

DESENVOLVIMENTO DE UM SISTEMA PARA A FORMAÇÃO DE FMC'S UTILIZANDO O MATLAB

ALVARO MICHEL TAMAYO, GUILHERME NOVAES RAMOS

E-mails: itotamayo@yahoo.com, helmetpackard@yahoo.com

Resumo – Neste artigo apresenta-se 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 improdutivo associados a processos de fabricação, implementando melhorias de fluxo e reduções de *setup* e também de inventários em processos. Como consequência tem-se o 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 os agrupamentos pela reordenação de uma matriz de incidência que representa os processos. Utilizando o MatLab, criou-se um programa de interface amigável para obter os resultados desejados.

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, production 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 this results.

Keywords –MatLab, Group Technology.

1. Introdução

A organização ótima dos componentes de uma FMC é o objetivo a ser atingido neste tipo de trabalho. Para tal, utilizam-se os métodos baseados em formulação matricial (rápidos e de baixo custo). A primeira etapa é a representação da FMC por uma matriz de incidência (MI), de forma que:

- ↔ $a_{ij}=1$, se a peça j deve ser processada pela máquina i ;
- ↔ $a_{ij}=0$, caso contrário.

Por exemplo, a $MI_{7 \times 9}$ a seguir:

	MÁQUINAS									
P E Ç A S		M1	M2	M3	M4	M5	M6	M7	M8	M9
	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		

Em seguida, é usado um algoritmo de identificação de agrupamentos, que ordena a matriz de forma que apareçam grupos de elementos '1' na matriz.

2. Algoritmo ROC

Existem diversos algoritmos de identificação de agrupamentos, cada um com uma lógica de funcionamento diferente:

Algoritmo	Lógica de Agrupamento
ROC (Rank Order Clustering)	Valor Binário
BEA (Bond Energy Algorithm)	Medida de Efetividade
CIA (Cluster Identification Algorithm)	Densidade de Matriz
SLC (Single Linkage Algorithm)	Medida de Similaridade
DCA (Direct Clustering Analysis)	Número de elementos i

Dentre estes, foi escolhido o ROC, que se baseia no arranjo de linhas e colunas como vetores que têm valor binários. As principais vantagens do ROC são: simplicidade e facilidade de implementação. Os passos deste algoritmo são:

- 1– Obter a MI (Matriz de Incidência).
- 2– Ordenar as linhas da MI em ordem decrescente, considerando-as como vetores binários.
- 3– Reescrever a MI ordenada.
- 4– Ordenar as colunas da nova MI em ordem decrescente, considerando-as como vetores binários.
- 5– Reescrever a MI ordenada.
- 6– Verificar se a nova MI necessita de novo ordenamento. Se for preciso, retornar ao passo 2.
- 7– Apresentar a matriz final.

Para a MI apresentada, tem-se como resultado a seguinte matriz:

P E Ç A		MÁQUINAS								
		M1	M8	M2	M4	M6	M7	M9	M3	M5
	P1	1	1	1	1					
	P5	1	1							
	P4			1	1	1				
	P7			1		1	1			
	P3							1	1	1
	P6							1	1	
	P2							1		1

Onde destacou-se três famílias distintas.

3. O Programa

Desta forma, o programa a ser implementado deve:

- ↔ executar a sequência de passos do algoritmo ROC, dada uma matriz de incidência definida pelo usuário;
- ↔ ter uma interface 'amigável' com o usuário, permitindo uma fácil entrada de dados (MI), apresentando os resultados de forma clara, podendo ser interrompido ou não a critério do usuário e permitindo acesso a todas as informações úteis fornecidas pela aplicação do algoritmo.

Optou-se pelo ambiente MatLab, pois este oferece uma série de vantagens tais como:

- ↔ Linguagem de programação de alto nível.
- ↔ Fácil manipulação de vetores e matrizes.
- ↔ Possui funções para interface gráfica.

4. Desenvolvimento do Ambiente

Primeiramente, implementou-se o algoritmo em linguagem de programação. Criou-se um programa com interface em modo texto, realiza todos passos propostos pelo algoritmo.

O usuário deve inserir as dimensões da matriz e então cada um de seus termos. O processo de ordenação escolhido, apesar de não o mais eficiente, é um dos mais facilmente implementáveis. Para os propósitos desejados, é bastante eficaz. O código proposto para a ordenação de uma matriz $MI_{m \times n}$ é:

```
troca ← true; % Variável de teste de ordenação
Enquanto troca faça
    troca ← false;
    % Ordenação das linhas
    Para i de 2 até m passo 1 faça
        Para j de m até i passo -1 faça
            Se vetor_binário_linha(j) < vetor_binário_linha(j-1) faça
                Troque_linhas j e j-1
                troca ← true
            fim_Se
        fim Para
    fim Para
    % Ordenação das colunas
    Para i de 2 até n passo 1 faça
        Para j de n até i passo -1 faça
            Se vetor_binário_coluna(j) < vetor_binário_coluna(j-1) faça
                Troque_colunas j e j-1
                troca ← true
            fim_Se
        fim Para
    fim Para
fim Enquanto
```

Onde se utiliza rotinas para a troca de linhas e colunas da matriz. É importante ressaltar que estas rotinas também atualizam a ordem dos vetores que representam a sequência de máquinas e peças do chão de fábrica.

