

**UNIVERSIDADE DE SÃO PAULO  
ESCOLA DE ENGENHARIA DE SÃO CARLOS  
DEPARTAMENTO DE ENGENHARIA DE PRODUÇÃO**

**OTIMIZAÇÃO DA MANUFATURA CELULAR POR SIMULAÇÃO COMPUTACIONAL:  
ESTUDO DE CASO EM UMA EMPRESA DE AUTOPEÇAS**

**TRABALHO DE CONCLUSÃO DE CURSO**

Murilo Melhado Baldi

Orientador: Walther Azzolini Junior

São Carlos

2014

MURILO MELHADO BALDI

**OTIMIZAÇÃO DA MANUFATURA CELULAR POR SIMULAÇÃO COMPUTACIONAL:  
ESTUDO DE CASO EM UMA EMPRESA DE AUTOPEÇAS**

Trabalho de Conclusão de Curso apresentado  
à Escola de Engenharia da Universidade de São  
Paulo para obtenção do título de Engenheiro de  
Produção Mecânica.

Orientador: Prof. Dr. Walther Azzolini Junior

São Carlos

2014

## **RESUMO**

Este trabalho possui como objetivo apresentar o uso de ferramentas computacionais de simulação para a otimização de fluxos de produtos, matérias e informações, em um ambiente de manufatura celular. Além disso, as ferramentas computacionais serão empregadas no estudo da concepção e validação das células adotadas pela empresa, objeto de estudo deste trabalho.

Devido ao aumento de complexidade dos sistemas produtivos e, consequentemente, dificultando as análises de processos de produção, fluxos de informação, produtos e pessoas, empresas vêm buscando ferramentas de simulação computacional para auxiliar o entendimento e comportamento de sistemas dinâmicos complexos, inclusive com a possibilidade de gerar cenários para suportar uma decisão.

O trabalho apresenta um estudo de simulação celular desenvolvido no *software* Plant Simulation®, além de vários algoritmos utilizados para a eficiente constituição de uma célula.

## ABSTRACT

This work presents the use of computational simulation tools to optimize flows, such as products, materials and information, in a cellular manufacturing environment. Furthermore, computational tools will be employed in the study of design and validation of cells adopted by the company, subject of this study.

Due to the increasing complexity of production systems and hence, hindering the analysis of production processes and information, products and people flows, companies are seeking computer simulation tools to ease the understanding and behavior of complex dynamic systems, including the possibility to generate scenarios that will be able to support a decision.

This paper presents a study of cell simulation developed with Plant Simulation®, and also shows various algorithms used for an efficient cell formation.

## LISTA DE FIGURAS E TABELAS

Figura 1. Exemplo de classificação *monocode*.

Figura 2. Matriz de itens e máquinas.

Figura 3. Representação gráfica da matriz produtos e máquinas.

Figura 4. *Layout* da célula A.

Figura 5. *Layout* da célula B.

Figura 6. Mapeamento do fluxo de valor da célula A.

Figura 7. Mapeamento do fluxo de valor da célula B.

Figura 8. Representação geral do modelo de simulação.

Figura 9. Representação geral do modelo simplificado.

Figura 10. Gráfico de aptidão.

Tabela 1. Matriz de itens e máquinas.

## SUMÁRIO

1. INTRODUÇÃO.....	7
1.1. OBJETIVOS .....	8
1.2. ESTRUTURA DO TRABALHO .....	9
2. REVISÃO BIBLIOGRÁFICA .....	10
2.1. TIPOS DE <i>LAYOUT</i> DE SISTEMAS DE MANUFATURA.....	10
2.2. FORMAÇÃO DE CÉLULAS DE MANUFATURA.....	11
2.2.1. MODELOS HEURÍSTICOS .....	12
2.2.2. MODELOS MATEMÁTICOS .....	16
2.2.3. OUTROS MODELOS .....	19
3. METODOLOGIA .....	21
4. DADOS COLETADOS .....	22
4.1. MODELO DE SIMULAÇÃO .....	30
5. ADEQUAÇÃO DO MODELO DE SIMULAÇÃO .....	33
6. ANÁLISE E DISCUSSÃO DOS RESULTADOS.....	34
7. REFERÊNCIAS BIBLIOGRÁFICAS.....	36

## 1. INTRODUÇÃO

Devido a fatores externos que afetam a competitividade e apresentam impacto cada vez maior em escala global, empresas buscam alternativas de adequação do tempo de resposta no fluxo de produção com custos operacionais e planos de investimento na manufatura exequíveis para a conquista de novos mercados. Entre as alternativas uma delas consiste no projeto consistente e operação eficiente de um sistema de produção.

Atualmente, a manufatura em lotes (*batch manufacturing*) responde por grande parte das atividades produtivas, entretanto a mesma possui limitações e apresenta ineficiências quando o lote produzido contém menos que 50 unidades.

Nesse caso, a elevada variedade de itens aliada ao baixo volume do lote produz efeitos indesejáveis como elevados custos de *setup*, assim como de equipamentos e ferramentas específicos de mais alto custo de aquisição e manutenção, elevados níveis de estoque e maior suscetibilidade a deficiências logísticas, como consequência desses efeitos há a baixa eficiência de ferramentas computacionais ou não envolvendo técnicas de planejamento e estimativas de custos, por exemplo.

O conceito de Tecnologia de Grupo (TG) tem como base a redução dos tempos de *setup*, do tamanho dos lotes e distâncias percorridas por itens. A Tecnologia de Grupo tenta conciliar a alta flexibilidade oferecida por um *layout jobshop* com a alta produtividade de um *layout flowshop*.

O pressuposto básico do conceito de TG reside no fato de que muitos problemas no contexto da manufatura são similares em muitos aspectos, e que se explorados, essas similaridades podem contribuir para que esses problemas sejam classificados de uma mesma forma e uma solução estratégica de manufatura pode ser aplicada com sucesso na resolução desses problemas (OZDEN et al., 1985; DURIE, 1970; SÉRIO, 1990).

Essa classificação com o foco na resolução dos problemas pode ser feita de diversas formas, por exemplo, um grupo de produtos pode ser classificado pelas suas características de tipo de produto, material, formato, dimensão, tipo de processo envolvido, aspectos do manuseio do produto, etc.

Esses exemplos de classificação buscam acima de tudo, auxiliar o planejamento e controle da produção, bem como atingir economia no uso dos recursos (OZDEN et al., 1985; FERNANDES & FILHO, 2010).

Atualmente, empresas vêm recorrendo à simulação computacional para auxílio em análises de um ambiente produtivo. Trata-se de uma prática pouco explorada na literatura, que possibilita a análise de sistemas dinâmicos e complexos, utilizando para isso modelos matemáticos, especialmente aqueles que apresentam componentes interativos, como ocorre em sistemas de manufatura.

A simulação computacional, aplicada à sistemas de manufatura, possibilita a otimização do mesmo pelo processo de atribuir diferentes valores à variáveis que podem ser controladas,

obtendo como resultado ótimo uma combinação de valores que provê a solução mais adequada do sistema.

O presente trabalho foi desenvolvido em conjunto com uma empresa do segmento de autopeças, localizada no interior do estado de São Paulo. A empresa parceira é responsável pela montagem de conjuntos de embreagens para veículos comerciais e de passeio leves e possui como principais clientes o mercado de reposição e montadoras.

Este trabalho aborda o fluxo de produção de um sistema de manufatura com *layout* celular, configuração esta adotada pela empresa estudada.

## 1.1 OBJETIVOS

O presente trabalho tem como objetivo criar um modelo computacional de uma célula de manufatura da empresa parceira que garanta um ótimo fluxo de materiais, assim como um agendamento de tarefas(*scheduling*) eficiente, definidos por algoritmos. Será realizada, posteriormente, a aplicação do projeto na empresa objeto de estudo.

## 1.2 ESTRUTURA DO TRABALHO

Com este propósito o trabalho foi realizado a partir das fases:

- 1) 1 Fase – definição de duas células de manufatura como objeto do estudo e na sequência o mapeamento do fluxo de valor de ambas com base nos roteiros de fabricação da família de produtos processados pelas células e uso da cronoanálise em casos específicos para validação dos tempos padrão;
- 2) 2 Fase – validação dos mapas de fluxo de valor atual e avaliação da possibilidade de adequação do fluxo de valor com o objetivo de propor o mapa futuro, caso necessário, com o apoio da engenharia de processos da empresa objeto do estudo;
- 3) 3 Fase – uso de heurísticas específicas para validação da tecnologia de grupo quanto à disposição das máquinas nas células de manufatura abordadas e atribuição das famílias de produtos de acordo com a estrutura atual de operação das mesmas;
- 4) 4 Fase – análise dos resultados quanto à validação da tecnologia de grupo;
- 5) 5 Fase – Estudo do modelo de simulação desenvolvido e utilizado pela empresa para outras células de manufatura a partir do *software* de simulação *Plant Simulation*;
- 6) 6 Fase – Treinamento do uso do *software* de simulação *Plant Simulation*;
- 7) 7 Fase – Adequação do modelo de simulação desenvolvido e utilizado pela empresa para as células objeto de estudo do presente trabalho e validação dos resultados obtidos na simulação com os dados reais de processo;
- 8) 8 Fase – Análise dos resultados obtidos a partir da simulação da operação das células de manufatura objeto do estudo do presente trabalho;
- 9) Considerações Finais.

