

DESENVOLVIMENTO EM AMBIENTE MATLAB® DE UM SISTEMA PARA FORMAÇÃO DE CÉLULAS FLEXÍVEIS DE MANUFATURA BASEADO NO ALGORITMO ROC

Guilherme Novaes Ramos

Universidade de Brasília, Departamento de Engenharia Mecânica e Mecatrônica, Grupo de Automação e Controle – Graco, 70.910-300, Brasília, email: helmetpackard@yahoo.com

Alvaro Michel Tamayo

Universidade de Brasília, Departamento de Engenharia Mecânica e Mecatrônica, Grupo de Automação e Controle – Graco, 70.910-300, Brasília, email: itotamayo@yawl.com.br

Alberto José Álvares

Universidade de Brasília, Departamento de Engenharia Mecânica e Mecatrônica, Grupo de Automação e Controle – Graco, 70.910-300, Brasília, email: alvares@AlvaresTech.com

Resumo. *Este trabalho apresenta os resultados de um projeto de pesquisa cujo tema visa a aplicação do algoritmo ROC (Rank Order Clustering) em uma interface amigável no ambiente MatLab® para obter a organização de um chão-de-fábrica baseando-se em Tecnologia de Grupo (GT). A organização das Células Flexíveis de Manufatura (FMC's) é um fator determinante para que se possa implementar a GT, que é uma tendência dominante na indústria atual. Esta filosofia busca reduzir tempos improdutivos associados aos processos de fabricação, através de melhorias de fluxo. Como consequência tem-se um aumento na produtividade. Para implementar a GT, é necessário reconhecer os grupos (ou famílias) formados por elementos que apresentam características semelhantes de dimensões ou processos de fabricação. Um método bastante simples de executar esta tarefa é aplicar o algoritmo ROC, que obtém as famílias pela reordenação de uma matriz de incidência que representa matematicamente os processos em questão. Utilizando o MatLab®, criou-se um programa de interface amigável para obter os resultados desejados.*

Palavras-chave: *MatLab®, tecnologia de grupo, algoritmo ROC, células flexíveis de manufatura, análise do fluxo de produção.*

1. INTRODUÇÃO

O layout do chão-de-fábrica normalmente é classificado em quatro tipos: Funcional, Produto, Posição e Celular (Álvares *et al*, 1991). Neste trabalho o maior interesse concentra-se nos layouts Funcional e Celular. A configuração funcional separa as máquinas levando em conta as tarefas que elas são capazes de realizar (torneamento, fresamento, furação, etc.), e foi muito utilizada nos princípios da indústria. Contudo, atualmente este tipo de organização foi praticamente substituído pela configuração em células de manufatura. Isto se deu devido às vantagens oferecidas pela Tecnologia de Grupo (GT), que é um fator decisivo na formação das células.

A GT é uma filosofia que busca reduzir tempos improdutivos associados a processos de fabricação, implementando melhorias de fluxo e reduções de *setup* e também de inventários em processos, tirando proveito das similaridades das peças. Sua implementação requer o

reconhecimento dos grupos (ou famílias) formados por elementos que apresentam características semelhantes de dimensões geometria (forma, dimensões, tolerâncias, etc) ou de processos de fabricação. Uma vez delimitadas as famílias de peças, pode-se organizar as máquinas envolvidas no processo para as células de manufatura.

A organização de um chão-de-fábrica em células de manufatura aumenta significativamente a produtividade, pois reduz grande parte dos tempos improdutivos na fabricação. Porém, ela exige que seja realizado um estudo da relação entre as máquinas disponíveis e as peças a serem fabricadas na planta. Busca-se então identificar as famílias de peças, de modo a agrupar as máquinas, minimizando a movimentação das peças entre as células. Existem diversos métodos para obter estes resultados, e estes podem apresentar soluções distintas entre si.

Este trabalho apresenta os aspectos referentes ao desenvolvimento de um sistema computacional, implementado em ambiente MatLab, para formação de Células Flexíveis de Manufatura utilizando-se o algoritmo ROC (*Rank Order Clustering*). Com a utilização do sistema é possível estudar a viabilidade da aplicação dos conceitos de Tecnologia de Grupo (GT) a fim de buscar um "layout" celular como forma de organização do chão-de-fábrica.

A organização das Células Flexíveis de Manufatura é um fator determinante para que se possa implementar a GT, que é uma tendência dominante na indústria atual que trabalha com médios e pequenos lotes de fabricação. Esta filosofia busca reduzir tempos improdutivos associados aos processos de fabricação, através de melhorias de fluxo. Como consequência tem-se um aumento na produtividade refletindo também na qualidade do produto final.

