinar o tempo total de utilização dos recursos e os tempos de espera. Esses dados foram utilizados para determinar a relevância

dos produtos. Após a categorização dos produtos os roteiros foram agrupados e a importância de cada ligação encontrada nos roteiros foi determinada. As dimensões dos recursos foram utilizadas nos cálculos para determinação do *layout*.

O modelo utilizou 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 que um par de recursos que possua ligação com classe um. Para o fluxo, foi considerada a quantidade total das ligações de cada par de recursos analisado. As dimensões dos recursos, incluindo as distâncias de segurança e de operação necessárias, foram coletadas da planta da empresa. O comprimento de cada recurso foi dividido por dois, encontrando-se a posição *x* de seu centro. Na resolução do modelo foram desprezadas as ligações de um recurso para ele mesmo, sendo a matriz 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 à ligação original, formando assim uma matriz diagonal superior.

As ligações foram formadas, a partir dos roteiros, com a especificação da sequência da passagem do produto pelos recursos. O custo de cada ligação foi determinado pela Equação 10:

$$c = \sum_{i=1}^{i=10} L_i i \quad (10)$$

onde:

- L_i – é a quantidade de ligações de cada classe i ($i = 1, \dots, 10$).

```
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 2. Pseudocódigo do algoritmo de geração de roteiros classificados.

Tabela 1. Quantidade de ligações entre os recursos por classe.

Recursos			Classe									
Origem	Destino	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

O modelo priorizou a aproximação das ligações das classes maiores que apresentaram maior custo. Na Tabela 1 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 nessa tabela são referentes às primeiras dez ligações encontradas de um total de 264 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 ligações referentes a cada classe definida na coleta de dados.

Devido à produção estudada ser um *job-shop*, os recursos não apresentaram ligações padrão. O recurso 4, por exemplo, mostrou ligação para ele mesmo, e para os recursos 37, 66, 81, 118 e 129. A sequência a ser utilizada foi determinada pelo roteiro do produto desenvolvido pela engenharia de processo. Através dessa análise foi possível estabelecer a relevância de cada combinação de sequência de recurso e, assim, estipular um peso de acordo com a classe de produtos de interesse.

A Tabela 2 mostra o resultado do algoritmo para o caso em análise. Os custos das comparações apresentaram seu valor de zero a dezoito uma vez que cada produto apresentou no máximo dezoito operações. O custo foi considerado equivalente ao número de operações diferentes de cada comparação, ou seja, cada unidade correspondeu ao número de operações diferentes encontradas entre os dois produtos de cada análise.

A Figura 3 representa os dados da Tabela 2 em forma de gráfico. Observe que a maior concentração das combinações está nos grupos de cinco, seis, onze, doze e treze operações diferentes, mostrando assim que a existência de roteiros similares é pequena em relação ao montante das combinações.

Tabela 2. Custos Levenshtein agrupados.

Custo da alteração	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	

Os dados de custos das ligações e de média de valor ponderado dos produtos de cada combinação foram comparados. Os dados relativos à combinação dos custos com as médias dos valores ponderados permitiram classificar os produtos com seu grau de semelhança com outros produtos. A partir dessa combinação surgem roteiros comuns que atendem ao processo de produção dos produtos.

A Tabela 3 mostra os roteiros e a quantidade de produtos que os utilizam para um conjunto composto por 343 produtos classificados (70,14% dos produtos).

Os roteiros apresentados na Tabela 3 são detalhados na Tabela 4, com suas sequências de operações e o recurso utilizado.