Desse modo o presente trabalho foi dividido nas seguintes etapas:

- Introdução;
- Revisão Bibliográfica;
- Aplicação da Tecnologia de Grupo para validação do fluxo de produção ideal;
- Adequação do modelo de simulação;
- Análise e Discussão dos resultados;
- Considerações Finais.

## 2. REVISÃO BIBLIOGRÁFICA

### 2.1 TIPOS DE *LAYOUT* DE SISTEMAS DE MANUFATURA

Existem, basicamente, quatro tipos de *layouts* amplamente aplicados a sistemas de manufatura, são eles:

- Posição fixa ou posicional;
- Por produto ou linear;
- Por processo ou funcional;
- Celular.

#### **Posição fixa:**

Este tipo de *layout* está associado à produção de itens impossibilitados de locomoção ou que possui tamanho e peso expressivos, como edifícios, navios, aeronaves, entre outros. Há, nesse caso, movimentação de componentes e equipamentos para o local de fabricação do produto resultante, uma vez que o mesmo é fixo.

A estrutura proporcionada por este tipo de *layout* apresenta grande flexibilidade e variabilidade na fabricação de itens e é possível de receber mudanças significativas desde o projeto do produto assim como no volume de produção. De fato, cada produto pode ser único.

Há demanda, neste tipo de sistema, por trabalhadores altamente capacitados já que a identificação de tarefas é mais complexa se comparada a outros tipos de *layout*. Muitas vezes, a abordagem por equipes multifuncionais é utilizada.

#### ***Layout* por produto:**

São características deste tipo de *layout* o alto volume de produção e a baixa variedade de itens. As operações são arranjadas de forma sequencial, e podem apresentar um fluxo contínuo ou discreto de itens produzidos.

*Layouts* orientados a produtos requerem equipamentos específicos, muitas vezes automatizados, os quais elevam substancialmente o investimento inicial. Caso haja necessidade de mudanças no produto, devido à baixa variabilidade deste tipo de sistema, pode ser requerida uma reorganização na disposição de equipamentos, operações e subsistemas.

Este tipo de *layout* apresenta tarefas simples e repetitivas, desta maneira não é necessária alta capacitação da mão-de-obra, além disso, o fluxo de materiais e informações é lógico e simples, de fácil controle.

#### ***Layout* por processo:**

Este tipo de *layout* está associado a um baixo volume de produção e alta variedade de produtos, como em sistemas *jobshop*. Ocorre, neste caso, um agrupamento de processos similares.

O *layout* organizado por processos favorece a flexibilidade em detrimento da eficiência o que, por sua vez, torna mais complicada a tomada de decisões operacionais, além disso, nota-se que o fluxo de materiais, muitas vezes, não é organizado, contribuindo para a situação anterior.

Devido à sua forma de organização, o *layout* por processos apresenta altos níveis de estoque e *leadtimes*.

#### **Layout celular:**

Slack et al. (1999) descreve que o arranjo físico celular é aquele em que os recursos transformados, entrando na operação, são pré-selecionados a partir de atividades comuns no processo produtivo. Nele é feito o agrupamento de tarefas com o objetivo de formar famílias de produtos. Tarefas diferentes podem ser agrupadas em famílias devido a sequências comuns de operações, materiais que as compõem, equipamentos necessários, ou similaridade de manuseio, estoque e controle.

Este tipo de layout proporciona alto volume de produção aliado a baixa variedade de produtos. Equipamentos específicos normalmente são utilizados e, assim como no layout por produto, o fluxo de materiais e informações é simples e lógico, contribuindo assim para facilitar o controle.

## **2.2 FORMAÇÃO DE CÉLULAS DE MANUFATURA**

Para a formação de células poder ser realizada de forma eficiente, necessita-se, primeiramente, do agrupamento inteligente de itens a serem manufaturados. Desse modo, deve-se considerar tipo de produto, assim como suas dimensões, forma, material, processo de fabricação, distribuição, segundo as premissas do conceito de Tecnologia de Grupo.

Primeiramente, os itens devem ser classificados e codificados para realizar o agrupamento em famílias, atributos como dimensões, tipo de material, processos de fabricação devem ser considerados. A Figura 1 mostra um exemplo de classificação baseado em uma sequência numérica em que cada dígito depende do anterior (*monocode*).

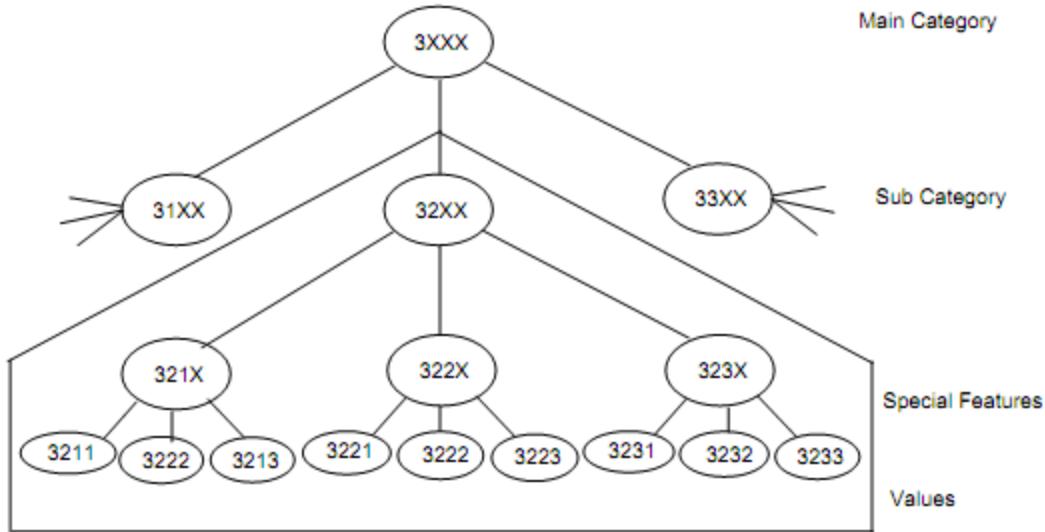


Figura 1 – Exemplo de classificação *monocode*

## 2.2.1 MODELOS HEURÍSTICOS

O principal objetivo apresentado pelos algoritmos descritos nesta sessão de modelos heurísticos reside na atribuição de itens às famílias de produtos, essencial para estudos de formação de células manufatureiras em si. Uma das possíveis maneiras de obter soluções, isto é famílias de itens, é minimizando a função distância, calculada segundo a equação de *Minkowski*:

$$dpq = (\sum |X_{pk} - X_{qk}|^r)^{1/r} \quad (1)$$

Em que  $p$  e  $q$  são itens,  $X$ , o código atribuído aos mesmos e  $k$ , a quantidade de dígitos do sistema de codificação.

Após esta etapa, segundo o algoritmo, devem-se agrupar as partes que apresentam menores distâncias e recalculá-las, segundo a equação 2.1. Após uma série de iterações é possível identificar o arranjo ótimo de itens de uma mesma família, considerando os atributos apresentados anteriormente e, dessa maneira, definir a família que minimiza as distâncias em questão.

Outro modelo, denominado *P-median*, consiste em identificar minimizar  $f$  famílias de itens e minimizar as distâncias entre as mesmas, e seus respectivos itens baseando-se em um valor médio relativo a cada família. Este algoritmo permite a melhor combinação de variáveis para obter a solução ótima, descrito pela função:

$$\text{Min } \sum \sum d_{pq} X_{pq}$$

Sujeito a:

$$\sum X_{pq} = 1 \text{ e,}$$

$$\sum X_{qq} = f$$

Em que,

$$d_{pq} \geq 0, \text{ para } p \neq q$$

$$d_{pq} = 0, \text{ para } p = q \text{ e}$$

$$X_{pq} = 1 \text{ caso o item } p \text{ pertença à família de } q.$$

Até então, os algoritmos apresentados têm o propósito de identificar e formar famílias de produtos, utilizando para isso, atributos como planejamento de processos, custos, vendas. Isto é feito mediante a definição de uma distância, a qual deve ser minimizada, que representa a similaridade entre itens de uma família.

Agora, como parte de um conceito mais amplo de célula de manufatura, os algoritmos apresentados nesta seção têm como ênfase a manufatura, identificando famílias de produtos e grupos de máquinas ótimos, isto é, que possuam a menor interação com outros grupos ou elementos externos, durante o processamento dos itens.

Um dos algoritmos utilizados denomina-se *Bond Energy Algorithm* (BEA), desenvolvido por McCormick, Schweitzer e White, e é capaz de identificar grupos em arranjos complexos de dados em forma matricial, distribuindo-os de maneira mais uniforme.