Para implementar a GT, é necessário reconhecer os grupos (ou famílias) formados por elementos que apresentam características semelhantes de dimensões/forma ou processos de fabricação. Um método clássico de executar esta tarefa é aplicar o algoritmo ROC, que obtém as famílias pela reordenação de uma matriz de incidência (MI) que representa matematicamente os processos em questão.

Utilizando o ambiente de desenvolvimento MatLab® (<http://www.mathworks.com>), criou-se um programa com uma interface gráfica amigável que permite a entrada de dados na forma da MI e seu processamento a fim de obter a MI diagonalizada, que representa a proposta da nova organização do "layout" em um contexto de Manufatura Celular, baseando-se no algoritmo ROC (*Rank Order Clustering*).

2. TECNOLOGIA DE GRUPO E CÉLULAS FLEXÍVEIS DE MANUFATURA

2.1. Considerações Iniciais

A organização funcional das máquinas em um chão-de-fábrica agrupa máquinas que desempenham a mesma função. Embora de fácil implementação, esta configuração não é muito eficiente. Por exemplo, uma peça que necessita de torneamento, furação e fresamento deverá ser levada de uma área a outra do chão-de-fábrica, aumentando os tempos improdutivos. Como a indústria contemporânea visa a melhor relação custo/benefício possível, busca-se novas técnicas e filosofias que aprimorem as condições de produção. Dentre as diversas soluções propostas, destaca-se a idéia da *Tecnologia de Grupo*, ou GT.

Esta filosofia busca reduzir tempos improdutivos associados a processos de fabricação, implementando melhorias de fluxo e reduções de *setup* e também de inventários em processos tirando proveito das similaridades das peças. Sua implementação requer o reconhecimento dos grupos (ou famílias) formados por elementos que apresentam características semelhantes de dimensões ou processos de fabricação. Uma vez delimitadas as famílias de peças, pode-se organizar as máquinas envolvidas no processo para formar as células de manufatura.

Álvares *et al* (1991) define uma célula de manufatura, ou Célula Flexível de Manufatura, como o menor conjunto indivisível na fabricação que garante, a partir do item a processar e dentro de uma família de peças pré-determinada, o cumprimento de uma etapa completa do processo. Esta configuração deixa todas as máquinas necessárias para um processo de uma família de peças

reunidas em uma única célula. O termo “flexível” indica que a célula pode facilmente ser adaptada para a fabricação de peças diferentes.

Para uma organização em FMC's, a alternativa mais conveniente é a que separa as peças em função dos processos pelos quais ela é fabricada. Desta forma, podem ser reunidas as máquinas capazes de realizar estes processos em uma mesma célula, formando assim agrupamentos máquinas-peças. Esquemáticamente, pode-se diferenciar a topologia das duas configurações de chão-de-fábrica, como mostrado na Fig. (1).

2.2. Formação de FMC's

Uma das aplicações mais interessantes da GT, e também sua principal dificuldade é a formação das FMC's. Isto exige um estudo das peças que serão produzidas, bem como da capacidade de cada máquina disponível, já que devem ser definidos agrupamentos máquinas-peças coerentes com a produção.

Diversos métodos capazes de realizar esta tarefa são propostos (Chan *et al*, 1999, Chandrasekharan *et al*, 1986, Chandrasekharan *et al*, 1987, Vieira *et al*, 1999), e cada um pode apresentar uma solução distinta. Pode-se separá-los em duas categorias: métodos orientados à peças e métodos orientados ao processo. Os métodos orientados à peças se baseiam nas similaridades de formato e funcionalidade das peças. Estes métodos geralmente não são muito difundidos, já que sua influência na configuração da FMC é limitada. Os métodos orientados ao processo se baseiam nas similaridades de processo das peças. Este tipo é mais utilizado na indústria atual. Ele pode ainda ser diferenciado em cerca de quatro categorias:

- Métodos descritivos;
- Métodos matriciais;
- Métodos de coeficiente de similaridade;
- Outros.

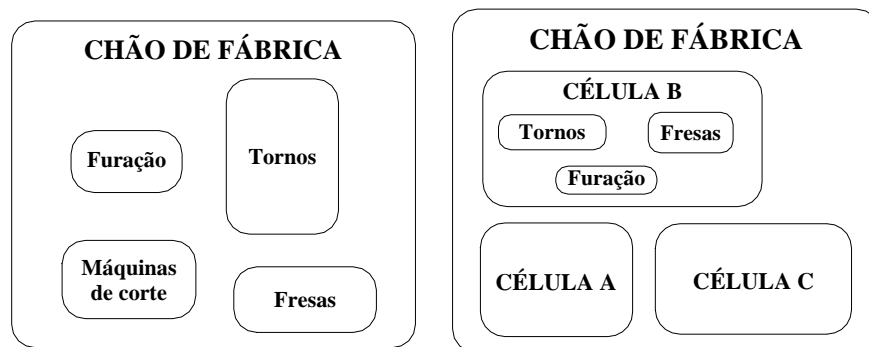


Figura 1. Exemplo de um chão-de-fábrica funcional e um com organização celular.

