

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA

SISTEMA PARA PROJETO DE CÉLULA DE MANUFATURA:
DEFINIÇÃO DE AGRUPAMENTOS

por

Elias Baldissera Amantino

Dissertação para obtenção do Título de
Mestre em Engenharia

Porto Alegre, outubro de 2005.

Livros Grátis

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

Milhares de livros grátis para download.

SISTEMA PARA PROJETO DE CÉLULA DE MANUFATURA:
DEFINIÇÃO DE AGRUPAMENTOS

por

Elias Baldissera Amantino
Engenheiro Mecânico

Dissertação submetida ao Corpo Docente do Programa de Pós-Graduação em Engenharia Mecânica, PROMEC, da Escola de Engenharia da Universidade Federal do Rio Grande do Sul, como parte dos requisitos necessários para a obtenção do Título de

Mestre em Engenharia

Área de Concentração: Processo de Fabricação

Orientador: Prof. Dr. Flávio José Lorini

Comissão de Avaliação:

Prof. Dr. Tarcício Abreu Saurim, PPGEP/UFRGS

Prof. Dr. Ney Francisco Ferreira, DEMEC/UFRGS

Prof. Dr. Joyson Luiz Pacheco, PROMEC/UFRGS

Prof. Dr. Flávio José Lorini
Coordenador do PROMEC

Porto Alegre, outubro de 2005.

À minha esposa Candice e aos meus filhos Matheus e Julia.

À minha avó Anita (*in memoriam*).

Aos meus avós Biaggio e Paulina.

A meus pais Manoel e Jane.

Ao professor e orientador Dr. Flávio José Lorini.

Aos amigos.

AGRADECIMENTOS

Ao professor Juan Carlos Sandoval Ortiz, pelo incentivo a ingressar no mestrado e por ter me indicado o professor Dr. Flávio José Lorini como orientador.

Ao professor Dr. Flávio José Lorini por ter me aceitado como orientado.

À Universidade Federal do Rio Grande do Sul (UFRGS), ao Programa de Pós-Graduação em Engenharia Mecânica da UFRGS (PROMEC) e ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) pela bolsa concedida.

Aos professores do Grupo de Projeto, Fabricação e Automação Industrial (Gpfai) e do Departamento de Engenharia Mecânica da UFRGS pelas oportunidades que me proporcionaram com cursos, viagens, estágios e de lecionar as disciplinas de Prática de Oficina, Projeto Assistido por Computador e Processos de Fabricação por Usinagem.

Aos membros da banca de avaliação, Prof. Tarcício Abreu Saurim, Prof. Ney Francisco Ferreira e Prof. Luiz Pacheco.

Aos funcionários do PROMEC, do DEMEC, Gpfai e da UFRGS.

Aos professores, amigos e colegas.

Elias Baldissera Amantino

RESUMO

O entendimento da manufatura celular passa pelo estudo das rotas de fabricação e pelos métodos de agrupamento de máquinas e de peças. Neste contexto, o objetivo principal deste trabalho é a implementação de uma ferramenta de auxílio ao projeto de células de manufatura, abordando uma metodologia fundamentada na Tecnologia de Grupo para agrupamento de máquinas em células de manufatura com as respectivas famílias de peças. A base de dados com as informações das peças, das máquinas e das rotas que compõe o fluxo de produção é implementada em um banco de dados relacional. A matriz incidência peça-máquina é montada a partir das rotas armazenadas no banco de dados através de um aplicativo desenvolvido em *Visual Basic*. Os agrupamentos em famílias de peças e células de manufatura são gerados por três diferentes algoritmos: o *Rank Order Clustering (ROC)*, o *Close Neighbor Algorithm (CNA)* e um algoritmo heurístico (*HEU*). São aplicadas restrições referentes a limite de carregamento, tamanho de célula e resolução de situações de “gargalo”. Processados os algoritmos de agrupamento, são analisados os resultados em função da densidade e da eficiência do agrupamento. O sistema apresenta o resultado final em planilhas no formato *MS Excel*. A primeira planilha, chamada *resultados*, exibe os valores das restrições de projeto das células (número de máquinas por célula, tempo limite de carregamento e tempo limite para duplicação da máquina), o número de peças (colunas) e de máquinas (linhas) da matriz incidência, os valores de eficiência do agrupamento de peças, do agrupamento de máquinas e do aproveitamento dos recursos, bem como o número de células independentes e a quantidade de máquinas duplicadas, obtidos por cada algoritmo no sistema, destacando os melhores resultados. A segunda planilha mostra a matriz incidência peça-máquina. As demais planilhas apresentam, respectivamente, a matriz diagonalizada com o algoritmo original (*ROC* e *CNA*), a matriz diagonalizada levando-se em consideração as restrições de projeto das células (*análise ROC*, *análise CNA* e *HEU*) e por fim, uma planilha relatório. A planilha relatório tabula os mesmos valores citados na primeira planilha e lista as peças associadas a cada família, as máquinas associadas a cada célula, as peças rejeitadas, por não se enquadrarem nos agrupamentos, e que devem ser tratadas fora do ambiente celular e as máquinas duplicadas.

A comparação dos resultados é efetuada considerando as características adicionadas aos algoritmos originais. Dos três estudados, as restrições de projeto são tratadas na implementação do *HEU*. Os demais, *ROC* e *CNA*, têm incorporado um pós processamento. Em análises comparativas observa-se a superioridade dos algoritmos *ROC* e *HEU* em relação ao *CNA* e os resultados do *ROC* são melhores ou iguais aos demais, dificilmente inferiores.

ABSTRACT

Cellular manufacturing design system: clustering definition

The understanding of the cellular manufacture goes through the production routes and machines and parts grouping methods study. In this context, the main goal of this work is a tool implementation to aid the manufacture cells design, approaching a applied methodology in Group Technology for machine grouping in manufacture cells with their respective part families. The database with the information on parts, machines and production flow routings is implemented in a relational database. The part-machine incidence matrix is built up from the routes stored in this database through an application developed in Visual Basic instruction code. The groupings in part families and manufacture cells are generated by three different algorithms: Rank Order Clustering (*ROC*), Close Neighbor Algorithm (*CNA*) and a heuristic algorithm (*HEU*). Restrictions are applied with regard to loading limit, cell size and resolution of bottleneck situations. Once the grouping algorithms are processed, the results in function of the density and of the efficiency of the grouping are analyzed. The system presents the final result in *MS Excel* spreadsheets. The first spreadsheet, called *resultados*, shows the values of the project restrictions of the cells (number of machines per cell, time shipment limit and time limits for machine duplication), parts numbers (columns) and machine numbers (rows) of the incidence matrix, the grouping efficiency values of parts, of machine grouping and of resource use, as well as the independent cells number and the duplicated machines amount, obtained by each algorithm in the system, making evident the best results. The second spreadsheet shows the part-machine incidence matrix. The other spreadsheets present, respectively, the block diagonal matrices with the original algorithm (*ROC* and *CNA*), the block diagonal matrices considering the cell project restrictions (*ROC*, *CNA* and heuristic analysis) and finally, a spreadsheet report. The latter tabulates the same values mentioned in the first spreadsheet and it lists the parts associated to each family, the machines associated to each cell, the rejected parts, for they don't frame in the groupings, and that should be treated outside of the cellular environment and the duplicated machines.

The results comparison is made considering the characteristics added to the original algorithms. In an algorithms studied, design restrictions are treated only in the *HEU* implementation. The remaining algorithms, *ROC* and *CNA*, had incorporated a postprocessing. Comparative analysis showed the superiority of the *ROC* and *HEU* algorithms in relation to *CAN*. The *ROC* algorithm presents worst results, been usually better than the remaining ones.

ÍNDICE

1	INTRODUÇÃO	1
2	ARRANJO FÍSICO NA ÁREA DE MANUFATURA.....	3
2.1	<i>Layout</i> em linha ou de produto.....	4
2.2	<i>Layout</i> funcional (<i>job shop</i>)	4
2.3	<i>Layout</i> celular ou de grupo.....	5
3	MANUFATURA CELULAR	6
4	MÉTODOS DE SOLUÇÃO NA FORMAÇÃO DE CÉLULAS.....	13
4.1	Procedimentos descritivos.....	13
4.2	Procedimentos baseados em análise de agrupamento	16
4.2.1	<i>Rank Order Clustering (ROC)</i>	17
4.2.2	<i>Close Neighbour Algorithm (CNA)</i>	18
4.2.3	Procedimento heurístico (<i>HEU</i>)	23
4.3	Aproximações por particionamento gráfico	25
4.4	Aproximações por inteligência artificial	25
4.5	Aproximações por programação matemática	26
4.6	Algoritmos genéticos (<i>GAs</i>) – estado da arte	26
5	RESTRICÇÕES NO PROJETO DA CÉLULA	31
5.1	O tamanho da célula	31
5.2	Carregamento das máquinas.....	31
5.3	Interdependência de máquinas	33
5.4	Balanceamento das células.....	33
6	MEDIDAS DE DESEMPENHO	34
7	TRABALHO IMPLEMENTADO	37
7.1	Considerações sobre a escolha de uma metodologia	37
7.2	Estrutura do sistema e sua implementação.....	38
7.2.1	Modo de inicialização principal	39
7.2.2	Módulo para cadastramento	39
7.2.3	Módulos de processamento do sistema	44
7.2.3.1	Algoritmos <i>ROC</i> e <i>CNA</i>	46
7.2.3.2	Procedimento heurístico (<i>HEU</i>)	57
7.2.3.3	Apresentação dos resultados	59
8	TESTES EXPERIMENTAIS – VALIDAÇÃO DO SISTEMA	62
8.1	Matriz <i>teste_2</i>	62
8.2	Matriz <i>teste_3</i>	65
8.3	Análise comparativa dos resultados	69
9	CONCLUSÕES E SUGESTÕES PARA TRABALHOS FUTUROS.....	74
9.1	Características operacionais importantes do sistema proposto	75
9.2	Limitações	76
9.3	Sugestões para continuidade do trabalho	76
	REFERÊNCIAS BIBLIOGRÁFICAS.....	77
	APÊNDICE I – Relacionamento entre tabelas.....	80

LISTA DE SÍMBOLOS

<i>BEA</i>	<i>Bond Energy Analysis</i>
<i>CAPP</i>	Sistemas de Auxílio ao Planejamento da Produção
<i>CIA</i>	<i>Cluster Identification Algorithm</i>
<i>CNA</i>	<i>Close Neighbour Algorithm</i>
<i>CNC</i>	Comando Numérico Computadorizado
<i>DCA</i>	<i>Direct Clustering Analysis</i>
<i>DCLASS</i>	<i>Design and Classification Information System</i>
<i>DP</i>	Programação Dinâmica
<i>FMS</i>	Sistemas de Manufatura Flexível
<i>GA</i>	Algoritmo Genético
<i>GP</i>	Programação Objetiva
<i>HEU</i>	Procedimento Heurístico
<i>JSPMI</i>	Sociedade Japonesa para a Produção de Máquinas Industriais
<i>LP</i>	Programação Linear
<i>LQP</i>	Programação Linear e Quadrática de Números Inteiros
<i>MGI</i>	Identificação de Grupos de Máquinas
<i>MICLASS</i>	<i>Metal Institute Classification System</i>
<i>MODROC</i>	<i>Modified Rank Order Clustering</i>
<i>OIR</i>	Organização para Pesquisa Industrial
<i>PCRS</i>	Probabilidade de Recombinação
<i>PF/MG</i>	Agrupamentos de Famílias de Peças e de Grupos de Máquinas
<i>PFA</i>	Análise do Fluxo de Produção
<i>PFI</i>	Identificação de Famílias de Peças
<i>ROC</i>	<i>Rank Order Clustering</i>
<i>ROC2</i>	<i>Rank Order Clustering 2</i>
<i>TNO</i>	<i>The Netherlands Organization for Applied Scientific Research</i>
<i>VCM</i>	Manufatura Celular Virtual

ÍNDICE DE FIGURAS

Figura 2.1 – Distribuição do tempo em processo.....	3
Figura 2.2 – Esquemas de <i>layout</i> em linha (a) e funcional (b).....	4
Figura 2.3 – <i>Layout</i> celular dedicado à manufatura de uma família de peças.	5
Figura 3.1 – Exemplo de <i>Layout</i> de sistema de manufatura celular virtual.....	11
Figura 4.1 – Classificação dos métodos de formação de células.....	13
Figura 4.2 – Matriz incidência peça-máquina.....	14
Figura 4.3 – Matriz incidência peça-máquina reordenada.....	15
Figura 4.4 – Matriz com ponderação nas linhas.....	17
Figura 4.5 – Matriz com ponderação nas colunas.....	18
Figura 4.6 – Matriz com linhas e colunas ordenadas.....	18
Figura 4.7 – Matriz incidência peça-máquina.....	19
Figura 4.8 – Matriz B máquina-máquina.....	19
Figura 4.9 – Reordenamento das linhas da matriz incidência.....	20
Figura 4.10 – Reordenamento das linhas da matriz incidência.....	20
Figura 4.11 – Sumário de resultados do estágio 2 aplicado ao exemplo.....	21
Figura 4.12 – Sumário de resultados do estágio dois aplicado ao exemplo.....	22
Figura 4.13 – Reorganização final da matriz de incidências.....	23
Figura 4.14 – Matriz com tempos de processo $t(i, j)$	24
Figura 4.15 – representação do cromossomo.....	28
Figura 6.1 – Exemplo de matriz para cálculo de GE	36
Figura 7.1 – Fluxograma do sistema.....	38
Figura 7.2 – Formulário <i>entrada</i>	39
Figura 7.3 – Formulário <i>f_disponibilidade</i>	40
Figura 7.4 – Formulário <i>f_maquinas</i>	41
Figura 7.5 – Formulário <i>f_pecas</i>	42
Figura 7.6 – Formulário <i>f_rotas</i> e subformulário <i>f_rotas sub</i>	43
Figura 7.7 – Formulário <i>matriz</i>	44
Figura 7.8 – Etapas da rotina <i>m04_mat_incidencia</i>	45
Figura 7.9 – Matriz <i>teste_1</i>	45
Figura 7.10 – Seqüência da rotina <i>m05_algoritmo_roc</i>	46
Figura 7.11 – Matriz <i>teste_1</i> ordenada pela rotina <i>m05_algoritmo_roc</i>	47
Figura 7.12 – Seqüência da rotina <i>06_algoritmo_cna_ep</i>	47
Figura 7.13 – Matriz <i>teste_1</i> ordenada pela rotina <i>m06_algoritmo_cna_ep</i>	48
Figura 7.14 – Seqüência da rotina <i>07_algoritmo_cna</i>	49
Figura 7.15 – Matriz <i>teste_1</i> ordenada pela rotina <i>m07_algoritmo_cna</i>	49
Figura 7.16 – Células candidatas a início de famílias de peças e grupos de máquinas.....	50
Figura 7.17 – Uma célula encontrada para a matriz <i>teste_1</i>	51
Figura 7.18 – Quatro células encontradas para a matriz <i>teste_1</i>	51
Figura 7.19 – Aplicação da restrição sobre a matriz <i>teste_1</i>	52
Figura 7.20 – Aplicação da restrição sobre a matriz <i>teste_1</i>	53
Figura 7.21 – Aplicação da restrição e ordenamento <i>ROC</i> sobre a matriz <i>teste_1</i>	54
Figura 7.22 – Aplicação da restrição sobre parte da matriz <i>teste_1</i>	55
Figura 7.23 – Aplicação da restrição sobre parte da matriz <i>teste_1</i>	56
Figura 7.24 – Aplicação da restrição sobre parte da matriz <i>teste_1</i>	56
Figura 7.25 – Aplicação da restrição sobre parte da matriz <i>teste_1</i>	57
Figura 7.26 – Aplicação da restrição sobre parte da matriz <i>teste_1</i>	57
Figura 7.27 – Matriz <i>teste_1</i> ordenada pelo procedimento <i>HEU</i>	58
Figura 8.1 – Matriz <i>teste_2</i>	62

Figura 8.2 – Matriz <i>teste_2</i> ordenada com <i>ROC</i>	63
Figura 8.3 – Matriz <i>teste_2</i> ordenada com <i>CNA EP</i>	63
Figura 8.4 – Matriz <i>teste_2</i> ordenada com <i>CNA</i>	64
Figura 8.5 – Matriz <i>teste_2</i> ordenada com o <i>HEU</i>	64
Figura 8.6 – Matriz ordenada de Lee <i>et al.</i>	65
Figura 8.7 – Matriz <i>teste_3</i>	65
Figura 8.8 – Matriz <i>teste_3</i> ordenada com <i>ROC</i>	66
Figura 8.9 – Matriz <i>teste_3</i> ordenada com <i>ROC</i> e tamanho de célula limitado.....	66
Figura 8.10 – Matriz <i>teste_3</i> ordenada com <i>CNA</i>	67
Figura 8.11 – Matriz <i>teste_3</i> ordenada com <i>CNA</i> e tamanho de célula limitado.....	67
Figura 8.12 – Matriz <i>teste_3</i> ordenada com <i>CNA</i> e tamanho de célula limitado.....	67
Figura 8.13 – Matriz <i>teste_3</i> ordenada com <i>CNA</i> e tamanho de célula limitado manualmente. ..	68
Figura 8.14 – Matriz <i>teste_3</i> ordenada com <i>HEU</i>	68
Figura 8.15 – Matriz <i>teste_3</i> ordenada com <i>HEU</i> e tamanho de célula limitado.	69
Figura 8.16 – Agrupamento <i>ROC</i> das peças restantes.	72
Figura 8.17 – Sobreposição das matrizes rearranjadas.	73
Figura AI.1 – Relacionamentos entre tabelas de dados.	81

ÍNDICE DE TABELAS

Tabela 7.1 – Planilha <i>resultados</i>	60
Tabela 7.2 – Planilha <i>relatório ROC</i>	61
Tabela 8.1 – Planilha <i>resultados</i> para a matriz <i>teste_2</i>	69
Tabela 8.2 – Resultados comparativos para a matriz <i>teste_3</i>	70
Tabela 8.3 – <i>Relatório ROC</i> para a matriz <i>teste_3</i>	71
Tabela 8.4 – <i>Relatório heurístico</i> para a matriz <i>teste_3</i>	71
Tabela 8.5 – Resultados comparativos para a matriz <i>teste_3</i>	72

1 INTRODUÇÃO

A acirrada competição internacional exerce crescente pressão sobre as indústrias de manufatura para adequarem seus sistemas produtivos no sentido de serem mais eficazes e competitivos. Este contexto manifesta-se pelo fato que mercados de consumidores de bens mostram um crescimento na variedade e uma diminuição no ciclo de vida dos produtos. Os sistemas de manufatura clássicos tais como *layout* de processo e *layout* de produto não tem a habilidade para responder rapidamente às mudanças necessárias nesse contexto de produção.

Recentemente, muitas organizações de manufatura foram inábeis para competir com um crescimento rápido das mudanças do mercado, perdendo posição e competitividade. A vida dos produtos tende a ser cada vez menor com ao passar do tempo (Drolet *et. al.*, 1996); isto força as organizações de manufatura a reduzirem ciclo de desenvolvimento de novos produtos, aumentarem a responsividade e a flexibilidade, além de diminuir o tempo de *setup* e baixarem os inventários entre processos enquanto mantém uma eficiência aceitável.

Diversas abordagens e conceitos têm sido aplicados para caracterização de novos sistemas de manufatura: agilidade, flexibilidade, inteligência, etc., para nomear alguns. Solberg apud Drolet *et al.* (1996) define sistemas de manufatura inteligentes como aqueles que são responsivos, indicando uma necessidade por sistemas de manufatura que possuam a “habilidade inata para responder, pronta e corretamente, a mudanças em exigências”.

Do ponto de vista organizacional, modernos sistemas de manufatura são concebidos e implantados como células, onde o arranjo físico e o fluxo lógico são definidos de maneira racional para a produção de um universo específico de produtos.

Diferentes metodologias podem ser adotadas para esse objetivo destacando-se os procedimentos de análise do fluxo de produção sobre os quais são aplicadas técnicas dos respectivos algoritmos de agrupamento. Na literatura, grande parte dos trabalhos com essa finalidade limita-se a uma abordagem matemática, aplicada sobre uma massa de dados, buscando apenas sugerir agrupamentos preliminares de peças e máquinas, sem considerar condições operacionais.

Neste contexto o sistema proposto tem objetivos mais amplos: implementar uma ferramenta de auxílio ao projeto de células de manufatura, buscando além da definição de famílias de peças e grupos de máquinas através de diferentes algoritmos de agrupamento, inserir características inerentes a um projeto específico adotadas condições em paralelo aos procedimentos analíticos para adequar os agrupamentos ao *layout* e balanceamento na implementação de um sistema de fabricação.

Para auxiliar ao processo decisório na atividade de projeto dos agrupamentos, o sistema

deve apoiar-se, ainda, em informações contidas em um banco de dados de implementação associado neste trabalho.

O tema abordado insere-se em um contexto de organização industrial que implica em uma nova concepção de *layout* para sistemas de fabricação, assunto estudado juntamente com os diferentes algoritmos e técnicas de agrupamento abordados na revisão bibliográfica a seguir.

2 ARRANJO FÍSICO NA ÁREA DE MANUFATURA

O tipo de arranjo físico para máquinas e demais equipamentos de um sistema de manufatura assume relevante destaque, não somente em função dos processos, mas principalmente pela racionalização dos tempos de passagem dos lotes das peças entre os postos de trabalho.

Luggen apud Chan *et al.* (1999) e Bedworth e Groover apud Lorini (1993) enfatizam, porém, que considerando fabricação por usinagem, só 5% do tempo total no processo de manufatura são utilizados pela peça na operação de corte do metal, ressaltando ainda que quando a peça está em contato com metal-cortante em uma máquina-ferramenta particular, aproximadamente 30% dos 5% do tempo de manufatura corresponde a trabalho de fato e agregando valor, considerando que o resto do tempo é empregado para preparação, carga e descarga. Os 95% restantes do tempo total são gastos com movimentações da peça pela fábrica ou espera na fila para a próxima operação (Figura 2.1).

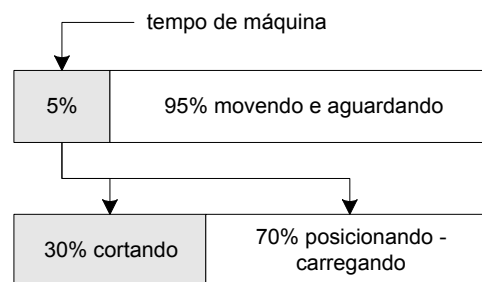


Figura 2.1 – Distribuição do tempo em processo.
Fonte: Lorini, 1993.

Essa constatação evidencia a importância de um bem definido fluxo de produção para execução em um projeto adequado de *layout*, de modo que atenda o fluxo com mínimos deslocamentos e dentro da lógica exigida na execução das tarefas de processo.

Em um enfoque mais abrangente (Shingo, 1996) identifica sete tipos de perda na produção: superprodução; espera; transporte; processamento; estoque; desperdício nos movimentos e na elaboração de produtos defeituosos. Afirmar também que melhorias obtidas utilizando técnicas de troca rápida de ferramentas reduzem em média 80 a 95% os tempos de *setup*; a eliminação das esperas do processo através do balanceamento de quantidades e sincronização pode reduzir o ciclo de produção em até 80%; a eliminação do estoque entre processos e pelo tamanho do lote, combinados, podem reduzir entre 86% a 93% do ciclo de produção. A redução destas perdas está diretamente associada a cada modelo de *layout* adotado para o sistema de fabricação.

Segundo Lorini (1993), considerando-se apenas o aspecto principal, que é a disposição das máquinas, têm-se três sistemas básicos de *layout* comumente encontrados nas instalações industriais, que são respectivamente: os arranjos em linha ou de produto, funcional e celular ou de grupo.

2.1 *Layout* em linha ou de produto

Trata-se do arranjo das máquinas na forma de uma linha de produção, cuja seqüência é adequada a determinado tipo de produto, conforme esquema da Figura 2.2(a). É um sistema utilizado para processos industriais relativamente simples, linhas de montagem ou de produção em massa, para grandes quantidades. Caracterizam-se, assim, por uma constituição de máquinas muito específicas, muitas vezes com sistema de transporte e manuseio bastante dedicado. A produtividade fica condicionada à velocidade da linha, que deve ser adequadamente balanceada, e a mão-de-obra aplicada é relativamente pequena. Considerando-se que a especificação e disposição das diferentes máquinas ficam condicionadas a determinado produto, caracteriza-se, neste tipo de arranjo, a pouca flexibilidade operacional.

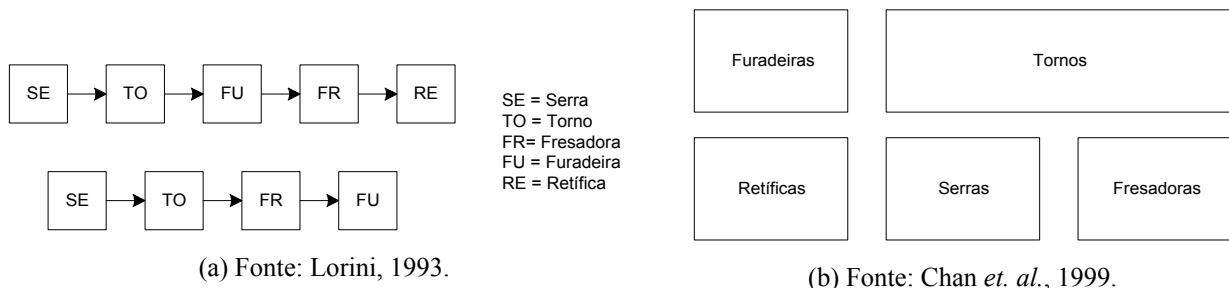


Figura 2.2 – Esquemas de *layout* em linha (a) e funcional (b).

2.2 *Layout* funcional (*job shop*)

O conceito de *layout* funcional corresponde à modalidade de arranjo de equipamentos mais comum e antiga encontrada na maioria das instalações industriais de manufatura. O *layout* envolve a disposição de máquinas do mesmo tipo próximas umas as outras (Lorini, 1993, e Chan *et. al.*, 1999, Ramos *et. al.*, 2004), constituindo-se assim, num arranjo típico de especialização por processo, porque as máquinas que realizam processos semelhantes ficam agrupadas numa mesma área física, no espaço da indústria. Como exemplo, para um ambiente de usinagem, uma instalação de produção pode ser organizada para acomodar uma seção de furadeiras, uma seção de retíficas e uma seção de fresas (Figura 2.2(b)).