O algoritmo inicia-se a partir de uma matriz binária de itens (produtos) e máquinas, estabelecida previamente, representada na Figura 2, e utiliza uma medida de eficiência para um determinado arranjo A, ME (A), calculada segundo:

$$ME(A) = \frac{1}{2} \sum \sum a_{pm}[a_{p,m+1} + a_{p,m-1} + a_{p,+1,m} + a_{p-1,m}]$$

Em que:

$$a_{0,m} = a_{p+1,m} = a_{p,0} = a_{p,M+1} = 0$$

$a_{pm} = 1$ , caso seja necessário processamento do item  $p$  na máquina  $m$ .

$a_{pm} = 0$ , caso contrário.

	P	A	R	T		
Machine	1	2	3	4	5	6
1	1		1			
2	1			1		
3			1		1	
4	1			1		
5		1				
6			1			1

Figura 2 – Matriz de itens e máquinas

A função ME deve ser maximizada considerando todas as permutações possíveis, ou seja,  $P!M!$  combinações.

A ordenação de elementos obtida no fim depende da escolha de linhas ou colunas iniciais, sendo assim, uma limitação do algoritmo.

Outro algoritmo, *Rank Order Clustering* (ROC), baseia-se em princípios do anterior (BEA), para cálculo de eficiência, entretanto é de mais fácil implementação computacional uma vez que a cada linha da matriz itens (produtos) e máquinas é atribuído um valor decimal ( $C_m$ ), convertido de seu valor binário equivalente, que deve ser reduzido a cada iteração até que não haja mudanças entre iterações subsequentes.  $C_m$  é calculado segundo:

$$C_m = \sum 2^{(P-p)} ap_m \quad (ap_m = 0 \text{ ou } 1)$$

Em que  $m$  representa o número de linhas da matriz.

Este algoritmo apresenta limitações computacionais, dessa forma pode representar até 47 linhas e colunas. Além disso, há dependência da matriz inicial para obtenção dos melhores resultados, perdendo-se a arbitrariedade da escolha inicial.

De maneira a superar algumas restrições do algoritmo anterior, foi desenvolvido o ROC 2 que elimina o cálculo de decimal equivalente e utiliza um rearranjo de linhas distinto, em que o usuário deve identificar o maior número de colunas que apresentem 1 como o elemento mais à direita e assim, realizar a mesma manipulação com as linhas, até que não ocorram mudanças na organização da matriz.

Outra variação de ROC encontrada na literatura é a *Modified Rank Order Clustering* (MODROC), que distingue-se do primeiro por remover o bloco de colunas que apresentam 1s, no canto superior esquerdo da matriz após terem sido realizadas as iterações de ROC. A iteração termina uma vez que não existam mais elementos como descritos na matriz.

Esse processo é utilizado para eliminação de famílias mutuamente exclusivas que podem ter sido formadas.

Em 1982, foi proposto por Chan e Milner o *Direct Clustering Algorithm* (DCA) que rearranja as linhas, da matriz binária de itens e máquinas, que possuem elementos 1s à esquerda para cima e as colunas com elementos 1s na parte superior, para a esquerda. Esse rearranjo é realizado de forma iterativa até que não existam mais modificações entre iterações subsequentes. O DCA não garante soluções aceitáveis, uma vez que não necessariamente produz soluções diagonais.

Outro método, proposto por Iri (1968), denominado *Cluster Identification Algorithm* (CIA) identifica se uma matriz possui blocos diagonais, que podem constituir uma solução ótima. O processo consiste em desconsiderar, esconder as colunas de elementos 1s em uma dada linha da matriz, em seguida, esconder as linhas em que se encontram os elementos 1s das colunas desconsideradas anteriormente. O algoritmo deve ser repetido até que o número de colunas e linhas pare de aumentar, como resultado, blocos diagonais podem ser obtidos constituindo assim, uma solução válida.

Alguns algoritmos utilizam-se de coeficientes para identificar grupos de máquinas para, posteriormente, utilizando outros métodos, atribuir famílias de produtos.

Dentre eles podemos destacar o *Single Linkage Clustering* (SLC), desenvolvido por McAuley (1972) que antes de analisar a matriz itens e máquinas define um coeficiente de similaridade entre duas máquinas, baseando-se no número de produtos que passam pelas mesmas, como apresentado na Tabela 1.

Máquina n \ Máquina m	1	0
1	a	b
0	c	d

Tabela 1 – Matriz de itens e máquinas

O cálculo do coeficiente é dado por:

$$S_{mn} = a / (a + b + c), \quad 0.0 \leq S_{mn} \leq 1.0.$$

Em que a é o número de itens que passam por ambas as máquinas, b apenas a máquina m, c apenas a máquina n e d, nenhuma das duas.

Este coeficiente apresenta máxima similaridade quando  $S_{mn} = 1$  e nenhuma quando  $S_{mn} = 0$ .

Desta forma, são reunidas, em um grupo de máquinas, aquelas que apresentam maior valor de  $S_{mm}$ . O processo continua até que um desejado número de grupos tenha sido formado.

$$S_{tv} = \text{Max} \{ S_{mn} \}, \text{ com } m \text{ pertencendo a } t \text{ e } n, \text{ a } v.$$

Além do SLC, existe o *Complete Linkage Clustering* (CLC), que atua da mesma forma, porém agrupa máquinas baseando-se no valor mínimo do coeficiente  $Smn$ .

$Stv = \text{Min}\{Smn\}$ , com  $m$  pertencendo a  $t$  e  $n$ , a  $v$ .

Além destes métodos há o *Average Linkage Clustering* (ALC), o qual, diferentemente de SLC e CLC que consideram valores extremos do coeficiente  $Smn$ , utiliza um valor médio, calculado baseando-se não apenas em um par de máquinas.

Dessa maneira, o grupo de máquinas é determinado segundo:

$$Astv = \sum \sum Smn / Nt*Nv$$

Em que  $Nt$  e  $Nv$  correspondem ao número de máquinas presentes no grupo  $t$  e  $v$ , respectivamente.

Outro algoritmo, baseado em coeficientes, é o *Linear Cell Clustering* (LCC), proposto por Wei e Kern (1989) que agrupa máquinas segundo um coeficiente de similaridade, semelhante aos métodos apresentados anteriormente, entretanto, este considera itens que não são processados em nenhuma das máquinas (representados por  $d$  na Tabela 1). Além disso, o algoritmo pode ser adaptado para considerar uma gama de valores pertinentes ao tamanho e quantidade de células que deverão ser formadas futuramente.

O coeficiente de LCC pode ser calculado segundo:

$$Cmn = \sum \delta(apm, apn)$$

Em que:

$$\delta(apm, apn) = (P-1), \text{ se } apm = apn = 1$$

$$\delta(apm, apn) = 1, \text{ se } apm = apn = 0$$

$$\delta(apm, apn) = 0, \text{ se } apm \neq apn$$

## 2.2.2 MODELOS MATEMÁTICOS

Os algoritmos que serão apresentados nesta sessão constituem modelos matemáticos que, diferentemente das heurísticas discutidas até este ponto, não são afetados pela natureza dos dados e soluções iniciais.

Serão apresentados modelos matemáticos que podem ser empregados na formação de famílias de produtos e em agrupamento de máquinas.

Proposto por Kusiak (1987), o *P-Median Model* identifica famílias de produtos maximizando o parâmetro de similaridade,  $Spq$ , ao invés de minimizar a distância, como no modelo *P-Median* anterior. Para isso, o modelo seleciona uma determinada quantidade de medianas e as atribui às famílias, de forma que a soma do parâmetro similaridade seja a maior possível.

$Spq$  é calculado segundo:

$$Spq = \sum \delta(apm, apn)$$

Em que:

$$\delta(apm, apn) = 1, \text{ se } apm = apn$$

$$\delta(apm, apn) = 0, \text{ caso contrário}$$

Uma das limitações do modelo é desconhecer o número ideal de medianas de forma a produzir a solução ótima. Limitação essa explorada pelo modelo seguinte, denominado *Assignment Model*, capaz de identificar famílias e grupos de máquina por meio da mesma função similaridade, expressa anteriormente.

Neste modelo temos os seguintes parâmetros:

$$Xpq = 1, \text{ se os produtos } p \text{ e } q \text{ estão conectados}$$

$$Xpq = 0, \text{ caso contrário}$$

$$Ymn = 1, \text{ se as máquinas } m \text{ e } n \text{ estão conectadas}$$

$$Ymn = 0, \text{ caso contrário}$$

Para identificação de famílias:

$$\text{Max } \sum \sum Spq Xpq$$

Para agrupamentos de máquinas:

$$\text{Max } \sum \sum Smn Ymn$$

Este modelo oferece maior qualidade da solução e menor tempo computacional se comparado ao *P-Median*.

Outro modelo, denominado *Quadratic Programming Model*, baseia-se nos anteriores, porém considera as interações entre elementos de uma mesma família ou grupo.