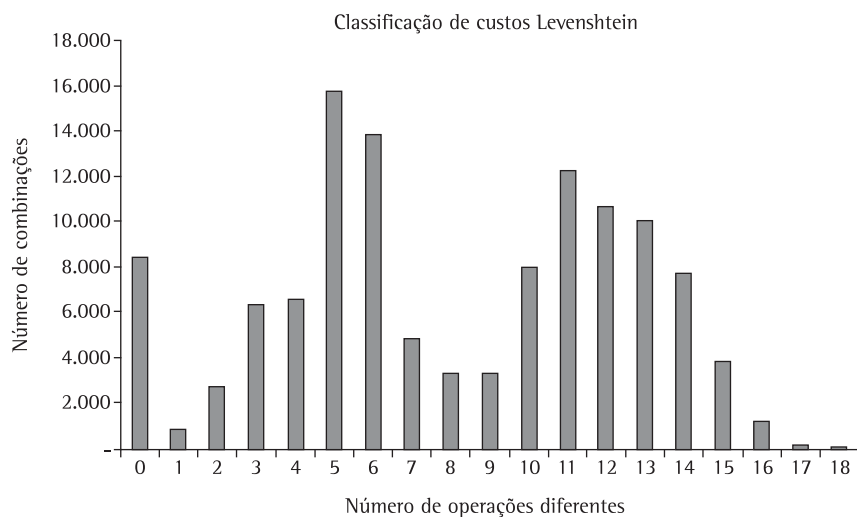


Figura 3. Gráfico dos custos apurados.

Tabela 3. Classificação dos roteiros gerados com o valor ponderado dos produtos.

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 4 não apresentam uniformidade completa. A partir de um roteiro único, o *layout* poderia ser definido conforme a sequência das operações desse roteiro (DAITA; IRANI; KOTAMRAJU, 1999). Procedeu-se então à análise das operações apresentadas no

roteiro que foi utilizado pelos produtos com maior valor ponderado. Além do roteiro selecionado foram escolhidos outros roteiros que apresentaram semelhança com este. As operações foram analisadas considerando o valor ponderado associado ao roteiro e, conseqüentemente, às suas operações.

Tabela 4. 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 produtos foram então classificados de acordo com o valor ponderado, gerando assim a classificação dos seus roteiros. Na base de dados, cada produto possui um roteiro, porém os roteiros apresentam semelhanças. Selecionado o roteiro que possui produtos com o valor ponderado 10 foi obtido o roteiro 34. A Tabela 5 mostra o roteiro selecionado e as semelhanças com outros roteiros – a quantidade de operações diferentes dos roteiros em relação ao roteiro 34 está apresentada na última coluna da tabela.

A semelhança dos roteiros ocorreu na seqüência produtiva. Observando que o objetivo dessa combinação é a definição de um *layout*, foram encontrados os roteiros que utilizam os mesmos recursos do roteiro 34. Os roteiros da Tabela 5 foram detalhados na Tabela 6, com a seqüência dos recursos utilizados.

Os produtos que possuem os roteiros selecionados são apresentados na Tabela 7, que mostra os produtos com valor ponderado e o roteiro que utilizam.

4.2. Resolvendo o modelo

A ferramenta utilizada para resolver o modelo apresentado na Equação 2 foi o LINGO/Win32

Tabela 5. Classificação dos roteiros gerados com o valor ponderado.

Roteiro	Valor ponderado					Total	Diferença de operações em relação ao roteiro 34
	10	7	4	2	1		
8					4	4	14
34	1	1	1	1	11	15	0
36				2	1	3	3
37					1	1	3
83					5	5	12
Total	1	1	1	3	22	28	

versão 11.0.0.11. Os valores de i e j variaram de 1 a 17 e correspondem à quantidade de recursos diferentes (n). O custo e o fluxo foram associados a cada combinação x^p para x^+ e x^- . A variável M recebeu o valor de 99999, correspondendo 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 valor um ou zero, e o valor da distância mínima entre os pontos centrais dos recursos foi calculada previamente.

Tabela 6. Roteiros selecionados e suas sequências.

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														

Tabela 7. Produtos com valor ponderado e roteiro 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

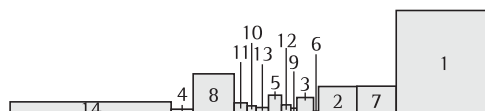


Figura 4. Resultado do modelo.