Pela caracterização de arranjos independentes de tipo de produto, é permitida uma grande flexibilidade para diferentes especificações de fabricação e variações na demanda (Lorini, 1993).

Sob o aspecto operacional, os problemas primários associados com o *layout* funcional são os tempos de fabricação longos, uma grande área de armazenamento (*buffer*), baixos índices de produtividade, longos tempos de percurso devidos às distâncias de fluxo de material caracterizando-se numa circulação de produto entre áreas distintas de equipamentos, em que deve ocorrer o processamento. Tipicamente, o *layout* funcional é usado para produzir um número limitado de peças (*job shop manufacturing* - fabricação que produz um lote de itens não repetitivos, produtos sob encomenda).

2.3 *Layout* celular ou de grupo

Layout celular ou de grupo é a configuração que conjuga características de uma família de peças aos equipamentos necessários exclusivamente a sua fabricação. No chão-de-fábrica, as máquinas são agrupadas e organizadas em uma ordem lógica para produzir uma família de peças e não mais de um determinado produto (Figura 2.3). É o tipo de *layout* com maior tendência de utilização na atualidade pela engenharia industrial, em que se busca, no suprimento dos processos necessários, a produção de determinadas peças, uma especialização por componentes, quando se determina o arranjo das máquinas. Caracteriza-se, assim, pela diversificação de produtos, ou seja, uma família de peças com diferentes tamanhos de lotes. Em relação a outros tipos de arranjo, diminuem consideravelmente os tempos improdutivos, como os de movimentação e manuseio das peças, além daqueles de preparação das máquinas. Reúnem-se, assim, as vantagens dos arranjos em linha e funcionais, dispondo-se de todas as máquinas próximas umas das outras, na seqüência mais adequada ao processo, ao mesmo tempo em que se tem a flexibilidade de execução, não mais para um só produto, mas para uma família de produtos (Lorini, 1993).

Em um senso amplo, a meta de uma célula de manufatura é melhorar a eficiência de uma operação industrial minimizando o trabalho como também maximizando qualidade do produto, a utilização dos equipamentos e a flexibilidade da produção (Chan *et. al.*, 1999).

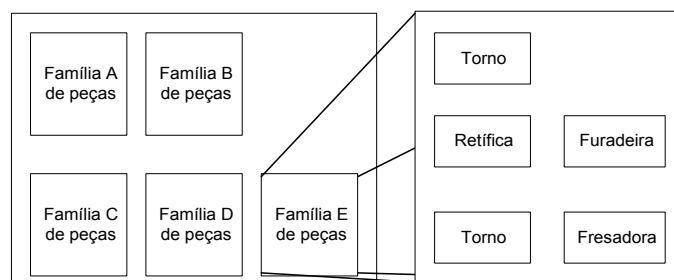


Figura 2.3 – *Layout* celular dedicado à manufatura de uma família de peças.

Fonte: Chan *et. al.*, 1999.

3 MANUFATURA CELULAR

A manufatura celular, como concebida atualmente, é uma técnica derivada de uma prática japonesa de organizar grupos de máquinas e de pessoas para produzir uma família de peças (Reynolds, 1998). Historicamente, os sistemas de manufatura celular clássicos resultam da aplicação da filosofia identificada como Tecnologia de Grupo, um conceito desenvolvido por um russo chamado Mitrofanow na década de 1940, que procurava buscar o agrupamento de peças que poderiam ser fabricadas em uma mesma máquina, racionalizando sua produção. Esse conceito passou a ser adotado e aperfeiçoado pelos fabricantes americanos somente nos anos 70 (Drolet *et. al.*, 1999).

Groover apud Silveira (1994) concebe Tecnologia de Grupo como um enfoque genérico, utilizável em qualquer área do pensamento humano como: “Tecnologia de Grupo é um enfoque que procura identificar os atributos de uma população com o objetivo de coletá-los em grupos, às vezes famílias”.

No entanto, como este trabalho está voltado para a área de manufatura, dois conceitos podem ser destacados:

“Tecnologia de Grupo é um tipo de organização fabril na qual os materiais processados são totalmente divididos entre unidades organizacionais (grupos), onde cada item completa todos os estágios de fabricação e montagem exigidos, sendo estes grupos aparelhados com todos os equipamentos e acessórios necessários a esta tarefa”. (Burbidge apud Silveira 1994).

“Tecnologia de Grupo é uma filosofia de manufatura que advoga a simplificação e padronização de entidades similares (peças, montagens, planejamento de processo, ferramentas, instruções, etc.) com o objetivo de redução da complexidade e obtenção de efeitos de economias de escala na produção em lotes” (Tatikonda e Wemmerlov apud Silveira 1994).

Em um ambiente de manufatura celular, do seu projeto a organização e gestão, a aplicação dos princípios da Tecnologia de Grupo constitui-se como uma das principais ferramentas de trabalho. A Tecnologia de Grupo é uma tentativa de aproximação da engenharia de produção e da engenharia de manufatura que ajuda a controlar a variedade através da reunião das similaridades básicas dos produtos e atividades. Dentro do contexto da manufatura, a Tecnologia de Grupo pode ser definida como uma filosofia de produção identificando as peças que apresentam características semelhantes de geometria (forma, dimensões, tolerâncias etc.) ou de processos de fabricação e agrupando-as em famílias para obter vantagens de suas similaridades na manufatura e projeto (Wang, 1998, Selim *et. al.*, 1998). Uma importante aplicação da filosofia da Tecnologia de Grupo na manufatura, dentre outras, é justamente na organização da manufatura celular. A manufatura celular, no seu conceito essencial, trata das operações de

manufatura dedicadas à produção de famílias de peças, associadas aos respectivos *layouts* de equipamentos (Moon e Gen, 1999).

Greene apud Drolet *et al.* (1996) define manufatura celular como divisão física das máquinas flexíveis de fabricação em células de produção, sendo cada célula projetada para produzir uma família de peças, onde um número de peças é definido como família por requerer máquinas semelhantes. Dentro da célula, as peças de uma família são processadas desde a matéria-prima até a peça acabada.

As vantagens derivadas da manufatura celular incluem redução dos tempos mortos associados a processos de fabricação, em razão de melhorias e racionalização no fluxo dos processos, redução do número de preparações (*setups*) e dos custos de manipulação de materiais. Outras vantagens decorrem da redução do trabalho indireto e de inspeção, da diminuição dos inventários em processo, da melhor utilização do espaço e simplificação do planejamento e programação da produção. Do ponto de vista operacional implica em um aumento nas habilidades do operador, além do moral e das relações humanas entre os mesmos devido a maior responsabilidade e diversidade de tarefas (Moon e Gen, 1999, Baker e Maropoulos, 1997, Chan *et al.*, 1999, Bedworth *et al.*, 1991, Ramos *et al.*, 2004).

Mukhopadhyay *et al.* (1994) cita que um sistema de manufatura celular é projetado para combinar a eficiência de uma linha de alta-produção e a flexibilidade de um *job shop*.

De acordo com Greene apud Drolet *et al.* (1996), sistemas de manufatura celular têm muitas vantagens sobre um *job shop*. Para nomear algumas, podem-se mencionar uma redução significativa do controle necessário nas atividades de manipulação de materiais, nos tempos de *setup*, no ferramental, no inventário em processo, do tempo de fluxo, bem como um aumento da habilidade dos operadores, e melhoria nas relações humanas. Algumas desvantagens também devem ser consideradas. O fato de atribuir a dedicação das estações de trabalho para operações específicas reduz a utilização das máquinas e também a flexibilidade global da fábrica (Drolet *et al.* 1996, Baker e Maropoulos, 1997).

Algumas das vantagens e desvantagens obtidas a partir da implantação de células de manufatura condensadas a partir de Harvey, Sundaram, Gaither, Frazier e Wei apud Silveira (1994), podem ser citadas como:

Vantagens:

1. A simplificação das rotas de produção e dos relacionamentos entre componentes e máquinas aumenta a acurácia dos sistemas de planejamento e controle da produção, simplificando sua execução.
2. A redução das distâncias entre as operações minimiza as atividades de transporte e manuseio

de materiais.

3. À medida que aumenta o grau de similaridade entre as peças processadas na mesma célula, o tempo de *setup* entre os lotes processados tende a reduzir-se automaticamente, justificando economicamente a redução destes lotes e aumentando a capacidade produtiva.
4. A similaridade entre os componentes normalmente possibilita a padronização de máquinas e ferramentas, reduzindo a diversidade e, conseqüentemente, a ociosidade dos meios de produção.
5. Como conseqüência da padronização, o processo produtivo torna-se mais simples, o número de relações é reduzido. Isto facilita enormemente a automação posterior das células.
6. O trajeto percorrido pelos componentes é reduzido consideravelmente, o que direta e indiretamente reduz tempos mortos entre as operações, além do tempo total de fabricação e os estoques intermediários.
7. A variedade das atividades desempenhadas por uma mesma máquina é reduzida. Isto possibilita treinamentos mais rápidos e ganhos de qualidade advindos de um conhecimento maior por parte dos operadores, além de facilitar a criação de cargos multifuncionais.
8. No que se refere à mão-de-obra, a organização do trabalho dentro da célula tende a melhorar o sistema de relacionamento em equipe, o grau de envolvimento e a participação dos trabalhadores.

Desvantagens:

1. A introdução de novos produtos ou componentes muito diferentes dos convencionais torna-se mais difícil, já que as células foram desenhadas a partir dos componentes existentes na fábrica. Em alguns casos, é necessária uma redefinição no *layout*. Para minimizar este problema, as atividades de projeto podem tomar como ponto de partida características de componentes já existentes, na definição de novos produtos.
2. A tarefa de modificar o *layout*, movimentando máquinas, envolve custos com instalações elétricas, hidráulicas ou pneumáticas, reformas, etc.
3. Em muitos casos onde se deseja minimizar o nível de movimentações de elementos entre células, exige-se investimento em máquinas redundantes, aumentando, no primeiro momento, a ociosidade geral.

Segundo Goldhar e Jelinek apud Silveira (1994), Seifoddini e Djassemi (1995), Chan *et al.* (1999), um sistema de produção celular oferece maior habilidade para adaptar a produção de pequenos lotes, viabilizando-a economicamente e reduzindo a necessidade por formação de lotes de grande tamanho para alcançar a economia de escala e ajudando os fabricantes a reagir ao ambiente variável de uma maneira efetiva e eficiente.

Os benefícios da manufatura celular tornam-se tanto maiores quanto mais independentes sejam as células (Moon e Gen, 1999, Baker e Maropoulos, 1997) desde que o desempenho de uma determinada célula não dependa do desempenho de outras. Porém, células totalmente independentes podem ser de difícil concepção, particularmente em ambientes de alta variedade de produtos.

Geralmente o projeto de células independentes implica na duplicação de máquinas gargalo, onde, muitas vezes, essa duplicação requer adicional investimento de capital. Uma das principais fontes de custos aumentados na indústria pode ser atribuída freqüentemente ao equipamento e às horas-extras do empregado. Estas horas-extras podem ser reduzidas pela otimização na programação de máquinas e operações (Moussa e Kamel, 1998, Goldratt, 1986) e a simplificação no planejamento e programação das funções da produção pode produzir suficiente economia para justificar um investimento adicional no projeto das células (Moon e Gen, 1999). Considerando que uma peça pode ter mais de um plano de processo e cada operação associada a uma peça possa ser processada em máquinas alternativas, é possível projetar células independentes mesmo sem investimento adicional. Também, considerando os planos de processo alternativos pode-se aumentar muito a eficiência do agrupamento (Moon e Gen, 1999).

As células podem ser projetadas para operar completamente com uma classe de equipamento que freqüentemente incorpora várias formas flexíveis de automação. Desta maneira essas células são tipicamente não manuais e a função do ser humano na operação das células é restrita ao carregamento e descarregamento de peças, troca de ferramentas, manutenção e inspeção (Selim *et. al.*, 1998).

As metodologias aplicadas na formação de células assumem que um conjunto de peças é identificado como adequado para a manufatura em um grupo específico de máquinas ou tipos de máquinas, selecionadas através dos respectivos planos de processo das peças. Isso implica em se identificar essa relação básica entre uma peça e um elenco de máquinas (uma rota de fabricação), como base de informação. As peças podem então ser associadas em famílias de tal maneira que essas agreguem todas as peças que possam ser processadas em um mesmo grupo de máquinas (Selim *et. al.*, 1998).

A proximidade das estações de trabalho, a sua limitação em número e as semelhanças dos produtos tornam a programação dos trabalhos mais fácil. Porém, quando o *mix* de peças muda, um sistema celular clássico tende a ser menos eficiente, e tipicamente mais máquinas são exigidas para compensar o balanceamento inadequado. Deve-se considerar ainda que não são comuns células de manufatura completamente independentes; freqüentemente as peças têm de passar por mais de uma célula quando economicamente não seja viável a duplicação de

máquinas simplesmente para aumentar a independência entre células.

A manufatura celular clássica pode não ser apropriada em situações onde seqüências de operações ou rotas são propensas a mudar no tempo. Na realidade, é difícil de encontrar uma medida para determinar quando uma configuração *job shop* seria melhor ou pior, em termos de eficiência, que um sistema celular.

Outro fator que deve ser considerado no projeto de sistemas de manufatura é aquele relacionado com o tipo de demanda a ser atendida. As razões para a fabricação de produtos, como bens de consumo, podem ser destinados a atender necessidades de montagem, reposição de estoques, ou mesmo criados para finalidades especiais. Pode-se assumir que em alguns casos, o pedido requer produção em quantidades muito pequenas. Considerando-se ainda situação de uma demanda-dirigida, alternativas para o sistema de manufatura celular clássico, que podem ser até mesmo mais responsivas para pequenos lotes de produção, como para produção tipo único, deve-se considerar os conceitos de sistema de manufatura *celular virtual* e sistema de manufatura *celular dinâmico*.

O conceito de célula virtual foi introduzido primeiro por McLean et al. apud Drolet (1996) no início dos anos 80. De acordo com a definição de McLean, uma célula virtual não é identificável como um agrupamento físico fixo de estações de trabalho, mas como arquivos de dados e processos em um controlador. Em outras palavras, uma célula virtual é um agrupamento lógico de recursos dentro de um controlador. Quando uma ordem de serviço necessitar de uma configuração de estações de trabalho, um controlador de células virtuais assume o controle destas estações e torna possível a comunicação entre estações de trabalho que fazem parte de um agrupamento de estações de trabalho disponíveis ou de uma célula virtual.

Drolet *et al.* (1996), propõe o caminho que permite manter sob controle, e especialmente explorar, esta organização celular virtual para atingir níveis altos de desempenho em termos de produtividade, do tempo de fluxo do pedido de serviço, do inventário em processo, e da flexibilidade em face de um ambiente rápido, volátil e variável. Para exemplificar considere-se a ilustração da Figura 3.1 que inclui 18 estações de trabalho sobre a fábrica, cada uma delas classificada em um dos quatro tipos de estações. Neste exemplo, o sistema tem três células virtuais ativas e nenhuma estação compartilhada. Desta forma, as células virtuais contêm somente as estações necessárias e a capacidade extra requerida em um sistema celular clássico não é mais necessária em um sistema de manufatura celular virtual.

Devem ser enfatizados três aspectos desta configuração física: a variedade de estações de trabalho, a freqüência de cada tipo de estação de trabalho, e finalmente sua distribuição ao longo da fábrica.

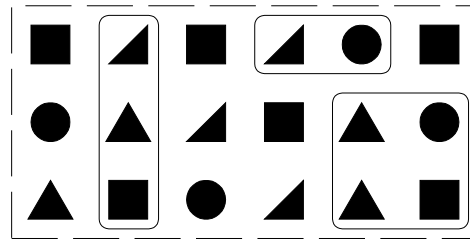


Figura 3.1 – Exemplo de *Layout* de sistema de manufatura celular virtual.
Fonte: Drolet *et. al.*, 1996.

A variedade de estações de trabalho é uma função das exigências tecnológicas das peças para as quais o sistema foi projetado e a flexibilidade das próprias estações de trabalho. A frequência de certo tipo de estação de trabalho depende da demanda e fornece a capacidade e a flexibilidade por rotas alternativas. A distribuição física das estações de trabalho na fábrica poderia ser um problema significativo para a manipulação de materiais. Pode parecer que aquelas estações de trabalho estão distribuídas fortuitamente sobre a fábrica, porém, este não é o caso. É mais favorável a dispersão de estações de trabalho idênticas ao longo da fábrica em vez de agrupar todas as máquinas semelhantes (*layout* funcional). Deste modo, quando a demanda cresce, a adição de estações de trabalho aumenta a flexibilidade de rotas. Entretanto, a distância global do fluxo necessariamente não aumenta. Como consequência as estações de trabalho são distribuídas de tal modo que aquelas similares não estão frequentemente perto umas das outras, isto melhorará a eficiência da manipulação de materiais (Drolet *et al.*, 1996).

Neste sistema é importante que as estações de trabalho estejam próximas, assim, estoques em processo não são aceitáveis. Muitas companhias têm realizado estudos e modificado suas instalações industriais para adotar a filosofia *Just-in-time*. Esta filosofia requer a participação estreita da maioria dos fornecedores para transportar matéria-prima e/ou peças na quantidade exata no momento exato e com qualidade assegurada. Entregas de fornecedores tendem a ser mais frequentes e em menores quantidades nos períodos considerados (Drolet *et al.*, 1996).

O conceito de *layout* de um sistema de manufatura celular virtual (*VCMS*) pode ser aplicado em harmonia com esta filosofia. Na realidade, pode-se conceber que sejam localizados galpões de recebimento/expedição ao redor do estabelecimento. Remessas de fornecedores podem ser programadas em determinado espaço e momento para abastecer uma célula virtual específica. O horário de entrega poderia incorporar um período de tempo que entregasse o material algumas horas mais cedo da criação de certa célula virtual. Também poderia ser definido o período de tempo como uma função da habilidade do fornecedor para entregar na hora certa e/ou a distância entre o centro de distribuição e o *VCMS*. Fornecedores se tornariam uma extensão flexível do *VCMS* e as suas operações estariam em harmonia com o *VCMS* (Drolet *et al.*, 1996).

Um sistema de manufatura celular virtual cresce naturalmente como aumentos de demanda. Ao contrário dos *layouts* de processos e de produto, a eficiência de um *VCMS* pode, às vezes, aumentar com o tamanho do estabelecimento. Quando o estabelecimento cresce, isto aumenta o número de rotas alternativas, como resultado, a densidade de cada célula virtual (por exemplo, proximidade de máquinas e de estações de trabalho dentro de uma célula virtual) pode tender a aumentar. Por causa da descontinuidade física e do aspecto dinâmico da configuração da célula virtual, é aceito que a imprevisibilidade da demanda e mudanças no *mix* de produtos não venha a afetar significativamente a eficiência nem a operação de um *VCMS* (Drolet *et al.*, 1996).

O conceito de célula virtual requer um sistema de manipulação de materiais verdadeiramente flexível para movimentação de peças dentro de cada célula virtual. Este poderia ser um operador dirigindo o sistema de movimentação de materiais ou um sistema de manipulação de materiais automatizado.

Para um grande número de fabricantes, os sistemas de manufatura celulares clássicos não são mais a principal vantagem tecnológica. Para ficar à frente da competição, eles têm que buscar algo novo, um sistema industrial projetado para máxima eficiência em todo o tempo e capaz de apoiar uma única produção. A resposta poderia vir, em alguns casos no conceito de manufatura celular dinâmica.

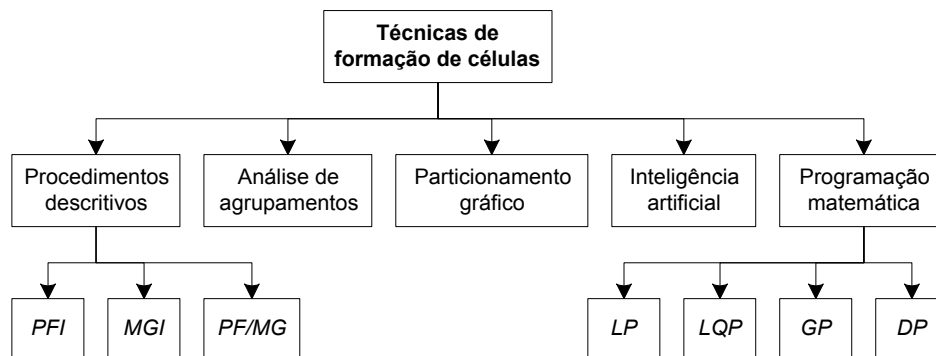
Esta aproximação parece ser muito eficiente em um ambiente altamente turbulento como é freqüentemente o caso de fabricantes sub-contratados. Tipicamente, estes fabricantes são especializados em alguns processos e vendem a capacidade e conhecimento para produzir uma variedade de peças para um número de clientes. Freqüentemente, estes não possuem as peças, ferramental especializado, fixações e moldes, e têm um controle limitado no projeto de peças e suas características. Apesar disto, têm que oferecer um ambiente industrial altamente flexível, mas ainda competitivo.

A configuração física do sistema de manufatura celular dinâmico é propensa a mudar no tempo, visando minimizar o custo marginal total de manipulação de materiais (peças e máquinas) considerado em certo horizonte de tempo. Acredita-se que um Sistema de Manufatura Celular Dinâmico seja mais eficiente em ambientes turbulentos do que numa configuração *job shop* clássica ou em um sistema de manufatura celular clássico (Drolet *et al.*, 1996).

4 MÉTODOS DE SOLUÇÃO NA FORMAÇÃO DE CÉLULAS

Há várias técnicas desenvolvidas para auxiliar a formação de famílias de peças, associadas às respectivas tarefas nas máquinas. Essas, em geral, são identificadas como: (a) inspeção visual; (b) técnicas de classificação e codificação; (c) técnicas de análise do fluxo de produção (*PFA*); (d) conjunto de algoritmos teóricos; (e) programação matemática; (f) técnicas baseadas em custo; (g) algoritmos genéticos; e (h) inteligência artificial (Chan *et. al.*, 1999, Silveira, 1994).

Diversas classificações para esses métodos de formação de células têm sido propostas por muitos pesquisadores. Uma classificação baseada em um tipo de metodologia geral de solução foi desenvolvida por Hassan *et al.* apud Selim (1998) e é ilustrada na Figura 4.1.



PFI – identificação de famílias de peças

MGI – identificação de grupos de máquinas

PF/MG – agrupamentos de famílias de peças e de grupos de máquinas simultaneamente

LP – programação linear

LQP – programação linear e quadrática de números inteiros

GP – programação objetiva

DP – programação dinâmica

Figura 4.1 – Classificação dos métodos de formação de células.

Fonte: Selim *et. al.*, 1998.

4.1 Procedimentos descritivos

De acordo com Wemmerlöv e Hyer apud Selim (1998), em geral, procedimentos descritivos podem ser classificados dentro de três classes principais. A primeira classe, que é referida à identificação de famílias de peças (*PFI*), cujo processo de formação de células tem início pela identificação das peças em prováveis famílias e posterior alocação das máquinas às famílias criadas. A segunda classe, que é referida à identificação prioritária de grupos de

máquinas (*MGI*), seguindo o caminho inverso das etapas da primeira classe. A terceira classe de procedimentos descritivos refere-se a procedimentos para gerar simultaneamente os agrupamentos de famílias de peças e dos grupos de máquinas (*PF/MG*).

Os métodos de identificação de famílias de peças podem ser ainda classificados como aqueles baseados em sistemas informais (investigação visual ou outro critério qualquer) e aqueles baseados em codificação formal e sistemas de classificação. A maior parte da literatura discute aproximações informais para identificar as famílias de peças descritas em experiências reais de empresas que têm implementado células (Selim *et. al*, 1998, Silveira, 1994). O papel da coleção de regras da Tecnologia de Grupo no contexto da manufatura celular é principalmente uma ajuda na identificação das famílias de peças que poderão ser produzidas em células dedicadas.

O procedimento de identificação de grupos de máquinas considera o problema da formação de células como um processo de dois estágios onde no primeiro estágio de análise, máquinas são agrupadas com base em informações disponíveis nas rotas das peças para num segundo estágio, as peças serem alocadas nos grupos de máquinas.

Dentre os trabalhos de maior importância, principalmente pelo seu caráter pioneiro, deve ser considerado o estudo apresentado por Burbidge (Selim *et. al*, 1998, Lorini, 1993, Chan *et. al.*, 1999, Singh e Rajamani, 1996), em que estabelece a conceituação da representação da matriz peça-máquina, avaliando o fluxo e a seqüência de produção, base de desenvolvimento da grande maioria dos sistemas de agrupamento existentes. O trabalho de Burbidge é referido como *Production Flow Analysis (PFA)*. Como exemplo, a Figura 4.2 mostra uma matriz incidência peça-máquina muito simples onde a *peça_1* precisa ser processada por duas máquinas (*máquina_1* e *máquina_2*) definindo o seu fluxo entre as máquinas. Esta matriz peça-máquina forma o núcleo do *PFA* e várias técnicas foram desenvolvidas para facilitar a permutação das colunas e linhas para formar uma matriz diagonalizada onde as células de máquinas e famílias de peças podem ser identificadas (Figura 4.3).

	peça 1	peça 2	peça 3	peça 4	peça 5	peça 6	peça 7	peça 8
máquina 1	1							1
máquina 2	1				1	1		
máquina 3			1	1			1	
máquina 4		1	1				1	
máquina 5					1	1		1

Figura 4.2 – Matriz incidência peça-máquina.

	peça 3	peça 7	peça 2	peça 4	peça 1	peça 8	peça 6	peça 5
máquina 4	1	1	1					
máquina 3	1	1		1				
máquina 1					1	1		
máquina 5						1	1	1
máquina 2					1		1	1

Figura 4.3 – Matriz incidência peça-máquina reordenada.

Fonte: Chan *et. al.*, 1999.