Os métodos matriciais têm base em uma Matriz de Incidência binária (MI) que descreve o fluxo de peças através das diferentes máquinas que poderiam compor as FMC's. Ou seja, a MI indica quais peças são processadas por quais máquinas. A partir deste ponto, algoritmos matemáticos de manipulação de matrizes são aplicados para a obtenção de uma nova configuração da matriz que permite uma fácil identificação dos agrupamentos máquinas-peças. Como os algoritmos de manipulação são vários, existem várias soluções diferentes para uma mesma MI. Estas podem (ou não) ser um resultado satisfatório (Chan *et al*, 1999).

2.3 O Algoritmo ROC (*Rank Order Clustering*)

O algoritmo ROC é uma técnica para formação de FMC's orientada à produção (método matricial) que se baseia na análise do fluxo de produção. Ele considera cada linha e cada coluna da MI como um número binário e realiza ordenações sucessivas até que o resultado desejado seja atingido. Seu princípio básico é arranjar as linhas e colunas da matriz de incidência de modo a obter uma matriz diagonal de blocos que determinam as FMC's.

- A MI é formada da seguinte forma:
- elemento a_{ij} é igual a 1, se a peça j deve ser processada pela máquina i ;
- elemento a_{ij} é igual a 0, caso contrário.

Ou seja, as colunas da MI representam as máquinas e as linhas representam as peças. A Figura (2) mostra um exemplo de uma MI (os elementos 0 são omitidos para melhor visualização) correspondente a um chão-de-fábrica composto por 5 máquinas (M1, M2, M3, M4, M5), para a produção de 8 peças diferentes (P1, P2, P3, P4, P5, P6, P7, P8).

Analisando a Fig. (2), percebe-se facilmente que a peça P1 deve ser processada pelas máquinas M1 e M2, a peça P2 apenas pela máquina M4, a peça P3 pelas máquinas M3 e M4, e assim por diante.

O algoritmo ROC segue os seguintes passos (Leal et. al, 1999):

- obter a MI;
- ordenar as linhas da MI em ordem decrescente, considerando-as como vetores binários;
- reescrever a MI ordenada;
- ordenar as colunas da nova MI em ordem decrescente, considerando-as como vetores binários;
- reescrever a MI ordenada;
- verificar se a nova MI necessita de uma nova ordenação. Se for preciso, retornar ao segundo passo;
- apresentar a matriz final.

| | | MÁQUINAS | | | | |
|-------|----|----------|----|----|----|----|
| | | M1 | M2 | M3 | M4 | M5 |
| PEÇAS | P1 | 1 | 1 | | | |
| | P2 | | | | 1 | |
| | P3 | | | 1 | 1 | |
| | P4 | | | 1 | | |
| | P5 | | 1 | | | 1 |
| | P6 | | 1 | | | 1 |
| | P7 | | | 1 | 1 | |
| | P8 | 1 | | | | 1 |

Figura 2. Exemplo de Matriz de Incidência 8x5.

Aplicando o algoritmo ROC à MI da Fig. (2), obtém-se a matriz da Fig. (3).

| | | MÁQUINAS | | | | |
|-------|----|----------|----|----|----|----|
| | | M1 | M2 | M5 | M3 | M4 |
| PEÇAS | P1 | 1 | 1 | | | |
| | P8 | 1 | | 1 | | |
| | P5 | | 1 | 1 | | |
| | P6 | | 1 | 1 | | |
| | P3 | | | | 1 | 1 |
| | P7 | | | | 1 | 1 |
| | P4 | | | | 1 | |
| | P2 | | | | | 1 |

Figura 3. Resultado da aplicação do algoritmo ROC na MI da Fig. (2).

A Figura (3) destaca duas FMC's que podem ser criadas no chão-de-fábrica: uma formada pelas máquinas M1, M2 e M5, para a produção das peças P1, P8, P5 e P6; e outra formada pelas máquinas M3 e M4, para a produção das peças P3, P7, P4 e P2. Ressalta-se que esta não é a única solução possível para a formação de FMC's neste mesmo chão-de-fábrica, já que outro algoritmo pode ser aplicado à MI e fornecer um resultado diferente.

O resultado obtido com a aplicação do algoritmo ROC nem sempre fornece agrupamentos mutuamente exclusivos. Uma alternativa para este caso é pesquisar entre soluções diferentes, obtidas por meio da aplicação dos diferentes métodos de formação de FMC's (uma inspeção visual de refinamento posterior à aplicação do ROC pode ser suficiente). Outra é a formação de células com repetição de máquinas, aumentando os custos de implementação. Ou ainda pode-se implementar o transporte de peças entre as FMC's, diminuindo a produtividade.

