Algoritmos de tecnologia de grupo para projetos de células de manufatura

Alexandre Augusto Massote
Faap/FEI/Unisanta. São Paulo – SP [Brasil]
massote@fei.edu.br

Este trabalho apresenta algoritmos aplicados para solucionar problemas relacionados à formação de células de máquinas e famílias de peças, que é uma das etapas fundamentais nos projetos de células de manufatura. Para isso foram selecionados três algoritmos heurísticos: o *rank order clustering* (ROC), o *direct clustering analysis* (DCA) e o *cluster identification algorithm* (CIA), além de um modelo de Programação Inteira de Kusiak (1987) e um algoritmo usando redes neurais de Malakooti e Yang (1995). Nesta pesquisa, são apresentadas também três medidas de desempenho para avaliar a qualidade do método de agrupamento, além de ilustrações numéricas e análise dos resultados obtidos com a aplicação de cada um deles. Finalmente, são discutidas as vantagens e as limitações da aplicação desses métodos para o projeto de células de manufatura, usando-se a simulação.

Palavras-chave: CIA. DCA. ROC. Tecnologia de grupo.

1 Introdução

A tecnologia de grupo (em inglês *group technology* [GT]) é um enfoque moderno aplicado ao estudo de sistemas de manufatura que vem sendo utilizado por muitas indústrias, tais como *job shops* e de produção tipo *batch*. Destaca-se a sua importância nos projetos de células de manufatura. Tradicionalmente, para a organização dos sistemas de produção em massa, eram usados *layouts* em linha e, para outros tipos de produção, *layouts* por processos. Nesses tipos de estruturas, a redução dos tamanhos dos lotes poderia acarretar uma elevação dos custos como os de *set-up*. A GT invalidou essa relação e, em decorrência disso, obteve economia mesmo nos pequenos lotes. A GT é, portanto, uma técnica que propicia, de maneira simplificativa, a melhoria da produtividade em sistemas de produção, tais como reduções de ciclos de fabricação e de material em processo, garantindo os prazos de entrega e menor movimentação de materiais.

O problema básico que envolve a GT é a identificação de células de máquinas e das famílias de peças. Em razão disso, procura-se identificar e explorar as semelhanças entre produtos, peças e processos de manufatura. Em relação às peças, segundo Groover (1987), existem três maneiras para identificá-las: inspeção visual, classificação e codificação, e análise do fluxo de produção. Outras técnicas de análise para formação de famílias e células, tais como as de King e Nakornchai (1982) e Kusiak (1987), chamadas máquinas-peça, com base nos atributos de fabricação, ou operações necessárias para a fabricação de uma família, têm sido desenvolvidas desde a análise do fluxo de produção proposta por Burbidge (1971).

As operações necessárias para a fabricação das peças nas máquinas podem ser representadas na forma de uma matriz, chamada matriz de incidência $\{a_{ij}\}$, que tem m linhas a representar as máquinas, enquanto n colunas referem-se às peças. Considera-se $a_{ij} = 1$ para a peça j, que requer operação na máquina; e $a_{ij} = 0$ para a peça i, que não precisa de operação.

Este é o ponto de partida tanto para o desenvolvimento quanto para a implementação das várias técnicas voltadas à formação de células. O objetivo dos algoritmos de formação de famílias máquinas-peça é rearranjar as linhas e as colunas da matriz de incidência, de tal maneira que a matriz resultante fique com todos os elementos iguais a 1 agrupados em blocos na diagonal, em que cada bloco na matriz rearranjada indique um grupo de peças e o correspondente grupo de máquinas. Dessa forma, as famílias de peças e as células de máquinas podem ser identificadas. Uma perfeita estrutura de blocos diagonais poderá ser constituída caso seja possível formar células de máquinas e famílias de peças de modo que cada família possa ser processada em uma única célula de máquinas. A Ilustração 1a mostra uma matriz de incidência inicial com 10 máquinas e 12 peças. A Ilustração 1b é a matriz resultante depois do agrupamento em três células de máquinas-peça distintas. Nessa ilustração, pode ser observado que a família de peças 1 é composta das peças 1, 9, 10, 11 e 6, e a célula de máquinas 1, das máquinas 1, 7 e 10. Assim, a família de peças 1 é processada pela célula de máquinas 1; a família de peças 2, pela célula de máquinas 2, e a família 3, pela célula 3.

Quando não se consegue perfeitamente tal estrutura, as peças e as máquinas são agrupadas com o objetivo de minimizar os elementos excepcionais para um dado número de células de máquinas. Esses elementos são os que estarão fora da diagonal 1, representando o número total de processamentos que as peças terão em máquinas de outras células. O agrupamento final, na Ilustração 2, mostra essa situação. Vale observar que a matriz final será diferente quando se usar em distintos métodos de agrupamento.