Assim:

$X_{pf} = 1$ , se o item  $p$  é atribuído à família  $f$   
 $X_{pf} = 0$ , caso contrário

A função a ser maximizada, neste caso, é dada por:

$$\sum \sum \sum S_{pq} X_{pf} X_{qf}$$

Alguns modelos matemáticos oferecem representação gráfica da matriz produtos e máquinas, formada por nós, que representam um conjunto de produtos e máquinas, e arcos, os quais podem ser entendidos como a relação de requerimento entre um item e uma máquina.

Nestes modelos gráficos são feitas decomposições de maneira que sejam formados famílias e grupos, como mostrado na figura 3, em que X representam produtos, itens e Y, máquinas.

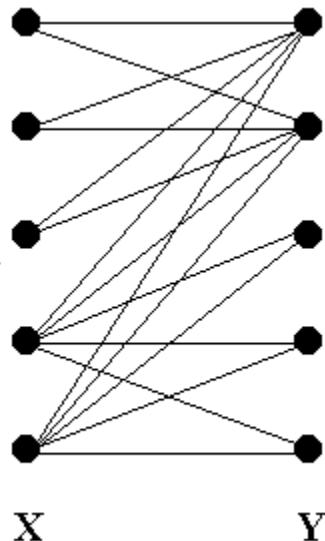


Figura 3 – Representação gráfica da matriz produtos e máquinas

Além destes, há o modelo de atribuição não-linear que considera elementos especiais não contemplados nos algoritmos anteriores, como uma máquina que deve processar todos os itens de várias famílias, ou então, itens que são processados em apenas determinadas máquinas de maneira a maximizar a ocupação. Isso pode ser feito mediante atribuição de peso(W) a determinados elementos que requerem atenção especial.

Assim, este modelo deve minimizar:

$$W * \sum \sum \sum a_{m} X_{pc} (1 - y_{mv}) + (1 - W) * \sum \sum \sum (1 - a_{pm}) X_{pc} Y_{mc}$$

Em que:

$a_{pm} = 1$ , se o item  $p$  necessita de processamento na máquina  $m$

$a_{pm} = 0$ , caso contrário

$X_{pc} = 1$ , se o item  $p$  for alocado na célula  $c$

$X_{pc} = 0$ , caso contrário

$Y_{mc} = 1$ , se a máquina  $m$  for atribuída à célula  $c$

$Y_{mc} = 0$ , caso contrário

### 2.2.3 OUTROS MODELOS

Além dos modelos matemáticos descritos anteriormente, o conceito de algoritmo genético também pode ser empregado para obter grupos de máquinas e famílias de produtos.

Um algoritmo genético (AG) inicia com uma população de indivíduos (cromossomos) que representam possíveis soluções do problema a ser resolvido. A população é avaliada e cada cromossomo recebe uma nota, denominada aptidão. Os melhores cromossomos (mais aptos) são selecionados e os piores descartados. Os selecionados podem sofrer cruzamento e mutação, gerando descendentes para a próxima geração. O processo continua até que uma solução satisfatória seja encontrada.

Aplicando o AG neste problema percebe-se que um cromossomo pode representar uma célula e cada gene, uma máquina ou produto. A ordenação dos genes no cromossomo representa o posicionamento dessas máquinas e produtos em uma célula.

Exemplificando, supondo que existam  $M$  máquinas, em que  $M = 5$ , além de  $P$  produtos, com  $P = 5$ , desejando-se formar 3 células.

O cromossomo resultante pode ser representado como:  $(2,1,2,1,3 | 2,2,3,3,1)$ . As primeiras cinco posições referem-se às máquinas e as demais, a produtos. Logo, temos as seguintes células:

Célula 1: Máquinas 2 e 4; Produto: 5.

Célula 2: Máquinas 1 e 3; Produtos: 1 e 2.

Célula 3: Máquinas 5 e 4; Produtos: 3 e 4.

A função aptidão deve maximizar a ocupação de elementos na célula, assim, ela pode ser representada por:

$F(t) = F_{max} - g(t)$ , quando  $g(t) < F_{max}$

$F(t) = 0$ , caso contrário

Em que  $g(t)$  corresponde ao valor objetivo, soma ponderada das ociosidades dentro da célula, e  $F_{max}$ , ao maior valor objetivo obtido em cada iteração.

Além da heurística algoritmo genético, o conceito de redes neurais também pode ser empregado na formação de células.

O modelo de redes neurais imita a forma com que decisões são geradas em neurônios, que combina reconhecimentos de padrão, *know-how* e até intuição. O modelo consiste em unidades de processamento conectadas entre si, essas conexões possuem ponderação, simulando ações inibidoras e estimuladoras. Cada unidade de processamento recebe sinais dessas conexões, aplica uma determinada função e envia o sinal a outras unidades a ela conectadas. Esta operação é realizada de forma dinâmica e contínua.

Neste algoritmo, cada conjunto de unidades de processamento representa um tipo de elemento (máquinas, produtos e células) que têm seus sinais (enviados e recebidos) combinados e analisados por uma função linear ou não-linear. Acrescentam-se ponderações ( $W_{ij}$ ) aos sinais dessas unidades de processamento, que podem ser calculadas segundo:

$$\begin{aligned} W_{ij} &= s_{ij} - A, & \text{para } i \neq j \\ W_{ij} &= 0, & \text{para } i = j. \end{aligned}$$

Em que  $i$  e  $j$  representam produtos e máquinas, respectivamente;  $s_{ij}$  é o coeficiente de similaridade de Jaccards e  $A$ , calculado por:

$$A = \sum \sum s_{ij} / n(n - 1)$$

A combinação dos sinais enviados e recebidos, acrescentando-se influências de outras unidades de processamento, pode ser representada por:

$$\text{netinput}(i) = \sum W_{ij} \text{output}(i) + \text{extinput}(i)$$

Todos os modelos anteriormente apresentados oferecem artifícios para formação de agrupamentos de máquinas e/ou famílias de produtos, além de sequenciá-los otimamente, que constituirão uma célula de manufatura, entretanto, para isso, não são considerados aspectos relativos à implantação dessas células como custos de processamento, capacidade de atendimento da demanda, planos de processo e outros atributos que compõe a estratégia organizacional de uma empresa. Então, foram desenvolvidos diversos algoritmos para proporcionar a inclusão dessas características.

Neste algoritmo, o agrupamento de máquinas é atribuído às famílias previamente delimitadas, e é dado pela seguinte função a ser minimizada:

$$\text{Min } \sum C_m Z_{mf} + \sum (\sum a_{ms}(lkp) c_{ms}(kp)) X(lkp)$$

Sujeito a:

$$\sum X(lkp) \geq d_k$$

$$\sum \beta_{kf} \sum (\sum a_{ms}(lkp) t_{ms}(kp)) X(lkp) \leq b_m Z_{mf}$$

Em que:

$b_m$  corresponde ao tempo disponível em cada tipo de máquina  $m$ ;

$C_m$  é o custo de cada máquina do tipo  $m$ ;

$c = 1, 2, \dots, C$  é o número de células;

$d_k$  corresponde a demanda pelo produto  $k$ ;

$k = 1, 2, \dots, K$  é o número de produtos;

$l = 1, 2, \dots, L$  ( $kp$ ) corresponde aos planos de produção, para  $kp$  combinações;

$m = 1, 2, \dots, M$ , o número de máquinas;

$p = 1, 2, \dots, P_k$ , planos de processo para o item/produto  $k$ ;

$s = 1, 2, \dots, S(kp)$ , operações para ( $kp$ ) combinações;

$X(lkp)$  corresponde à quantidade de itens  $k$  produzidos segundo o plano de processos  $p$  e plano de produção  $l$ ;

$Z_{mf}$  corresponde ao número de máquinas  $m$  utilizadas para produzir itens na família  $f$ .

$a_{ms}(lkp) = 1$ , se no plano  $l$  a máquina  $m$  é atribuída a operação  $s$  para todas ( $kp$ ) combinações

$a_{ms}(lkp) = 0$ , caso contrário

$c_{ms}(kp)$  = custo para a máquina  $m$  realizar a operação  $s$  para todas ( $kp$ ) combinações

$c_{ms}(kp) = \infty$ , se a máquina  $m$  não puder realizar esta operação

$t_{ms}(kp)$  = tempo para a máquina  $m$  realizar a operação  $s$  para todas as ( $kp$ ) combinações

$t_{ms}(kp) = \infty$ , caso a máquina  $m$  não possa realizar a operação em questão

$\beta_{kf} = 1$ , se o produto  $k$  pertence à família  $f$

$\beta_{kf} = 0$ , caso contrário

### 3. METODOLOGIA

Para a realização do objetivo proposto na empresa estudada serão aplicados alguns modelos e algoritmos apresentados na seção de revisão bibliográfica do presente trabalho.

Dentre os diversos algoritmos mencionados serão selecionados aqueles mais apropriados, considerando os dados obtidos durante o processo de coleta de informações e as condições gerais da organização.

#### 4. DADOS COLETADOS

Nesta sessão serão apresentadas, como objeto de estudo, características principais de duas células de manufatura da empresa de autopeças estudada.