O resultado apresentado na coluna *posição central* foi utilizado para encontrar a posição inferior e superior de x para cada recurso. Como mostrado na segunda coluna da Tabela 9, cada recurso foi associado a um índice. Foram acrescentados dois recursos aos roteiros: o recurso 0, que corresponde ao estoque de matéria-prima e que representa o espaço necessário para esse setor, e o recurso 99999, que corresponde ao estoque de material acabado.

Utilizando os resultados da Tabela 9, foi gerado o desenho do *layout* conforme Figura 4. Os códigos atribuídos a cada recurso correspondem ao índice da matriz de recursos da Tabela 9. Conforme discutido por vários autores (TOMPKINS, 2003; DAITA; IRANI; KOTAMRAJU, 1999; HERAGU, 1997), após a definição do *layout* por meio de um 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 e, por esse motivo, não foi necessário modificar o posicionamento dos recursos determinado pelo modelo. As formas retangulares na Figura 4 representam o espaço real dos recursos.

4.3. Validação do modelo

O resultado do modelo depende diretamente dos valores do custo e do fluxo. A Tabela 8 apresenta os valores do custo multiplicado pelo fluxo dispostos de forma a considerar o fluxo em um único sentido.

O fluxo de um recurso para ele mesmo é desconsiderado para o cálculo do *layout*. As combinações que não apresentaram ligações receberam valor igual a zero.

O resultado do modelo são os valores para x que representam o ponto central de cada um dos recursos apresentados na Tabela 9.

A posição original dos recursos na planta está apresentada na Tabela 10 com os valores de x e y da posição inferior e superior de cada recurso. A distância entre os recursos foi determinada a partir do ponto central de cada um dos recursos.

A distância encontrada (em milímetros) foi multiplicada pelo valor ponderado e pela quantidade de produtos que transita na ligação. Os valores do cálculo estão apresentados nas Tabelas 11 e 12.

Procedendo-se a somatória dos valores das Tabelas 11 e 12, e dividindo estes valores por mil,

Tabela 8. Custo multiplicado pelo fluxo.

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		

Tabela 9. Resultados do modelo.

Recurso	Índice	Dimensão horizontal	$\frac{l}{2}$	Solução		
				Posição central	Posição inferior	Posição superior
				x	x	X
0	1	16.000	8.000	72.950	64.950	80.950
24	2	6.500	3.250	55.200	51.950	58.450
33	3	2.730	1.365	49.585	48.220	50.950
37	4	3.610	1.805	28.935	27.130	30.740
55	5	2.250	1.125	44.655	43.530	45.780
63	6	1.000	500	51.450	50.950	51.950
65	7	6.500	3.250	61.700	58.450	64.950
66	8	6.900	3.450	34.190	30.740	37.640
73	9	1.040	520	47.700	47.180	48.220
75	10	1.500	750	40.660	39.910	41.410
77	11	2.270	1.135	38.775	37.640	39.910
87	12	1.400	700	46.480	45.780	47.180
93	13	2.120	1.060	42.470	41.410	43.530
99.999	14	27.130	13.565	13.565	0	27.130

Tabela 10. 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	16.000	8.000	17.850	8.925	37.714	23.301	29.714	14.376	45.714	32.226
24	4.680	2.340	5.300	2.650	53.117	49.512	50.777	46.862	55.457	52.162
33	2.730	1.365	3.150	1.575	36.440	69.349	35.075	67.774	37.805	70.924
37	3.609	1.805	1.230	615	78.923	44.250	77.118	43.635	80.727	44.865
55	2.250	1.125	3.500	1.750	60.528	34.939	59.403	33.189	61.653	36.689
63	1.000	500	1.000	500	86.593	57.100	86.093	56.600	87.093	57.600
65	6.500	3.250	5.000	2.500	37.483	11.885	34.233	93.85	40.733	14.385
66	6.900	3.450	7.200	3.600	19.605	61.333	16.155	57.733	23.055	64.933
73	1.040	520	1.370	685	88.018	48.533	87.498	47.848	88.538	49.218
75	1.500	750	1.800	900	86.180	26.924	85.430	26.024	86.930	27.824
77	2.270	1.135	2.190	1.095	57.738	54.243	56.603	53.148	58.873	55.338
87	1.400	700	1.902	951	87.684	45.613	86.984	44.662	88.384	46.564
93	2.120	1.060	1.530	765	85.777	44.341	84.717	43.576	86.837	45.106
99.999	27.130	13.565	2.420	1.210	60.590	18.703	47.025	17.493	74.155	19.913

Tabela 11. 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							

Tabela 12. Custo de cada ligação multiplicado pela distância na posição original.