O algoritmo ROC nem sempre é a melhor escolha, apesar da simplicidade e a facilidade de implementação computacional. Ele apresenta limitações computacionais (praticamente desprezíveis com a tecnologia atual): as linhas e colunas das matrizes sendo interpretadas como números binários, uma MI 50x50 pode exigir que o processador manipule inteiros de até 2^{50} , ou 1.125.899.906.842.624. Evidentemente, isto implicaria que uma das máquinas participe da fabricação de todas as peças, ou que uma das peças tenha que ser processada por todas as máquinas, o que raramente ocorre. No entanto, esta limitação só deve ser levada em conta se o algoritmo for implementado com o uso de um processador não muito recente. Deve-se destacar também que o algoritmo ROC desconsidera uma ordem obrigatória (se existir) de operações a serem realizadas nas peças, e geralmente requer um especialista para identificar os grupos formados.

3. ARQUITETURA DO SISTEMA MATROC

3.1 Sistema Proposto

O sistema concebido, denominado MatROC (MatLab[®] ROC), utiliza o algoritmo ROC para ordenar uma matriz de incidência conhecida, visando a formação de FMC's. Desta forma, ele deve executar as tarefas básicas, mas específicas, de aquisição e ordenação da matriz, bem como, a apresentação do resultado obtido. Contudo, como objetiva-se uma interface amigável, o sistema também deve oferecer algumas funções extras, que facilitem sua utilização. Portanto, além das três tarefas essenciais, estabeleceu-se que ele deve oferecer opções de manipulação de arquivos (para uma eventual evolução para o trabalho com uma base de dados), de impressão e de ajuda ao usuário. A Figura (4) sintetiza esquematicamente as funções do MatROC.

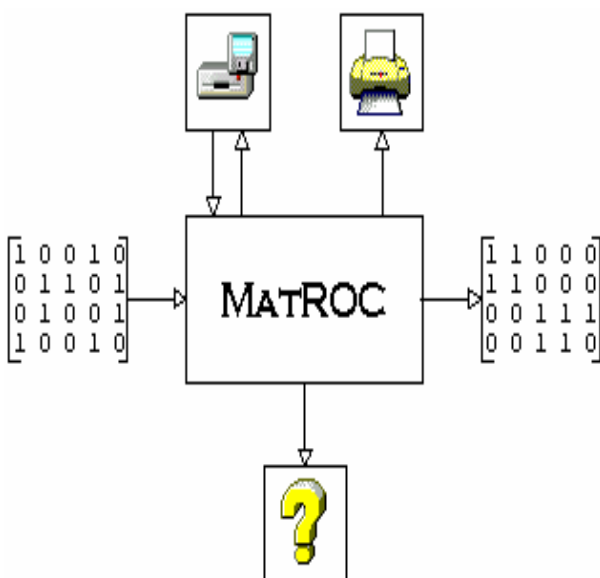


Figura 4. Arquitetura do sistema.

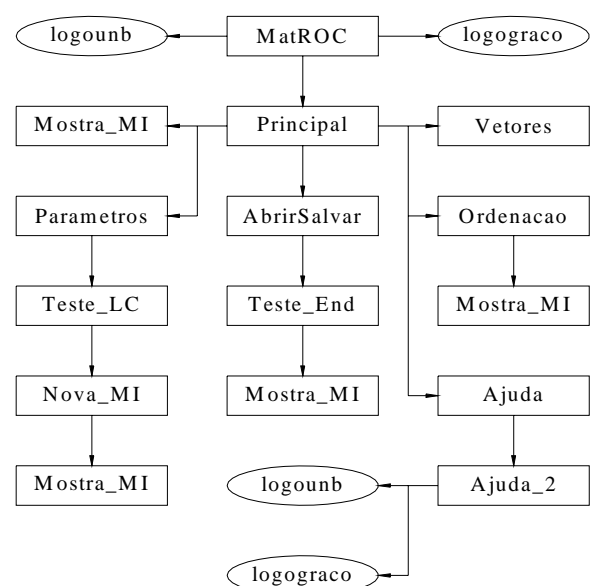


Figura 5. Estrutura de arquivos do MatROC.

A Figura (4) indica as opções oferecidas pelo sistema MatROC, que devem ser executadas por meio de uma interface “amigável” com o usuário. Assim, ele deve trabalhar de tal forma que:

- Permita uma fácil entrada de dados (MI).
 - Receber uma matriz de incidência, via usuário ou por uma base de dados.
- Executar a seqüência de passos do algoritmo ROC, ordenando a matriz.
- Apresente os resultados de forma clara.
 - Mostrar ao usuário a nova matriz de incidência gerada.
 - Possibilitar que a mesma seja salva na base de dados.
- Permita acesso a todas as informações úteis fornecidas pela aplicação do algoritmo.
 - Oferecer instruções específicas do modo de operação do sistema.
 - Possibilitar a impressão em papel dos resultados obtidos.