Inspeção visual versus métodos de classificação e de codificando

Em geral, inspeção visual é o método mais imediato e de menor custo para definir agrupamentos, embora aplicável somente em situações especiais. Um especialista é exigido para executar inspeções, determinar as famílias de peças e associar grupos de máquinas. Nas técnicas de classificação e de codificando, códigos numéricos e alfanuméricos são determinados para representar o atributo do projeto e/ou da manufatura de uma peça. A estrutura de um sistema de codificação pode seguir uma concepção do tipo: cadeia, hierárquico ou híbrido. Embora o esquema hierárquico normalmente contenha mais informações que os outros e também agrupa as peças de uma maneira minuciosa, tendendo a ser mais complicado. Além disso, também há ocasiões onde o atributo de projeto pode entrar em conflito com o atributo da manufatura, e isto causa dificuldades de fundir ambos os atributos em um sistema hierárquico. A aproximação híbrida pode superar tal problema, mas a classificação atual e sistemas de codificação tendem a ser mais atribuídos ao projeto orientado e neste falta a característica para ajudar a identificação de famílias de peças. Em adição, o método de classificação e codificação parece consumir mais tempo e ser um método complicado devido à exigência de uma pesquisa completa na população de peças. Diversos sistemas de codificação têm sido desenvolvidos seja por Instituições de Ensino e Pesquisa ou indústrias no sentido de permitir aplicações em ambiente de manufatura, dentre os quais podem ser destacados segundo Melo (2003), Chan (1999), Lorini (1993), Silveira (1994) e Bedworth *et al.* (1991): *DCLASS (Design and Classification Information System*, desenvolvido na *Brigham Young University*); *MICLASS (Metal Institute Classification System*, desenvolvido pela *TNO – The Netherlands Organization for Applied Scientific Research*, Holanda); *OPITZ* (desenvolvido por H. Opitz, na Universidade de Aachen, Alemanha); *KK-3* (desenvolvido pela *JSPMI – Sociedade Japonesa para a Produção de Máquinas Industriais*); *CODE (Manufacturing Data Systems, Inc., EUA)*; *SCC/GRUCON* (desenvolvido pelo Grupo de Pesquisas e Treinamento em Comando Numérico e Automatização Industrial na Universidade Federal de Santa Catarina); *Tecla* e o *MultiClass (OIR – Organização para Pesquisa Industrial, EUA)*.

4.2 Procedimentos baseados em análise de agrupamento

O principal objetivo desta ferramenta estatística é agrupar de modo idêntico objetos ou entidades ou seus atributos dentro de grupos (*clusters*) de modo que os elementos individuais dentro de um grupo tenham um alto grau de “associação natural” entre eles e que não haja “associação natural” entre os grupos. Procedimentos de agrupamento podem ser classificados como: técnicas de agrupamento baseadas em arranjos (*array-based clustering techniques*), técnicas de agrupamento hierárquicas, e técnicas de agrupamento não hierárquicas.

No agrupamento **baseado em arranjos**, o elenco de processos necessários aos componentes nas máquinas pode ser representado por uma matriz incidência, assim é chamada a matriz peça-máquina, cuja representação dos elementos (a_{ij}) pode ser formada por zeros e uns. Uma entrada 1 em uma linha i e coluna j ($a_{ij}=1$) da matriz indica que a peça j tem uma operação na máquina i , enquanto que um 0 indica que não tem. As técnicas baseadas em arranjos tentam alocar máquinas a grupos e associar peças às famílias pelo rearranjo apropriado na disposição das linhas e das colunas para encontrar uma forma diagonalizada em blocos de $a_{ij}=1$ na matriz peça-máquina. Outra simbologia também pode ser usada além de zeros e uns. Neste trabalho os zeros não são apresentados nas figuras das matrizes incidência e reordenadas para maior clareza.

Na literatura encontram-se inúmeros algoritmos com esse propósito, de agrupamento baseados em arranjos, dentre eles o *Bond Energy Analysis (BEA)* de McCormick *et al.* (Selim *et al.*, 1998, Lorini, 1993, Chan *et al.*, 1999, Silveira, 1994, Singh e Rajamani, 1996), *Rank Order Clustering (ROC)* de King e *Rank Order Clustering 2 (ROC2)* de King e Nakornchai, *Modified Rank Order Clustering (MODROC)* de Chandrasekharan e Rajagopalan (Selim *et al.*, 1998, Lorini, 1993, Singh e Rajamani, 1996), *Direct Clustering Analysis (DCA)* de Chan e Milner (Selim *et al.*, 1998, Silveira, 1994, Singh e Rajamani, 1996), *Cluster Identification Algorithm (CIA)* e *Modified CIA* de Kusiak e Chow (Singh e Rajamani, 1996), *Close Neighbour Algorithm (CNA)* de Boe e Cheng (Dalmas, 2004, Boe e Cheng, 1991, Silveira, 1994), *Occupancy Value method* de Khator e Irani, *Cluster Identification method* de Kusiak e Chow, e *Hamiltonian Path Heuristic* de Askin *et al.* (Selim *et al.*, 1998).

No **agrupamento hierárquico**, os dados na matriz peça-máquina não são separados dentro de grupos ou células em uma etapa. Primeiramente eles são separados dentro de pequenas células, sendo cada uma delas posteriormente dividida em grupos menores, e cada um destes particionados ainda mais, e assim sucessivamente até a formação de grupos que não possam ser subdivididos. As técnicas hierárquicas essencialmente podem ser subdivididas em métodos aglomerativos que se originam de uma série de sucessivas fusões de M máquinas ou P peças dentro de grupos, e métodos divisores que particionam o conjunto de M máquinas (P peças)

sucessivamente dentro de grupos muito pequenos. Todas as técnicas hierárquicas aglomerativas, em síntese, reduzem os dados para um único agrupamento contendo todas as máquinas (peças), e técnicas analíticas irão finalmente separar o conjunto de entradas de máquinas (peças) em células M (P) que contém uma única máquina (peça). Classificações hierárquicas podem ser representadas por três estruturas ou dendogramas invertidos, que são diagramas bidimensionais que ilustram as fusões ou as divisões obtidas em cada estágio sucessivo da análise (Selim *et. al.*, 1998, Silveira, 1994).

Os métodos de **agrupamento não hierárquicos** são procedimentos iterativos e também iniciam com uma divisão inicial do conjunto de dados ou a escolha de alguns pontos semente. Arbitrariamente na escolha dos pontos semente (ou divisões iniciais de dados) pode moldar resultados insatisfatórios. Alguns procedimentos não hierárquicos têm sido desenvolvidos por Chandrasekharan e Rajagopalan, Lemoine e Mutel, e Srinivasan e Narendran (Selim *et. al.*, 1998).

4.2.1 Rank Order Clustering (ROC)

Este algoritmo foi desenvolvido e apresentado por King e posteriormente aperfeiçoado por King e Narkornchai com a versão ROC2 (Lorini, 1993, Singh e Rajamani, 1996). Na sua concepção original, o método estabelece uma ponderação sobre uma matriz binária, onde 1 representa a incidência de uma peça na máquina e 0, o caso oposto. Um valor decimal correspondente a cada palavra binária é atribuído para linhas e colunas. Com estes valores, o algoritmo ROC rearranja as linhas e colunas da matriz de modo iterativo, em um número finito de alterações, até que linhas e colunas fiquem dispostas em ordem decrescente.

Para melhor ilustrar a metodologia, considere-se a matriz da Figura 4.4. Primeiramente é estabelecido o respectivo valor de cada linha e sua posição correspondente a uma ordem de grandeza decrescente dos valores. Assim, tomando-se a linha 1, o valor da palavra binária (001010) seria o decimal 10, ou seja;

$$(0 \times 2^5) + (0 \times 2^4) + (1 \times 2^3) + (0 \times 2^2) + (1 \times 2^1) + (0 \times 2^0) = 10$$

<u>peso -</u>	2^5	2^4	2^3	2^2	2^1	2^0	<u>valor</u>	<u>ordem</u>
1	1	2	3	4	5	6	10	5
2		1	1				24	4
3	1			1			36	2
4		1	1		1		26	3
5	1			1		1	37	1

Figura 4.4 – Matriz com ponderação nas linhas.

Fonte: Lorini, 1993.

Alternando-se a posição das linhas, de modo que os valores fiquem posicionados em ordem decrescente, passa-se a calcular o valor para as colunas.

	1	2	3	4	5	6	peso
5	1			1		1	2^4
3	1			1			2^3
4		1	1		1		2^2
2		1	1				2^1
1			1		1		2^0
	24	6	7	24	5	16	valor
	1	5	4	2	6	3	ordem

Figura 4.5 – Matriz com ponderação nas colunas.
Fonte: Lorini, 1993.

Alterando-se a posição das colunas para ordem decrescente, e repetindo-se o processo, se necessário, até que todas as linhas e colunas estejam ordenadas, resulta uma matriz classificada, ou seja, com uma definição de agrupamento.

	1	4	6	3	2	5	valor	ordem
5	1	1	1				56	1
3	1	1					48	2
4				1	1	1	7	3
2				1	1		6	4
1				1		1	5	5
valor	24	24	16	7	6	5		
ordem	1	2	3	4	5	6		

Figura 4.6 – Matriz com linhas e colunas ordenadas.
Fonte: Lorini, 1993.

Obtém-se, assim, por meio de um ordenamento decrescente dos valores decimais calculados, tanto para as linhas, quanto para as colunas, o agrupamento das peças em famílias. Neste caso foram obtidas duas famílias distintas, quais sejam:

- família 1 (peças 1 – 4 – 6) para a célula 1 (máquinas 5 – 3)
- família 2 (peças 2 – 2 – 5) para a célula 2 (maquinas 1 – 2 – 4)

4.2.2 Close Neighbour Algorithm (CNA)

O CNA, desenvolvido por Boe e Cheng (1991), é um algoritmo dos métodos de arranjos de matrizes, de implementação simples cuja lógica de convergência conduz ao resultado em apenas uma interação. O desenvolvimento do CNA, de acordo com os autores, é apresentado na seqüência:

Uma matriz inicial **A** de máquinas e peças é gerada com máquinas (ou processos) listadas nas linhas e produtos (ou componentes) nas colunas da matriz. Sempre que um produto

necessitar de uma máquina em seu processamento, assinala-se com 1 no cruzamento correspondente; caso contrário, o cruzamento recebe um zero. Somente fluxos primários são indicados na matriz. A Figura 4.7 traz um exemplo de matriz de incidências para o caso de 8 máquinas (linhas) e 20 produtos (colunas).

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1		1	1					1	1		1		1	1		1	1		1	
2			1	1		1	1							1				1		1
3		1						1	1		1		1	1		1	1		1	
4			1	1		1	1			1								1		1
5	1				1	1				1		1			1		1			
6	1				1				1	1		1			1					1
7			1	1		1	1				1	1						1		1
8			1	1		1	1											1		1

Figura 4.7 – Matriz incidência peça-máquina.
Fonte: Boe e Cheng, 1991.

Uma matriz **B** de máquinas versus máquinas é gerada, indicando o número de componentes comuns a cada par de máquinas. O elemento b_{ij} na matriz indica quantos produtos necessitam das máquinas i e j para o seu processamento. A última coluna da matriz designada por S_i , traz o somatório dos números das linhas (idêntico ao somatório dos números nas colunas). A Figura 4.8 traz a matriz **B** obtida da matriz **A** da Figura 4.7.

	1	2	3	4	5	6	7	8	S_i
1	-	2	9	1	1	1	2	1	17
2	2	-	1	6	1	1	6	6	23
3	9	1	-	0	1	1	1	0	13
4	1	6	0	-	2	2	6	6	23
5	1	1	1	2	-	5	2	1	13
6	1	1	1	2	5	-	2	1	13
7	2	6	1	6	2	2	-	6	25
8	1	6	0	6	1	1	6	-	21

Figura 4.8 – Matriz B máquina-máquina.
Fonte: Boe e Cheng, 1991.

O *CNA* é implementado em dois estágios. O estágio um tem por objetivo reordenar as linhas da matriz **A**; no estágio dois, reorganizam-se as colunas da matriz. Ao final do estágio dois, espera-se identificar aglomerações de incidências ao longo da diagonal principal da matriz reorganizada.

Para definir a nova ordem das linhas na matriz **A** (estágio um), analisa-se a matriz **B** relativamente a alguns critérios. Para selecionar a primeira linha a ser reordenada na matriz **A**, seleciona-se a linha com maior valor de S_i na matriz **B**; no caso de empate, escolhe-se a linha de

menor índice i . As linhas seguintes a serem reordenadas são determinadas pela utilização sequencial de três regras: (i) identificação do maior valor de b_{ij} na última linha trabalhada, onde índice j (da coluna) indica a próxima linha a ser trabalhada; no caso de empate, utiliza-se o próximo critério; (ii) seleciona-se a linha com maior S_i dentre as empatadas; no caso de novo empate, aplica-se o próximo critério; e (iii) seleciona-se a linha com menor índice i dentre as empatadas. Reorganizam-se as linhas da matriz inicial A , seguindo a ordem definida acima.

A Figura 4.9 apresenta os resultados do estágio um aplicado ao exemplo numérico. Na última coluna, identifica-se o critério adotado para reordenação de cada linha. A matriz A com linhas reordenadas é apresentada na Figura 4.10. Essa matriz de resultados intermediários é designada por A' .

Índice linhas	Máquinas possíveis	Máquina selecionada	Razão
$v = 1$	7	7	$S7 = \text{máximo}$
$v = 2$	2, 4, 8	2	$b72 = b74 = b78 = \text{máximos}; S2 = S4 > S8; 2 < 4$
$v = 3$	4, 8	4	$b24 = b28 = \text{máximos}; S4 > S8$
$v = 4$	8	8	$b48 = \text{máximo}$
$v = 5$	1, 5, 6	1	$b81 = b85 = b86 = \text{máximos}; S1 > S5 = S6$
$v = 6$	3	3	$b13 = \text{máximo}$
$v = 7$	5, 6	5	$b35 = b36 = \text{máximos}; S5 = S6; 5 < 6$
$v = 8$	6	6	$b56 = \text{máximo}$

Figura 4.9 – Reordenamento das linhas da matriz incidência.
Fonte: Boe e Cheng, 1991.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
7			1	1		1	1				1	1						1		1
2			1	1		1	1							1				1		1
4			1	1		1	1			1								1		1
8			1	1		1	1											1		1
1		1	1					1	1		1		1	1		1	1			1
3		1						1	1		1		1	1		1	1			1
5	1				1	1				1		1			1		1			
6	1				1				1	1		1			1					1

Figura 4.10 – Reordenamento das linhas da matriz incidência.
Fonte: Dalmas, 2004.

O desenvolvimento do estágio dois, para definir a nova ordem das colunas da matriz A' , é apresentado de duas formas: a primeira forma é apresentada de acordo Fogliatto (2004) e Dalmas (2004) e a segunda forma, de acordo com Boe e Cheng (1991).

De acordo com Fogliatto (2004) e Dalmas (2004), para definir a nova ordem das colunas da matriz A' (estágio dois), divide-se a matriz A' em duas metades, superior e inferior. Quando o número de linhas da matriz for ímpar, deve-se fazer com que a metade superior fique com um

número ímpar de linhas. Denomina-se, no caso, a metade superior da matriz dividida de *SEQ1* e a metade inferior de *SEQ2*.

Analisa-se as colunas da matriz *A'* dividida, contando-se o número de 1's em cada uma das metades (*SEQ1* e *SEQ2*). Sempre que $SEQ1 > SEQ2$, assinala-se a coluna e reordena-se a numeração da primeira à última coluna assinalada. Colunas não assinaladas vão para a segunda fase do estágio dois. Divide-se a metade inferior (*SEQ2*) da matriz *A'* em duas metades, *SEQ1* e *SEQ2*, novamente. Analisa-se as colunas que não foram reordenadas na fase quanto à incidência de células assinaladas com 1. Sempre que $SEQ1 > SEQ2$, assinala-se a coluna e reordena-se a numeração da primeira à última coluna assinalada. Colunas não assinaladas passam para a fase seguinte do estágio dois e o procedimento é repetido até que todas as colunas sejam reordenadas ou a divisão da matriz não seja mais possível. Neste último caso, em que remanescerem colunas não assinaladas, estas devem ser ordenadas livremente, na seqüência da última coluna assinalada.

A Figura 4.11 traz o sumário de resultados do estágio dois, aplicado ao exemplo. O índice *v* na tabela indica a linha inicial da análise. Três divisões da matriz *A'* foram necessárias para convergência do estágio dois.

1ª Fase:

v = 1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
SEQ1	0	0	4	4	0	4	4	0	0	1	1	1	0	1	0	0	0	4	0	4
SEQ2	2	2	1	0	2	1	0	2	3	2	2	2	2	2	2	2	3	0	2	1
SEQ1 > SEQ2			X	X		X	X											X		X
Coluna			1	2		3	4											5		6

2ª Fase:

v = 5	1	2	5	8	9	10	11	12	13	14	15	16	17	19
SEQ1	0	2	0	2	2	0	2	0	2	2	0	2	2	2
SEQ2	2	0	2	0	1	2	0	2	0	0	2	0	1	0
SEQ1 > SEQ2	X			X	X		X		X	X		X	X	X
Coluna		7		8	9		10		11	12		13	14	15

3ª Fase:

v = 7	1	5	10	12	15
SEQ1	1	1	1	1	1
SEQ2	1	1	1	1	1
SEQ1 > SEQ2	X	X	X	X	X
Coluna	16	17	18	19	20

Figura 4.11 – Sumário de resultados do estágio 2 aplicado ao exemplo.
Fonte: Dalmas, 2004.

De acordo com Boe e Cheng (1991), para definir a nova ordem das colunas da matriz *A'* (estágio dois), a linha corrente é definida como 1. Cada iteração no estágio dois verifica a seqüência ininterrupta mais longa de 1's iniciando na linha corrente para cada peça não

ordenada. Uma peça que tem a seqüência ininterrupta mais longa de 1's a partir da linha corrente será ordenada mais à esquerda das colunas não ocupadas da matriz final. Então a linha corrente é incrementada por um para a próxima iteração. O arranjo das peças é armazenado e usado para transformar a matriz intermediária do estágio um.

Analisa-se as colunas da matriz A' , contando-se o número de 1's pertencente a uma seqüência ininterrupta a partir da linha corrente v ($NNOV$). Contam-se o número de 1's pertencente a uma seqüência ininterrupta a partir da linha w ($NNOW$), onde $w > NN OV + v - 1$. Sempre que $NN OV > NN OW$, assinala-se a coluna e reordena-se a numeração da primeira à última coluna assinalada. Colunas não assinaladas vão para a segunda fase do estágio dois. Incrementa-se v em um, novamente. Analisa-se as colunas que não foram reordenadas na fase quanto à seqüência ininterrupta de 1's a partir da linha corrente mais longa. Sempre que $NN OV > NN OW$, assinala-se a coluna e reordena-se a numeração da primeira à última coluna assinalada. Colunas não assinaladas passam para a fase seguinte do estágio dois e o procedimento é repetido até que todas as colunas sejam reordenadas ou que o incremento da linha corrente não seja mais possível. Neste último caso, em que remanescerem colunas não assinaladas, estas devem ser ordenadas livremente, na seqüência da última coluna assinalada.

A Figura 4.12 traz o sumário de resultados do estágio dois, aplicado ao exemplo. O índice v na tabela indica a linha da análise. Aqui são apresentadas somente as fases onde ocorreram ordenamentos de colunas.

v = 1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
<i>NNOV</i>	0	0	5	4	0	4	4	0	0	0	1	1	0	0	0	0	0	4	0	4
<i>NNOW</i>	2	2	0	0	2	1	0	2	2	2	2	2	2	2	2	2	3	0	2	1
<i>NNOV > NNOW</i>			X	X		X	X											X		X
Coluna			1	2		3	4											5		6
v = 5	1	2			5			8	9	10	11	12	13	14	15	16	17		19	
<i>NNOV</i>	0	2			0			2	2	0	2	0	2	2	0	2	3		2	
<i>NNOW</i>	2	0			2			0	1	2	0	2	0	0	2	0	0		0	
<i>NNOV > NNOW</i>		X						X	X		X		X	X		X	X		X	
Coluna		7						8	9		10		11	12		13	14		15	
v = 7	1				5			10			12				15					
<i>NNOV</i>	2				2			2			2				2					
<i>NNOW</i>	0				0			0			0				0					
<i>NNOV > NNOW</i>	X				X			X			X				X					
Coluna	16				17			18			19				20					

Figura 4.12 – Sumário de resultados do estágio dois aplicado ao exemplo.
Fonte: Boe e Cheng, 1991.

O algoritmo definido para obter o agrupamento das peças em família baseia-se em um procedimento iterativo, que busca a formação das células e conseqüentes famílias, como resultado da primeira solução encontrada, de acordo com um determinado procedimento heurístico. A seleção é orientada, de modo a se obter, com a escolha, uma maximização na utilização dos recursos de produção dentro da célula, ou seja, que se aproveitem ao máximo possível os recursos disponíveis, no ambiente celular que está sendo definido.

A metodologia do agrupamento trabalha com a análise do fluxo de produção (*PFA*), onde a incidência das peças nas máquinas é assinalada com o correspondente tempo de processo para cada etapa da seqüência de fluxo. O tratamento heurístico, para agrupar os dados, baseia-se em duas simples observações sobre a matriz de incidências da relação peça-máquina, que correspondem a observações sobre linhas *horizontais e verticais* aleatoriamente identificadas na matriz de incidência como geradoras da análise de densidade.

Determinação de linhas horizontais h_i

Uma horizontal h_i , traçada por qualquer linha da matriz peça-máquina, escolhida arbitrariamente, ou definida por um critério qualquer, irá cruzar os elementos que definem as relações entre uma peça j numa máquina i , onde $t(i, j)$ é o respectivo tempo de processo da peça j na máquina i . Quando esta relação estiver assinalada por um valor não nulo, significa que a peça j deve ser processada na máquina i . Isto significa que a máquina i é uma potencial candidata à inclusão na célula.

Determinação de linhas verticais v_j

Do mesmo modo, verticais v_j traçadas pelas colunas da matriz, nos elementos não nulos, cruzados pelas horizontais, sobre $t(i, j)$, encontram as máquinas utilizadas pela peça j .

Na figura 4.14, a linha h_2 cruza dois elementos diferentes de zero, respectivamente, nas posições $t(2,3)$ e $t(2,5)$ que representam os tempos de processo 7,5 e 1,6 das peças 3 e 5, que são peças a serem fabricadas na máquina 2.

		peças					
		1	2	3	4	5	
máquinas	1	3,5	3,0	3,2	1,5	-	h_2
	2	-	-	7,5	-	1,6	
	3	3,8	-	4,2	-	4,3	
	4	-	2,7	-	-	-	
	5	-	3,4	-	-	2,2	
				v_3		v_5	

Figura 4.14 – Matriz com tempos de processo $t(i, j)$.
Fonte: Lorini, 1993.

4.3 Aproximações por particionamento gráfico

Os métodos de particionamento gráfico tratam as máquinas e/ou as peças como vértices e o processamento das peças com arcos conectando estes nós. Estes modelos ajudam na obtenção de sub-gráficos desconectados dos gráficos máquina-máquina ou dos gráficos peça-máquina para identificar células de manufatura. Rajagopalan e Batra apud Silveira (1994) e Selim (1998) sugerem o uso dos coeficientes de similaridade de Jaccard e a teoria gráfica para formar grupos de máquinas. Vanelli e Kumar desenvolveram um método que busca identificar e minimizar o número de máquinas gargalo, duplicando-as quando necessário. Wu e Salvendy apresentam um algoritmo baseado em redes que busca formar as células minimizando o número final de elementos extracelulares. Faber e Carter desenvolveram um algoritmo gráfico teórico para agrupamento de máquinas e peças dentro de células de manufatura pela transformação da matriz de similaridade dentro de uma rede de grupos (*cluster network*). A rede de grupos é particionada em células pela resolução de um problema de fluxo de custo mínimo (*minimum cost flow problem*). Kumar *et al.* apud Selim (1998), desenvolveu uma programação quadrática de 0-1 com restrições lineares para resolver o problema de agrupamento de peças. Askin e Chiu propõem uma formulação matemática baseada no custo e uma solução heurística para o problema de formação de células. Vohra *et al.* propõem um algoritmo baseado em rede (*network-based algorithm*) para minimizar a soma dos tempos de usinagem de peças fora das células primárias. Wu e Salvendy desenvolveram um modelo de rede (*network model*) para particionar o gráfico máquina-máquina em células considerando as seqüências de operações (Selim *et al.*, 1998).

4.4 Aproximações por inteligência artificial

Elmagharaby e Gu apud Selim (1998) apresentaram uma aproximação usando regras de domínio específico de conhecimento e uma característica protótipo baseada em sistema de modelamento para automatizar processos de identificação de atributos de peças e determinação de peças as mais apropriadas células de manufatura. O sistema de determinação técnica é baseado nas características geométricas das peças, características das células de manufatura formadas, características funcionais das peças e atributos, bem como conhecimento da manufatura de domínio específico. Kusiak apud Selim (1998) desenvolveu uma identificação padrão baseada em agrupamentos de peças que é similar ao agrupamento na Tecnologia de Grupo. A diferença básica entre estas duas aproximações é no grau de automação. Aplicações de redes neurais artificiais para problemas de formação de células são propostas por Rao e GU e Karapathi e Suresh apud Selim (1998).

4.5 Aproximações por programação matemática

Os métodos de programação matemática buscam formar famílias a partir da definição de funções objetivo e da formulação de restrições ao atingimento destas funções. Exemplos de funções objetivo encontradas na literatura são a maximização dos coeficientes de similaridade, minimização de custos de investimentos e movimentações e minimização dos tempos de atravessamento.

Podem ser ainda classificados em quatro grupos principais baseados no tipo de formulação: 1) programação linear (*LP*), 2) programação linear e quadrática de números inteiros (*LQP – linear and quadratic integer programming*), 3) programação dinâmica (*DP*), e 4) programação objetiva (*GP – goal programming*). A programação linear baseada nos métodos de formação celular é proposta por Olivia-Lopez e Purcheck. Eles essencialmente aplicam a técnica de agrupamento combinatório e programação linear ao problema de formação celular. Os modelos de programação linear e quadrática de números inteiros são propostos por Ballakur, Kumar *et al.*, Kusiak, Kusiak e Heragu, Choobineh, Elzinga *et al.*, Kasilingam e Bhole, Vakharia e Chang, Vakharia *et al.*, Vakharia e Kaku, Wei e Gaither, e Boctor. Os modelos de programação dinâmica são desenvolvidos por Ballakur enquanto que os modelos de programação objetiva são propostos por Sankaran e Shafer e Rogers (Selim *et. al.*, 1998, Lorini, 1993, Silveira, 1994).