Peças

Máquinas	1	2	3	4	5	6	7	8	9	10	11	12
1									1	1	1	
2		1		1			1					1
3	1				1			1				
4	1		1					1				
5		1		1			1					1
6	1		1		1			1				
7						1			1	1	1	
8		1		1			1					1
9			1		1			1				
10						1			1	1	1	

(a)

Peças

Máquinas	1	9	10	11	6	2	4	7	12	1	5	8	3
1	1	1	1	1									
7	1	1	1	1	1								
10	1	1	1	1	1								
2						1	1	1	1				
5						1	1	1	1				
8						1	1	1	1				
3										1	1	1	
4										1		1	1
6										1	1	1	1
9											1	1	1

(b)

Ilustração 1: Matriz de incidência

Obs.: antes do agrupamento (a); depois do agrupamento (b).

Fonte: O autor.

Peças

Máquinas	2	7	11	12	5	8	9	1	3	6	4	10	13
1	1	1	1	1									
6	1	1	1	1	1							1	
7	1	1	1	1							1		
2						1	1	1	1				
5						1	1	1	1				
9				1						1	1	1	
3										1		1	
4		1								1	1	1	
8												1	1

Ilustração 2: Matriz de incidência final

Fonte: O autor.

Durante as duas últimas décadas, muitos métodos foram desenvolvidos para resolver o problema referente à formação de famílias máquinas-peça. Também são considerados muitos critérios de desempenho, tais como custo total de movimentação, utilização média das máquinas, tempo médio de *set-up* e de fabricação e outros. Chu (1989) classifica esses métodos como manipulação de matriz, agrupamento hierárquico e não-hierárquico, programação matemática, técnicas gráfica e heurística. Os primeiros algoritmos desenvolvidos basearam-se, principalmente, na manipulação de matriz. Nesse método, linhas e colunas são rearranjadas para a obtenção da diagonal de blocos, da qual as células de máquinas e famílias de peças são obtidas. Pode-se citar o *rank order clustering* (ROC) de King (1980), o *direct clustering analysis* (DCA) de Chan e Milner (1982) e o *cluster identification algorithm* (CIA) de Kusiak e Chow (1987). No agrupamento hierárquico, as similaridades e as diferenças entre máquinas são computadas e agrupadas de modo que minimizem as diferenças e maximizem as similaridades. Como exemplos de algoritmos de agrupamento hierárquico, têm-se o *single linkage algorithm* (SLA) de McAuley (1972) e o *average linkage algorithm* (ALA) de Seifoddini (1989). Métodos de agrupamento não-hierárquicos diferentemente dos de agrupamento hierárquico são, por natureza, repetitivos. O não-hierárquico é usado nos algoritmos Zodiac de Chandrasekharan e Rajagopalan (1987) e Grafics de Srinivasan e Narendran (1991).

Os problemas de formação de células têm sido analisados também usando-se o modelo *p-median* de Kusiak (1987), a programação dinâmica de Steudel e Ballakur (1987) e a programação inteira de Boctor (1991). Essas técnicas são empregadas para resolver problemas de pequenas dimensões, enquanto, nos de grandes dimensões, o método heurístico tem sido usado com mais eficiência, porque os algoritmos existentes para resolver problemas de otimização demandam um grande tempo de processamento, e na maioria das vezes, um tempo inviável do ponto de vista prático. Vohra e colaboradores (1990) sugeriram um enfoque com base em redes para formação de células, e Askin (1991) vem usando o caminho hamiltoniano para rearranjar a matriz de incidência de máquinas e peças.

Vale observar que todos esses algoritmos apontam para o agrupamento de um conjunto de máquinas em células e de famílias de peças. Esse enfoque tem dado boas soluções para problemas em que as famílias de peças e as células de máquinas existam naturalmente. Entretanto, apresenta falha quando existem vários elementos excepcionais após a diagonalização dos blocos. Nesses casos, a melhor alternativa seria tentar a formação de *fractional cell*, que inclui parte das máquinas agrupadas em células e as demais agrupadas em uma célula de "resto", que funciona como um *job shop*. O uso prático deste método pode ser observado no estudo de Wemmerlov e Hyer (1989).

Segundo Murthy e Srinivasan (1995), embora a idéia de formação de *fractional cell* venha sendo mencionada na literatura de GT, como em Burbidge (1971) e Greene e Sadowski (1983), que consideram a idéia de célula-resto, observa-se pouca pesquisa nessa área. Pela constituição de célula-resto é possível reduzir o número de elementos excepcionais existente em uma grande quantidade de problemas. Na formação tipo *fractional cell,* apesar da movimentação de peças de uma célula para uma célula-resto ser permitida, ela deve ser evitada, principalmente, entre células. Na análise, enquanto o movimento de uma célula para outra é tratado como um elemento excepcional o de uma célula para uma célula-resto não é considerado como um elemento excepcional. O objetivo é minimizar os elementos excepcionais. Se uma solução é obtida sem nenhum elemento excepcional, a matriz é perfeitamente conveniente para formação de *fractional cell*, desde que exista movimentação de material somente entre células e células resto.