	73	75	77	87	93	99.999
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

Tabela 13. 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							

Tabela 14. Custo de cada ligação multiplicado pela distância no novo *layout* proposto.

	73	75	77	87	93	99.999
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

obtêm-se o valor total de 25.786.051 unidades de custo, que é o valor de referência utilizado para comparar o custo total do *layout* antes e depois do cálculo do modelo.

Após o posicionamento dos recursos, foi calculado o valor do custo de cada ligação relativo à posição de cada par de recursos no novo *layout*. Os valores das ligações estão apresentados nas Tabelas 13 e 14.

Procedeu-se então ao 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.

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

Embora alguns autores (GUINET; LEGRAND, 1998; FRAMINAN, 2007) tenham proposto modelos para transformar um *layout* tipo *job-shop* em um *layout* por produto, de modo a implementar as vantagens do *layout* por produto em um sistema *job-shop*, é importante observar que tais modelos utilizam o algoritmo de Johnson, o qual não permite o retorno do produto para um determinado recurso (máquina). Essa situação, porém, raramente é encontrada em um ambiente *job-shop* e, por conseguinte, invalida a aplicação de tais modelos para a definição de um *layout* industrial.

5. Conclusões

O presente artigo apresentou um modelo para solução de problemas de *layout* industrial por processo (*job-shop*) considerando os dados históricos referentes ao volume de produtos manufaturados, quantidade distinta de itens, lucratividade, frequência de produção, os recursos disponíveis e os roteiros e características de cada produto. O método proposto baseou-se na análise de dados para qualificar as ligações entre cada recurso encontrado em um *job-shop* e na aplicação de um modelo matemático para definir a posição dos recursos.

O método utilizou como procedimentos a coleta de dados dos produtos, processos e recursos, o balanceamento dos roteiros, a classificação de produtos e roteiros, a seleção dos roteiros relevantes e, finalmente, a determinação do *layout*.

O algoritmo de Levenshtein, aplicado para classificar a diferença dos roteiros quantitativamente, mostrou que pode haver roteiros utilizando recursos em comum, porém isso não significa que esses roteiros são iguais. A análise da igualdade entre os

roteiros foi desenvolvida utilizando um algoritmo para gerar os dados do diagrama de Pareto e, a partir dele, foram estabelecidos novos roteiros padrões e definidos 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. As ligações desses recursos foram submetidas ao modelo matemático LMIP1 para a determinação da posição linear dos recursos no *layout*.

Para validação do método foram utilizados dados de posicionamento dos recursos antes e depois da aplicação do modelo. A comparação dos resultados mostrou que o *layout* definido pelo modelo apresenta o menor custo de operação.

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

Referências

- ASKIN, R. G.; STANDRIDGE, C. R. *Modeling and analysis of manufacturing systems*. New Jersey: John Wiley & Sons, 1993.
- BAYBARS, I. A Survey of Exact Algorithms for the Simple Assembly Line Balancing Problem. *Management Science*, v. 32, p. 909-932, 1986.