3.2. Implementação

Para implementar este sistema escolheu-se a ferramenta MatLab[®] (*Matrix Laboratory*), por apresentar condições de trabalho tais como:

- Linguagem de programação de alto nível.
- Desenvolvimento em *scripts* em arquivos do tipo *.m*, de pequeno tamanho.
- Fácil manipulação de vetores e matrizes.
- Funções para interface gráfica.

O resultado obtido é um conjunto de doze arquivos tipo *.m*, que contêm as funções e comandos necessários para o funcionamento adequado do programa e dois arquivos tipo *.jpg* para as figuras de apresentação do programa. A Figura (5) ilustra as chamadas de arquivos executadas pelo MatROC (inicializado pelo arquivo *MatROC.m*), evidenciando as relações existentes entre os quatorze arquivos do sistema. Cada arquivo execute tarefas específicas, que são descritas a seguir:

- *MatROC.m*: apresenta o sistema e chama a função de controle principal.
- *Principal.m*: cria a janela principal do programa, gerenciando as demais funções.
- *Mostra_MI.m*: mostra de forma gráfica a matriz de incidência.
- *Parametros.m*: recebe os parâmetros da nova matriz de incidência.
- *Teste_LC.m*: testa se estes parâmetros são válidos.
- *Nova_MI.m*: recebe do usuário, graficamente, uma matriz de incidência.
- *Abrir_Salvar.m*: abre de ou salva em um arquivo a matriz de incidência.
- *Teste_End.m*: testa se o endereço do arquivo é válido.
- *Vetores.m*: mostra a ordenação da matriz de incidência.
- *Ordenacao.m*: aplica o algoritmo ROC, ordenando a matriz de incidência.
- *Ajuda.m*: fornece os tópicos de ajuda do programa.
- *Ajuda_2.m*: fornece as instruções referentes ao tópico escolhido.
- *logounb.jpg* e *logograco.jpg*: são imagens dos logotipos da Universidade de Brasília (UnB) e do Grupo de Automação e Controle (GRACO), respectivamente.

O tamanho total do programa (soma dos quatorze arquivos) não ultrapassa 60KB, facilitando a disposição do mesmo para *download* (via *ftp* ou *http*). Esta característica é de grande valia pois possibilita o uso do MatROC por qualquer usuário que tenha acesso à *internet* e ao MatLab[®], de forma a independar da plataforma usada em seu computador (Linux, Solaris, FreeBSD ou Windows).

As funcionalidades do MatROC estão resumidas na da Fig. (6), assim como as etapas necessárias para a execução das tarefas.

A Figura (7) mostra a janela principal, que gerencia as demais, apresenta a matriz de incidência em questão e botões para ativar as opções do programa. A janela principal permite ao usuário acessar qualquer função do MatROC, sempre tendo à sua esquerda a representação da MI. O sistema está disponível em ftp://graco.unb.br/pub/alvares/sim/Trabalhos_sim_2000/algoritmo_roc_tecnologia_grupo/MatROC.zip.

4. RESULTADOS

Diversos testes foram realizados para avaliar o desempenho do sistema MatROC. Partindo de matrizes de incidência cujo resultado da aplicação do algoritmo ROC era conhecido (MI's tiradas de referências), foram comparadas a solução fornecida pela referência e a obtida com o MatROC. Seguem dois exemplos destas comparações, Fig. (8) e Fig. (9).

No exemplo da Fig. (8), nota-se que não evidenciam-se diferenças entre a solução tirada da referência e a solução obtida através do uso do MatROC.

A Figura (9) mostra um exemplo no qual as matrizes finais não são idênticas. Porém, mesmo apresentando agrupamentos máquinas-peças diferentes, os dois resultados são válidos, já que fornecem configurações de FMC's de fácil implementação e perfeitamente aceitáveis do ponto de vista produtivo (agrupamentos mutuamente exclusivos).