Recentemente, novas técnicas, tais como reconhecimento do padrão de Wu e colaboradores (1988) e de Harhalakis Nagi e Proth (1990), agrupamento *fuzzy* e sistemas especialistas (KUSIAK, 1990) têm sido aplicadas para resolver esses problemas. E, mais recentemente ainda, pesquisas com redes neurais artificiais (em inglês *artifitial neural network* [ANN]) têm mostrado a sua eficiente aplicação na solução de problemas de agrupamento máquinas-peça, especialmente os de grandes dimensões, como mostram McLave e Ramachardran (1991) e Kaparthi e Suresh (1992).

Sistemas paralelos, tais como redes neurais, podem ser usados para observar e comparar diferentes soluções em um intervalo de tempo muito pequeno e mesmo para solucionar problemas de grandes dimensões. Uma revisão detalhada dessa proposta é feita por Dagli e Huggahalli (1991; 1993).

Moon (1990) propôs uma ativação iterativa e um modelo de competição em que a rede neural estabelece, inicialmente, as similaridades das peças, das máquinas e a relação máquinas-peça obtida da matriz máquinas-peça; por isso, são agrupadas usando-se redes neurais que ativam unicamente cada grupo de peça e de máquinas. Em razão disso, esse modelo requer a computação de similaridades entre cada par de peças e de máquinas. Outra aplicação de redes neurais com o intuito de solucionar o problema de formação de células é a busca da melhoria da eficiência.

2 Alguns algoritmos aplicados em GT

A seguir serão apresentados alguns algoritmos aplicados em GT. Para verificar melhor o potencial dos métodos aplicados na solução dos problemas de formação de grupos máquinas-peça, os algoritmos foram selecionados de modo que abrangessem três grupos distintos: heurístico, otimizante e usando redes neurais. Os algoritmos heurísticos estão descritos na ordem cronológica de seus respectivos desenvolvimentos, o otimizante é um modelo de programação inteira de Kusiak (1987) enquanto o algoritmo, usando redes neurais, foi proposto por Malakooti e Yang (1995).

2.1 ROC

King (1980) desenvolveu este algoritmo que consiste nos seguintes passos:

1) Para cada linha da matriz de incidência, designar um peso binário e calcular o peso decimal equivalente;

2) Rearranjar as linhas da matriz na ordem decrescente dos valores dos pesos decimais equivalentes;

3) Para cada coluna da matriz obtida no passo 2, designar um peso binário e calcular o peso decimal equivalente;

4) Rearranjar as colunas da matriz na ordem decrescente dos valores dos pesos decimais equivalentes;

5) Repetir os passos de 1 a 4 até não haver mais mudanças de posições dos elementos em cada linha ou coluna.

Os pesos para cada linha i e coluna j são calculados da seguinte forma:

n = nº de peças e m = nº de máquinas

$$\text{linha } i: \sum_{k=1}^{n} a_{ik} \cdot 2^{n-k}$$

$$\text{coluna } j: \sum_{k=1}^{n} a_{ik} \cdot 2^{n-k}$$

2.2 DCA

Chan e Milner (1982) desenvolveram este algoritmo que consiste nos seguintes passos:

1) Determinar o número total de elementos 1 em cada linha e em cada coluna na matriz de incidência;

2) Rearranjar as linhas na ordem crescente do número total de elementos 1;

3) Rearranjar as colunas na ordem decrescente do número total de elementos 1;

4) Repetir os passos de 1 a 3 até que não haja nenhuma mudança de posição dos elementos da matriz.

2.3 CIA

Kusiak e Chow (1987) aplicaram o conceito apresentado por Iri (1968) no desenvolvimento deste algoritmo. O CIA permite checar a existência de grupos mutuamente separáveis na matriz de incidência. O algoritmo consiste nos seguintes passos:

1) Estabelecer o número da iteração k = 1;

2) Selecionar uma linha i na matriz de incidência e desenhar uma linha horizontal $h_{i;}$

3) Para cada elemento igual a 1 encontrado na linha $h_{i,}$ desenhar uma linha vertical v_j passando por esse elemento;

4) Para cada elemento igual a 1 encontrado na linha $v_{j,}$ desenhar uma linha horizontal h_k passando por esse elemento;

5) Repetir os passos 3 e 4 até não ser mais possível desenhar linhas passando por um elemento igual a 1 pertencente a uma linha v_j ou h_k. Todos os elementos que estão no cruzamento de uma linha h_k com uma linha v_j formam a célula de máquinas MC-k e a família de peças PF-k;

6) Definir uma nova matriz de incidência inicial, retirando as linhas e as colunas (máquinas e

peças) que já fazem parte de uma célula e família, obtidas nos passos de 2 a 5;

7) Se a matriz obtida contiver todos os elementos iguais a zero, significa que todas as máquinas já foram alocadas em uma célula, e todas as peças, em uma família ou obtiveram-se elementos excepcionais, que constitui o fim do algoritmo. Caso contrário, incrementar a variável k (k = k + 1) que controla o número de iterações e voltar para o passo 3.