Os dados coletados são parte integrante deste trabalho e serão analisados com base nos algoritmos apresentados anteriormente.

##### CÉLULA A:

Célula de manufatura responsável pela montagem do conjunto disco de embreagem pequeno.

##### OPERAÇÕES:

10 - Montar pré-amortecimento e dispor para o posto de montagem manual. (Operação realizada fora da célula A).

20 - Rebitar disco de torção e molas de guarnição utilizando rebites maciços:

Nessa operação utilizam-se os seguintes componentes: discos de torção, molas de guarnição e rebites maciços para realizar a montagem do subconjunto. Há um operador que realiza tanto a montagem do conjunto com molas de guarnição (inserir componentes e retirar subconjuntos do equipamento) como o abastecimento, e participa também da operação seguinte.

Trata-se de uma operação automatizada cujo tempo de ciclo é de 18,36 segundos.

30 - Rebitar guarnições no subconjunto da operação anterior utilizando rebites tubulares e colocar anel auto centrante no subconjunto:

Nesta operação há colocação do componente anel auto centrante no subconjunto anterior, realizado, de forma automatizada, pela estação de trabalho SAX, o subconjunto resultante dirige-se, por meio de uma esteira, até o posto de trabalho seguinte.

Ambos operadores da operação 20 participam desta etapa, cujo tempo de ciclo corresponde a 12,03 segundos.

40 - Montar pacote torcional e dispor conjunto para próxima operação:

40.1 - Montagem 1:

Há abastecimento dos seguintes componentes para a montagem do conjunto: anel de atrito, anel ondulado, anel auto centrante, molas helicoidais externas, flange do cubo, placa de controle e

anel de pressão. Esses componentes são montados no subconjunto anterior por um operador exclusivo, na mesa 1 da bancada de montagem.

Tempo de ciclo: 12,89 segundos.

#### 40.2 - Montagem 2:

São adicionados ao subconjunto anterior os seguintes componentes: pinos distanciadores, pacote de pré-amortecimento e disco de retenção. Assim como a suboperação 40.1, há um operador exclusivo para a montagem.

Tempo de ciclo: 13,77 segundos.

#### 50 - Prensar subconjunto:

Nesta operação é utilizada a prensa para realizar a prensagem do subconjunto anterior. Devido ao balanceamento da célula e aos processos automatizados, as operações 50, 70 e 80 são realizadas por um único operador, em um layout disposto em U.

O tempo de ciclo desta operação corresponde a 8,80 segundos.

#### 60 – Verificação de Balanceamento:

Esta operação consiste no balanceamento do subconjunto utilizando uma balança que é dividida com outra célula de manufatura. A operação em questão é eventual, e realizada apenas quando requisitada pelo cliente, o que pode ocorrer caso haja inconformidades nas especificações de determinado lote, ou mesmo para garantir a conformidade de todos os conjuntos.

#### 70 - Medir características funcionais:

Esta operação consiste em realizar medições de características funcionais do conjunto, além disso, atua como um dispositivo à prova de falhas, *Poka Yoke*.

O tempo de ciclo desta etapa corresponde a 18,09 segundos.

#### 80 - Gravar conjunto a laser:

Nesta etapa ocorre a gravação a laser do conjunto advindo da operação anterior, segundo codificação que varia conforme o cliente. O processo é automatizado e possui tempo de ciclo igual a 7,69 segundos.

#### 90 - Fazer inspeção visual no conjunto:

Esta operação corresponde a uma etapa de verificação final das características do conjunto, é realizada manualmente por um operador. O conjunto é então disposto em um cesto que será expedido posteriormente.

O tempo de ciclo dessa operação corresponde a 17,75 segundos.

### CÉLULA B:

Responsável pela montagem e usinagem do platô pequeno.

#### OPERAÇÕES:

##### 10 – Usinagem da placa de pressão:

Nesta operação ocorre a usinagem da placa de pressão a partir de uma peça bruta, assim como a inspeção visual e medições de características críticas, relativas ao controle de qualidade. O abastecimento da célula é realizado conforme a figura 5, e esta operação é realizada em duas estações de trabalho distintas.

Há um operador que realiza as funções supracitadas em cada uma das estações de trabalho e dispõe os subconjuntos resultantes, manualmente, para a próxima operação. Estes mesmos operadores são também responsáveis pelas operações 15 e 20.

Caso a peça seja reprovada no Controle de Qualidade, a mesma será segregada das demais e disposta em local apropriado.

O tempo de ciclo médio desta etapa corresponde a 17,88 segundos.

##### 15 - Furação:

A etapa de furação é normalmente realizada como suboperação da operação 10, entretanto, devido à manutenção no ferramental do torno, esta foi desvinculada e figura como uma operação independente e subsequente, em relação a 10.

O fluxo de subconjuntos ocorre, manualmente, segundo a figura.

##### 20 – Oleamento da placa de pressão:

Nesta etapa ocorre o oleamento dos subconjuntos provenientes das operações anteriores e a disposição dos mesmos para a etapa seguinte, por meio de esteira.

##### 30 - Montagem do conjunto platô:

Esta operação é composta por duas suboperações, são elas: montagem da placa de pressão com mola de retrocesso e rebitação do subconjunto resultante; e medição de características críticas e disposição da peça, por meio de esteira, para a operação seguinte. Os componentes utilizados na montagem estão estocados na estação de trabalho.

São empregados dois operários, em uma operação semi-automática cujo tempo de ciclo corresponde a 15,50 segundos.

##### 40 - Balanceamento conjunto platô:

Nesta etapa é medido e corrigido o nível de desbalanceamento do conjunto, caso sejam constatadas inconformidades. Existem, também, duas estações de trabalho.

Há o abastecimento da estação de trabalho com rebites de balanceamento e disposição, por meio de esteira, para a operação seguinte.

É empregado um operador para colocação da peça no dispositivo e para acrescentar os rebites de balanceamento, sendo que as suboperações de medição e conferência são automáticas.

O tempo de ciclo médio corresponde a 15,70 segundos.

50 - Regular mola membrana:

Nesta operação é realizada a regulagem do conjunto platô anterior e a disposição, manual, do mesmo para a próxima etapa. Verifica-se a altura e oscilação da mola membrana. Trata-se de um processo manual realizado por um operador cujo tempo de ciclo corresponde é de 15,78 segundos.

60 - Medir carga:

Nesta etapa são medidas características críticas do conjunto platô mediante aplicação de carga, de forma a satisfazer a estreita faixa de tolerância do conjunto. Trata-se de uma operação é automática e o operador (mesmo da operação 70) realiza a disposição das peças para a etapa seguinte, manualmente, e o setup da máquina de medição de carga e Laser.

O tempo de ciclo corresponde a 18,69 segundos.

70 – Gravar de identificação a laser e oleamento do conjunto:

Essa operação é responsável por realizar a gravação a laser do conjunto anterior. O operador (mesmo da operação anterior) é responsável pelo trânsito da peça, a fixação da mesma no Laser e oleamento do conjunto para proteção da gravação de identificação.

O tempo de ciclo desta etapa corresponde a 15,03 segundos.

80 - Inspeção final, e embalagem (Montagem de *Kit*):

É realizada a inspeção final (visual) do conjunto platô, além da verificação da gravação do código de barras, por um operador, assim como a embalagem final do item. Os conjuntos são dispostos, manualmente, em locais apropriados (pontos de estoque) para a futura expedição.

O tempo de ciclo desta operação corresponde a 12,83 segundos.