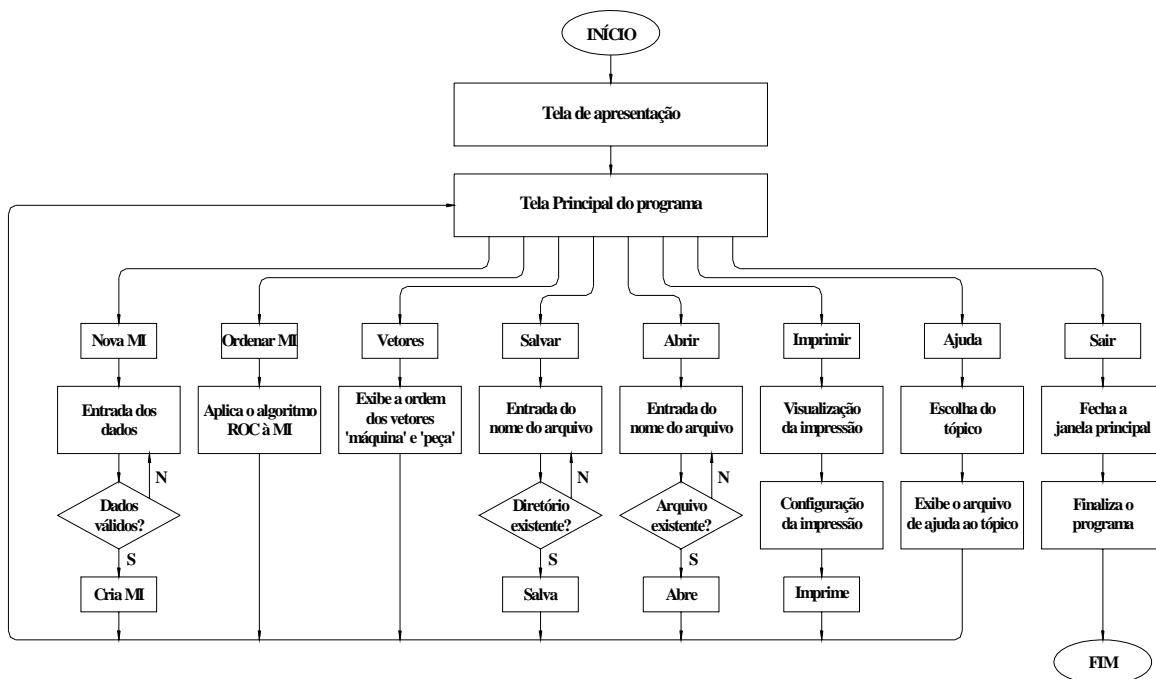


Figura 6. Fluxograma do MatROC.

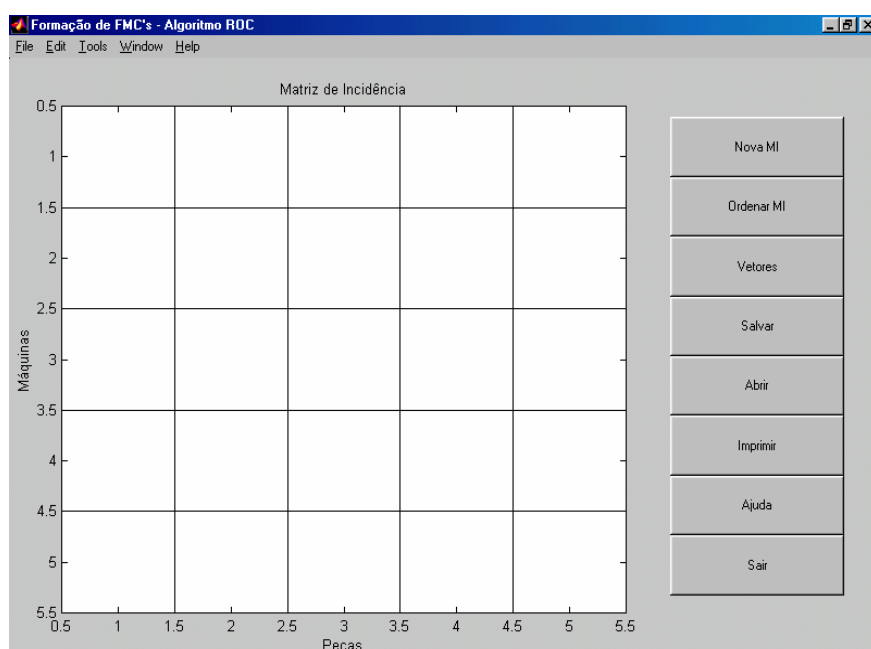


Figura 7. Janela principal do MatROC.

Nos exemplos citados, bem como em outros testados, o sistema forneceu o resultado correto, comprovando a eficiência do código de ordenação implementado.