4.6 Algoritmos genéticos (*GAs*) – estado da arte

Apesar de estarem sendo desenvolvidos desde 1962, a primeira conquista significativa em Algoritmos Genéticos foi feita em 1975 com a publicação de *Adaptation in Natural and Artificial System* por John Holland, seus colegas e seus alunos na Universidade de Michigan (Zuben, 2004).

Os algoritmos genéticos são descritos por John Holland e Kenneth DeJong nos seus trabalhos pioneiros (DeJong 1975 e Holland 1975 apud Gulsen *et al.*, 1995 e apud Singh e Rajamani, 1996).

De acordo com Holland apud Wang e Tai (2005) e apud Singh e Rajamani (1996), o algoritmo genético é um método estocástico de procura global que imita a teoria da evolução biológica natural. Segundo Goldberg apud Wang e Tai (2005), o *GA* está baseado no princípio da sobrevivência-pela-adaptação de Darwin e opera em uma população de soluções potenciais para produzir a melhor ou as melhores aproximações para a ótima solução.

Algoritmos genéticos, segundo Gulsen (1995), são procedimentos heurísticos de otimização global determinados pelo processo de seleção natural em sistemas biológicos. Sob o paradigma do algoritmo genético, mais importante que a geração de uma seqüência de soluções

candidatas de cada vez, uma população de soluções candidatas é mantida. As características distintivas primárias do *GA* são: uma codificação (*encoding*), um mecanismo de seleção, um mecanismo de recombinação (*crossover*), um mecanismo de mutação, e um mecanismo de refugo (*culling*). A codificação é uma estrutura de dados que descreve uma única solução para a otimização do problema, em uma representação compacta. Para dar uma analogia biológica, a codificação é semelhante a um cromossomo, na qual genes específicos descrevem vários aspectos do indivíduo em questão. O mecanismo de seleção leva a codificação de uma solução individual como entrada (*input*) e computa o valor correspondente daquela solução com relação ao(s) objetivo(s) da otimização. Este valor é então usado para selecionar soluções para recombinações (*crossover*). O mecanismo de recombinação produz uma ou mais novas codificações que usam dois códigos “pai” selecionados de uma população atual como a fonte de “genes”. A nova codificação descreve uma solução com algumas características em comum a essas de cada pai. O mecanismo de mutação é um algoritmo para perturbar uma codificação para produzir uma codificação “vizinha”, e é usado para ajudar a manter um reservatório de genes diversos para que sejam construídas soluções novas. O mecanismo de refugo (*culling*) é um procedimento para remover soluções conhecidas da população e as substituir por soluções novas. Isto pode ser feito substituindo os pais pelos seus filhos, ou apagando as piores soluções na população quando são criados os novos indivíduos, ou pela combinação de um pouco dos dois.

Para Gupta *et al.* (1996), o projeto de um algoritmo genético depende de seis conceitos fundamentais: representação, inicialização, função avaliação, reprodução, recombinação (*crossover*) e mutação.

Representação

Um problema pode ser resolvido uma vez que possa ser representado na forma de uma solução tipo cadeia (*solution string*). As menores unidades de informação contidas no cromossomo poderiam ser binárias, inteiras ou uma combinação de caracteres. Para a formação celular considerada, cada gene representa um número de célula e o posicionamento do gene no cromossomo representa o número da máquina ou número da peça. As primeiras M posições correspondem às máquinas e as últimas P posições correspondem às peças (M e P são os números de máquinas e peças). Para o exemplo da Figura 4.15, quando $M=5$ e $P=5$ o cromossomo(2,1,2,1,3,2,2,3,3,1) representa uma solução de três células com as seguintes máquinas e peças em cada célula (Gupta e Rajamani, 1994 apud Singh e Rajamani, 1996):

célula = 1: máquinas 2, 4; peça 5

célula = 2: máquinas 1, 3; peças 1, 2

célula = 3: máquina 5; peças 3, 4

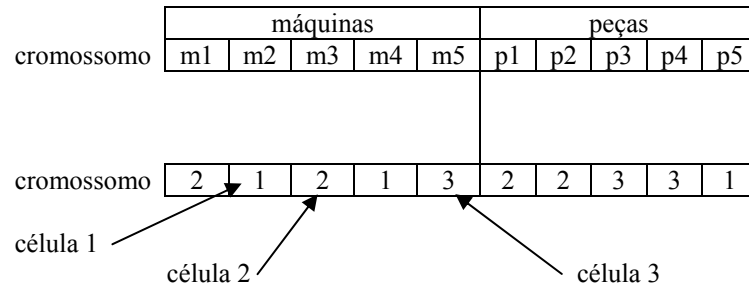


Figura 4.15 – representação do cromossomo.

Inicialização

O processo de inicialização pode ser executado com uma população criada ao acaso ou uma população bem adaptada (semeada).

Evolução ou função adaptação

Em um algoritmo genético um valor da função de aptidão é computado para cada cadeia na população e o objetivo é achar uma cadeia com o máximo valor. O objetivo do problema da formação celular é a minimização da soma ponderada dos vazios e dos elementos excepcionais. É necessário traçar este objetivo a uma função aptidão por um ou mais mapeamentos, como a transformação (Goldberg, 1989):

$$f(t) \begin{cases} f_{\max} - g(t), & \text{quando } g(t) < f_{\max} \\ 0, & \text{caso contrário} \end{cases} \quad (1)$$

onde $g(t)$ é o valor objetivo (soma ponderada dos vazios e dos elementos excepcionais) e f_{\max} é o maior valor da função objetiva na atual geração.

Seleção e reprodução

São selecionados cadeias com valores de aptidão mais altos para recombinação e mutação usando o processo de seleção estocástico sem substituição (Goldberg, 1989 apud Singh e Rajamani, 1996).

Recombinação e mutação

Os cromossomos a serem cruzados e o(s) ponto(s) de cruzamento são selecionados ao acaso. A técnica de recombinação de ponto-único é usada para ilustrar os conceitos. A recombinação é terminada com uma probabilidade chamada probabilidade de recombinação (*PCRS*). Por exemplo, considere dois cromossomos pai com um ponto de cruzamento de 3:

pai 1: 1 2 3 | 2 2 1 2 2

pai 2: 2 3 1 | 1 1 1 3 3

O operador de recombinação gera duas descendentes:

descendente 1: 1 2 3 | 1 1 1 3 3

descendente 2: 2 3 1 | 2 2 1 2 2

Para construir ao acaso, a mutação é feita com uma baixa probabilidade (*PMUTI*). Dois números ao acaso são gerados $r1$ e $r2$ tal que $1 \leq r1 \leq M + P$ (número total de peças mais máquinas) e $1 \leq r2 \leq C$ (número de máximo de células especificadas). O número de célula corresponde à máquina ou à peça especificada através de $r1$ é substituído com o número de célula $r2$.

Gulsen (1995) resume o processo genérico de procura nos passos descritos abaixo:

1. ao acaso gere uma solução inicial (população) e calcule o valor de aptidão (isto é, o valor da função objetiva) para cada membro da população;
2. repita os procedimentos abaixo:
 - a. selecionar os pais para recombinação (*crossover*);
 - b. gerar a descendência;
 - c. transformar alguns dos membros da população original (mutação);
 - d. fundir mutantes e descendentes na população;
 - e. selecionar alguns membros da população para manter o tamanho da população constante;
3. continuar até o critério de terminação desejado seja encontrado.

Os aspectos distintivos de procura genética são:

1. a procura é altamente paralela, com cada membro da população definindo muitas direções possíveis de procura. Potencialmente, a procura do *GA* poderia ser implementada eficazmente em hardware maciçamente paralelo.

2. nenhuma informação especial sobre a superfície de solução, como gradiente ou curvatura local, necessita ser identificada. A função objetivo não necessita ser polida, contínua ou unimodal.
3. algoritmos genéticos têm provado serem bastante robustos sob conjuntos de parâmetros variados e problemas particulares. Dada uma correlação fraca até mesmo entre codificações e valores da função objetivo, algoritmos genéticos normalmente acham soluções próximas às ótimas.

Desde os trabalhos de Holland e DeJong, foram aplicados algoritmos genéticos a muitos problemas combinatórios de otimização, inclusive programação para *job shop* (Beand 1994, Davis 1985, Storer *et al.* 1992), o problema do caixeiro viajante (Jog *et al.* 1991, Tate *et al.* 1994, Whitley *et al.* 1991), embalagem de caixas (House and Dagli 1992), carga de palete (Juliff 1993), problema quadrático de atribuição (Levitin e Rubinovitz 1993, Nissen 1994, Tate and Smith 1994), balanceamento de linha de montagem (Anderson e Ferris 1994) e plano de instalações (Tate e Smith 1993). Estes esforços são altamente diversos no seu uso da aproximação genética, mas freqüentemente os relatórios de resultados são favoráveis quando comparados a outros problemas heurísticos específicos. Os resultados encorajadores parecem ter origem da habilidade do *GA* para escapar dos mínimos ou máximos locais criados (isto é, recombinação e mutação), a robustez do *GA* para parâmetros de evolução e codificações, e a eficiência do *GA* na pesquisa global.

5 RESTRIÇÕES NO PROJETO DA CÉLULA

No projeto de uma célula, de acordo com Lorini (1993) e Rajamani *et al.* (1996), na aplicação de um algoritmo sobre as informações das rotas das peças visando minimizar a soma de custos de investimentos, processo e movimentação de materiais, alguns parâmetros de projeto podem ser estabelecidos como restrições para a criação das famílias e células, sendo, respectivamente, considerados, dentre outros, como: o tamanho da célula; o limite de carga para cada máquina; a interdependência entre as máquinas.

5.1 O tamanho da célula

O tamanho da célula é medido pelo número de processos ou máquinas alocadas na mesma. Na aplicação de um algoritmo de agrupamento para definição de células, serve como ponto de corte para o processo matemático ou heurístico de iteração. É um parâmetro que deve ser controlado por várias razões, entre as quais podem ser citadas como as mais importantes:

- a limitação do espaço físico disponível;
- o tipo de sistema de movimentação desejado;
- o tamanho dos lotes de fabricação;
- o número de operadores integrados a cada célula, para que estes possam acompanhar o seqüenciamento e ter uma visão adequada do conjunto, facilitando a supervisão.

5.2 Carregamento das máquinas

O planejamento operacional de uma célula implica na análise de dois enfoques que são: o carregamento e a programação das máquinas em estudo. A busca de uma solução otimizada simultaneamente para os dois aspectos é um tanto complexa, principalmente num período de planejamento de curto prazo. Para facilitar a busca de uma solução, normalmente procura-se resolver os dois aspectos separadamente. Uma vez estando o problema do carregamento definido, a programação fica condicionada a atender, de forma mais adequada, o equacionamento estabelecido.

Nos sistemas tradicionais, muitos estudos e metodologias são aplicados para resolver o problema da programação com enfoque nas peças a produzir. Nos sistemas celulares, notadamente para os mais automatizados, como os *FMS's*, a conotação muda um pouco devido ao alto custo dos equipamentos, tanto das máquinas, quanto dos sistemas de manuseio e movimentação dos materiais e de controle. As peças passam a ser apenas mais um item de programação em um universo que compreende os paletes e fixações, as ferramentas e os demais

sistemas. Mesmo assim, embora se tenham vários itens a considerar em programações separadas, normalmente as peças têm um maior significado, devido a se desejar uma alta utilização das máquinas-ferramenta pelo seu elevado custo. Estima-se que nos *FMS's*, as máquinas-ferramenta representem em torno de 65% dos custos do sistema (Kusiak apud Lorini, 1993). Por esta razão, é importante considerar o aspecto de carregamento das células, representado pela alocação dos tempos de processo das peças da família em sua definição, com as máquinas correspondentes, definidas pelas respectivas rotas, dentro de um determinado período de tempo considerado.

Devem ser conhecidos os tempos de processo das peças e estabelecidos limites para o carregamento de cada máquina, para fins de alocação destes tempos de processo das peças durante a definição das famílias. O limite de carregamento de cada máquina deve considerar as características específicas que definem a capacidade média de utilização da mesma no intervalo considerado. O somatório dos tempos de processo de todas as peças de determinada família deve ser mantido em níveis no máximo iguais ao limite definido para cada máquina correspondente. Considerando-se todas as n peças de uma família com seus respectivos tempos de processo $t(i, j)$, alocados em uma determinada máquina i , o somatório destes tempos deve ser compatível com a capacidade de processamento daquela máquina, para um período de tempo definido e suas características operacionais (T_i).

$$t(i, j) = (\text{tempo de processo da peça}(j) \text{ na máquina}(i)) \cdot (\text{tamanho do lote}) \quad (2)$$

$$\sum_{j=1}^n t(i, j) \leq T_i \quad (3)$$

O limite de carregamento (T_i) pode ser estabelecido levando-se em conta as variáveis que definem o horizonte de planejamento e o aproveitamento da máquina, pela seguinte expressão:

$$T_i = K \cdot n \cdot c_i \cdot p \quad (4)$$

K = valor de tempo de um turno de operação

n = número de turnos de trabalho de uma jornada

p = período de tempo considerado

c_i = coeficiente de utilização da máquina

Quando o somatório superar o limite (T_i), duas medidas podem ser tomadas para contornar o problema: aumentar a disponibilidade daquele processo com a inclusão de uma nova máquina para aliviar o “gargalo” identificado, ou reduzir-se o tamanho da família, excluindo-se algumas das peças agrupadas, até compatibilizar a demanda de processo com a capacidade de absorção da

máquina. Estas medidas, embora simples, exigem critérios adequados para soluções racionais. Uma nova máquina significa um custo que deve ser considerado do ponto de vista tanto de investimento inicial, quanto de retorno operacional, que irá depender do seu percentual de aproveitamento na célula. É necessário definir um parâmetro de utilização aceitável, compatível com o ambiente do contexto em estudo.

A exclusão de peças de uma família significa que estas deverão ser fabricadas fora da célula, num ambiente convencional, ou então ser encomendadas a terceiros. Admitindo-se que sejam solicitadas a fabricantes externos, algum critério deve ser adotado também para definir qual peça deva ser separada. Estes critérios devem ser de interesse de cada ambiente, podendo se voltar a aspectos econômicos, como o de eliminar as peças de menor custo de processo, para minimizar o dispêndio financeiro com a encomenda. Do ponto de vista de operações, pode-se também separar aquelas de maior complexidade de manufatura, ou considerar outro critério específico qualquer.

5.3 Interdependência de máquinas

Na definição de grupos de máquinas, pode-se desejar manter agregadas, algumas delas, em uma mesma célula. Esta consideração deve ser observada devido a uma interdependência estabelecida por características tecnológicas, físicas, ou por outra razão estratégica. Mesmo o aproveitamento de equipamentos complementares existentes pode impor o vínculo, entre outras justificativas, como aquelas relativas ao porte, à precisão, às compatibilidades de tecnologias como comandos *CNC*, etc. Uma vez uma máquina sendo definida para uma célula, aquelas outras que mantiverem alguma dependência deverão, automaticamente, constar naquele agrupamento, evidentemente para atender a processos compatíveis e necessários.

5.4 Balanceamento das células

O processo de formação de famílias fornece duas informações básicas para o projeto das células de manufatura: a relação dos componentes que serão processados em cada célula, bem como das máquinas que aí serão alocadas para realizar o processamento destes componentes, sem considerar aspectos operacionais.

O balanceamento do fluxo das operações, entre as diversas máquinas de uma célula deve procurar a maximização no aproveitamento da capacidade instalada.

Em fase de projeto da célula, deve-se buscar em primeiro plano atender a carga de processo. O problema de balanceamento, no sentido de evitar máquinas ociosas ou sobrecarregadas deve ser equacionado com adequado Planejamento e Programação da Produção.

6 MEDIDAS DE DESEMPENHO

Devido à crescente diversidade de métodos de agrupamento desenvolvidos, surge a necessidade de criação de parâmetros objetivos de avaliação dos resultados obtidos pela aplicação de diversos métodos sobre um mesmo problema, a fim de permitir avaliar-se sobre qual das soluções apresentadas seria a mais interessante.

A avaliação da qualidade de um determinado agrupamento, partindo-se dos resultados obtidos na matriz incidência reordenada, é dada por dois fatores:

- O número de elementos extra-celulares, isto é, o número de incidências não agrupadas em qualquer uma das células identificadas, o qual tem de ser minimizado e
- O grau de similaridade observado entre os diversos elementos alocados a uma mesma célula, o qual deve ser maximizado.

O que ocorre, em termos práticos, é que há uma espécie de troca entre estes dois parâmetros. Se, através da criação de células adicionais, duplicação de máquinas, ou por qualquer outro meio, tenta-se minimizar o número de elementos extra-celulares, o grau de similaridade médio intra-celular vai provavelmente diminuir. Assim, um coeficiente que meça o grau de qualidade de um determinado agrupamento, calculado a partir dos resultados observados em relação a estas duas variáveis, deve se concentrar em uma certa medida de equilíbrio entre as mesmas.

Chandrasekaran e Rajagolapan apud Singh e Rajamani (1996) desenvolveram um indicador de eficiência do agrupamento (*Group Efficiency*, ou *GE*), baseando-se nesta análise. Uma primeira apresentação deste indicador vem dada a seguir:

$$GE = p\theta_1 + (1-p)\theta_2, \quad 0 \leq p \leq 1 \quad (5)$$

Enquanto θ_1 e θ_2 são computados como:

$$\theta_1 = \frac{e_1}{\sum_{r=1}^k Gr Cr} \quad (6)$$

$$\theta_2 = 1 - \frac{e_0}{MN - \sum_{r=1}^k Gr Cr} \quad (7)$$

onde:

k = número total de células identificadas;

M = número total de linhas (máquinas) na matriz incidência;

N = número total de colunas (componentes) na matriz incidência;

e_1 = número total de incidências intra-celulares após reordenamento;

e_0 = número total de incidências extra-celulares após reordenamento;

Gr = número total de linhas (máquinas) na célula 'r';

Cr = número total de colunas (componentes) na célula 'r';

p = fator de ponderação da média;

Além disso, θ_1 e θ_2 podem ser definidos, respectivamente, como a taxa de incidências registradas dentro das células sobre o total de incidências (valor 1) possíveis e como a taxa de não incidências (valor 0 ou 'branco') registradas fora das células identificadas em relação ao total de não incidências – ou espaços – possíveis.

De forma objetiva, o GE calculado significa a média ponderada entre estas duas taxas que, simplificada, representam respectivamente o segundo e o primeiro parâmetros de avaliação da qualidade do agrupamento definidos anteriormente.

Há duas formas interessantes de utilização deste indicador, segundo Silveira (1994), na primeira pode-se calcular os diferentes GE 's para cada resultado gerado por um diferente método de agrupamento para uma mesma matriz. Quanto maior o GE , melhor o resultado teórico do método. Assim, pode-se escolher o resultado de maior GE . Este é o método objetivo de comparar, inclusive, a eficácia dos diferentes métodos utilizados.

Em segundo lugar, é comum que, após o agrupamento da matriz segundo a heurística proposta, façam-se ajustes 'manuais' sobre esta. Ajustes como o deslocamento de determinado componente ou máquina de uma para outra célula, ou a eliminação ou inclusão de incidências dentro das células, quando existem roteiros alternativos inicialmente não considerados. Para cada modificação destas, pode-se recalculá-lo GE da matriz para, objetivamente, concluir se o resultado desta modificação 'forçada' é melhor que o anterior.

Fogliatto (2004) apresenta o mesmo cálculo da eficiência do agrupamento utilizando uma abordagem mais simples. A eficiência depende de dois fatores: a proporção de elementos extra-celulares (Ex) e a proporção de espaços ocupados nas células (Es).

O primeiro deve ser minimizado enquanto o segundo deve ser maximizado.

$$GE = \frac{[(1 - Ex) + Es]}{2} \quad (8)$$

onde Ex é o número de incidências extra-células dividido pelo número total de incidências e Es é

o número de espaços ocupados dividido pelo número de espaços intra-celulár. Assim, na matriz da Figura 6.1, o valor da eficiência do agrupamento é 80,1%.

	1	2	3	4	5	6	7	8
A	1	1	1	1		1	1	
B	1	1		1				
C	1	1		1			1	
D					1		1	1
E		1			1	1	1	1
F						1		1

Figura 6.1 – Exemplo de matriz para cálculo de GE
Fonte: Fogliatto, 2004.

$$Ex = \frac{4}{23} = 0,1739$$

$$Es = \frac{19}{24} = 0,7917$$

$$GE = \frac{[(1 - Ex) + Es]}{2} = \frac{[(1 - 0,1739) + 0,7917]}{2} = 0,8089 = 80,1\%$$

Um outro parâmetro, a eficácia do agrupamento (*Group efficacy*), proposta por Kumar e Chandrasekaran apud Singh e Rajamani (1996), é uma medida de desempenho para superar o baixo poder de discriminação da eficiência do agrupamento entre matrizes bem-estruturadas e mal-estruturadas. Ao contrário da eficiência de agrupamento, a eficácia de agrupamento não é afetada pelo tamanho da matriz.

Outra medida de agrupamento (*Grouping measure*), proposta por Miltenburg e Zhang apud Singh e Rajamani (1996) e segue do trabalho de Chandrasekaran e Rajagopalan. Trata-se de uma medida direta da efetividade de um algoritmo para obter uma matriz agrupada final. O valor da medida de agrupamento é alto se a utilização das máquinas for alta (menos vazias) e poucas peças requerem processamento de máquinas em mais de uma célula (menos elementos excepcionais). Este foi considerado como uma medida primária por Miltenburg e Zhang ao comparar o desempenho de vários algoritmos, que também criaram duas outras medidas que poderiam ser usadas para enriquecer comparações: *clustering measure* e a *bond energy measure*.

Geralmente as matrizes rearranjadas não possuem o mesmo número de peças e máquinas da matriz incidência depois de aplicadas as restrições de projeto da célula, e assim pode-se adotar, como critério de melhor solução, o arranjo que obtiver o maior número de peças agrupadas e o maior número de peças fabricadas com o menor número de máquinas agrupadas.

7 TRABALHO IMPLEMENTADO

A implantação da manufatura celular em uma empresa leva em consideração questões como a análise de sua especificidade, a diversidade de seus produtos e componentes fabricados, o volume de produção, a cultura técnica como o grau de motivação e envolvimento dos funcionários dentre outros fatores.

Após a análise e definição do problema, com objetivos bem definidos segue-se para a fase de formação das famílias, balanceamento e projeto definitivo das células aplicando-se de ferramentas técnicas adequadas.

Cada técnica de formação de famílias compreende um conjunto específico de dados de entrada, no caso de técnicas baseadas em características de processo, as informações utilizadas são de três fontes:

- produtos e seus componentes (lista de materiais);
- lista de máquinas;
- rotas de produção.

Adicionalmente, algumas técnicas utilizam outros parâmetros, como tempos de processamento e de preparação, custos com transporte, investimento em novas máquinas, volumes de produção e características físicas (forma e propriedade) dos componentes.

7.1 Considerações sobre a escolha de uma metodologia

Para a definição dos agrupamentos de peças e máquinas, apesar de estudos comparativos demonstrarem a superioridade de algumas técnicas em relação a outras, cabe ao analista definir qual ou quais métodos utilizar, dependendo dos objetivos e das características do estudo.

Alguns dos parâmetros, segundo Silveira (1994), que podem servir de base para a escolha do método são:

- os objetivos do estudo: a implantação de células manuais ou autônomas, de banco de dados para projetos, de *FMS*, com sistemas de auxílio ao planejamento da produção (*CAPP*);
- o nível de complexidade do problema: a diversidade de componentes, diversidade de máquinas, quantidade e acurácia dos dados utilizados, seqüenciamento da produção, rotas alternativas, etc.;
- a relação custo versus eficiência que se deseja obter no estudo: o tempo disponível para o aprendizado e aplicação do método, a capacitação das pessoas envolvidas direta e indiretamente na análise; seu conhecimento da técnica e capacidade de compreensão da realização, os recursos computacionais disponíveis, etc.

De modo geral, não é errado afirmar que qualquer método, se bem aplicado, pode trazer resultados interessantes. No entanto, a utilização da Tecnologia de Grupo na implantação de sistemas caros como Sistemas de Manufatura Flexível deve ser mais rigorosa, pois o custo posterior da baixa eficácia dos resultados gerados pode ser alto, ou seja, quanto maiores os recursos envolvidos no projeto, maior deve ser o rigor na escolha e aplicação da técnica. Além disso, em muitos casos, a utilização de vários métodos simultaneamente, em diversas etapas do trabalho, é um exercício interessante.

7.2 Estrutura do sistema e sua implementação

Este trabalho propõe um sistema, como ferramenta para projeto de células, implementado com suporte de um banco de dados (*MS Access*) programado em linguagem *MS Visual Basic for Application*.

Das técnicas de agrupamento estudadas, optou-se por implementar os algoritmos *Rank Order Clustering (ROC)*, *Close Neighbor Algorithm (CNA)* e um procedimento heurístico (*HEU*).

Os parâmetros de projeto estabelecidos como restrições para a criação das famílias e células, respectivamente considerados, são: o tamanho da célula; o limite de carga para cada máquina; e o limite de carga para duplicação da máquina gargalo.

O sistema apresenta os resultados em um arquivo em *MS Excel* com várias planilhas. O fluxograma da Figura 7.1 representa a estrutura geral do sistema.