2.4 Programação inteira

A maioria dos modelos matemáticos desenvolvidos para a solução de problemas de formação de grupos considera a medida de distância d_{ij} entre as peças i e j (Kusiak, 1987). A distância d_{ij} é uma função simétrica real com as seguintes características:

Reflexiva: $d_{ii} = 0$
Simétrica: $d_{ij} = d_{ji}$
Desigualdade triangular: $d_{iq} \leq d_{ip} + d_{pq}$

As medidas de distância mais usadas em problemas de formação de grupos são:

1) Minkowski

$$d_{ij} = \left(\sum_{k=1}^{m} |a_{ki} - a_{kj}|^{r} \right)^{1/r}$$

Onde:
r = inteiro positivo
m = número de máquinas

Dois casos especiais de r são os mais usados:

Distância absoluta: r = 1
Distância euclidiana: r = 2

2) Minkowski ponderado

$$d_{ij} = \left(\sum_{k=1}^{m} w_k |a_{ki} - a_{kj}|^{r} \right)^{1/r}$$

3) Hamming

$$d_{ij} = \sum_{k=1}^{m} \delta(a_{ki}, a_{kj})$$

Onde:
$\delta(a_{ki}, a_{kj}) = 1$ se $a_{ki} \; \pi a_{kj}$
$\delta(a_{ki}, a_{kj}) = 0$, caso contrário

Kusiak (1987) desenvolveu um modelo de programação inteira, conhecido como *p-median model*, que é usado para agrupar n peças e p em famílias. As seguintes variáveis são definidas:

m = número de máquinas
n = número de peças
p = número de famílias de peças
$x_{ij} = 1$ se a peça i pertencer à família j
$x_{ij} = 0$ caso contrário
d_{ij} = medida de distância entre as peças i e j

A função do objetivo é minimizar a soma total das distâncias entre as peças i e j

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} \cdot x_{ij}$$

sujeito a:

$$\sum_{j=1}^{n} x_{ij} = 1 \text{ , para todos } i = 1, \ldots, n$$

$$\sum_{j=1}^{n} x_{jj} = p$$

$x_{ij} \leq x_{jj}$, para todo i = 1, ..., n e j = 1, ..., n
x_{ij} = 0 ou 1, para todo i = 1, ..., n e j = 1, ..., n

Esse modelo foi desenvolvido considerando-se a possibilidade de haver apenas um plano de operações para cada peça i. Kusiak (1987) desenvolveu um outro modelo, chamado *generalized p-median model*, que relaxa em relação a essa restrição, ao considerar a existência de mais que um plano de operações para cada peça i, sendo, adicionalmente, os custos de operação associados a cada um desses planos.

2.5 *Clustering neural network* (CNN)

Malakooti e Yang (1995) desenvolveram um algoritmo usando redes neurais de aprendizado não-supervisionado para problemas de formação de grupos. O algoritmo é composto de duas fases. Na fase 1, os centros das células são identificados. Na fase 2, o agrupamento das máquinas é feito com base na distância entre o vetor de máquinas e os centros e entre os limites superior e inferior do número de máquinas designado para cada célula. O agrupamento das peças é feito de acordo com o mesmo procedimento.

Definições:

K_i: Coeficiente na fórmula euclidiana

R: Número de nós de saída (número de centros de células)

m: Número de entradas-padrão (vetor de máquinas)

$w_r(t)$: Vetor peso (vetor do centro da célula)

a_s: Vetor de entrada (vetor de máquinas)

Ed_r: Distância euclidiana entre o vetor peso $w_r(t)$ e o vetor de entrada a_s

$\beta(t)$: Taxa de aprendizagem

δ: Parâmetro de parada

t: Índice de tempo de treinamento (índice de iteração)

g_r: Número de máquinas pertencentes à célula r

$g_{r,min}$: Número mínimo de máquinas na célula r

$g_{r,max}$: Número máximo de máquinas na célula r

Os passos deste algoritmo são os seguintes:

Fase 1: Para encontrar os centros das células.

1) Colocar k_i, β, δ, p e R;

2) Estabelecer t = 1. Gerar vetor peso inicial $w_r(1)$; $\Delta w_r(1) = 0$, r = 1, 2, ..., R. Colocar s = 1;

3) Entrar com o vetor a_s;

4) Computar a distância euclidiana entre a entrada-padrão e todos os vetores peso

$$Ed_r = \| a_s - w_r(t) \| k^2 = k_1 (a_{s1} - w_{r1}(t))^2 + k_2 (a_{s2} - w_{r2}(t))^2 + \dots + k_n (a_{sn} - w_{rn}(t))^2$$

para

$$r = 1, 2, \dots, R$$

5) Encontrar a menor distância $Ed_{r^*} = \min \{Ed_r\}$, onde r = 1, 2, ..., R;

6) Desde então, a entrada-padrão a_s pertence ao r^*, atualizar

$$\Delta w_{r^*}(t+1) = \beta (w_s - w_{r^*}(t)) + (1 - \beta) \Delta w_{r^*}(t)$$

$$\Delta w_{r^*}(t + 1) = \Delta w_{r^*}(t)$$

para

$$r = 1, 2, \dots, R \text{ e } r \neq r^*$$

7) Estabelecer s = s + 1

Se s < p, t = t + 1 e retornar para o passo 3

Caso contrário, seguir para o passo 8

8) s = p

se $\Delta w_r(t + 1) < \delta$ para r = 1, 2, ..., R fim

Caso contrário, estabelecer s = 1, t = t + 1, e retornar para o passo 3.