| | | MÁQUINAS | | | | | | | | |
|-------------|----|----------|----|----|----|----|----|----|----|----|
| | | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | M9 |
| MI original | P1 | 1 | 1 | | 1 | | | | 1 | |
| | P2 | | | | | 1 | | | | 1 |
| | P3 | | | 1 | | 1 | | | | 1 |
| | P4 | | 1 | | 1 | | 1 | | | |
| | P5 | 1 | | | | | | | 1 | |
| | P6 | | | 1 | | | | | | 1 |
| | P7 | | 1 | | | | 1 | 1 | | |

| | | MÁQUINAS | | | | | | | | |
|-------------------------|----|----------|----|----|----|----|----|----|----|----|
| | | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | M9 |
| Resultado da referência | P1 | 1 | 1 | 1 | 1 | | | | | |
| | P2 | 1 | 1 | | | | | | | |
| | P3 | | | 1 | 1 | 1 | | | | |
| | P4 | | | 1 | | 1 | 1 | | | |
| | P5 | | | | | | | 1 | 1 | 1 |
| | P6 | | | | | | | 1 | 1 | |
| | P7 | | | | | | | 1 | | 1 |

| | | MÁQUINAS | | | | | | | | |
|---------------------|----|----------|----|----|----|----|----|----|----|----|
| | | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | M9 |
| Resultado do MatROC | P1 | 1 | 1 | 1 | 1 | | | | | |
| | P2 | 1 | 1 | | | | | | | |
| | P3 | | | 1 | 1 | 1 | | | | |
| | P4 | | | 1 | | 1 | 1 | | | |
| | P5 | | | | | | | 1 | 1 | 1 |
| | P6 | | | | | | | 1 | 1 | |
| | P7 | | | | | | | 1 | | 1 |

Figura 8. Exemplo 1, proposto por Groover (2000).

As demais opções, assim como os testes de controle das variáveis de entrada funcionam corretamente. Portanto, consideradas as limitações da utilização deste método de formação de FMC's (de hardware, de seqüência de operações, necessidade de uma análise posterior para identificação de grupos), o sistema MatROC proposto proporciona, por meio de uma interface amigável e simples (com recursos adicionais de manipulação de arquivos, impressão e ajuda), resultados coerentes.

5. CONCLUSÕES E SUGESTÕES PARA TRABALHOS FUTUROS

O algoritmo ROC é de fácil implementação e, dependendo da matriz de incidência, pode fornecer bons resultados. O MatLab[®] apresenta muitas ferramentas para manipulação de matrizes e figuras, possibilitando diversas aplicações.

O sistema MatROC, para formação de FMC's utilizando o MatLab[®], apresenta algumas características:

➤ Desvantagens:

- Cadastro de cada peça e máquina do sistema e estabelecimento de relações de passagem/não passagem de cada peças por cada máquina para formar a matriz de incidência.

- Necessidade de um analista para identificar possíveis grupos.
- Vantagens:
 - Fácil utilização.
 - Fornece um resultado rápido.
 - Não carrega o sistema.
 - Apresenta funções de manipulação de arquivos.
 - Pode ser continuamente desenvolvido.

Para obter um melhor desempenho do sistema MatROC, pode-se implementar um de algoritmo de ordenação mais eficiente. A utilização de outras rotinas para obter um refinamento do resultado apresentado pelo algoritmo ROC pode dispensar a presença de um especialista para identificar as famílias.

Pode-se ainda implementar um controle do tamanho da matriz, de modo a formatar a entrada/saída de dados para obter a melhor visualização da mesma. Por fim, deve-se buscar a compilação em um arquivo tipo “stand-alone”, que possa ser executado fora do ambiente MatLab®. Desta forma, possibilita-se um maior acesso às aplicações do sistema MatROC, atualmente disponível em formato compactado .zip (aproximadamente 20KB) no endereço ftp://graco.unb.br/pub/alvares/sim/Trabalhos_sim_2000/algoritmo_roc_tecnologia_grupo/MatRC.zip.

| | | MÁQUINAS | | | | | | | | | |
|-------------|----|----------|----|----|----|----|----|----|----|----|-----|
| | | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | M9 | M10 |
| MI original | P1 | 1 | | 1 | 1 | 1 | | | 1 | 1 | |
| | P2 | | | | 1 | | | | | | |
| | P3 | 1 | | 1 | | 1 | | | 1 | 1 | |
| | P4 | | 1 | | | | | | | | 1 |
| | P5 | | 1 | | | | | | | | |
| | P6 | | | | | | 1 | 1 | | | 1 |
| | P7 | 1 | | | | | | | | | |
| | P8 | | | | | | | | | | 1 |

| | | MÁQUINAS | | | | | | | | | |
|-------------------------|----|----------|----|----|----|----|----|----|----|----|-----|
| | | M4 | M3 | M5 | M8 | M9 | M1 | M2 | M6 | M7 | M10 |
| Resultado da referência | P2 | 1 | | | | | | | | | |
| | P7 | | | | | | 1 | | | | |
| | P3 | | 1 | 1 | 1 | 1 | 1 | | | | |
| | P1 | 1 | 1 | 1 | 1 | 1 | 1 | | | | |
| | P5 | | | | | | | 1 | | | |
| | P8 | | | | | | | | | | 1 |
| | P4 | | | | | | | 1 | | | 1 |
| | P6 | | | | | | | | 1 | 1 | 1 |