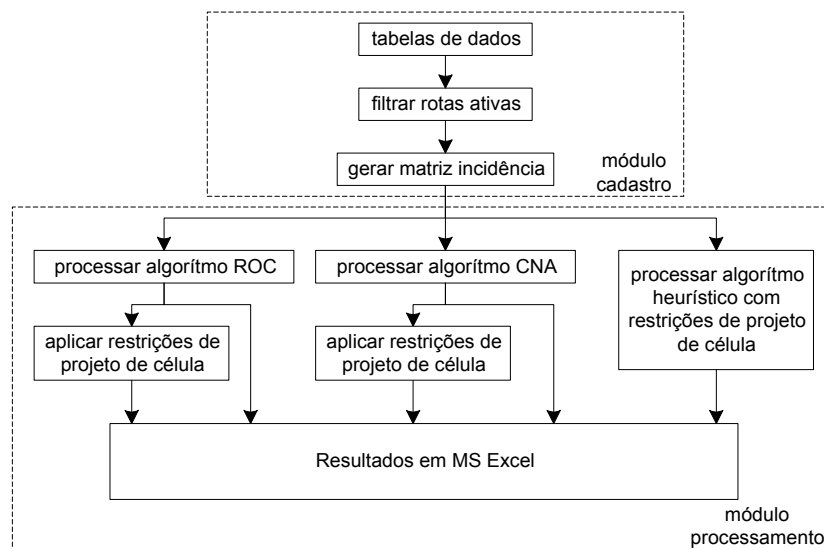


Figura 7.1 – Fluxograma do sistema.

Os módulos implementados são descritos a seguir e ilustrados pelas respectivas interfaces com o usuário.

7.2.1 Modo de inicialização principal

Uma abordagem eficaz para organizar e apresentar tarefas em um aplicativo é o uso de um formulário principal como centro de controle do sistema. A interface *entrada* proporciona a navegação para todas as tarefas e é o primeiro formulário carregado quando o aplicativo é iniciado. Este funciona como um painel de controle onde estão inseridos botões para a chamada dos formulários *f_pecas*, *f_maquinas*, *f_rotas*, *f_disponibilidades* e *matriz*.

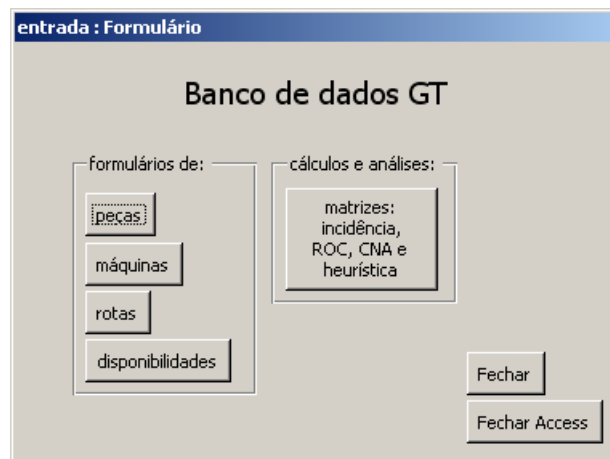


Figura 7.2 – Formulário *entrada*.

7.2.2 Módulo para cadastramento

Para cadastramento e posterior processamento das informações relativas a peças, equipamentos e restrições de projeto, os dados são armazenados em cinco tabelas com informações relativas às peças (*t_pecas*), às máquinas (*t_maquinas*), ao seqüenciamento e fluxo das peças através das máquinas (*t_rotas* e *t_rotas sub*) e as restrições de projeto (*t_disponibilidade*). Os formulários criados tornam a manipulação de dados mais intuitiva.

Cadastro das restrições de projeto

A tabela *t_disponibilidade* armazena as informações relativas às restrições de projeto das células. Existe um único registro com sete campos numéricos responsáveis pelo armazenamento das seguintes informações:

- *maq_por_grupo*: quantidade limite de máquinas para cada célula;
- *n_turnos_trab*: número *n* de turnos de trabalho de uma jornada;
- *k_tempo_turno*: tempo *k*, em minutos, de um turno de operação;
- *p_perodo_considerado*: período *p*, em dias, considerado;
- *percentual*: percentual limite para que, a partir dele, a máquina seja duplicada;
- *t_limite*: valor obtido a partir da equação 13;

- $t_{duplica}$: valor obtido a partir da equação 14.

A partir da tabela $t_{disponibilidade}$, com as informações das restrições de projeto, é criado formulário $f_{disponibilidade}$, apresentado na Figura 7.3.

Figura 7.3 – Formulário $f_{disponibilidade}$.

O valor *limite de carregamento* (t_{limite}) é obtido a partir da modificação das equações 3 e 4, apresentadas no capítulo 5. A alteração corresponde a divisão das equações pelo coeficiente de utilização da máquina (c_i) como mostrado a seguir:

$$\frac{\sum_{j=1}^n t(i, j)}{c_i} \leq \frac{T_i}{c_i} \quad (9)$$

$$\frac{\sum_{j=1}^n t(i, j)}{c_i} \leq \frac{K n c_i p}{c_i} \quad (10)$$

$$\frac{\sum_{j=1}^n t(i, j)}{c_i} \leq K n p \quad (11)$$

$$t_{limite} = \frac{T_i}{c_i} \quad (12)$$

$$t_{limite} = K n p \quad (13)$$

Esta modificação das equações 3 e 4 para a equação 13 torna o tempo limite de carregamento independente da máquina, ou seja, ao invés de armazenar i valores de T_i , um para cada linha i , o sistema armazena um único valor t_{limite} válido para todas as máquinas. Ao

mesmo tempo, o valor $t(i,j)$ não é armazenado no cruzamento da linha i na coluna j mas sim o valor $t(i,j)$ dividido pelo valor de c_i daquela máquina. Dessa forma o sistema não necessita associar o valor de T_i a linha reordenada tornando assim a matriz incidência independente dos valores limite de carregamento das respectivas máquinas.

O valor *limite de carregamento para duplicação* é obtido a partir da multiplicação da equação 13 pelo valor do campo *percentual*, assim:

$$t_duplica = t_limite \cdot percentual \quad (14)$$

Cadastro das máquinas

A tabela *t_maquinas* é estruturada para conter informações básicas relativas às máquinas e necessárias ao sistema de projeto. Cada registro está estruturado em seis campos, contendo as seguintes informações:

- ID_maquinas*: indexador numérico automático do registro da máquina (não é permitida a sua duplicação);
- codigo*: texto usado para identificar a máquina na montagem da matriz incidência (não é permitida a sua duplicação);
- descricao*: descrição do equipamento;
- coeficiente_utilizacao*: coeficiente de utilização da máquina;
- custo_hora_maq*: valor do custo da hora de trabalho nesta máquina;
- status_da_maq*: indicação do “status” do registro.

O formulário *f_maquinas*, Figura 7.4, permite inserir, pesquisar, editar e excluir registros com as informações de cada máquina.

The screenshot shows a software window titled 'f_maquinas' with a sub-header 'Máquinas'. On the left, there are input fields for: ID_maquinas (1), código (maq001), descrição (torno mecânico), coeficiente de utilização (0,78), custo hora máquina (R\$30,00), and status da máquina (Ativa). On the right, there is a table titled 'máquinas cadastradas' with columns 'codigo', 'descricao', and 'status_da_maq'. The table lists 13 records, all with 'Ativa' status. At the bottom, there is a pagination control showing 'Registro: 1 de 40'.

codigo	descricao	status_da_maq
maq001	torno mecânico	Ativa
maq002	fresadora	Ativa
maq003	furadeira	Ativa
maq004	lixadeira	Ativa
maq005	plana	Ativa
maq006	máquina 006	Ativa
maq007	máquina 007	Ativa
maq008	máquina 008	Ativa
maq009	máquina 009	Ativa
maq010	máquina 010	Ativa
maq011	máquina 011	Ativa
maq012	máquina 012	Ativa
maq013	máquina 013	Ativa

Figura 7.4 – Formulário *f_maquinas*.

Cadastro das peças

Na tabela *t_pecas* são armazenadas as informações relativas à identificação das peças referenciadas na matriz peça-máquina. Cada registro é estruturado em cinco campos, contendo as seguintes informações:

- ID_pecas*: índice numérico que indexa automaticamente o registro da peça (não é permitida a sua duplicação);
- codigo*: texto usado para identificar a peça na montagem da matriz incidência (não é permitida a sua duplicação);
- descricao*: descrição da peça;
- tamanho do lote de fabricação*: tamanho do lote de fabricação;
- status da peça*: indicação do “status” do registro.

O formulário *f_pecas* (Figura 7.5) possui os mesmos recursos de inserção, pesquisa, edição e exclusão de registros de peças disponíveis no formulário *f_maquinas*.

The screenshot shows a software window titled "f_pecas" with a sub-header "Peças". On the left, there are input fields for "ID_pecas" (value: 1), "código" (value: p001), "descrição" (value: chapeleta), "tamanho do lote de fabricação" (value: 50), and "status da peça" (value: Ativa). On the right, a table titled "peças cadastradas" lists 13 records. At the bottom, there is a "Registro:" field showing "1" of "100" total records, with navigation buttons.

codigo	descrição	status da peça
p001	chapeleta	Ativa
p002	peça 002	Ativa
p003	peça 003	Ativa
p004	peça 004	Ativa
p005	peça 005	Ativa
p006	peça 006	Ativa
p007	peça 007	Ativa
p008	peça 008	Ativa
p009	peça 009	Ativa
p010	peça 010	Ativa
p011	peça 011	Ativa
p012	peça 012	Ativa
p013	peça 013	Ativa

Figura 7.5 – Formulário *f_pecas*.

Cadastro das rotas de fluxo

As tabelas (*t_rotas* e *t_rotas sub*) reúnem as informações relativas a uma determinada matriz de fluxo de processo, ou seja, um conjunto de rotas de peças nas respectivas máquinas em que devem ser processadas. As informações são registradas de tal modo que, para cada peça, correspondam tantos registros quantas forem as máquinas em que é processada, indicando também o respectivo tempo de processo. Cada registro da tabela *t_rotas* tem o seguinte conteúdo:

- ID_rota*: índice numérico que indexa automaticamente o registro da rota (não é permitida a sua duplicação);

- b) *codigo*: texto usado para identificar a rota na montagem da matriz incidência;
- c) *descricao*: descrição da rota;
- d) *peca*: campo que relaciona as tabelas *t_rotas* e *t_pecas* através da *ID_pecas* (ver *Relacionamento entre tabelas* no Apêndice I);
- e) *status da rota*: indicação do “status” do registro;
- f) *tempo_rota*: indica o tempo total de processamento desta peça nesta rota;
- g) *custo_rota*: indica a soma do produto *custo_hora_maquina* e *tempo_rota* que representa o custo de utilização das máquinas pela peça na rota (ver *Relacionamento entre tabelas* no Apêndice I);
- h) *seleccionado*: botão para desativar manualmente a rota;

Em cada registro da tabela *t_rotas sub* contém as seguintes informações:

- a) *ID_montarota*: índice numérico que indexa automaticamente o registro (não é permitida a sua duplicação);
- b) *ID_rota*: campo que relaciona numericamente as tabelas *t_rotas* e *t_rotas sub* (ver *Relacionamento entre tabelas* no Apêndice I);
- c) *maquina*: campo que relaciona as tabelas *t_maquinas* e *t_rotas sub* através do campo *ID_maquinas* (ver *Relacionamento entre tabelas* no Apêndice I);
- d) *tempo de processamento*: campo para indicar o tempo de processamento da peça na máquina.

O formulário *f_rotas*, Figura 7.6, possui os mesmos recursos de inserção, pesquisa, edição e exclusão de registros de rotas disponíveis nos formulário *f_maquinas* e *f_pecas*.

f_rotas

Rotas

Fechar

ID_rota: 1

código: r01

descrição: rota 01

peça: p001 Ativa

status da rota: Ativa

tempo_rota: 20,00 minutos

custo_rota: R\$700,00

inativar manualmente

rotas cadastradas

descrição	t_rotas	t_pecas	status da rota
artigo boe & cheng	r07	p001	Ativa
artigo boe & cheng	r07	p002	Ativa
artigo boe & cheng	r07	p003	Ativa
artigo boe & cheng	r07	p004	Ativa
artigo boe & cheng	r07	p005	Ativa
artigo boe & cheng	r07	p006	Ativa
artigo boe & cheng	r07	p007	Ativa
artigo boe & cheng	r07	p008	Ativa

máquinas pertencentes a essa rota:

máquina	tempo de processamento	custo máq. de processamento	status da máquina:
maq002	10,00	500,00	Ativa
maq003	10,00	200,00	Ativa
maq005	10,00	200,00	Ativa
maq007	10,00	100,00	Ativa
maq010	10,00	100,00	Ativa

Atualizar dados

Registro: 1 de 5

Registro: 1 de 180

Figura 7.6 – Formulário *f_rotas* e subformulário *f_rotas sub*.

7.2.3 Módulos de processamento do sistema

O processamento de uma matriz de incidência peça-máquina por diferentes opções de algoritmos, é realizado através da interface do sistema identificado como *matriz*. As opções do módulo permitem processar uma matriz de dados e verificar a formação dos agrupamentos através dos algoritmos *ROC*, *CNA EP*, *CNA* e *HEU*. A interface é ilustrada na Figura 7.7.



Figura 7.7 – Formulário *matriz*.

Procurando um nível de interatividade, potencialidade e controle maior sobre o aplicativo, rotinas foram escritas em *MS Visual Basic*.

Rotina definição de variáveis

Esta rotina define as variáveis públicas utilizadas pelo sistema. As variáveis responsáveis pelo armazenamento dos dados da matriz apresentam a seguinte notação:

$$\text{array} (I, J, K)$$

onde *array* é o nome da variável, *I* representa as linhas da matriz (máquinas) e *J* representa as colunas da matriz (peças), *K* representa níveis (*layers*) que armazenarão dados da matriz, variando de 0 a 2, assim, se $K = 0$ – armazena valores binários, $K = 1$ – armazena os tempos de processo de cada peça *J* na máquina *I*, $K = 2$ – armazena o número da célula.

Rotina *m04_mat_incidencia*

Toda a metodologia do sistema baseia-se na matriz de incidência peça-máquina, montada de tal modo que a cada linha corresponda a uma máquina, e nas colunas sejam representadas as peças. A matriz é montada pela rotina *m04_mat_incidencia* em cinco etapas, conforme ilustra a Figura 7.8.

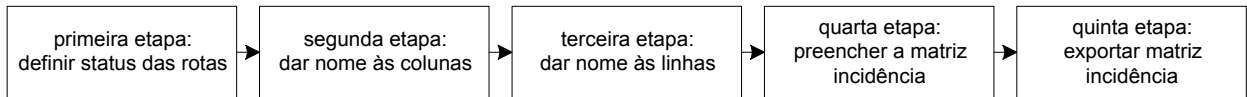


Figura 7.8 – Etapas da rotina *m04_mat_incidencia*.

A primeira etapa define o status da rota em função do status da peça e da máquina, isto é, se a peça e a máquina forem iguais a “*Ativa*”, então a rota será “*Ativa*”. Se um dos status, da peça ou da máquina, for igual a “*Inativa*”, então a rota será “*Inativa*”. As etapas 2, 3 e 4, a partir das rotas ativas, definem o tamanho da matriz, os nomes das linhas e das colunas e seu preenchimento. O último passo efetua uma cópia da matriz para ser processado em cada algoritmo.

Adotou-se as primeiras 40 máquinas e 45 peças da matriz de Wang (1998) para a realização de testes iniciais, identificada como matriz *teste_1*, representada na Figura 7.9.

	p001_JW	p002_JW	p003_JW	p004_JW	p005_JW	p006_JW	p007_JW	p008_JW	p009_JW	p010_JW	p011_JW	p012_JW	p013_JW	p014_JW	p015_JW	p016_JW	p017_JW	p018_JW	p019_JW	p020_JW	p021_JW	p022_JW	p023_JW	p024_JW	p025_JW	p026_JW	p027_JW	p028_JW	p029_JW	p030_JW	p031_JW	p032_JW	p033_JW	p034_JW	p035_JW	p036_JW	p037_JW	p038_JW	p039_JW	p040_JW	p041_JW	p042_JW	p043_JW	p044_JW	p045_JW							
maq001				1	1																																															
maq002																																																				
maq003				1	1																																															
maq004			1							1																																										
maq005						1																																														
maq006	1	1																																																		
maq007				1	1					1																																										
maq008						1																																														
maq009			1																																																	
maq010													1	1																																						
maq011							1																																													
maq012	1	1																																																		
maq013							1																																													
maq014																																																				
maq015																																																				
maq016																																																				
maq017																																																				
maq018																																																				
maq019																																																				
maq020																																																				
maq021																																																				
maq022																																																				
maq023																																																				
maq024																																																				
maq025																																																				
maq026	1																																																			
maq027																																																				
maq028																																																				
maq029																																																				
maq030																																																				
maq031																																																				
maq032																																																				
maq033																																																				
maq034																																																				
maq035																																																				
maq036																																																				
maq037																																																				
maq038	1	1																																																		
maq039																																																				
maq040	1	1																																																		

Figura 7.9 – Matriz *teste_1*.

Fonte: Wang (1998).

7.2.3.1 Algoritmos *ROC* e *CNA*

Os algoritmos *ROC* e *CNA* são apresentados neste capítulo, sem restrições e posteriormente tratados com a aplicação das restrições e com um procedimento para definição das fronteiras das células. O algoritmo *HEU*, por ter as restrições e a definição das fronteiras das células dentro de seu processamento, é tratado separadamente.

Processamento do algoritmo *ROC*

A rotina *m05_algoritmo_roc* é responsável pelo rearranjo da matriz incidência através da aplicação do algoritmo *Rank Order Clustering* na sua concepção original, sem restrições de projeto da célula. O procedimento de ordenamento das linhas e das colunas segue exatamente a seqüência apresentada no capítulo 4.2.1 e é ilustrado na Figura 7.10. Após efetuado o arranjo da matriz, uma cópia da mesma é salva para posterior análise.

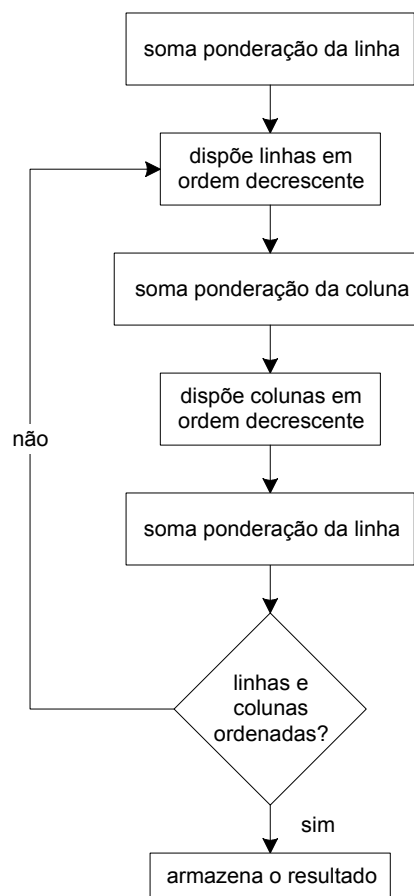


Figura 7.10 – Seqüência da rotina *m05_algoritmo_roc*.

O algoritmo *ROC*, executado para ordenar as linhas e colunas da matriz incidência *teste_1*, obtém a matriz da Figura 7.11.

	p001_JW	p030_JW	p043_JW	p045_JW	p002_JW	p015_JW	p040_JW	p014_JW	p029_JW	p038_JW	p044_JW	p037_JW	p021_JW	p022_JW	p012_JW	p013_JW	p007_JW	p011_JW	p018_JW	p042_JW	p032_JW	p041_JW	p023_JW	p020_JW	p031_JW	p026_JW	p003_JW	p010_JW	p019_JW	p036_JW	p006_JW	p017_JW	p027_JW	p034_JW	p028_JW	p008_JW	p016_JW	p025_JW	p035_JW	p039_JW	p004_JW	p005_JW	p024_JW	p009_JW	p033_JW												
maq038	1	1	1	1	1	1	1	1	1	1	1																																														
maq040	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1											
maq006	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1										
maq012	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1									
maq026	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1									
maq017					1							1	1	1																																											
maq028					1																																																				
maq016						1									1	1																																									
maq013							1										1	1	1	1																																					
maq021								1								1	1																																								
maq036									1																																																
maq011										1																																															
maq014											1																																														
maq035												1	1																																												
maq002																	1	1																																							
maq010																	1	1																																							
maq030																																																									
maq031																																																									
maq018																																																									
maq034																																																									
maq033																																																									
maq019																																																									
maq015																																																									
maq004																																																									
maq009																																																									
maq025																																																									
maq008																																																									
maq005																																																									
maq022																																																									
maq037																																																									
maq039																																																									
maq023																																																									
maq020																																																									
maq027																																																									
maq024																																																									
maq029																																																									
maq007																																																									
maq032																																																									
maq003																																																									
maq001																																																									

Figura 7.11 – Matriz teste_1 ordenada pela rotina m05_algoritmo_roc.

Processamento do algoritmo CNA EP

A rotina *06_algoritmo_cna_ep* rearranja a matriz incidência através da aplicação do algoritmo *Close Neighbour Algorithm (CNA)*, na interpretação de Fogliatto (2004) e Dalmas (2004) e sem restrições de projeto da célula. O procedimento de ordenamento das linhas e das colunas segue exatamente a seqüência apresentada no capítulo 4.2.2 e é ilustrado na Figura 7.12.

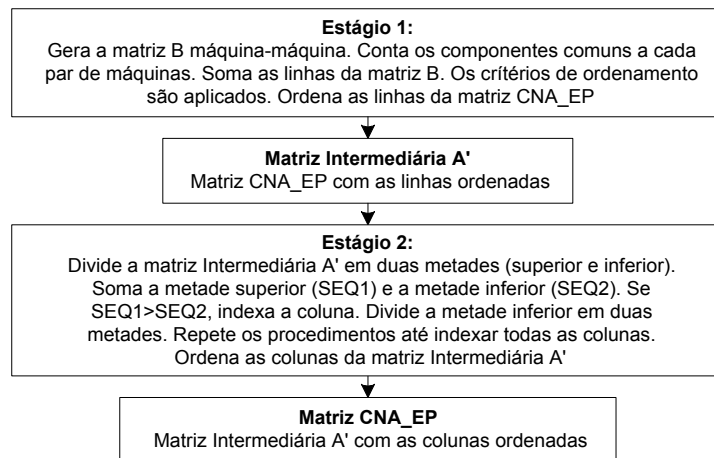


Figura 7.12 – Seqüência da rotina 06_algoritmo_cna_ep.

Executado o algoritmo *CNA* sobre a matriz incidência teste_1 tomada para teste, obtém-se como resultado a matriz da Figura 7.13.

	p001_JW	p002_JW	p006_JW	p014_JW	p015_JW	p017_JW	p023_JW	p026_JW	p027_JW	p028_JW	p029_JW	p030_JW	p031_JW	p032_JW	p033_JW	p038_JW	p040_JW	p041_JW	p043_JW	p044_JW	p045_JW	p004_JW	p005_JW	p008_JW	p009_JW	p012_JW	p013_JW	p016_JW	p024_JW	p025_JW	p034_JW	p035_JW	p039_JW	p003_JW	p010_JW	p019_JW	p020_JW	p021_JW	p022_JW	p036_JW	p007_JW	p011_JW	p018_JW	p037_JW	p042_JW								
maq006	1	1																																																			
maq040	1	1		1	1																																																
maq026	1			1	1																																																
maq038	1	1		1	1																																																
maq012	1	1																																																			
maq036							1																																														
maq018								1							1	1																																					
maq034								1							1	1																																					
maq033								1							1	1																																					
maq015								1							1	1																																					
maq025								1							1	1																																					
maq005		1				1			1	1	1																																										
maq022		1				1			1	1	1																																										
maq037		1				1			1	1	1																																										
maq008		1				1			1	1	1																																										
maq023		1				1			1	1	1																																										
maq039		1				1			1	1	1																																										
maq019						1												1																																			
maq007																																																					
maq003																																																					
maq032																																																					
maq001																																																					
maq027																																																					
maq024																																																					
maq029																																																					
maq016																																																					
maq021																																																					
maq002																																																					
maq010																																																					
maq030																																																					
maq004																																																					
maq009																																																					
maq020																																																					
maq017																																																					
maq014																																																					
maq035																																																					
maq013																																																					
maq011																																																					
maq028																																																					
maq031																																																					

Figura 7.13 – Matriz *teste_1* ordenada pela rotina *m06_algoritmo_cna_ep*.

Nota-se que a aplicação desta interpretação do algoritmo ordena as linhas, porém não ordena as colunas da matriz *teste_1*.

Realizadas revisões, não foram encontradas contradições entre os procedimentos apresentados por Fogliatto (2004) e Dalmas (2004), cálculos manuais para ordenamento da matriz e a rotina programada. Optou-se, então, por buscar a concepção original do algoritmo *CNA* proposto por Boe e Cheng (1991). Aquele procedimento difere no ordenamento das colunas (estágio 2) o que poderia significar uma interpretação diferente deste estágio pelos autores citados. Embora implementado em fase de desenvolvimento do sistema, o algoritmo não é utilizado para comparação entre algoritmos diferentes devido à incapacidade de rearranjar as colunas de matrizes de maior dimensão e recebeu o nome de *CNA EP*. O algoritmo concebido originalmente é apresentado a seguir, na rotina *m07_algoritmo_cna*.

Processamento do algoritmo *CNA*

A rotina *07_algoritmo_cna* executa o arranjo de matrizes através do algoritmo *Close Neighbour Algorithm (CNA)* na concepção original descrita por Boe e Cheng (1991), sem restrições de projeto da célula. A seqüência do processo de diagonalização da matriz, é efetuada conforme metodologia apresentada no capítulo 4.2.2, é ilustrada na Figura 7.14. Após efetuado o arranjo da matriz, uma cópia é salva para posterior análise.

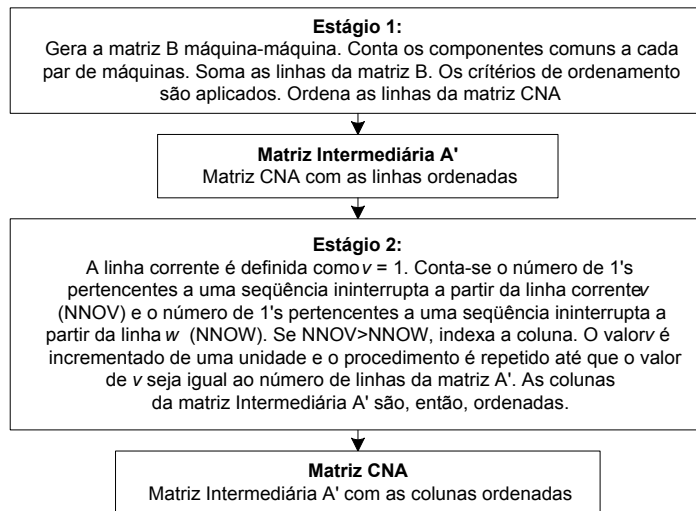


Figura 7.14 – Seqüência da rotina *07_algorithm_cna*.