- BENJAAFAR, S.; HERAGU, S. S.; IRANI, S. A. Next generation factory layouts: Research challenges and recent progress. *Interfaces*, v. 32, p. 58-78, 2002.
- BOYSEN, N.; FLIEDNER, M.; SCHOLL, A. *A Classification of Assembly Line Balancing Problems*. Arbeits und Diskussionspapiere der Wirtschaftswissenschaftlichen Fakultät der Friedrich-Schiller-Universität Jena, 2006.
- CHUTIMA, P.; SUPHAPRUKSAPONGSE, H. Practical Assembly Line Balancing in a Monitor Manufacturing Company. *Thammasat International Journal of Science and Technology*, v. 9, n. 2, April-June 2004.
- DAITA, S. T.; IRANI, S. A.; KOTAMRAJU, S. Algorithms for Production Flow Analysis. *International Journal of Production Research*, v. 37, n. 11, p. 2609-2638, 1999.
- DRIRA, A.; PIERREVAL, H.; HAJRI-GABOUJ, S. Facility layout problems: A survey. *Annual Reviews in Control*, v. 31, n. 2, p. 255-267, 2007.
- FRAMINAN, J. M. An adaptative branch and bound approach for transforming job shops into flow shops. *Computers & Industrial Engineering*, v. 52, n. 1, p. 1-10, 2007.
- FRANCIS, R. L.; WHITE, J. A. *Facility Layout and Location - An Analytical Approach*. New Jersey: Prentice-Hall, 1974.

- FRANCIS, R. L.; MCGINNIS Jr., L. F.; WHITE, J. A. *Facility Layout and Location – An Analytical Approach*. 2 ed. New Jersey: Prentice-Hall, 1992.
- GOPALAKRISHNAN, B.; WENG, L.; GUPTA, D. P. Facilities design using a split departmental layout configuration. *Facilities*, v. 21, n. 3/4, p. 66-73, 2003.
- GROOVER, M. P. *Automation, Production Systems, and Computer Integrated Manufacturing*. 2 ed. New Jersey: Prentice-Hall, 2000.
- GUINET, A.; LEGRAND, M. Reduction of job-shop problems to flow-shop problems with precedence constraints. *European Journal of Operational Research*, v. 109, p. 96-110, 1998.
- HERAGU, S. S. *Facilities Design*. 2 ed. Boston: PWS Publishing Co., 1997.
- LEVENSHTEIN, V. *Efficient Implementation of the Levenshtein-Algorithm, Fault-tolerant Search Technology, Error-tolerant Search Technologies*. 2007. Disponível em: <http://www.levenshtein.net/>. Acesso em: dez. 2009.
- SCHOLL, A.; BECKER, C. State of-the art exact heuristic solution procedures for simple assembly line balancing. *European Journal of Operations Research*, v. 168, p. 666-693, 2006.
- SULE, D. R. *Manufacturing facilities – location, planning, and design*. 3 ed. Boca Raton: CRC Press, 2009.
- TONGE, F. M. Summary of a Heuristic Line Balancing Procedure. *Management Science*, v. 7, p. 21-42, 1960.
- TOMPKINS, J. A. et al. *Facilities Planning*. 3 ed. New Jersey: John Wiley & Sons, 2003.

Historical data-based job-shop layout systems definition method

Abstract

This paper proposes a method for solving the job-shop layout problem definition, based on historical production data, available resources and production characteristics. The products were classified according to total sales, sales frequency, profitability, total processing time and stand-by time. The production path was compared and the degree of importance of each link between resources was determined. Path comparisons were made to quantify the same routes and the route differences, generating similar routes. The links were submitted to a mathematical model to determine the position of resources in the layout. The method developed was applied to a machining company with a job-shop profile. When the company's data was subjected to the method, it enabled the development of a layout, the main goal of which was to minimize distances between product-processing resources.

Keywords

Optimization. Layout. Mathematical model. Job-shop.