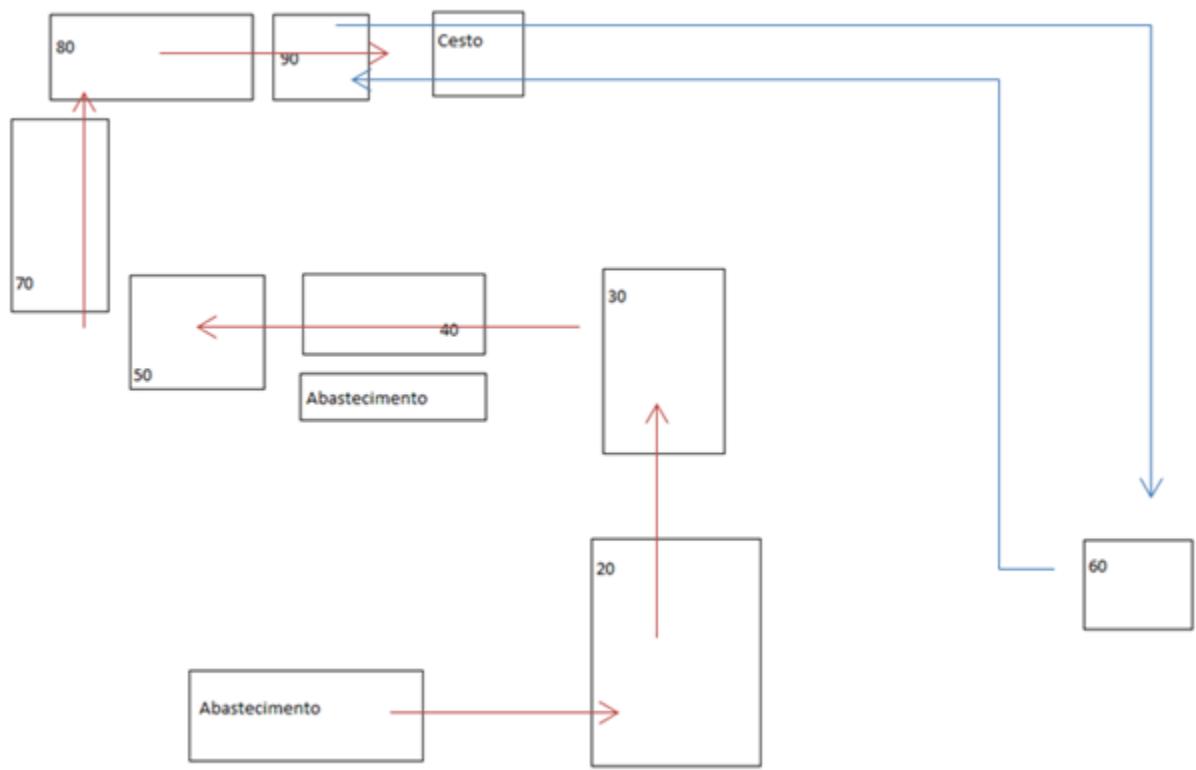


Figura 4 - *Layout* da célula A

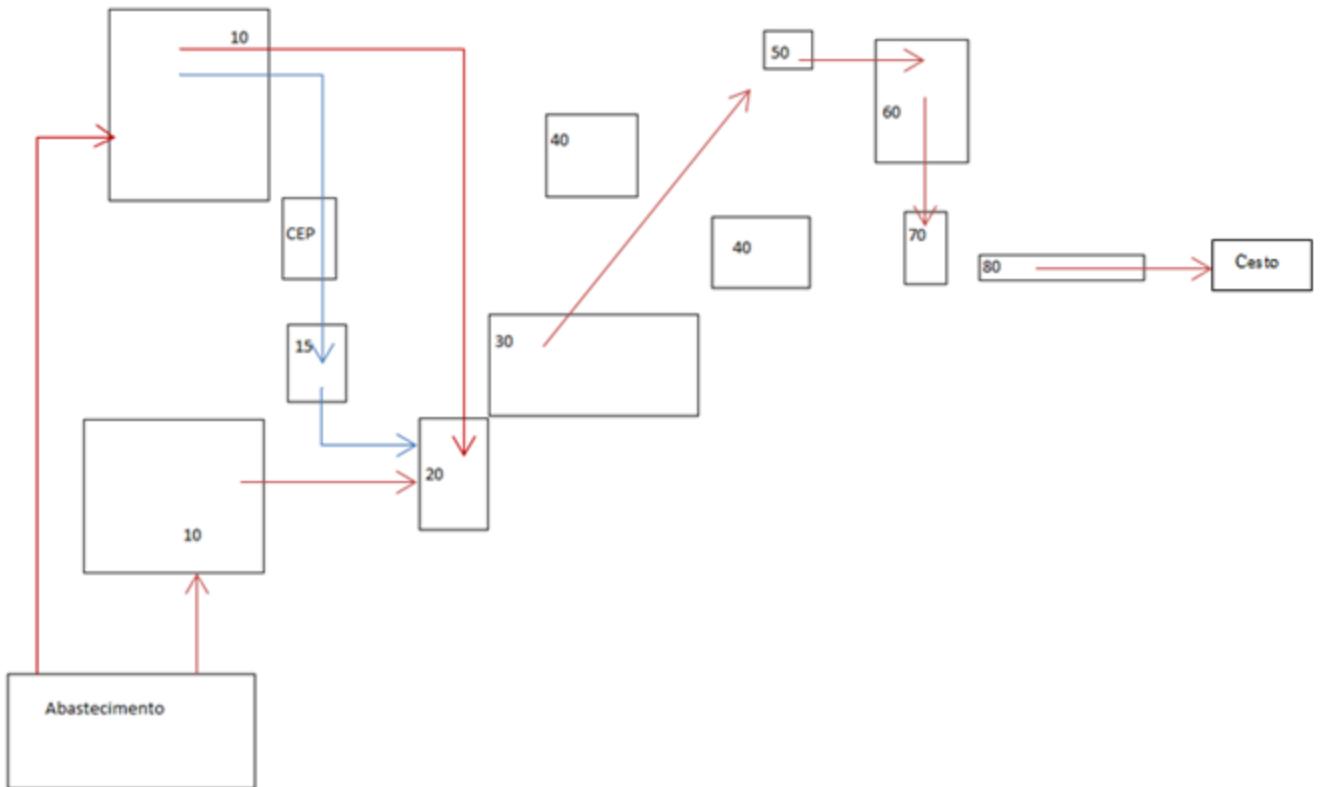


Figura 5 - *Layout* da célula B

Legenda das figuras 4 e 5:

Fluxo de produtos →

Fluxo eventual de produtos →

Para a complementação do estudo de otimização das células apresentadas anteriormente, realizou-se o mapeamento do fluxo de valor de ambas as células, os quais estão representados nas figuras 6 e 7.