Diferente do resultado obtido pelo sistema através da rotina *m06_algorithm_cna_ep*, a rotina *07_algorithm_cna* mostra-se capaz de diagonalizar a matriz *teste_1*, tomada para teste, cujo resultado é ilustrado na Figura 7.15.

	p001_JW	p015_JW	p030_JW	p038_JW	p040_JW	p043_JW	p044_JW	p045_JW	p014_JW	p029_JW	p002_JW	p023_JW	p032_JW	p041_JW	p031_JW	p006_JW	p017_JW	p026_JW	p027_JW	p028_JW	p004_JW	p005_JW	p024_JW	p033_JW	p039_JW	p009_JW	p008_JW	p016_JW	p025_JW	p034_JW	p035_JW	p012_JW	p013_JW	p020_JW	p003_JW	p010_JW	p019_JW	p036_JW	p021_JW	p022_JW	p007_JW	p011_JW	p018_JW	p037_JW	p042_JW													
maq006	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1												
maq040	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1												
maq026	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1												
maq038	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1											
maq012	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1											
maq036												1	1	1																																												
maq018												1	1	1	1																																											
maq034												1	1	1	1																																											
maq033												1	1	1	1																																											
maq015												1	1	1	1																																											
maq025												1	1	1	1																																											
maq005																1	1	1	1	1	1																																					
maq022																1	1	1	1	1	1																																					
maq037																1	1	1	1	1	1																																					
maq008																																																										
maq023																																																										
maq039																																																										
maq019																																																										
maq007																																																										
maq003																																																										
maq032																																																										
maq001																																																										
maq027																																																										
maq024																																																										
maq029																																																										
maq016	1																																																									
maq021																																																										
maq002																																																										
maq010																																																										
maq030																																																										
maq004																																																										
maq009																																																										
maq020																																																										
maq017																																																										
maq014																																																										
maq035																																																										
maq013																																																										
maq011																																																										
maq028																																																										
maq031																																																										

Figura 7.15 – Matriz *teste_1* ordenada pela rotina *m07_algorithm_cna*.

A matriz ordenada de Wang (1998), a qual não sofre análise de restrições, é representada simbolicamente sem a identificação das máquinas e peças e por esse motivo não ficam claros os agrupamentos. Porém esta matriz resposta é bastante semelhante à apresentada na Figura 7.15.

Os resultados diretos dos algoritmos, no entanto, devem ser refinados, através de restrições de projeto para um efetivo suporte ao processo de implementação de um sistema de fabricação

celular. No sistema implementado estão inseridos procedimentos de análise para refinamento dos resultados nos aspectos referentes a uma efetiva interdependência das células, através da delimitação de suas fronteiras na matriz resultado; na imposição de um número máximo de máquinas por célula, bem como na limitação do carregamento por máquina. Essas restrições são tratadas separadamente no sistema conforme ilustrado na respectiva análise descrita a seguir.

Análise das fronteiras da célula

A definição das fronteiras de cada célula normalmente é realizada por intervenção de um analista em fase posterior ao ordenamento decorrentes da aplicação dos algoritmos sobre as matrizes de incidência. Porém para viabilizar uma análise otimizada de um grande número de opções de rotas é desejável que isto seja feito automaticamente.

Baker e Maropoulos (1997) propõem, em seu algoritmo *Black Box Clustering*, um procedimento para determinar as fronteiras das células, definição realizada pela identificação de cantos candidatos a limite da célula, proposta conforme ilustração da Figura 7.16.

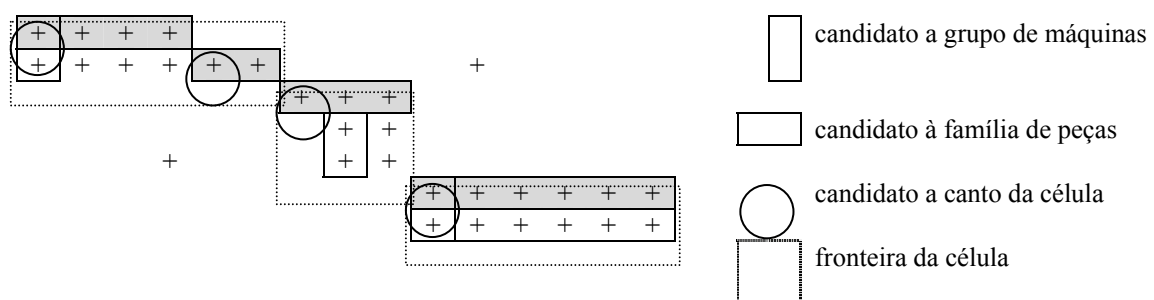


Figura 7.16 – Células candidatas a início de famílias de peças e grupos de máquinas.

Fonte: Baker e Maropoulos (1997)

Apesar deste procedimento apresentar resultados convincentes, não é tão simples sua implementação computacional. Uma proposta de delimitação de fronteiras de células independentes, semelhante, é apresentada pela análise das seguintes etapas:

1. definir o valor da célula como 1;
2. procurar a última incidência na linha corrente, a partir do canto superior esquerdo da matriz diagonalizada, e definir a fronteira com o número da coluna;
3. repetir o procedimento para a linha seguinte;
4. caso a primeira incidência ocorra fora do limite estabelecido, incrementar o valor da célula em uma unidade e retornar ao passo 2.

Assim, buscando definir as fronteiras das células independentes nas matrizes das Figuras 7.11 e 7.15, diagonalizadas respectivamente pelos algoritmos *ROC* e *CNA*, obtém-se os resultados das Figuras 7.17 e 7.18.

Análise para limitação do tamanho da célula

Esta restrição está inserida nas rotinas *m09_analise_roc* e *m10_analise_cna*, responsáveis pela aplicação das restrições para os resultados obtidos pelos algoritmos *ROC* e *CNA*.

O tamanho da célula é definido pelo *número de máquinas por célula* do formulário *f_disponibilidades* ou pelo *limite máq/célula* do formulário *matriz*, vinculados ao campo *maq_por_grupo* da tabela *t_disponibilidade*.

Para aplicar a restrição sobre uma matriz, é identificado o valor que define o número limite de máquinas por célula e verificado, para cada coluna e a partir do canto superior esquerdo, se o número de máquinas está dentro do intervalo estabelecido. Identificando-se uma incidência fora do limite predefinido, é efetuada a exclusão da peça. Quando efetuada a exclusão de uma peça procede-se a verificação e eliminação de linhas em branco. Os limites impostos para a célula levam em consideração, além do tamanho da célula, o índice que identifica a que grupo a máquina ou a peça pertence. Isso evita a exclusão de um número maior de peças uma vez que torna mais flexível a definição das fronteiras da célula.

A Figura 7.19 apresenta o resultado da aplicação da restrição limitando o tamanho da célula em 5 máquinas para a matriz processada por *ROC* (Figura 7.17). Nota-se que foram excluídas 25 peças e configuradas 5 células na matriz reordenada com o algoritmo *ROC*.

	p001_JW	p030_JW	p043_JW	p045_JW	p040_JW	p029_JW	p007_JW	p011_JW	p018_JW	p042_JW	p017_JW	p008_JW	p016_JW	p025_JW	p035_JW	p004_JW	p005_JW	p024_JW	p009_JW	p033_JW	
maq038	1	1	1	1	1	1															
maq040	1	1	1	1	1																
maq006	1	1	1	1	1	1															
maq012	1	1	1	1		1															
maq026	1	1	1	1	1	1															
maq013							2	2	2	2											
maq011							2	2	2	2											
maq008																				3	
maq005																				3	
maq022																				3	
maq037																				3	
maq039																				3	
maq027													4	4	4	4					
maq024													4	4	4	4					
maq029													4	4	4	4					
maq007																	5	5	5	5	5
maq032																	5	5	5	5	
maq003																	5	5	5	5	
maq001																	5	5	5	5	

Figura 7.19 – Aplicação da restrição sobre a matriz *teste_1*.

A Figura 7.20 apresenta o resultado da aplicação da restrição limitando o tamanho da célula em 5 máquinas para a matriz processada por *CNA* (Figura 7.18). Comparando-se os resultados das matrizes rearranjadas pelos algoritmos *ROC* e *CNA*, nota-se um número menor de exclusões de peças na matriz processada pelo *CNA*, 18 no total, configurando 7 células.

	p001_JW	p030_JW	p040_JW	p043_JW	p045_JW	p029_JW	p023_JW	p027_JW	p004_JW	p005_JW	p024_JW	p033_JW	p039_JW	p009_JW	p013_JW	p020_JW	p003_JW	p010_JW	p019_JW	p036_JW	p021_JW	p022_JW	p007_JW	p011_JW	p018_JW	p037_JW	p042_JW	
maq006	1	1	1	1	1	1																						
maq040	1	1	1	1	1																							
maq026	1	1	1	1	1	1																						
maq038	1	1	1	1	1	1																						
maq012	1	1		1	1	1																						
maq036							2																					
maq018							2																					
maq034							2																					
maq033							2																					
maq015							2																					
maq005								3																				
maq022								3																				
maq037								3																				
maq008								3																				
maq023								3																				
maq007									4	4	4	4	4	4	4													
maq003									4	4	4	4	4	4														
maq032									4	4	4		4	4														
maq001									4	4	4	4		4	4													
maq027													4															
maq016																5												
maq021																5												
maq002																5												
maq010																5												
maq030																	6											
maq004																	6	6	6	6	6							
maq009																	6	6	6	6	6							
maq020																	6	6	6	6								
maq017																					7	7				7		
maq014																					7	7						
maq035																					7	7						
maq013																								7	7	7	7	7
maq011																								7	7	7	7	7

Figura 7.20 – Aplicação da restrição sobre a matriz *teste_1*.

Se aplicado o algoritmo *ROC* após cada exclusão de uma peça, eliminando-se linhas vazias quando ocorrerem, obtém-se um resultado diferente do obtido anteriormente (Figura 7.19). A Figura 7.21 apresenta a nova matriz ordenada, levando em consideração o critério adotado acima. Neste caso foram identificados automaticamente 8 grupos independentes e excluídas 18 peças, ou seja, nesta configuração é possível agrupar 8 peças a mais do que na configuração exibida pela Figura 7.19.

Esse procedimento é uma exceção à regra. Se executado o *CNA* após a eliminação de uma peça, toda a matriz novamente, inclusive as peças que já foram processadas pela restrição, são novamente reordenadas, piorando o resultado. O algoritmo *ROC* não modifica a ordem da peça já processada pela restrição quando uma outra peça é excluída e a matriz rearranjada, porque mantém à esquerda e à cima os itens que possuem o maior valor da ponderação.

Definida a fronteira esquerda, limites inferior e superior, falta definir o limite à direita de cada célula independente. Adicionamos, então, a restrição de carregamento da máquina.

	p001_JW	p030_JW	p043_JW	p045_JW	p040_JW	p029_JW	p032_JW	p037_JW	p007_JW	p011_JW	p018_JW	p042_JW	p013_JW	p020_JW	p003_JW	p010_JW	p019_JW	p036_JW	p017_JW	p008_JW	p016_JW	p025_JW	p035_JW	p004_JW	p005_JW	p024_JW	p009_JW	p033_JW	
maq026	1	1	1	1	1	1																							
maq006	1	1	1	1	1	1																							
maq038	1	1	1	1	1	1																							
maq040	1	1	1	1	1																								
maq012	1	1	1	1		1																							
maq014							2																						
maq018							2																						
maq034							2																						
maq036							2																						
maq013								3	3	3	3	3																	
maq011								3	3	3	3	3																	
maq017								3																					
maq021													4																
maq016													4																
maq002													4																
maq010													4																
maq004														5	5	5	5	5											
maq009														5	5	5	5	5											
maq030														5															
maq020															5	5	5	5											
maq005																												6	
maq022																												6	
maq037																												6	
maq039																												6	
maq008																												6	
maq027																													7
maq024																													7
maq029																													7
maq007																													8
maq001																													8
maq032																													8
maq003																													8

Figura 7.21 – Aplicação da restrição e ordenamento *ROC* sobre a matriz *teste_1*.

Análise para verificação do carregamento

Esta restrição também está inserida nas rotinas *m09_analise_roc* e *m10_analise_cna*, responsáveis pela aplicação das restrições para os resultados obtidos pelos algoritmos *ROC* e *CNA*.

Para sua aplicação, faz-se necessário o preenchimento do formulário *f_disponibilidade* e dos campos *tamanho de lote de fabricação* no formulário *f_pecas*, *coeficiente de utilização* no formulário *f_maquinas*, e *tempo de processamento* nos formulários *f_rotas* e *f_rotas sub*. Assim, criada a matriz incidência peça-máquina, reordena-se com os algoritmos *ROC* e *CNA*, aplica-se a restrição de tamanho da célula, com a limitação no tempo de processamento e a condição para duplicação das máquinas gargalo.

A restrição é aplicada obedecendo a seguinte seqüência de passos:

1. somar o tempo necessário para cada máquina processar todas as peças nela alocada;
2. se a soma ultrapassar o tempo de duplicação, a linha é assinalada com “duplicar” indicando que aquela máquina deve ser duplicada e o novo tempo limite passa a ser o dobro do limite de carregamento; se a soma for maior que o tempo limite e menor que o tempo para

duplicação, assinalar a linha com “excluir” informando a necessidade de excluir uma peça para que a soma do tempo seja menor que o tempo limite de carregamento; se a soma dos tempos for menor que o tempo limite de processamento, assinalar com “ok” informando que a linha satisfaz a restrição;

3. aplicar o critério de exclusão e eliminar uma peça;
4. repetir o procedimento a partir do primeiro passo.

Por exemplo, definindo-se como 3 o *número de turnos de trabalho de uma jornada*, 480 minutos como *tempo de um turno de trabalho* e 1 dia como *período de tempo considerado* tem-se 1440 minutos como *tempo limite de carregamento* (equação 10). Se o tempo de carregamento exceder em 25% o tempo limite, ou seja, 1800 minutos como *limite de carregamento para duplicação* (equação 11), duplica-se a máquina gargalo. A Figura 7.22 ilustra a aplicação da restrição sobre as primeiras 16 linhas e 13 colunas da matriz incidência reordenada com o algoritmo ROC ilustrada na Figura 7.21.

	p001_JW	p030_JW	p043_JW	p045_JW	p040_JW	p029_JW	p032_JW	p037_JW	p007_JW	p011_JW	p018_JW	p042_JW	p013_JW	soma	
maq026	625	62,5	62,5	62,5	62,5	62,5								937,5	ok
maq006	625	62,5	125	62,5	62,5	62,5								1000	ok
maq038	625	62,5	62,5	62,5	62,5	62,5								937,5	ok
maq040	625	62,5	62,5	62,5	62,5									875	ok
maq012	625	62,5	62,5	62,5		62,5								875	ok
maq014							62,5							62,5	ok
maq018							62,5							62,5	ok
maq034							50							50	ok
maq036							50							50	ok
maq013								375	375	412,5	675	125		1962,5	duplicar
maq011								375	375	137,5	675	125		1687,5	excluir
maq017								375						375	ok
maq021													130	130	ok
maq016													162,5	162,5	ok
maq002													162,5	162,5	ok
maq010													162,5	162,5	ok

Figura 7.22 – Aplicação da restrição sobre parte da matriz teste_1.

Vários critérios de exclusão de peças podem ser adotados para maximizar a utilização dos recursos de produção dentro da célula: minimização dos custos de transporte ou das distâncias entre máquinas, rejeição das peças de menor custo de fabricação para que estas sejam manufaturadas por terceiros ou em um outro sistema de *layout*, minimização de investimentos em equipamentos, entre outros. Adota-se como critério excluir a peça que ultrapassar, na ordem de disposição das colunas, o tempo limite de carregamento. Mesmo o sistema apresentando o campo *custo_hora_maquina*, na tabela *t_maquinas*, e calculando o *custo_rota*, no formulário *f_rotas*, para essa análise optou-se apenas pelo critério de exclusão da peça que ultrapassar, na ordem de disposição das colunas, o tempo limite de carregamento. Este critério é adotado pois

não necessita a adição de mais uma camada de dados na matriz incidência ($K=3$), responsável pelo armazenamento dos custos envolvidos no processamento das peças, tornando a análise simplificada. Assim, no exemplo ilustrado, percorrendo a linha da máquina *maq011*, a qual indica a necessidade de eliminação de uma peça, identificando-se para exclusão a peça *p018_JW*. Verifica-se, a seguir, sobre a matriz da Figura 7.22, a soma e a aplicação das restrições obtendo a seguinte matriz.

	p001_JW	p030_JW	p043_JW	p045_JW	p040_JW	p029_JW	p032_JW	p037_JW	p007_JW	p011_JW	p042_JW	p013_JW	soma	
maq026	625	62,5	62,5	62,5	62,5	62,5							937,5	ok
maq006	625	62,5	125	62,5	62,5	62,5							1000	ok
maq038	625	62,5	62,5	62,5	62,5	62,5							937,5	ok
maq040	625	62,5	62,5	62,5	62,5								875	ok
maq012	625	62,5	62,5	62,5		62,5							875	ok
maq014							62,5						62,5	ok
maq018							62,5						62,5	ok
maq034							50						50	ok
maq036							50						50	ok
maq013								375	375	412,5	125		1287,5	ok
maq011								375	375	137,5	125		1012,5	ok
maq017								375					375	ok
maq021												130	130	ok
maq016												162,5	162,5	ok
maq002												162,5	162,5	ok
maq010												162,5	162,5	ok

Figura 7.23 – Aplicação da restrição sobre parte da matriz *teste_1*.

No caso da matriz reordenada pelo algoritmo *CNA*, há a necessidade de excluir duas peças: *p037_JW* e *p018_JW*. A Figura 7.24 mostra as últimas linhas e colunas da matriz rearranjada indicando a necessidade de excluir uma peça na linha da máquina *maq011*. Após a remoção da peça *p037_JW*, como mostra a Figura 7.25, é a máquina *maq013* que indica a necessidade de excluir uma peça: *p018_JW*. Rejeitadas essas duas peças, as demais células estão dentro dos limites pré-estabelecidos pelas restrições, como mostra a Figura 7.26.

	p020_JW	p003_JW	p010_JW	p019_JW	p036_JW	p021_JW	p022_JW	p007_JW	p011_JW	p018_JW	p037_JW	p042_JW	soma	
maq030	200												200	ok
maq004	250	125	125	237,5	62,5								800	ok
maq009	250,03	125	125	237,5	62,5								800,03	ok
maq020		125	125	237,5	62,5								550	ok
maq017						25	25				375		425	ok
maq014						25	25						50	ok
maq035						20	20						40	ok
maq013								375	412,5	675	375	125	1962,5	duplicar
maq011								375	137,5	675	375	125	1687,5	excluir

Figura 7.24 – Aplicação da restrição sobre parte da matriz *teste_1*.

	p020_JW	p003_JW	p010_JW	p019_JW	p036_JW	p021_JW	p022_JW	p007_JW	p011_JW	p018_JW	p042_JW	soma	
maq030	200											200	ok
maq004	250	125	125	237,5	62,5							800	ok
maq009	250,03	125	125	237,5	62,5							800,03	ok
maq020		125	125	237,5	62,5							550	ok
maq017						25	25					50	ok
maq014						25	25					50	ok
maq035						20	20					40	ok
maq013								375	412,5	675	125	1587,5	excluir
maq011								375	137,5	675	125	1312,5	ok

Figura 7.25 – Aplicação da restrição sobre parte da matriz *teste_1*.

	p020_JW	p003_JW	p010_JW	p019_JW	p036_JW	p021_JW	p022_JW	p007_JW	p011_JW	p042_JW	soma	
maq030	200										200	ok
maq004	250	125	125	237,5	62,5						800	ok
maq009	250,03	125	125	237,5	62,5						800,03	ok
maq020		125	125	237,5	62,5						550	ok
maq017						25	25				50	ok
maq014						25	25				50	ok
maq035						20	20				40	ok
maq013								375	412,5	125	912,5	ok
maq011								375	137,5	125	637,5	ok

Figura 7.26 – Aplicação da restrição sobre parte da matriz *teste_1*.

As rotinas *m09_analise_roc* e *m10_analise_cna* tornam-se, então, responsáveis pela aplicação das restrições de tamanho da célula, limite de carregamento e limite para duplicação da máquina gargalo para as matrizes rearranjadas pelos algoritmos *Rank Order Clustering* e *Close Neighbour Algorithm*.

7.2.3.2 Procedimento heurístico (HEU)

A metodologia de geração de agrupamento através do algoritmo *HEU* é implementado no sistema pela rotina *m11_algoritmo_heuristico* que rearranja as linhas e as colunas da matriz incidência ao mesmo tempo em que aplica as restrições de projeto da célula, diferentemente dos outros procedimentos.

Como descrito no capítulo 4.2.3, define-se na matriz incidência uma linha horizontal escolhida arbitrariamente ou por um critério qualquer para iniciar o processo de agrupamento. Neste trabalho adotou-se como critério de início a maior soma de peças por linha, ou seja, o maior número de peças exigindo ser processadas por uma determinada máquina. A seguir são apresentadas as etapas de reordenamento das linhas e colunas para o algoritmo *HEU*.

1. somar o número de peças que exigem ser manufaturadas para cada máquina;
2. definir o número da célula e o número da linha como sendo igual a 1;

3. definir como linha inicial a ser ordenada aquela que apresentar o maior número de incidências;
4. para cada entrada na matriz em que $heurístico(I, J, 0)$ for igual a 1, seleccionar a coluna e contar o número de máquinas que a respectiva peça exigir; se o número de máquinas for superior ao limite imposto pela restrição de tamanho de célula, a peça é expurgada; para cada coluna e para cada linha acrescentada ao agrupamento, incrementar o valor que indexa a ordem das colunas ou das linhas em uma unidade;
5. constatar a existência de elementos com cruzamentos nas máquinas já verificadas. Em caso negativo, incrementar o valor da célula e passar para a etapa 3, ou não havendo mais elementos a serem verificados, passar para a etapa seguinte;
6. aplicar a restrição do limite de carregamento.

Aplicando o procedimento *HEU* sobre a matriz incidência *teste_1*, tomada como matriz teste, com tamanho de célula igual a 5, limite de carregamento de 1440 minutos e limite para duplicação de 1800 minutos, obtém-se a matriz da Figura 7.27. Neste caso foram rejeitadas 20 peças das 45 iniciais e configurados 7 agrupamentos celulares. Note-se que é possível formar duas células distintas a partir da célula 3, o que indica que este procedimento não identifica células totalmente independentes.

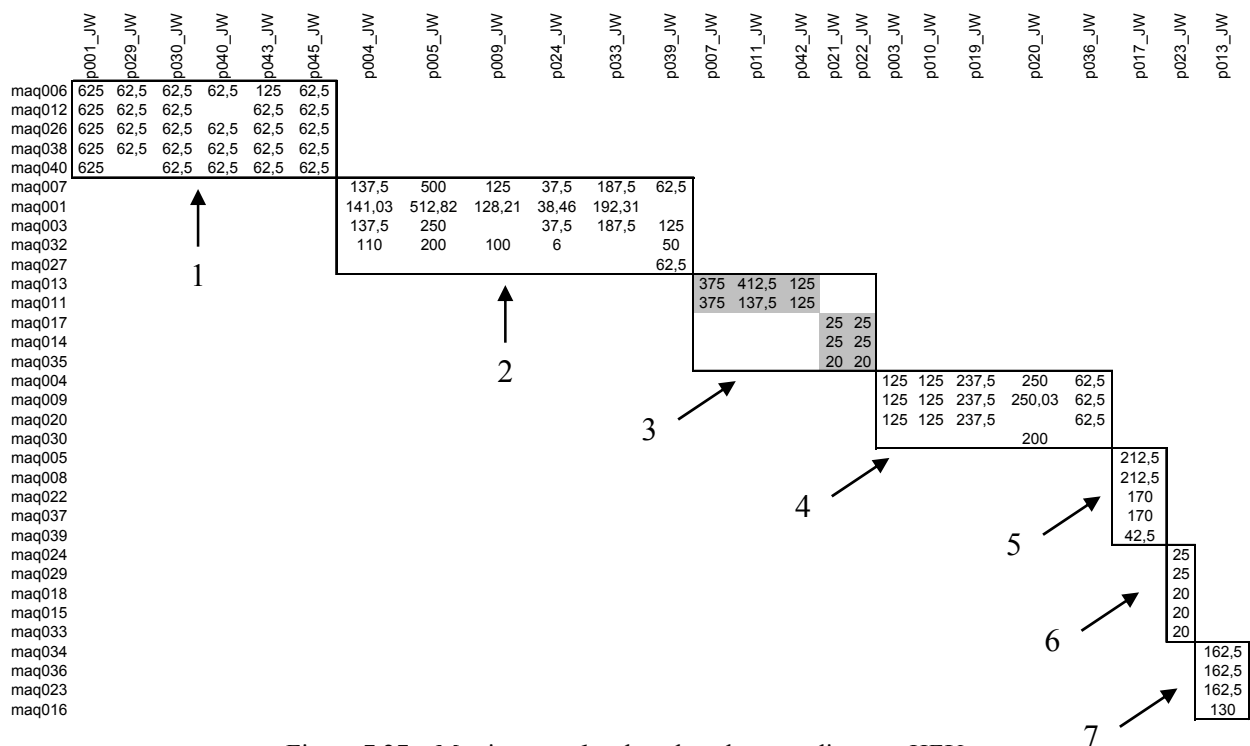


Figura 7.27 – Matriz *teste_1* ordenada pelo procedimento *HEU*.

7.2.3.3 Apresentação dos resultados

Os resultados finais de processamento e análise para os diversos algoritmos aplicados são apresentados através da rotina *m12_exporta_para_excel*, indicando os resultados em arquivo *MS Excel* com onze planilhas. As opções *binária*, *tempos* e *células* permitem a forma de visualização e de exportação da matriz para o arquivo.