Este programa executa todas as funções corretamente, mas sua interface não é amigável. Procurou-se então desenvolver a interface para que esta seja o mais simples possível, e proporcionando o máximo de controle dos procedimentos do programa ao usuário.

De início, buscou-se o método mais rápido e fácil para inserção da matriz de incidência, já que esta etapa é a mais demorada e monótona. Utilizando laços de repetição e funções gráficas matriciais específicas do MatLab, obteve-se o resultado desejado. A matriz agora é posta de forma gráfica, o usuário indica, com o mouse, a localização dos elementos 1's da matriz (automaticamente se supões que os outros são elementos 0's). Além de mais fácil, esta aplicação permite a correção de algum dado errado sem a necessidade de se inserir toda a matriz novamente.

De posse da matriz, tem-se a opção de ordená-la, chamando o algoritmo de ordenação proposto. Há também a opção de visualizar graficamente a ordenação, verificando a nova seqüência dos vetores que definem as máquinas e peças existentes.

Implementou-se também funções para salvar/abrir os dados (MI e vetores de ordenação), de modo que se possa formar um banco de dados ou realizar o intercâmbio de informações, sem que seja necessário todo o procedimento de inserir a matriz. Há ainda uma opção para imprimir a matriz de incidência.

Para garantir o bom funcionamento do programa, criou-se testes para verificar a validade dos dados de entrada. Estes testes conferem se as dimensões da MI são válidas e existem o diretório e/ou o arquivo especificados (para salvar/abrir). Caso os dados não sejam válidos, o programa os desconsidera e avisa ao usuário.

Por fim, criou-se uma opção de ajuda, que esclarece qualquer dúvida em relação ao programa e como utilizar suas funções.

O programa funciona seguindo um esquema de janelas. Há uma janela principal que fornece acesso à todas as opções de comandos citadas, e estas são chamadas através de botões. Um procedimento chamado pode (ou não) abrir uma janela secundária, que será fechada quando este terminar de ser executado. A janela principal permanece aberta pois o programa atua em um laço de repetição, e só será fechada caso o usuário deseje. Assim, as funções ficam disponíveis.

Criou-se então uma interface amigável, que proporciona todos os passos do ROC (e alguns a mais), de tal maneira que o usuário tem total controle, mas de modo bastante simples e didático.

5. Resultados

Diversos testes foram realizados para avaliar o desempenho do ambiente. O método de inserção da MI é eficiente, garantindo que a forma gráfica de obtenção dos elementos da matriz funciona. Partindo de matrizes de incidências conhecidas (inclusive o resultado do processamento destas pelo ROC), obteve-se as MI reordenadas. Em todos os casos o ambiente forneceu o resultado correto, comprovando a eficiência do código de ordenação implementado.

A apresentação dos vetores de ordenação, assim como a opção de impressão destes e da MI também funcionam corretamente. Testou-se ainda os controles das variáveis de entrada, e verificou-se que este também é válido. As dimensões da MI só são aceitas se

forem números inteiros e positivos. Os diretórios só são válidos se realmente existirem, assim como um arquivo a ser aberto. Caso algum dado não seja aceito, há um alerta indicando ao usuário esta falha e os dados são, em seguida, descartados.

Por fim, os tópicos de ajuda não apresentam problemas, e descrevem cada procedimento que se pode utilizar adequadamente.

6. Conclusões

O algoritmo ROC é de fácil implementação, e dependendo da MI pode fornecer bons resultados (podem ou não surgir famílias destacadas). O MatLab apresenta muitas ferramentas para manipulação de matrizes e figuras, possibilitando diversas aplicações.

A implementação do programa para "Desenvolvimento de um Sistema para a 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.

7. Sugestões

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

Sugere-se também a padronização dos arquivos de dados utilizados para registro. Adotando um tipo padrão (por exemplo, .txt) facilita-se o intercâmbio de informações. Por fim, deve-se buscar a obtenção de um arquivo tipo "stand-alone", que possa ser executado fora do ambiente MatLab, expandindo as aplicações do ambiente proposto.

8. Referências

- [1] LEAL, R. A., FILHO, A. N. C., MAESTRELLI, N. C., BATOCCHIO, A. – "Uma Aplicação do Algoritmo OC em Manufatura Celular"
- [2] VIEIRA, M., MAESTRELLI, N. C., BATOCCHIO, A. – "Utilização de Tecnologia de Grupo e Manufatura Celular Nas Empresas Da Região De Piracicaba/SP"

- [3] CHANDRASEKHARAN, M. P., RAJAGOPALAN, R. – "An Ideal Seed Non-hierarchical Clustering Algorithm For Cellular Manufacturing"
- [4] GROOVER – "Automation, Production Systems and Computer-Integrated Manufacturing"
- [5] CHANDRASEKHARAN, M. P., RAJAGOPALAN, R. – "ZODIAC – An Algorithm For Concurrent Formation Of Part-families And Machine Cells"
- [6] [online] Disponível em:
<http://iesu5.ieem.ust.hk/dfaculty/ajay/courses/ieem513/GT/GT.html>
- [7] [online] Disponível em: <http://www.mathworks.com>