Fase 2: Para agrupar o vetor de máquinas m em R células com um dado limite inferior e superior para o número de máquinas em uma célula.

1) Agrupar m máquinas em R células

Estabelecer s = 1, e $g_r = 0$, para r = 1, 2,..., R

A máquina s pertencerá à célula r se $d(a_{s,}\ x_r) \leq d(a_{s,}\ x_p)$ para p = 1, 2,..., R, $p \neq r$

Estabelecer $g_r = g_r + 1$

Se s = m, seguir para o passo 2, caso contrário estabelecer s = s + 1 e retornar ao início do passo 1.

2) Se o decisor estiver satisfeito com o agrupamento fim, solicitar para ele o fornecimento de $g_{r,max}$ e $g_{r,min,}$ que correspondem aos limites superior e inferior do número de máquinas na célula r, r = 1, 2,..., R.

3) Checar se o limite superior está satisfeito.

Para r = 1, 2,..., R, se $g_r \leq g_{r,max}$, seguir para o passo 4.

Para cada célula cujo $g_r > g_{r,max}$, encontrar as $(g_r - g_{r,max})$ máquinas na célula r mais distantes do centro da célula $x_{r,}$ e designá-las para outra célula (cujo $g_r < g_{r,max}$) de acordo com as suas distâncias dos centros das células $x_{1,}\ x_{2,}..., x_{r-1,}\ x_{r+1,}..., x_r$. Retornar para o início do passo 3.

4) Checar se o limite inferior está satisfeito.

Para r = 1, 2,..., R, se $g_r \geq g_{r,min}$, fim.

Para cada célula cujo $g_r < g_{r,min}$, encontrar as $(g_{r,min} - g_r)$ máquinas de outra célula (cujo $g_r > g_{r,min}$) mais próximas do centro da célula x_r, e designá-las para a célula x_r. Retornar para o início do passo 4.

3 Ilustração numérica

Para entender melhor a aplicação dos algoritmos ou métodos apresentados no item (2), tomemos como exemplo o problema cuja matriz de incidência é a seguinte:

Máquinas \ Peças	1	2	3	4	5	6	7
1		1		1			1
2			1		1		
3	1	1		1			1
4	1		1			1	
5			1	1	1	1	

Ilustração 3: Matriz de incidência aplicada em GT

Fonte: O autor.

3.1 Aplicando o ROC

1) Para cada linha da matriz de incidência, designar um peso binário e calcular o peso decimal equivalente:

Máquinas \ Peças	1	2	3	4	5	6	7	
Peso binário	2^6	2^5	2^4	2^3	2^2	2^1	2^0	Peso decimal
1		1		1			1	41
2			1		1			20
3	1	1		1			1	105
4	1		1			1		82
5			1	1	1	1		30

Ilustração 4: Passo 1 da aplicação de ROC

Fonte: O autor.

2) Rearranjar as linhas da matriz na ordem decrescente dos valores dos pesos decimais equivalentes:

Máquinas \ Peças	1	2	3	4	5	6	7
3	1	1		1			1
4	1		1				1
1		1		1			1
5			1	1	1	1	
2			1		1		

Ilustração 5: Passo 2 da aplicação de ROC

Fonte: O autor.

3) Para cada coluna da matriz obtida no passo 2, designar um peso binário e calcular o peso decimal equivalente:

Máquinas		Peças 1	2	3	4	5	6	7
3	2^4	1	1		1			1
4	2^3	1		1				1
1	2^2		1		1			1
5	2^1			1	1	1	1	
2	2^0			1		1		
Peso decimal		24	20	11	22	3	10	20

Ilustração 6: Passo 3 da aplicação de ROC

Fonte: O autor.

4) Rearranjar as colunas da matriz na ordem decrescente dos valores dos pesos decimais equivalentes:

Máquinas	Peças 1	4	2	7	3	6	5
3	1	1	1	1			
4	1				1	1	
1		1	1	1			
5		1			1	1	1
2					1		1

Ilustração 7: Passo 4 da aplicação de ROC

Fonte: O autor.

5) Repetir os passos de 1 a 4 até não haver mais mudanças de posições dos elementos em cada linha ou coluna.

Como não há mais mudanças de posição, a solução encontrada é:

Máquinas	Peças 1	4	2	7	3	6	5
3	1	1	1	1			
4	1				1	1	
1		1	1	1			
5		1			1	1	1
2					1		1

Ilustração 8: Passo 5 da aplicação de ROC

Fonte: O autor.