A primeira planilha, chamada *resultados*, exhibe os valores das restrições de projeto das células (número de máquinas por célula, tempo limite de carregamento e tempo limite para duplicação da máquina), o número de peças (colunas) e de máquinas (linhas) da matriz incidência, os valores de eficiência do agrupamento de peças, do agrupamento de máquinas e do aproveitamento dos recursos, bem como o número de células independentes e a quantidade de máquinas duplicadas, obtidos por cada algoritmo no sistema, destacando os melhores resultados.

Assim, para a matriz *teste_1*, o sistema gera a Tabela 7.1, interpretada da seguinte maneira:

- referente às restrições de projeto, impõe que:
 - uma célula independente contenha um número igual ou inferior a cinco máquinas;
 - uma peça deve ser eliminada quando seu tempo de processamento tornar uma máquina gargalo com tempo de carregamento superior a 1440 minutos e inferior a 1800 minutos;
 - uma máquina gargalo deve ser duplicada se esta necessitar de tempo superior a 1800 minutos para processar todas as peças a ela associadas;
- referente à matriz incidência:
 - dos dados processados para peças, máquinas e rotas ativas cadastradas, o sistema gerou uma matriz incidência com 45 peças (colunas) e 40 máquinas (linhas);
- referente à análise dos algoritmos:
 - o algoritmo *ROC* agrupa 27 das 45 peças da matriz incidência original associando em famílias 60,00% das peças da matriz incidência enquanto os algoritmos *CNA* e *HEU* agrupam, cada um, 25 das 45 peças associando somente 55,56% das peças. Como o primeiro critério para a escolha do melhor algoritmo de agrupamento é o maior número de peças associadas em famílias, o *ROC* é a melhor escolha entre os algoritmos implementados para esta matriz e a linha que possui o maior percentual de agrupamento recebe a anotação “*maior aproveitamento de peças*”;
 - o algoritmo *ROC* associa 32 máquinas para processar as 27 peças agrupadas, ou seja, agrupa 80% das máquinas das rotas ativas e apresenta uma relação do número de peças agrupadas sobre o número de máquinas necessárias para processá-las igual

a 84,38% (27 peças/32 máquinas). Já os algoritmos *CNA* e *HEU* associam, cada um, 33 máquinas para processar 25 peças ou 82,5% das máquinas ativas apresentando uma relação peças/máquinas igual a 75,76%. O segundo critério para definir o melhor algoritmo é a relação peças/máquinas e, para esta matriz, o algoritmo ROC continua sendo a melhor escolha recebendo a anotação “melhor aproveitamento dos recursos”;

- os algoritmos ROC e CNA identificam 8 células independentes enquanto o arranjo heurístico identifica 7;
- nenhum dos algoritmos implicou em duplicação de máquinas.

Tabela 7.1 – Planilha *resultados*.

Resultados	6:12:31 PM
	29/08/2005
Restrições de projeto:	
5 máquinas/célula	
1440 minutos - limite de carregamento	
1800 minutos - limite p/ duplicação	
Matriz incidência	
45 peças	
40 máquinas	
análise ROC	
27	60,00% peças
32	80,00% máquinas
	84,38% relação peças/máquinas
	8 células independentes
	nenhuma máquina duplicada
	maior aproveitamento de peças
	melhor aproveitamento dos recursos
análise CNA	
25	55,56% peças
33	82,50% máquinas
	75,76% relação peças/máquinas
	8 células independentes
	nenhuma máquina duplicada
Arranjo heurístico	
25	55,56% peças
33	82,50% máquinas
	75,76% relação peças/máquinas
	7 células independentes
	nenhuma máquina duplicada

A segunda planilha mostra a matriz incidência peça-máquina, no caso a *teste_1*.

As planilhas *ROC*, *análise ROC* e *relatório ROC* apresentam, respectivamente, a matriz rearranjada pelo algoritmo na sua concepção original (sem restrições), a matriz reordenada e aplicadas as restrições e, por fim, o relatório com os dados já apresentados na planilha *resultados* e com listas de famílias de peças, células de máquinas agrupadas, lista de peças não agrupadas e máquinas duplicadas. A Tabela 7.2 é a planilha *relatório ROC* obtida nesta análise.

A planilha seguinte apresenta a matriz reordenada pelo algoritmo *CNA EP*. Este algoritmo não é comparado aos outros e é ilustrado somente para efeito informativo.

As demais planilhas, *CNA*, *análise CNA*, *relatório CNA*, *heurístico* e *relatório heurístico*, apresentam as matrizes e os agrupamentos obtidos pelos algoritmos *CNA* e *HEU*.

Tabela 7.2 – Planilha *relatório ROC*.

Resultados ROC							6:12:35 PM
							29/08/2005
Restrições de projeto:							
5 máquinas/célula							
1440 minutos - limite de carregamento							
1800 minutos - limite p/ duplicação							
Matriz incidência							
45 peças							
40 máquinas							
análise ROC							
27 60,00% peças							
32 80,00% máquinas							
84,38% relação peças/máquinas							
8 células independentes							
nenhuma máquina duplicada							
família 1 (6):	p001_JW	p030_JW	p043_JW	p045_JW	p040_JW	p029_JW	
família 2 (1):	p032_JW						
família 3 (4):	p037_JW	p007_JW	p011_JW	p042_JW			
família 4 (1):	p013_JW						
família 5 (5):	p020_JW	p003_JW	p010_JW	p019_JW	p036_JW		
família 6 (1):	p017_JW						
família 7 (4):	p008_JW	p016_JW	p025_JW	p035_JW			
família 8 (5):	p004_JW	p005_JW	p024_JW	p009_JW	p033_JW		
célula 1 (5):	maq026	maq006	maq038	maq040	maq012		
célula 2 (4):	maq014	maq018	maq034	maq036			
célula 3 (3):	maq013	maq011	maq017				
célula 4 (4):	maq021	maq016	maq002	maq010			
célula 5 (4):	maq004	maq009	maq030	maq020			
célula 6 (5):	maq005	maq022	maq037	maq039	maq008		
célula 7 (3):	maq027	maq024	maq029				
célula 8 (4):	maq007	maq001	maq032	maq003			
18 peças não agrupadas:	p002_JW	p006_JW	p012_JW	p014_JW	p015_JW	p018_JW	
	p021_JW	p022_JW	p023_JW	p026_JW	p027_JW	p028_JW	
	p031_JW	p034_JW	p038_JW	p039_JW	p041_JW	p044_JW	
nenhuma máquina duplicada							

As planilhas *relatórios* são montadas a partir da comparação entre as matrizes rearranjadas, sem as linhas e colunas excluídas, e a matriz incidência.

Adotou-se o critério de melhor solução para o arranjo que obtiver o maior número de peças agrupadas e o maior número de peças fabricadas com o menor número de máquinas agrupadas.

8 TESTES EXPERIMENTAIS – VALIDAÇÃO DO SISTEMA

Com o intuito de validar o funcionamento do aplicativo realizaram-se diversos testes a partir de matrizes obtidas na literatura. Essas matrizes são processadas para os algoritmos implementados no sistema, inicialmente sem a imposição de restrições na tentativa de obter agrupamentos distintos na matriz incidência. Para os casos onde não seja constatada diretamente a formação de células totalmente independentes, são aplicados critérios de análise como a limitação no tamanho da célula.

A aplicação da análise de carregamento que eliminaria a peça conforme sua disposição na matriz não é avaliada como critério para verificação do melhor resultado de cada algoritmo. Assim, somente o tamanho da célula é adotado como restrição de projeto da célula para comparação entre os procedimentos de agrupamento. Para efeito de avaliação são utilizadas duas matrizes identificadas como matriz *teste_2* e *teste_3*, analisadas a seguir.

8.1 Matriz *teste_2*

A matriz *teste_2*, de Lee *et al.* (1997), tem dimensões correspondente a 30 máquinas e 40 peças sobre a qual são comentados os resultados relativos aos algoritmos aplicados.

	p001_LLA	p002_LLA	p003_LLA	p004_LLA	p005_LLA	p006_LLA	p007_LLA	p008_LLA	p009_LLA	p010_LLA	p011_LLA	p012_LLA	p013_LLA	p014_LLA	p015_LLA	p016_LLA	p017_LLA	p018_LLA	p019_LLA	p020_LLA	p021_LLA	p022_LLA	p023_LLA	p024_LLA	p025_LLA	p026_LLA	p027_LLA	p028_LLA	p029_LLA	p030_LLA	p031_LLA	p032_LLA	p033_LLA	p034_LLA	p035_LLA	p036_LLA	p037_LLA	p038_LLA	p039_LLA	p040_LLA				
maq001				1																																								
maq002				1											1		1											1	1															
maq003							1																						1			1												
maq004							1																						1			1												
maq005							1																																					
maq006									1												1						1																	
maq007				1				1														1				1																		
maq008	1											1						1							1																			
maq009	1										1		1			1		1																										
maq010		1																			1						1																	
maq011									1					1																														
maq012		1	1							1		1															1	1																
maq013		1	1							1		1									1						1																	
maq014									1													1						1																
maq015	1										1							1																										
maq016	1										1							1																										
maq017		1	1							1		1									1							1																
maq018									1													1						1																
maq019	1																																											
maq020	1																																											
maq021																																												
maq022				1																																								
maq023																																												
maq024				1																																								
maq025																																												
maq026																																												
maq027																																												
maq028																																												
maq029																																												
maq030																																												

Figura 8.1 – Matriz *teste_2*.
Fonte: Lee *et al.* (1997).

Procedendo-se com o ordenamento através do algoritmo *ROC* obtém-se a matriz ilustrada na Figura 8.2. Observa-se a formação natural de células totalmente independentes sem a imposição de restrições no projeto da célula.

	p009_LLA	p027_LLA	p021_LLA	p039_LLA	p024_LLA	p014_LLA	p018_LLA	p001_LLA	p013_LLA	p035_LLA	p016_LLA	p011_LLA	p002_LLA	p031_LLA	p020_LLA	p026_LLA	p003_LLA	p010_LLA	p012_LLA	p022_LLA	p029_LLA	p023_LLA	p015_LLA	p004_LLA	p017_LLA	p019_LLA	p028_LLA	p025_LLA	p008_LLA	p005_LLA	p033_LLA	p038_LLA	p030_LLA	p040_LLA	p006_LLA	p007_LLA	p032_LLA	p037_LLA	p034_LLA	p036_LLA						
maq006	1																																													
maq014	1	1																																												
maq018	1	1	1																																											
maq011	1	1		1		1																																								
maq008							1	1	1																																					
maq019							1	1	1	1																																				
maq020							1	1	1	1	1																																			
maq009							1	1	1	1	1	1																																		
maq015							1	1	1	1	1	1																																		
maq016							1	1	1	1	1	1																																		
maq010													1	1	1	1	1																													
maq012													1	1	1	1	1	1																												
maq013													1	1	1	1	1	1	1																											
maq017													1	1	1	1	1	1	1																											
maq021																					1																									
maq001																				1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1			
maq022																				1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1			
maq024																				1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1			
maq002																				1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1			
maq007																																														
maq030																																														
maq026																																														
maq028																																														
maq029																																														
maq005																																														
maq023																																														
maq004																																														
maq027																																														
maq025																																														
maq003																																														

Figura 8.6 – Matriz ordenada de Lee *et al.*

8.2 Matriz teste_3

A matriz *teste_3*, de Boe e Cheng (1991), com diferentes características de adensamento inicial e limitada a 20 máquinas e 35 peças analisada pelo sistema é ilustrada a seguir.

	p001_B&C2	p002_B&C2	p003_B&C2	p004_B&C2	p005_B&C2	p006_B&C2	p007_B&C2	p008_B&C2	p009_B&C2	p010_B&C2	p011_B&C2	p012_B&C2	p013_B&C2	p014_B&C2	p015_B&C2	p016_B&C2	p017_B&C2	p018_B&C2	p019_B&C2	p020_B&C2	p021_B&C2	p022_B&C2	p023_B&C2	p024_B&C2	p025_B&C2	p026_B&C2	p027_B&C2	p028_B&C2	p029_B&C2	p030_B&C2	p031_B&C2	p032_B&C2	p033_B&C2	p034_B&C2	p035_B&C2												
maq001	1																																														
maq002		1																																													
maq003	1		1		1					1		1	1		1			1	1																												
maq004		1					1					1	1											1	1																						
maq005					1									1			1										1																				
maq006							1																																								
maq007	1		1		1		1					1	1		1		1		1	1							1			1	1	1	1	1	1	1	1	1	1	1	1	1	1	1			
maq008	1							1						1		1		1		1																											
maq009									1						1																																
maq010								1								1																															
maq011				1		1			1							1																															
maq012				1		1			1																																						
maq013		1										1	1																																		
maq014		1						1			1	1																																			
maq015				1		1			1																																						
maq016				1		1			1																																						
maq017	1		1		1									1		1																															
maq018		1											1	1																																	
maq019				1				1																																							
maq020														1																																	

Figura 8.7 – Matriz *teste_3*.
Fonte: Boe e Cheng (1991).

Executando o procedimento *ROC* na matriz *teste_3*, sem a imposição de restrições, obtém-se:

	p001_B&C2	p015_B&C2	p005_B&C2	p003_B&C2	p020_B&C2	p017_B&C2	p031_B&C2	p029_B&C2	p024_B&C2	p030_B&C2	p026_B&C2	p012_B&C2	p007_B&C2	p019_B&C2	p022_B&C2	p025_B&C2	p035_B&C2	p023_B&C2	p009_B&C2	p014_B&C2	p018_B&C2	p032_B&C2	p006_B&C2	p011_B&C2	p034_B&C2	p002_B&C2	p013_B&C2	p027_B&C2	p010_B&C2	p004_B&C2	p028_B&C2	p016_B&C2	p033_B&C2	p008_B&C2	p021_B&C2		
maq007	1																																				
maq017	1	1														1	1																				
maq003	1	1	1	1				1	1																												
maq008	1	1			1	1										1		1	1																		
maq001	1	1			1				1	1						1	1	1		1	1	1	1	1	1	1											
maq005			1			1					1									1																	
maq018				1			1		1			1	1														1	1									
maq014						1		1	1			1	1									1					1	1	1	1							
maq002						1		1				1		1							1						1	1	1	1							
maq011							1	1								1		1				1	1							1	1	1					
maq004								1				1	1					1									1	1	1								
maq013								1				1															1	1							1		
maq016									1	1	1							1		1	1	1	1							1							
maq006									1					1	1	1		1		1		1													1		
maq019									1										1				1							1	1					1	
maq015									1										1				1	1						1	1					1	
maq010										1				1	1					1									1	1						1	
maq020										1										1	1															1	
maq009														1		1	1				1														1		
maq012																			1				1	1					1							1	

Figura 8.8 – Matriz *teste_3* ordenada com ROC.

A matriz resultante do processamento não apresenta células distintas e sua diagonal principal possui baixa densidade. Trata-se de uma matriz que deve ser tratada sob critérios de análise associados a restrições para gerar grupos distintos. Limitando-se, por exemplo, em 6 máquinas o tamanho da célula obtém-se o resultado ilustrado na Figura 8.9.

	p001_B&C2	p015_B&C2	p003_B&C2	p017_B&C2	p029_B&C2	p034_B&C2	p021_B&C2	p028_B&C2	p016_B&C2	p002_B&C2	p013_B&C2	p027_B&C2	p010_B&C2	p033_B&C2
maq007	1	1	1	1	1									
maq017	1	1	1	1										
maq003	1	1	1		1									
maq008	1	1		1										
maq001	1	1				1								
maq005				1										
maq015							2	2						
maq019							2	2						
maq012							2							
maq011								2	2					
maq010								2						
maq014									3	3	3	3		
maq002									3	3	3	3		
maq004									3	3	3			
maq013									3	3			3	
maq018									3	3				

Figura 8.9 – Matriz *teste_3* ordenada com ROC e tamanho de célula limitado.

Aplicando-se por sua vez o procedimento *CNA*, sem a imposição de restrições, obtém-se como resposta novamente uma matriz que não apresenta células distintas porém sua diagonal principal possui densidade maior que a da matriz rearranjada com o ROC. A Figura 8.10, obtida pelo sistema, corresponde à matriz reordenada de Boe e Cheng (1991). Limitando-se também em 6 máquinas o tamanho da célula obtém-se o resultado ilustrado na Figura 8.11.

	p001_B&C2	p003_B&C2	p015_B&C2	p017_B&C2	p020_B&C2	p035_B&C2	p006_B&C2	p030_B&C2	p032_B&C2	p034_B&C2	p004_B&C2	p009_B&C2	p011_B&C2	p021_B&C2	p028_B&C2	p023_B&C2	p025_B&C2	p008_B&C2	p014_B&C2	p019_B&C2	p026_B&C2	p016_B&C2	p022_B&C2	p005_B&C2	p029_B&C2	p031_B&C2	p010_B&C2	p018_B&C2	p027_B&C2	p007_B&C2	p024_B&C2	p002_B&C2	p012_B&C2	p013_B&C2	p033_B&C2		
maq007	1	1	1	1	1	1																															
maq017	1	1	1	1	1	1																															
maq001	1		1	1	1	1	1	1	1	1			1			1	1		1					1				1									
maq011						1	1	1	1	1												1															
maq016						1	1	1	1	1																											
maq019						1	1	1	1	1																											
maq015						1	1	1	1	1																											
maq012						1																															
maq008	1		1	1	1											1	1																				
maq006							1	1								1	1	1	1	1				1													
maq009						1			1							1	1	1	1	1																	
maq020																1	1	1	1	1																	
maq010																	1	1	1	1	1	1	1														
maq005				1																					1												
maq003	1	1	1																					1	1	1											
maq002																					1							1	1	1	1	1	1	1	1	1	
maq014																												1	1	1	1	1	1	1	1	1	
maq018					1																							1	1	1	1	1	1	1	1	1	
maq004																																					
maq013																																					1

Figura 8.10 – Matriz *teste_3* ordenada com *CNA*.

	p034_B&C2	p029_B&C2	p013_B&C2	p033_B&C2
maq007	1			
maq001	1			
maq003	1			
maq002		2		
maq014		2		
maq018		2		
maq004		2		
maq013		2	2	

Figura 8.11 – Matriz *teste_3* ordenada com *CNA* e tamanho de célula limitado.

A matriz da Figura 8.11 apresenta quatro peças agrupadas em duas famílias. Fazendo-se um descarte manual das peças, obedecendo o limite de máquinas estabelecido, obtém-se a matriz da Figura 8.12.

	p006_B&C2	p034_B&C2	p004_B&C2	p011_B&C2	p021_B&C2	p028_B&C2	p008_B&C2	p010_B&C2	p027_B&C2	p002_B&C2	p013_B&C2	p033_B&C2
maq001	1	1										
maq011	1		1			1						
maq016	1		1									
maq019			1	1	1	1						
maq015	1		1	1	1	1						
maq012	1		1	1	1							
maq006							1					
maq009							1					
maq020							1					
maq010							1					
maq002								1	1	1	1	
maq014								1	1	1	1	
maq018									1	1		
maq004									1	1	1	
maq013										1	1	1

Figura 8.12 – Matriz *teste_3* ordenada com *CNA* e tamanho de célula limitado.

A diferença de resultados é devida ao método de eliminação das peças. Aplicando-se a análise para limitação do tamanho da célula à matriz rearranjada da Figura 8.11 eliminam-se as

peças *p001*, *p003*, *p015* e *p017* e *p020*. A peça *p035* é a próxima a ser excluída porém, antes deveriam ser analisadas aquelas que utilizam a máquina *maq007*, como pode ser observado na Figura 8.13.

	p035_B&C2	p006_B&C2	p030_B&C2	p032_B&C2	p034_B&C2	p004_B&C2	p009_B&C2	p011_B&C2	p021_B&C2	p028_B&C2	p023_B&C2	p025_B&C2	p008_B&C2	p014_B&C2	p019_B&C2	p026_B&C2	p016_B&C2	p022_B&C2	p005_B&C2	p029_B&C2	p031_B&C2	p010_B&C2	p018_B&C2	p027_B&C2	p007_B&C2	p024_B&C2	p002_B&C2	p012_B&C2	p013_B&C2	p033_B&C2
maq007																														
maq017	1											1																		
maq001	1	1	1	1	1			1			1	1		1									1			1				
maq011	1	1	1	1		1	1			1						1										1				
maq016		1	1	1		1	1								1								1						1	
maq019			1	1		1	1	1	1	1																				
maq015		1	1			1	1	1	1	1																				
maq012		1				1	1	1	1																					
maq008							1				1	1																		
maq006			1	1							1	1	1	1	1				1											
maq009	1			1							1	1	1	1																
maq020									1			1	1	1	1	1														
maq010												1	1	1	1	1	1	1												
maq005													1						1											
maq003																			1	1		1								
maq002															1							1	1	1	1	1	1	1	1	1
maq014																						1	1	1	1	1	1	1	1	1
maq018																					1				1	1	1	1	1	1
maq004										1														1	1	1	1	1	1	1
maq013																										1	1	1	1	1

Figura 8.13 – Matriz *teste_3* ordenada com *CNA* e tamanho de célula limitado manualmente.

Como terceiro critério de tratamento a matriz é processada pelo procedimento *HEU* sem restrições, cujo resultado é ilustrado na Figura 8.14. Não são identificadas células independentes e sua diagonal principal possui baixa densidade, semelhante à diagonal obtida pelo arranjo *ROC*. Limitando-se em 6 o tamanho da célula, obtém-se a matriz resposta indicada na Figura 8.15.

	p001_B&C2	p003_B&C2	p005_B&C2	p007_B&C2	p012_B&C2	p015_B&C2	p017_B&C2	p019_B&C2	p020_B&C2	p022_B&C2	p024_B&C2	p026_B&C2	p029_B&C2	p030_B&C2	p031_B&C2	p006_B&C2	p011_B&C2	p014_B&C2	p018_B&C2	p023_B&C2	p025_B&C2	p032_B&C2	p034_B&C2	p035_B&C2	p009_B&C2	p002_B&C2	p013_B&C2	p027_B&C2	p010_B&C2	p033_B&C2	p008_B&C2	p016_B&C2	p028_B&C2	p021_B&C2		
maq007	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1			
maq001	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		
maq003	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		
maq008	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		
maq017	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		
maq005			1									1																								
maq004				1	1							1															1	1	1							
maq014				1	1							1															1	1	1	1						
maq018				1	1			1				1															1	1	1	1						
maq002				1	1			1				1															1	1	1	1						
maq013				1	1			1				1															1	1	1	1						
maq016				1	1			1				1															1	1	1	1						
maq006						1		1				1																								
maq009						1		1				1																								
maq010						1		1				1																								
maq020						1		1				1																								
maq011								1				1																								
maq015								1				1																								
maq019								1				1																								
maq012								1				1																								

Figura 8.14 – Matriz *teste_3* ordenada com *HEU*.

	p001_B&C2	p003_B&C2	p005_B&C2	p029_B&C2	p034_B&C2	p002_B&C2	p010_B&C2	p013_B&C2	p027_B&C2	p033_B&C2	p016_B&C2	p028_B&C2	p021_B&C2
maq007	1	1	1	1									
maq001	1				1								
maq003	1	1	1	1									
maq008	1												
maq017	1	1	1										
maq005		1											
maq002						2	2	2	2				
maq004						2		2	2				
maq013						2	2	2		2			
maq014						2	2	2	2				
maq018						2		2					
maq011										3	3		
maq012												3	
maq015										3	3		
maq019										3	3		
maq010										3			

Figura 8.15 – Matriz *teste_3* ordenada com *HEU* e tamanho de célula limitado.

8.3 Análise comparativa dos resultados

Para comparar os resultados obtidos pelo rearranjo das matrizes *teste_2* (Figura 8.1) e *teste_3* (Figura 8.7), sem considerar os tempos de processo, são montadas tabelas apresentando os aproveitamentos de cada algoritmo.

Para a matriz *teste_2*, com 40 peças e 30 máquinas, não é necessária a imposição das restrições para identificar as células independentes pois os arranjos obtidos não apresentam elementos excepcionais. Neste caso, somente é ilustrada, na Tabela 8.1, a planilha *resultados* apresentando o número de células identificadas e o total aproveitamento das populações de peças e máquinas.

Tabela 8.1 – Planilha *resultados* para a matriz *teste_2*.

análise ROC			
40	100,00%	peças	maior aproveitamento de peças
30	100,00%	máquinas	
	133,33%	relação peças/máquinas	melhor aproveitamento dos recursos
		6 células independentes	
		nenhuma máquina duplicada	
análise CNA			
40	100,00%	peças	maior aproveitamento de peças
30	100,00%	máquinas	
	133,33%	relação peças/máquinas	melhor aproveitamento dos recursos
		6 células independentes	
		nenhuma máquina duplicada	
Arranjo heurístico			
40	100,00%	peças	maior aproveitamento de peças
30	100,00%	máquinas	
	133,33%	relação peças/máquinas	melhor aproveitamento dos recursos
		6 células independentes	
		nenhuma máquina duplicada	

Para a matriz *teste_3*, com 35 peças e 20 máquinas, o número mínimo de máquinas requerido por uma peça é igual a um. Submetendo-se o processamento dos agrupamentos com o tamanho de célula de um até 8 são obtidos os resultados comparativos indicados na Tabela 8.2.