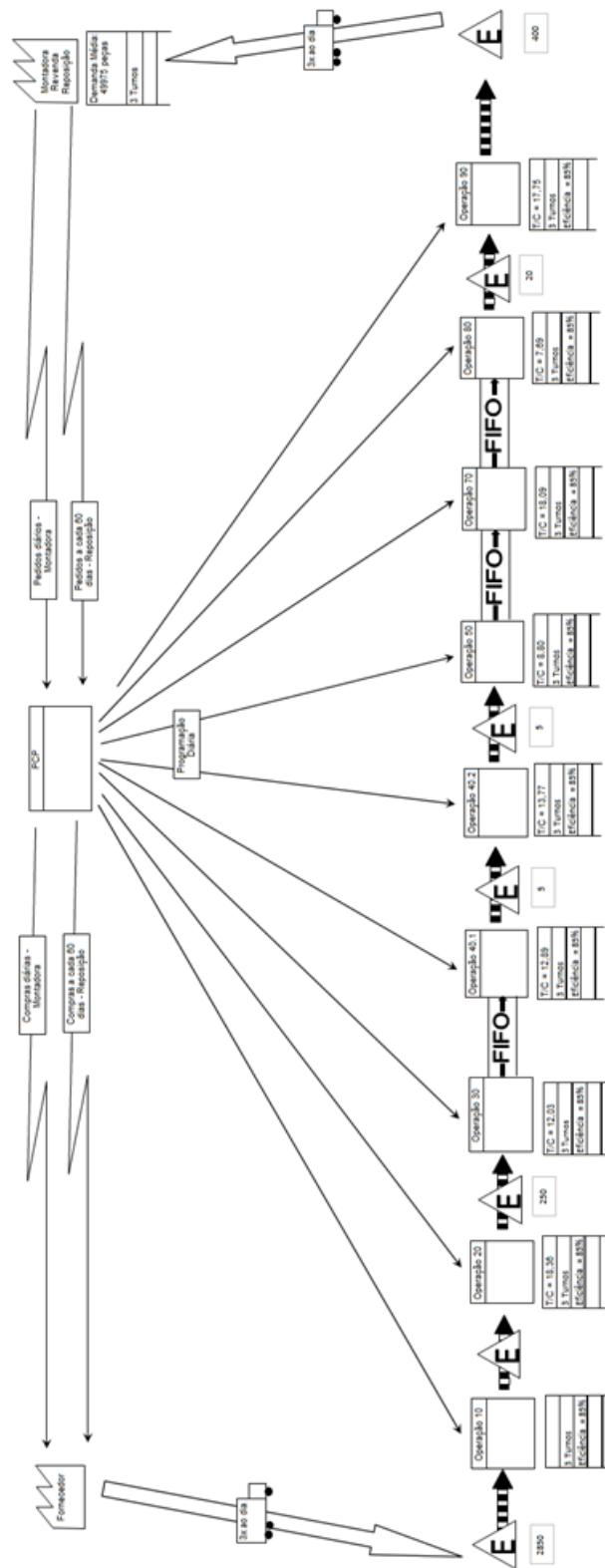


Figura 6 – Mapeamento do fluxo de valor da célula A

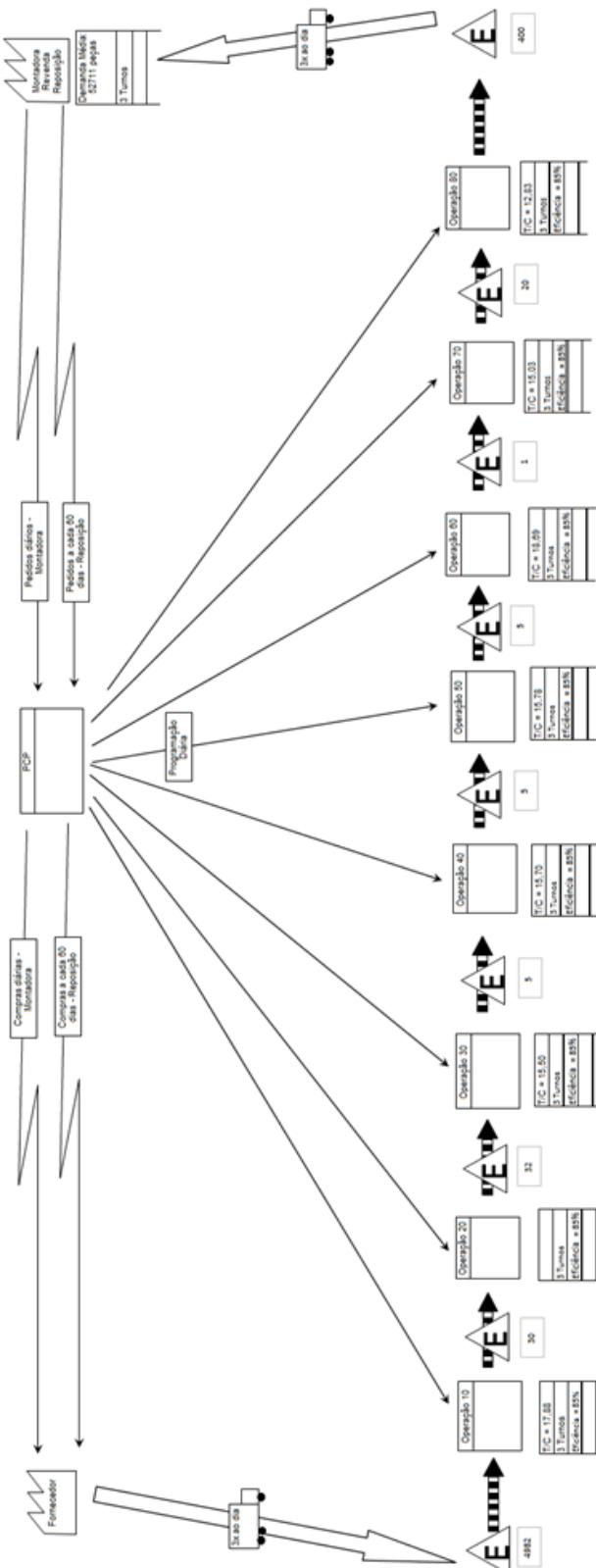


Figura 7 – Mapeamento do fluxo de valor da célula B

## 4.1 MODELO DE SIMULAÇÃO

Conjuntamente com a empresa objeto de estudo deste trabalho, foi desenvolvido um modelo utilizando o *software* Plant Simulation®, que visa retratar o estado atual das células A e B, apresentadas anteriormente.

O modelo considera três processos básicos, são eles: o estoque de componentes; manufatura (montagem dos componentes) e o estoque de produtos acabados. Esses processos são auxiliados por seis métodos (rotinas de programação), os quais fornecem informações como ordens de produção, critérios de sequenciamento de ordens, tempos de ciclo, *setup* de operações da manufatura, entre outras.

A representação geral do modelo pode ser observada na Figura 8.



Figura 8 – Representação geral do modelo

Entre os métodos (rotinas de programação) desenvolvidos, pode-se citar o responsável pela atribuição de ordens à manufatura, detalhado em seguida:

```
for i:= 1 to tabProgrDoDia.yDim loop
    if tabProgrDoDia["status", i] = "A" and pos(aCelula,
tabProgrDoDia["celula", i]) > 0 then
        k:= tf_progrCel.yDim + 1;

        tf_progrCel["OrdemProd", k]:= tabProgrDoDia["OrdemProd", i];
        tf_progrCel["Material", k]:= tabProgrDoDia["Material", i];
        tf_progrCel["idxCli", k]:= tabProgrDoDia["idxCli", i];
        tf_progrCel["idxProgr", k]:= i;
        tf_progrCel["diaOP", k]:= tabProgrDoDia["diaOP", i];
        tf_progrCel["diaPlan", k]:= tabProgrDoDia["diaPlan", i];
        tf_progrCel["diaProgr", k]:= tabProgrDoDia["diaProgr", i];
        tf_progrCel["lTime", k]:= tabProgrDoDia["lTime", i];
        tf_progrCel["qtdOP", k]:= tabProgrDoDia["qtdOP", i];
        tf_progrCel["embOP", k]:= tabProgrDoDia["embOP", i];
```

```
end;
```

Este método está interligado ao programa de coordenação de ordens da fábrica, desta forma, o trecho de código irá executar uma varredura por ordens abertas e atribuí-las à tabela `tf_progrCel`, utilizada pelo modelo, para que se possa iniciar a montagem dos componentes.

Outro método de extrema utilidade é aquele responsável pelos critérios de sequenciamento de ordens, em que os diversos tipos de clientes são discriminados e suas importâncias relativas são atribuídas. Além disso, estão inclusos critérios de priorização de pedidos em atraso assim como de retardo, ou suspensão temporária, daqueles em que haja falta de componentes.

```
for i:= 1 to tf_progrCel.yDim loop
    tf_progrSeq["OrdemProd", i]:= tf_progrCel["OrdemProd", i];
next;

for i:= 1 to tf_progrCel.yDim loop
    idxCli:= tf_progrCel["idxCli" , i];
    sOP   := tf_progrCel["OrdemProd", i];
    sMat  := tf_progrCel["Material" , i];

    sSuf  := tfCli["Mercado", idxCli];
```

O trecho de código executa os diferentes tipos de critérios escolhidos pelo usuário, neste modelo são empregados critérios como tipo de cliente, representados por `idxCli`; prazo de entrega do produto acabado, `sOP`; e disponibilidade de componentes, `sMat`.

Este outro método a seguir é responsável por providenciar informações relativas à produção, tais como, tempos de ciclo, *setup*, tempo *takt*, tamanho de lote, descrição dos componentes e produtos finais, e filtrá-las de acordo com cada célula produtiva, fornecendo assim informações indispensáveis para o funcionamento das células A e B.

```
tf_procPadrao.delete({0,0}..{*,*});
tf_procPadrao[ 0, 0]:= "Material";
tf_procPadrao[ 1, 0]:= "Descricao";
tf_procPadrao[ 2, 0]:= "tact time";
tf_procPadrao[ 3, 0]:= "tact efic";
tf_procPadrao[ 4, 0]:= "set up";
tf_procPadrao[ 5, 0]:= "lote min";
tf_procPadrao[ 6, 0]:= "lote mult";

for i:= 4 to tf_SAP_procPadrao.yDim loop
if tf_SAP_procPadrao[ 0, i] = "" then
exitLoop;
end;

j:= tf_procPadrao.yDim + 1;
```

```

tf_procPadrao["Material", j]:= tf_SAP_procPadrao[ 0, i] +
tf_SAP_procPadrao[ 1, i];
tf_procPadrao["Descricao", j]:= tf_SAP_procPadrao[ 2, i];
tf_procPadrao["tact time", j]:= str_to_num(tf_SAP_procPadrao[11, i]);
tf_procPadrao["tact efic", j]:= str_to_num(tf_SAP_procPadrao[12, i]);
tf_procPadrao["set up", j]:= str_to_num(tf_SAP_procPadrao[28, i]);
tf_procPadrao["lote min", j]:= str_to_num(tf_SAP_procPadrao[22, i]);

```

O trecho de código em questão é responsável por criar a tabela `tf_procPadrao`, contendo informações selecionadas e filtradas da tabela `tf_SAP_procPadrao`. Esta última contempla dados obtidos do sistema SAP® da planta da empresa.

Além disso, há o método responsável pela produção, o qual irá atribuir à manufatura, todas as ordens selecionadas e sequenciadas pelos métodos anteriores.

```

tOP:= root.fr_dados.tf_ordemProd;
k:= tOP.yDim + 1;
tOP["OrdemProd", k]:= sOP;
tOP["Material" , k]:= sMat;
tOP["celula" , k]:= current.name;
tOP["idxCli" , k]:= idxCli;
tOP["status" , k]:= "A";
tOP["qtdOP" , k]:= iQtd;
tOP["embOP" , k]:= iEmb;
tOP["lTime" , k]:= lTime;
tOP["diaOP" , k]:= iHoje;
tOP["diaPlan" , k]:= diaPlan;
tOP["diaProgr" , k]:= diaProgr;
tOP["diaInic" , k]:= iHoje;
tOP["timeInic" , k]:= eventController.absSimTime;

```

Este trecho de código irá copiar as ordens de produção, definidas pelos métodos anteriores, para as tabelas de programação diária.

```

tfEstq:= root.fr_dados.tf_infoPlan;
tfBOM := root.fr_dados.tf_BOM.mtd_buscaComp(sMat);
for j:= 1 to tfBOM.yDim loop
  sComp:= tfBOM["Componente", j];
  jQtd := tfBOM["qtd" , j] * iQtd;

  tfEstq["reservARA", sComp]:= jQtd;
next;

```

Este código é responsável pela busca dos componentes a serem produzidos, assim como suas quantidades, utilizando as informações de planejamento.

```

var_emProducao:= true;
tOP:= root.fr_dados.tf_progrDoDia;
tOP["status" , sOPt]:= "I";
tOP["OrdemProd", sOPt]:= sOP;

```

Finalmente, há a sinalização para o modelo de que a célula está produzindo, assim é possível gerar gráficos e relatórios para a aferição do desempenho de determinada célula.