3.2 Aplicando o DCA

1) Determinar o número total de elementos 1 em cada linha e em cada coluna na matriz de incidência:

Máquinas	Peças 1	2	3	4	5	6	7	
1		1		1			1	3
2			1		1			2
3	1	1		1			1	4
4	1		1			1		3
5			1	1	1	1		4
	2	2	3	3	2	2	2	

Ilustração 9: Passo 1 da aplicação DCA

Fonte: O autor.

2) Rearranjar as linhas na ordem crescente do número total de elementos 1:

Máquinas	Peças 1	2	3	4	5	6	7
2			1		1		
1		1		1			1
4	1		1			1	
3	1	1		1			1
5			1	1	1	1	

Ilustração 10: Passo 2 da aplicação DCA

Fonte: O autor.

3) Rearranjar as colunas na ordem decrescente do número total de elementos 1:

Máquinas	Peças 1	2	5	6	7	3	4
2			1			1	
1		1			1		1
4	1			1		1	
3	1	1			1		1
5			1	1		1	1

Ilustração 11: Passo 3 da aplicação DCA

Fonte: O autor.

4) Repetir os passos de 1 a 3 até que não haja nenhuma mudança de posição dos elementos da matriz.

Fazendo-se as mudanças necessárias, com a repetição dos passos 2 e 3, a solução encontrada é:

Máquinas \ Peças	2	4	7	1	3	5	6
1	1	1	1				
3	1	1	1	1			
5		1			1	1	1
2					1	1	
4				1	1		1

Ilustração 12: Passo 4 da aplicação DCA

Fonte: O autor.

3.3 Aplicando o CIA

1) Selecionar uma linha i na matriz de incidência e desenhar uma linha horizontal h_i:

Máquinas \ Peças	1	2	3	4	5	6	7
1		1		1			1
2			1		1		
3	1	1		1			1
4	1		1			1	
5			1	1	1	1	

Ilustração 13: Passo 1 da aplicação CIA

Fonte: O autor.

2) Para cada elemento igual a 1 encontrado na linha $h_{i,}$ desenhar uma linha vertical v_j passando por esses elementos:

Máquinas \ Peças	1	2	3	4	5	6	7
1		1		1			1
2			1		1		
3	1	1		1			1
4	1		1			1	
5			1	1	1	1	

Ilustração 14: Passo 2 da aplicação CIA

Fonte: O autor.

3) Para cada elemento igual a 1 encontrado na linha $v_{j,}$ desenhar uma linha horizontal h_k passando por esses elementos:

Máquinas \ Peças	1	2	3	4	5	6	7
1		1		1			1
2			1		1		
3	1	1		1			1
4	1		1			1	
5			1	1	1	1	

Ilustração 15: Passo 3 da aplicação CIA

Fonte: O autor.

4) Repetir os passos 2 e 3 até não ser mais possível desenhar linhas passando por um elemento igual a 1 pertencente a uma linha v_j ou h_k. Todos os elementos que estão no cruzamento de uma linha h_k com uma linha v_j formam a célula de máquinas MC-k e a família de peças PF-k:

Máquinas \ Peças	1	2	3	4	5	6	7
1		1		1			1
2			1		1		
3	1	1		1			1
4	1		1			1	
5			1	1	1	1	

Ilustração 16: Passo 4 da aplicação CIA

Fonte: O autor.

Como se pôde observar, o algoritmo não encontrou uma solução para esse problema, ou seja, a solução final indica a existência de uma única célula de máquinas e uma única família de peças, iguais à matriz de incidência.

3.4 Aplicando a programação inteira

Usando-se Hamming para determinar a medida de distância, obtém-se a seguinte matriz:

p = 2 (duas famílias de peças)

Peças

	1	2	3	4	5	6	7	
	1	0	2	3	3	4	2	2
	2	2	0	5	2	4	4	0
	3	3	5	0	4	1	1	5
d_{ij} =	4	3	2	4	0	3	3	1
	5	4	4	1	3	0	2	4
	6	2	4	1	3	2	0	4
	7	2	0	5	1	4	4	0

Ilustração 17: Passo 1 da aplicação de programação inteira

Fonte: Os autores.