Tabela 8.2 – Resultados comparativos para a matriz teste_3.

máquinas/célula		análise ROC		análise CNA		Arranjo heurístico	
		peças	máquinas	relação peças/máquinas	células independentes	peças	máquinas
1	peças	2	5,71%	2	5,71%	2	5,71%
	máquinas	2	10,00%	2	10,00%	2	10,00%
	relação peças/máquinas		100,00%		100,00%		100,00%
	células independentes		2		2		2
2	peças	5	14,29%	3	8,57%	5	14,29%
	máquinas	8	40,00%	4	20,00%	8	40,00%
	relação peças/máquinas		62,50%		75,00%		62,50%
	células independentes		5		3		5
3	peças	8	22,86%	2	5,71%	8	22,86%
	máquinas	13	65,00%	2	10,00%	13	65,00%
	relação peças/máquinas		61,54%		100,00%		61,54%
	células independentes		6		2		6
4	peças	10	28,57%	2	5,71%	10	28,57%
	máquinas	17	85,00%	2	10,00%	14	70,00%
	relação peças/máquinas		58,82%		100,00%		71,43%
	células independentes		5		2		5
5	peças	13	37,14%	4	11,43%	13	37,14%
	máquinas	18	90,00%	8	40,00%	18	90,00%
	relação peças/máquinas		72,22%		50,00%		72,22%
	células independentes		4		2		4
6	peças	15	42,86%	4	11,43%	15	42,86%
	máquinas	16	80,00%	8	40,00%	16	80,00%
	relação peças/máquinas		93,75%		50,00%		93,75%
	células independentes		3		2		3
7	peças	14	40,00%	8	22,86%	14	40,00%
	máquinas	16	80,00%	11	55,00%	16	80,00%
	relação peças/máquinas		87,50%		72,73%		87,50%
	células independentes		4		2		4
8	peças	15	42,86%	11	31,43%	15	42,86%
	máquinas	17	85,00%	13	65,00%	17	85,00%
	relação peças/máquinas		88,24%		84,62%		88,24%
	células independentes		4		2		4

Observa-se resultados melhores obtidos com os arranjos *ROC* e *HEU* comparados o *CNA*. A diferença de resultados é devida ao método de eliminação das peças quando aplicada a análise das fronteiras da célula para o algoritmo *CNA* (ver capítulo 8.2).

Constata-se, também, que o número de peças agrupadas não é superior a 15 mesmo quando o limite de máquinas por célula for superior a 6. Isso ocorre devido ao grande número de elementos excepcionais.

Adotando-se, como melhor solução, células que possuem limite de 6 máquinas, obtém-se os relatórios de agrupamentos para os arranjos *ROC* e *HEU* ilustrados nas Tabelas 8.3 e 8.4, respectivamente.

Tabela 8.3 – Relatório ROC para a matriz teste_3.

Resultados ROC								2:04:34 PM
								14/09/2005
Restrições de projeto:								
6 máquinas/célula								
análise ROC								
15	42,86%	peças						
16	80,00%	máquinas						
		93,75% relação peças/máquinas						
		3 células independentes						
		nenhuma máquina duplicada						
família 1 (7):	p001	p015	p005	p003	p017	p029	p034	
família 2 (3):	p021	p028	p016					
família 3 (5):	p002	p013	p027	p010	p033			
célula 1 (6):	maq007	maq017	maq003	maq008	maq001	maq005		
célula 2 (5):	maq015	maq019	maq012	maq011	maq010			
célula 3 (5):	maq014	maq002	maq004	maq013	maq018			
20 peças não agrupadas:		p004	p006	p007	p008	p009	p011	
		p012	p014	p018	p019	p020	p022	
		p023	p024	p025	p026	p030	p031	
		p032	p035					

Tabela 8.4 – Relatório heurístico para a matriz teste_3.

Resultados heurístico								2:04:38 PM
								14/09/2005
Restrições de projeto:								
6 máquinas/célula								
heurístico								
15	42,86%	peças						
16	80,00%	máquinas						
		93,75% relação peças/máquinas						
		3 células independentes						
		nenhuma máquina duplicada						
família 1 (7):	p001	p003	p005	p015	p017	p029	p034	
família 2 (5):	p002	p010	p013	p027	p033			
família 3 (3):	p016	p028	p021					
célula 1 (6):	maq007	maq001	maq003	maq008	maq017	maq005		
célula 2 (5):	maq002	maq004	maq013	maq014	maq018			
célula 3 (5):	maq011	maq012	maq015	maq019	maq010			
20 peças não agrupadas:		p004	p006	p007	p008	p009	p011	
		p012	p014	p018	p019	p020	p022	
		p023	p024	p025	p026	p030	p031	
		p032	p035					

Comparando-se as listas das peças não agrupadas para as duas tabelas, é possível verificar que se trata das mesmas 20 peças: *p004, p006, p007, p008, p009, p011, p012, p014, p018, p019, p020, p022, p023, p024, p025, p026, p030, p031, p032* e *p035*.

Considerando-se que esse resultado não satisfaça a necessidade esperada, uma vez que é capaz de associar em famílias somente 15 das 35 peças da população, e que as máquinas agrupadas estão ociosas ou que seja possível a movimentação intercelular, processando-se as peças não agrupadas, obtendo-se, para tamanhos da célula de 4 a 8 máquinas, os resultados apresentados na Tabela 8.5.

Tabela 8.5 – Resultados comparativos para a matriz *teste_3*.

máquinas/célula		análise ROC		análise CNA		Arranjo heurístico	
		peças	máquinas	relação peças/máquinas	células independentes	peças	máquinas
4	peças	3	15,00%	1	5,00%	3	15,00%
	máquinas	12	60,00%	4	20,00%	12	60,00%
	relação peças/máquinas		25,00%		25,00%		25,00%
	células independentes	3		1		3	
5	peças	2	10,00%	1	5,00%	3	15,00%
	máquinas	9	45,00%	4	20,00%	13	65,00%
	relação peças/máquinas		22,22%		25,00%		23,08%
	células independentes	3		1		3	
6	peças	5	25,00%	3	15,00%	4	20,00%
	máquinas	18	90,00%	11	55,00%	12	60,00%
	relação peças/máquinas		27,78%		27,27%		33,33%
	células independentes	3		2		2	
7	peças	6	30,00%	2	10,00%	5	25,00%
	máquinas	13	65,00%	5	25,00%	13	65,00%
	relação peças/máquinas		46,15%		40,00%		38,46%
	células independentes	2		1		2	
8	peças	6	30,00%	3	15,00%	5	25,00%
	máquinas	13	65,00%	8	40,00%	13	65,00%
	relação peças/máquinas		46,15%		37,50%		38,46%
	células independentes	2		1		2	

Com o novo processamento obtém-se o aproveitamento de 6 das 20 peças não agrupadas anteriormente como mostra a Figura 8.16.

	p009	p006	p011	p019	p008	p022
maq012	1	1	1			
maq011	1	1				
maq016	1	1				
maq015	1	1	1			
maq019	1		1			
maq008	1					
maq001		1	1			
maq006				2	2	2
maq010				2	2	2
maq009				2	2	
maq020				2	2	
maq007				2		2
maq002				2		

Figura 8.16 – Agrupamento ROC das peças restantes.

Admitindo-se movimentação intercelular e sobrepondo as matrizes obtidas no primeiro e no segundo agrupamento, obtém-se o resultado ilustrado na Figura 8.17. O primeiro resultado, onde o rearranjo da matriz *teste_3* original, é destacado nos blocos sombreados.

	p001	p003	p005	p015	p017	p029	p034	p019	p008	p022	p016	p021	p028	p009	p006	p011	p002	p010	p013	p027	p033	
maq003	1	1	1	1		1																
maq005			1		1																	
maq017	1	1	1	1	1																	
maq008	1			1	1									2								
maq001	1			1			1								2	2						
maq007	1	1	1	1	1	1		1		1												
maq006								1	1	1												
maq009								1	1													
maq020								1	1	1												
maq010								1	1													
maq011											2											
maq012											2	2		2	2							
maq015											2	2		2	2	2						
maq019											2	2		2		2						
maq016														2	2							
maq002								1											3	3	3	3
maq004																			3			
maq013																			3	3	3	3
maq014																			3			
maq018																			3			

Figura 8.17 – Sobreposição das matrizes rearranjadas.

Agrupa-se, assim, 21 das 35 peças da matriz *teste_3* representando 60% de aproveitamento.

Considerando-se que as peças tenham rotas alternativas de fabricação é possível obter resultados com maior número de peças agrupadas em células distintas, sem a necessidade de movimentação intercelular.

As peças rejeitadas dos agrupamentos em famílias poderão ser fabricadas em um sistema de *layout* funcional ou mesmo ser encomendadas a fornecedores externos.

9 CONCLUSÕES E SUGESTÕES PARA TRABALHOS FUTUROS

Analisando-se os resultados dos testes com diferentes matrizes para os algoritmos implementados algumas considerações podem ser feitas.

O *Rank Order Clustering (ROC)* é o algoritmo de mais fácil compreensão e implementação computacional, porém é o de processamento mais demorado. Na presença de elementos excepcionais, o reordenamento da matriz torna baixa a densidade da diagonal, como verificado para as matrizes *teste_1* (Figuras 7.11) e *teste_3* (Figura 8.8).

O *Close Neighbor Algorithm*, na interpretação de Fogliatto (2004), consegue ordenar as colunas de matrizes de pequena ordem agrupando adequadamente os elementos quando o número de máquinas em uma célula for igual ou superior a um terço do número de máquinas da matriz. Para matrizes de dimensões maiores, a aplicação desta interpretação do algoritmo não se mostra capaz de ordenar as colunas da matriz, como verificado nas Figuras 7.13 e 8.3, obtendo-se melhores resultados com a substituição deste pela versão original de Boe e Cheng (1991).

A concepção original do *CNA*, também de fácil compreensão e implantação computacional, mostra-se capaz de reordenar as linhas e colunas de uma matriz em um intervalo de tempo menor que aquele de processamento do algoritmo *ROC*. Há, porém, devido a sua lógica de agrupamento, uma deficiência em agrupar alguns elementos na matriz *teste_1*, da Figura 7.18, como a máquina *maq028* que poderia estar inserida na célula 1 e a máquina *maq031*, na célula 3. A densidade da diagonal é maior no arranjo *CNA* do que nos demais algoritmos estudados neste trabalho. Essa constatação pode ser verificada comparando-se os arranjos da matriz *teste_1* apresentados nas Figuras 7.11 e 7.15 e os arranjos da matriz *teste_3* ilustrados nas Figuras 8.8, 8.10 e 8.14.

O algoritmo *HEU* é o arranjo de concepção lógica mais difícil. No entanto, ao contrário dos algoritmos *ROC* e *CNA*, leva em consideração as restrições durante o arranjo da matriz, tornando-o uma ferramenta mais poderosa. Dos três procedimentos, o heurístico é o que mais rapidamente apresenta os resultados no processamento.

O tempo de processamento depende do tamanho da matriz trabalhada e quão maior é a matriz, maior é a diferença dos tempos de processamento. No caso da matriz *teste_1*, o *HEU* apresenta o resultado aproximadamente duas vezes mais rápido que o *CNA* e quatro vezes mais rápido que o *ROC*. Isso se deve ao fato de que o *HEU* leva em consideração as restrições de projeto durante o arranjo da matriz incidência e executa o processamento uma única vez, enquanto que o *CNA* analisa as restrições de projeto máquinas após diagonalização da matriz. O algoritmo *ROC* é o mais demorado pois está implementado de forma que, após o ordenamento da

matriz e aplicadas as restrições de projeto, para cada peça eliminada durante a análise das restrições um novo ordenamento da matriz é executado.

A ordem com que são agrupadas e apresentadas as peças influencia no resultado quando aplicada a restrição de carregamento de acordo com o critério de exclusão adotado neste trabalho. Para outro critério, como o de eliminar a peça de menor custo de processamento, todos os algoritmos apresentariam o mesmo resultado, rejeitando as mesmas peças. Por este motivo, o tamanho da célula mostra-se como a restrição responsável pelo melhor resultado de cada algoritmo e é adotado como critério comparativo.

Observa-se, devido ao critério de eliminação das peças, que quando o número de máquinas superar o valor imposto para o tamanho da célula e ao procedimento de demarcação de fronteiras do agrupamento, que o número de peças excluídas é maior para o algoritmo *CNA*. Uma pequena mudança no critério de eliminação pode vir a agrupar um maior número de peças.

Apesar dos estudos comparativos terem demonstrado a superioridade dos algoritmos *ROC* e *HEU* em relação ao *CNA*, este não deve ser descartado pois, em alguns casos, pode trazer melhores resultados de agrupamento. Deve-se observar que os resultados obtidos pelo algoritmo *ROC* são superiores ou iguais aos demais algoritmos, dificilmente inferiores. Isso ocorre devido a condição implementada no sistema que faz com que o algoritmo reordene a matriz incidência toda a vez que uma peça é descartada durante a análise das restrições.

Quanto maior o número de critérios adotados como suporte ao processo de refinamento dos agrupamentos tanto maior a probabilidade de encontrar uma solução mais eficiente.

9.1 Características operacionais importantes do sistema proposto

- trabalha com banco de dados;
- contabiliza o tempo de execução de cada algoritmo;
- apresenta três modos de visualização das matrizes:
 - na forma binária;
 - com tempos de processamento;
 - indicando a qual agrupamento os itens estão inseridos;
- identifica as células e suas fronteiras, isto é, não necessita de um analista para identificar possíveis grupos pois possui uma rotina interna que numera as células independentes;
- impõe restrições:
 - de limitação no tamanho da célula;
 - para o limite do tempo de processamento; e
 - para o limite de tempo para duplicação de máquinas gargalo;

- possibilita a adição e processamento de rotas alternativas de processamento de uma peça;
- exporta os resultados para um arquivo em *MS Excel* possibilitando outras análises;
- apresenta uma listagem de células, famílias e peças não agrupadas;
- apresenta as matrizes incidência e diagonalizadas pelos algoritmos e tratadas pelos critérios de análise.

9.2 Limitações

O sistema proposto apresenta algumas limitações operacionais por não considerar os seguintes aspectos:

- custos envolvidos em processamento, movimentação ou investimento em equipamentos;
- interdependência de máquinas;
- balanceamento da linha;
- número de operadores envolvidos.

9.3 Sugestões para continuidade do trabalho

- adicionar outras restrições que levam em consideração:
 - os custos envolvidos;
 - interdependência de máquinas;
 - balanceamento da linha;
 - número de operadores envolvidos;
- somar células pequenas para formar uma única célula maior, contornando situações de ociosidade de operador, espaço físico ou movimentação e armazenamento de peças desde que o fluxo de processo e o balanceamento das máquinas não sejam prejudicados;
- gerar uma curva *número de peças agrupadas versus tamanho de célula* destacando o ponto que apresente o melhor resultado;
- adotar outros procedimentos de definição das fronteiras das células;
- acrescentar outros algoritmos como os que utilizam coeficientes de similaridade ou de processamento genético;
- acrescentar módulo para gerar o *layout* em ambiente *CAD* das células;
- escrever o aplicativo em uma linguagem que independa do sistema operacional ou de um *software* proprietário.

REFERÊNCIAS BIBLIOGRÁFICAS

BAKER, R. P. e MAROPOULOS, P. G., 1997. “**An automatic clustering algorithm suitable for use by a computer-based tool for the design, management and continuous improvement of cellular manufacturing systems**”. Computer Integrated Manufacturing Systems Vol. 10, No. 3, p. 217-230.

BEDWORTH, D. D., HENDERSON, M. R. e WOLFE, Philip M., 1991. “**Computer Integrated Design and Manufacturing**”. USA: McGraw-Hill, 653p.

BOE, W. J., CHENG, C. H., 1991. “**A close neighbour algorithm for designing cellular manufacturing systems**”. International Journal of Production Research, Vol. 29 No 10, pp. 2097-2216.

CHAN, C. Y., LAM, F. W. e LEE C. P., 1999. “**Considerations for using cellular manufacturing**”. Journal of Materials Processing Technology, pp. 182-187.

DALMAS, V., 2004. “**Avaliação de um layout celular implementado: um estudo de caso em uma indústria de autopeças**”. Dissertação (mestrado profissionalizante na Eng. Produção) - Universidade Federal do Rio Grande do Sul. Escola de Engenharia - Mestrado Profissionalizante em Engenharia, Porto Alegre, 120 p.

DROLET, J., ABDULNOUR, G. e RHEAULT, M., 1996. “**The cellular manufacturing evolution**”. Computers ind. Engng Vol. 31, No. 1/2, pp. 139-142.

FOGLIATTO, F. S., 2004. “**PCP – Planejamento e Controle da Produção**” – Apostila da disciplina de Planejamento e Controle da Produção do Programa de Pós-Graduação em Engenharia de Produção da Universidade Federal do Rio Grande do Sul. Porto Alegre: PPGEP/UFRGS.

GOLDRATT, E. M. e COX, J., 1986. “**A Meta**”. São Paulo: Ed. Educador, 221 p.

GULSEN, M., SMITH, A. E. e TATE, D. M., 1995. “**A genetic algorithm approach to curve fitting**”. International Journal Production Research, Vol. 33, No. 7, p. 1911-1923

GUPTA, Y., GUPTA, M. KUMAR, A. e SUNDARAM, C., 1996. “**A genetic algorithm-based approach to cell composition na layout design problems**”. International Journal Production Research, Vol. 34, No. 2, p. 447-482.

KUSIAK, A., 1985. “**Flexible Manufacturing System: A Strutral Approach**”. International Jounal Production Research, Vol. 23, No. 6, p. 1057-1073.

LEE, M.K., Luong, H.S. e Abhary, K., 1997. “**A genetic algorithm based cell design considering alternative routing**”. Computer Integrated Manufacturing Systems, Vol. 10, No. 2, p. 93-107.

LORINI, F. J, 1993. “**Tecnologia de Grupo e Organização da Manufatura**”. Florianópolis: Editora da UFSC, 105 p.

MASSAY, L. L., UDOKA, S. J. e BENJAMIN, C. O., 1995. “**A simulator-based approach to cellular manufacturing system design**”. Computers Industrial Engineering Vol. 29, No. 1-4, pp. 327-331.

MELO, S. P., 2003. “**Desenvolvimento e aplicação de um sistema de planejamento de processo auxiliado por computador em um ambiente de usinagem**”. Dissertação de Mestrado. Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Engenharia Mecânica, Porto Alegre, BR-RS, 107 f.:il.

LEE, M. K., LUONG, H. S. e ABHARY, K., 1997. “**A genetic algorithm based cell design considering alternative routing**”. Computer Integrated Manufacturing Systems Vol. 10, No. 2, pp. 93-107.

MOON, C. e GEN, M., 1999. “**A genetic algorithm-based approach for design of independent manufacturing cells**”. Int. J. Production Economics 60-61, 421-426.

MOUSSA, S. E. e KAMEL, M., 1998. “**A part-machine assignment algorithm for cellular manufacturing with machine capacity constraints**”. Computers Industrial Engineering Vol. 35, No. 3-4, pp. 483-486.

MUKHOPADHYAY, S. K., MAITI, B. e GARG, S., 1991. “**Heuristic solution to the scheduling problems in flexible manufacturing system**”. International Journal of Production Research, Vol. 29 No 10, pp. 2003-2024.

RAMOS, G. N., TAMAYO, A. M e ÁLVARES, A. J., 2004. “**Desenvolvimento de um sistema informático para formação de células flexíveis de manufatura**”. Máquinas e Metais, Ano XLI n° 465. Aranda Editora, pp. 76-91.

RAJAMANI, D., SINGH, N. e ANEJA, Y. P., 1996. “**Design of cellular manufacturing systems**”. International Journal of Production Research, Vol. 34 No 7, pp. 1917-1928.

REYNOLDS, K. T., 1998. “**Cellular manufacturing & the concept of total quality**”. Computers Ind. Engineering Vol. 35, Nos. 1-2, pp. 89-92.

SEIFODDINI, H. e DJASSEMI, M., 1995. “**A simulation study of quality index in machine-component grouping**”. Computers Industrial Engineering Vol. 29, No. 1-4, pp. 449-453.

SELIM, H. M., ASKIN, R. G. e VAKHARIA, A. J., 1998. “**Cell formation in group technology: review, evaluation and directions for future research**”. Computers Industrial Engineering Vol. 34, No. 1, pp. 3-20.

SHINGO, S., 1996. “**O sistema Toyota de Produção do ponto de vista da Engenharia de Produção**”/Shingeo Shingo; trad. Eduardo Schaan. 2ª ed. Porto Alegre: Artes Médicas. 291p.

SILVEIRA, G., 1994. “**Uma metodologia de implantação da manufatura celular**”. Dissertação de Mestrado em Engenharia, Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Engenharia de Produção, Porto Alegre, BR-RS. 106p.: il.

SINGH, N. e RAJAMANI, D., 1996. “**Cellular manufacturing systems: design, planning and control**”. London: Chapman & Hall, 277 p.

WANG, J., 1998. “**A linear assignment algorithm for formation of machine cells and part families in cellular manufacturing**”. Computers Industrial Engineering Vol. 35, No. 1-2, pp. 81-84.

WANG, S. Y., 2005. “**Structural topology design optimization using Genetic Algorithms with a bit-array representation**”. Computer Methods in Applied Mechanics and Engineering. 194, pp. 3749-3770.

ZUBEN, F. J., 2004. <http://www.dca.fee.unicamp.br/~vonzuben/courses/ia707.html>, acessado em 22 de agosto de 2005.

APÊNCICE I – Relacionamento entre tabelas

O Microsoft Access é um banco de dados relacional, ou seja, usa os dados de mais de uma tabela ao mesmo tempo. Depois de criar as tabelas no seu banco de dados e definir a chave primária de cada tabela, você pode criar relacionamentos entre as tabelas. Um relacionamento pode ajudar o Microsoft Access a associar os dados de qualquer consulta, formulário, ou relatório novos que incluam as duas tabelas relacionadas.

Segundo a Microsoft (1996), pode-se criar dois tipos de relacionamentos no Microsoft Access: um relacionamento um-para-muitos e um relacionamento um-para-um. O tipo mais comum é o relacionamento um-para-muitos. Neste tipo de relacionamento, um registro de uma tabela pode ter muitos registros relacionados em outra tabela. Por exemplo, um cliente pode fazer muitos pedidos. Analogamente, um registro de uma tabela Clientes (chamada tabela primária no relacionamento) pode ter muitos registros coincidentes em uma tabela Pedidos (chamada tabela relacionada).

Em um relacionamento um-para-um, por outro lado, um registro da tabela primária pode ter apenas um registro coincidente na tabela relacionada. Esse tipo de relacionamento é menos comum do que o relacionamento um-para-muitos. Um motivo pelo qual é usado um relacionamento um-para-um seria quando fosse necessário separar as informações sobre empregados em dados públicos e privados. Por exemplo, é possível colocar informações públicas, tais como os nomes e cargos, em uma tabela e as informações restritas, tais como salários, em outra tabela. Essas duas tabelas teriam um relacionamento um-para-um, porque cada registro da tabela pública teria apenas um registro coincidente na tabela restrita.

Quando se avalia um relacionamento entre duas tabelas, é importante ver ambos os lados do relacionamento. A princípio, pode-se achar que há um relacionamento um-para-muitos quando, na verdade, o que há é um relacionamento muitos-para-muitos. Um relacionamento muitos-para-muitos ocorre quando um registro em uma das tabelas pode ter mais do que um registro coincidente na outra tabela. Nesses casos, antes de poder criar os relacionamentos, é necessário criar uma terceira tabela que vincule as duas.

As tabelas *t_rotas* e *t_maquinas* do banco de dados, à primeira vista, podem parecer ter um relacionamento um-para-muitos, porque uma rota pode conter muitas máquinas diferentes. Mas analisando o relacionamento do ponto de vista das máquinas, uma máquina pode aparecer em mais de uma rota.

Haveria problemas se fosse criado um relacionamento um-para-muitos entre as tabelas *t_rotas* e *t_maquinas*.

Supondo que a tabela primária do relacionamento fosse *t_rotas*. Quando acrescentado um campo *ID_rotas* à tabela *t_maquinas*, o qual conteria os valores coincidentes. Mas, no registro das máquinas, seria necessário inserir os *IDs* das rotas de todas as peças, porque uma máquina aparece em mais de uma rota. Caso isso acontecesse, o Microsoft não poderia relacionar o registro da máquina às rotas certas, e só haveria um valor em cada campo coincidente. O mesmo aconteceria se fosse colocado um campo *ID_maquinas* na tabela *t_rotas*.

A solução é criar uma tabela de associação contendo as chaves primárias das duas tabelas que se deseja relacionar. Em uma tabela de associação, é possível acrescentar outro campo que não esteja em nenhuma das tabelas originais, mas que lhe dê uma informação adicional relevante para as outras tabelas. Neste banco de dados, a tabela de associação chama-se *t_rotas sub*. A chave primária da tabela *t_rotas sub* consiste do *ID_rotas* e do *ID_maquinas* as quais são as chaves primárias das duas tabelas que estão sendo relacionadas. A tabela *t_rotas sub* também contém um campo *tempo de processamento*, o que informa o tempo de processamento de cada peça naquela máquina (Figura AI.1).

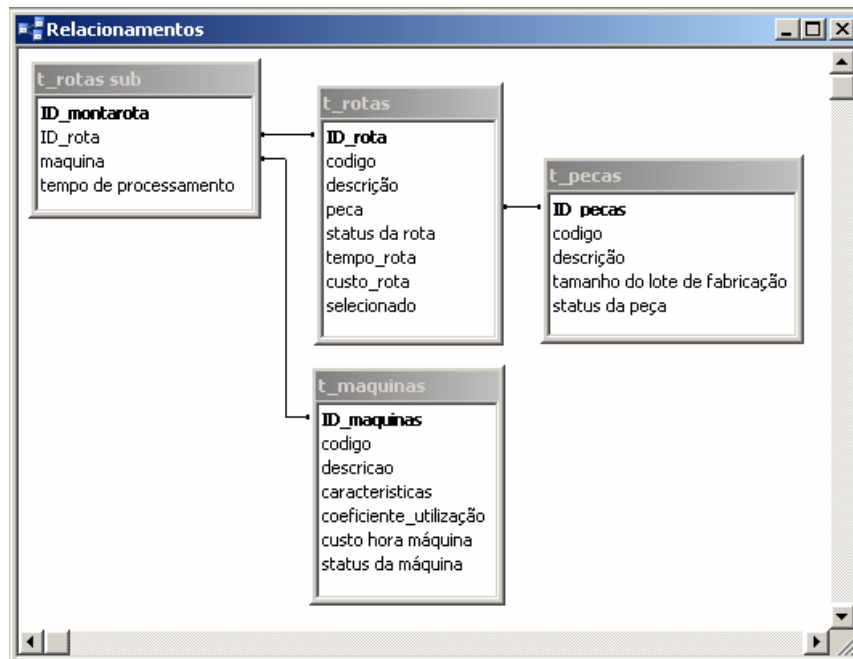


Figura AI.1 – Relacionamentos entre tabelas de dados.

Livros Grátis

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

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

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

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

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