Os outros dois métodos ainda existentes são responsáveis pelo ajuste do tempo de ciclo e da quantidade produzida, conferindo ao modelo características mais precisas de um cenário real de manufatura.

## 5. ADEQUAÇÃO DO MODELO DE SIMULAÇÃO

O modelo já existente, descrito na seção anterior e desenvolvido pela empresa parceira, foi simplificado de forma a apresentar mais didaticamente e com maior clareza, o entendimento geral do modelo e dos processos de fabricação, assim como proporcionar melhor entendimento dos resultados obtidos na simulação.

Dessa maneira, o novo modelo, simplificado, utilizado está representado na Figura 9.

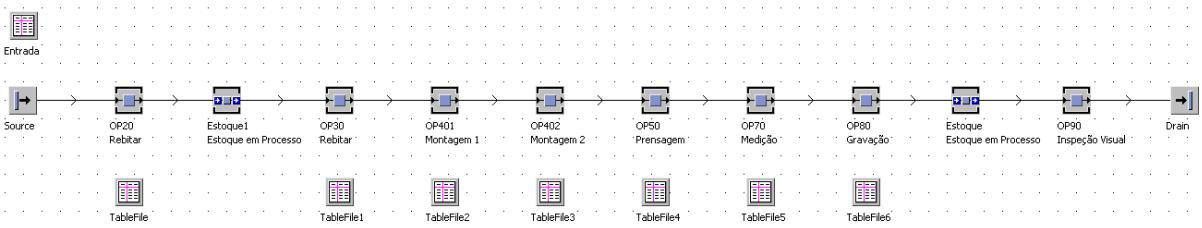


Figura 9 – Representação geral do modelo simplificado

Este modelo retrata uma das células manufatureiras, a célula A, apresentada e discutida na seção 4, e proporciona maior nível de detalhamento dos processos intracelulares. Além disso, são consideradas todas as operações efetivamente realizadas nesta célula, calibradas com os dados obtidos nas etapas anteriores do projeto.

O modelo considera a produção de quatro itens distintos baseando-se em dados como tempos processamento e *setup*.

Os arquivos de tabela, denominados *TableFile* na figura 9, representam os diferentes tempos de *setup* dos quatro itens (1, 2, 3 e 4) em cada operação. Esses dados foram estimados e extrapolados para três itens baseando-se no tempo de *setup* conhecido de um dos itens. As demais informações como tempos de processamento, formações de estoque em processo entre outras, foram cadastrados segundo observado na planta da empresa, durante a etapa de coleta de dados.

## 6. ANÁLISE E DISCUSSÃO DOS RESULTADOS

Uma vez pronto o modelo simplificado da célula estudada, aplicou-se o algoritmo genético (AG) com o intuito de determinar o sequenciamento ótimo dos itens produzidos, considerando para isso o tempo de processamento de cada operação, assim como o tempo de *setup* exigido por cada item em cada máquina.

Neste caso, a função a ser minimizada é o tempo total de simulação, representada pelo comando:

```
root.EventController.SimTime
```

Ou seja, quanto menor o tempo de simulação que uma solução ou indivíduo apresentar, maior será sua aptidão. Dessa maneira, a solução ótimo desejada corresponde àquela com o menor tempo de simulação.

Foram consideradas, durante a otimização com AG, uma população inicial composta por trinta indivíduos, que corresponde ao tamanho da geração, e cinquenta gerações de descendentes. Além disso, foram feitas cinco observações por indivíduo para maior confiabilidade estatística.

Os resultados observados estão representados na figura 10.

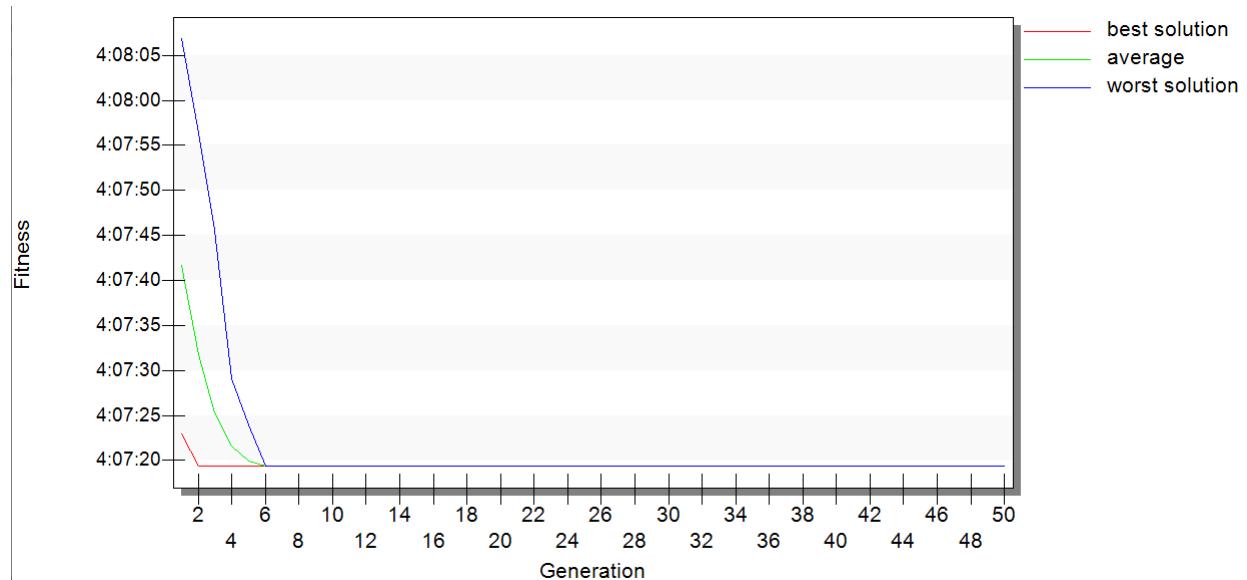


Figura 10 – Gráfico de aptidão

Nota-se que a melhor solução, apresentada em vermelho no gráfico da figura 10, converge mais rapidamente para a solução ótima, enquanto que as demais soluções necessitam de uma maior quantidade de iterações, ou seja, geração de descentes para atingir o mesmo resultado.

Dessa forma, avaliando a melhor solução, verifica-se que o sequenciamento ótimo dos itens considerados corresponde a: itens 4, 2, 3 e 1.

O ganho de quase 1 minuto pode ser considerado baixo, uma vez que representa apenas 0,4% do tempo total de simulação. Isso ocorre devido à pequena variação nos tempos de *setup* dos itens considerados, entretanto sua aplicação é extremamente bem vindas em ambientes de manufatura complexos já que o volume de dados inibe qualquer tentativa de otimização utilizando ferramentas tradicionais. Para efeito de comparação, apenas neste modelo foram gerados quase 3000 indivíduos, cada um correspondendo a uma solução distinta.

#### Observações:

- Devido à algumas limitações do método utilizado pelo *software*, a presença de falhas não foram consideradas durante a otimização, ou seja, os processos apresentam 100% de eficiência. Além disso, turnos também não puderam ser acrescentados;
- A otimização do modelo, por extensão, da célula de manufatura representada, está configurada baseando-se apenas nos fluxos de materiais e agendamento de tarefas (*scheduling*), ou seja, pode não representar a configuração ótima da célula se forem considerados mais parâmetros como estoque em processo, custos de *setup*, entre outros fatores.

## 7. REFERÊNCIAS BIBLIOGRÁFICAS

DURIE, F.R.E. *A survey of group technology and its potential for user application in UK*. Production Engineering, v.49, n.2, 1970

FERNANDES, J. *Estudo da Implantação de um Layout Celular*. Universidade do Estado de Santa Catarina, 2007.

FERNANDES, F.C.F.; FILHO, M.G. *Planejamento e Controle da Produção: Dos fundamentos ao essencial*. São Paulo: Editora Atlas, 2010.

OZDEN, M.; EGBELU, P.J.; IYER, A.V. *Job scheduling in a group technology environment for a single facility*. Comput. & Indust. Engineering, v.9, n.1, pp. 66-72, 1985.

SÉRIO, L.C. *Tecnologia de grupo no planejamento de um sistema produtivo*. São Paulo: Ícone, 1990.

SINGH, N.; RAJAMANI, D. *Cellular Manufacturing Systems. Design, planning and control*. Chapman & Hall, 1996.

SLACK, N.; CHAMBERS, S.; HARLAND, C.; HARRISON, A.; JOHNSTON, R. *Administração da Produção*. São Paulo: Atlas, 1999.