O modelo é:

Min 2X12 + 3X13 + 3X14 + 4X15 + 2X16 + 2X17 +
2X21 + 5X23 + 2X24 + 4X25 + 4X26 +
3X31 + 5X32 + 4X34 + 1X35 + 1X36 + 5X37 +
3X41 + 2X42 + 4X43 + 3X45 + 3X46 + 1X47 +
4X51 + 4X52 + 1X53 + 3X54 + 2X56 + 4X57 +
2X61 + 4X62 + 1X63 + 3X64 + 2X65 + 4X67 +
2X71 + 5X73 + 1X74 + 4X75 + 4X76

Sujeito a:

X11 + X12 + X13 + X14 + X15 + X16 + X17 = 1
X21 + X22 + X23 + X24 + X25 + X26 + X27 = 1
X31 + X32 + X33 + X34 + X35 + X36 + X37 = 1
X41 + X42 + X43 + X44 + X45 + X46 + X47 = 1
X51 + X52 + X53 + X54 + X55 + X56 + X57 = 1
X61 + X62 + X63 + X64 + X65 + X66 + X67 = 1
X71 + X72 + X73 + X74 + X75 + X76 + X77 = 1
X11 + X22 + X33 + X44 + X55 + X66 + X77 = 2
X21 - X11 ≤ 0
X31 - X11 ≤ 0
X41 - X11 ≤ 0
X51 - X11 ≤ 0
X61 - X11 ≤ 0
X71 - X11 ≤ 0
X12 - X22 ≤ 0
X32 - X22 ≤ 0
X42 - X22 ≤ 0
X52 - X22 ≤ 0
X62 - X22 ≤ 0
X72 - X22 ≤ 0
X13 - X33 ≤ 0
X23 - X33 ≤ 0
X43 - X33 ≤ 0
X53 - X33 ≤ 0
X63 - X33 ≤ 0
X73 - X33 ≤ 0
X14 - X44 ≤ 0
X24 - X44 ≤ 0
X34 - X44 ≤ 0
X54 - X44 ≤ 0
X64 - X44 ≤ 0
X74 - X44 ≤ 0
X15 - X55 ≤ 0
X25 - X55 ≤ 0
X35 - X55 ≤ 0
X45 - X55 ≤ 0
X65 - X55 ≤ 0
X75 - X55 ≤ 0
X16 - X66 ≤ 0
X26 - X66 ≤ 0
X36 - X66 ≤ 0
X46 - X66 ≤ 0
X56 - X66 ≤ 0
X76 - X66 ≤ 0
X17 - X77 ≤ 0
X27 - X77 ≤ 0
X37 - X77 ≤ 0
X47 - X77 ≤ 0
X57 - X77 ≤ 0
X67 - X77 ≤ 0

Resolvendo-se o modelo anterior, obtém-se a seguinte solução:

X17 = X27 = X47 = X77 = 1

X33 = X53 = X63 = 1

Portanto:

Família 1 { 1, 2, 4 e 7}

Família 2 { 3, 5 e 6}

Peças

Máquinas	1	2	4	7	3	5	6
3	1	1	1	1			
1		1	1	1			
4	1				1		1
5			1		1	1	1
2					1	1	

Ilustração 18: Matriz final de programação inteira

Fonte: O autor.

3.5 Aplicando o CNN

a_1 = [0 1 0 1 0 0 1]
a_2 = [0 0 1 0 1 0 0]
a_3 = [1 1 0 1 0 0 1]
a_4 = [0 1 0 1 0 0 1]
a_5 = [1 0 1 1 1 1 0]
p = 5

Fase 1

Interação 1

1) Estabelecer $k_1 = k_2 = \ldots = k_n = 1$, $\beta = 0{,}5$, $\pi = 0{,}5$

$\mu = 4$, $m = p = 5$

2) $w_1(1) = a_1 = [0\ 1\ 0\ 1\ 0\ 0\ 1]$, $s = 1$; $\Delta w_i(1) = 0$, para $i = 1, 2, \ldots, 5$

3) $t = 2$

$a_2 = [0\ 0\ 1\ 0\ 1\ 0\ 0]$

4) Computar $Ed_1 = \| a_2 - w_1(1) \|^2 = 5$

5) Estabelecer $Ed_{1^*} = Ed_1 = 5$

6) $Ed_{1^*} > \mu = 4$

$s = s + 1 = 2$, produz um novo centro de grupo $w_2(2) = a_2 = [0\ 0\ 1\ 0\ 1\ 0\ 0]$

$\Delta w_1(2) = \Delta w_2(2) = \ldots = \Delta w_5(2) = [0\ 0\ 0\ 0\ 0\ 0\ 0]$

$w_1(2) = w_1(1) + \Delta w_1(2) = [0\ 1\ 0\ 1\ 0\ 0\ 1]$

7) $t = 2 < p = 5$, retornar para o passo 3

Interação 2

3) $t = 3$, $a_3 = [1\ 1\ 0\ 1\ 0\ 0\ 1]$

4) Computar $Ed_1 = \| a_3 - w_1(2) \|^2 = 1$

$Ed_2 = \| a_3 - w_2(2) \|^2 = 6$

5) $Ed_{i^*} = Ed_1 = 1$

6) $Ed_1 = 1 < \mu = 4$

Então, a_3 pertence ao grupo 1

$\Delta w_1(3) = 0{,}5(x_3 - w_1(2)) + 0{,}5\Delta w_1(2) = [0{,}5\ 0\ 0\ 0\ 0\ 0\ 0]^T$