| | | MÁQUINAS | | | | | | | | | |
|---------------------|----|----------|----|----|----|----|----|----|-----|----|----|
| | | M1 | M9 | M3 | M8 | M5 | M4 | M2 | M10 | M7 | M6 |
| Resultado do MatROC | P1 | 1 | 1 | 1 | 1 | 1 | 1 | | | | |
| | P3 | 1 | 1 | 1 | 1 | 1 | | | | | |
| | P7 | 1 | | | | | | | | | |
| | P2 | | | | | | 1 | | | | |
| | P4 | | | | | | | 1 | 1 | | |
| | P5 | | | | | | | 1 | | | |
| | P6 | | | | | | | | 1 | 1 | 1 |
| | P8 | | | | | | | | 1 | | |

Figura 9. Exemplo 2 - <http://iesu5.ieem.ust.hk/dfaculty/ajay/courses/ieem513/GT/GT.html>.

6. REFERÊNCIAS

- Alvares, A. J., Lafratta, F. H., Volkmer, R., Almeida, M. C., Bernard, G. S., 1991, "CIM – Computer Integrated Manufacturing, Aspectos Relevantes Para Sua Implantação" .
- Chan, C. Y., Lam, F. W., Lee, C. P., 1999, "Considerations for Using Cellular Manufacturing", *Journal of Materials Processing Technology*, Vol. 96, pp. 182-187.
- Chandrasekharan, M. P., Rajagopalan, R., "An Ideal Seed Non-hierarchical Clustering Algorithm For Cellular Manufacturing", *Int. J. Prod. Res.*, Vol 24, n° 2, pp. 451-464.
- Chandrasekharan, M. P., Rajagopalan, R., "ZODIAC - An Algorithm for Concurrent Formation of Part-families and Machine Cells", *Int. J. Prod. Res.*, Vol 25, n° 6, pp. 835-850.
- Ferreira, J.C. E., Planejamento de Processo Assistido por Computador (CAPP), Apostila do Curso de CAPP, UFSC, 1996.
- Groover, M. P., 2000, "Automation, Production Systems and Computer-Integrated Manufacturing", Ed. Prentice Hall, Upper Saddle River, NJ, Estados Unidos, 856p.
- Leal, R. A., Filho, A. N. C., Maestrelli, N. C., Batocchio, A., 1999, "Uma Aplicação do Algoritmo ROC em Manufatura Celular", *Anais do 6º Congresso Nacional dos Estudantes de Engenharia Mecânica CREEM 99*, Brasília, Brasil.
- On-line, <http://iesu5.ieem.ust.hk/dfaculty/ajay/courses/ieem513/GT/GT.html>
- On-line, <http://www.mathworks.com>
- Vieira, M., Maestrelli, N. C., Batocchio, A., 1999, "Utilização de Tecnologia de Grupo e Manufatura Celular Nas Empresas da Região de Piracicaba/SP", *Anais do 6º Congresso Nacional dos Estudantes de Engenharia Mecânica CREEM 99*, Brasília, Brasil.

DEVELOPMENT OF A FLEXIBLE MANUFACTURING CELL FORMATION SYSTEM USING MATLAB®

Guilherme Novaes Ramos

Universidade de Brasília, Departamento de Engenharia Mecânica e Mecatrônica, Grupo de Automação e Controle – Graco, 70.910-300, Brasília, email: helmetpackard@yahoo.com

Alvaro Michel Tamayo

Universidade de Brasília, Departamento de Engenharia Mecânica e Mecatrônica, Grupo de Automação e Controle – Graco, 70.910-300, Brasília, email: itotamayo@yawl.com.br

Alberto José Álvares

Universidade de Brasília, Departamento de Engenharia Mecânica e Mecatrônica, Grupo de Automação e Controle – Graco, 70.910-300, Brasília, email: alvares@AlvaresTech.com

Abstract: This paper presents the results of a research project that seeks the application of the ROC (Rank Order Clustering) algorithm using a friendly user interface in the MatLab® environment to obtain the organization of a fabric based on Group Technology (GT). The organization of Flexible Manufacturing Cells (FMC's) is a determining factor for GT implementation, which is a dominant tendency in today's industry. This philosophy seeks to reduce unproductive times associated to fabrication processes by implementing flow improvement and setup as well as inventory reductions. Consequently, productivity is increased. In order to implement GT, it is necessary to recognize the groups (or families) formed by elements that present similar characteristics of dimension or fabrication processes. A simple method to do so is to use the ROC algorithm, which obtains the groupings by re-ordering a matrix that represents the processes. Using MatLab® a friendly user interface program was created to achieve the desired results.

Keywords: MatLab®, group technology, rank order clustering, FMC, production flow analysis.