$\Delta w_2(3) = [0\ 0\ 0\ 0\ 0\ 0\ 0]$

$w_1(3) = w_1(2) + \Delta w_1(3) = [0{,}5\ 1\ 0\ 1\ 0\ 0\ 1]$

$w_2(3) = w_2(2) = [0\ 0\ 1\ 0\ 1\ 0\ 0]$

7) $t = 3 < p = 5$, retornar para o passo 3

Interação 3

3) $t = 4$, $a_4 = [1\ 0\ 1\ 0\ 0\ 1\ 0]$

4) $Ed_1 = 5{,}25$

$Ed_2 = 3$

5) $Ed_{i^*} = Ed_2 = 3$

6) $Ed_2 = 3 < \mu = 4$

então, a_4 pertence ao grupo 2

$\Delta w_1(4) = \Delta w_1(3) = [0{,}5\ 0\ 0\ 0\ 0\ 0\ 0]$

$\Delta w_2(4) = 0{,}5(a_4 - w_2(3)) + 0{,}5\Delta w_2(3) = [0{,}5\ 0\ 0\ 0\ -0{,}5\ 0{,}5\ 0]$

$w_1(4) = w_1(3) = [0{,}5\ 1\ 0\ 1\ 0\ 0\ 1]$

$w_2(4) = w_2(3) + \Delta w_2(4) = [0{,}5\ 0\ 1\ 0\ 0{,}5\ 0{,}5\ 0]$

Fase 2: No exemplo, não serão considerados os limites superior e inferior. O agrupamento resultante para as células de máquinas é:

Célula 1: a_1 e a_3 (máquinas 1 e 3)

Célula 2: a_2, a_4 e a_5 (máquinas 2, 4 e 5)

As peças são agrupadas em famílias, usando-se o mesmo procedimento.

O agrupamento resultante para as famílias de peças é:

Família 1: peças 1, 2, 4 e 7.

Família 2: peças 3, 5 e 6.

Máquinas \ Peças	1	2	4	7	3	5	6
1		1	1	1			
3	1	1	1	1			
2					1	1	
4	1				1		1
5			1		1	1	1

Ilustração 19: Matriz final de CNN

Fonte: O autor.

4 Medidas de desempenho

Existem vários parâmetros ou metodologias para avaliar o desempenho de um algoritmo. Para os algoritmos aplicados na solução de problemas de formação de famílias e células, chamados máquinas-peça, Malakooti e Yang (1995), em seus estudos, propuseram três medidas.

4.1 Porcentagem de elementos excepcionais (PEE)

A qualidade do método de agrupamento pode ser avaliada pelo número de elementos excepcionais (CHAN; MILNER, 1982). A porcentagem de elementos excepcionais pode ser obtida dividindo-se o número de elementos excepcionais (NEE) pelo número total de elementos com valor 1 (N) na matriz final.

$$PEE = NEE / N$$

4.2 Utilização das máquinas (UM)

É a porcentagem do tempo em que as máquinas de cada grupo estão em produção (CHANDRASEKHARAN; RAJAGOPALAN; MODROC, 1986). Sendo N1 o número total de 1 nos grupos, R o número de grupos, m_r o número de máquinas pertencentes ao grupo r e n_r o número de peças que compôem o grupo r, o UM pode ser calculado da seguinte forma:

$$UM = N1 / \left(\sum_{r=1}^{R} m_r n_r \right)$$

4.3 Eficiência de agrupamento (EA) (em inglês *grouping efficiency*)

É uma medida de desempenho de agrupamento agregada, definida por Chandrasekharan, Rajagopalan e Modroc (1986). Sendo NEE o número de elementos excepcionais, MN as dimensões da matriz de incidência e m_r e n_r como já definidos anteriormente, o EA pode ser calculado da seguinte forma:

$$EA = 0{,}5\, UM + 0{,}5\, [1 - NEE / (MN - \sum_{r=1}^{R} m_r n_r)]$$

4.4 Aplicando as medidas de desempenho no problema da ilustração numérica

Tabela 1: Desempenho dos algoritmos

	PEE (%)	UM (%)	EA (%)
ROC	18,8	72,2	77,3
DCA	12,5	82,4	85,6
CIA	–	–	–
Programação inteira	12,5	82,4	85,6
CNN	12,5	82,4	85,6

Fonte: O autor.

5 Considerações finais

Os algoritmos heurísticos ROC e DCA, a programação inteira (método otimizante) e o CNN obtiveram os mesmos desempenhos para as três medidas consideradas (porcentagem de elementos excepcionais, utilização das máquinas e eficiência de agrupamento). O CIA, embora para o exemplo apresentado não tenha obtido solução, tem apresentado um bom desempenho em várias aplicações práticas. Os aspectos computacionais requeridos, apesar de não serem o enfoque deste trabalho, mesmo para a programação inteira (utilizado o *software* Lindo) foram compatíveis e viáveis quando aplicados ao problema estudado. Entretanto, deveriam ser estudados, de forma mais profunda, problemas de dimensões maiores para que se pudesse determinar o método de solução (ou algoritmo), ideal do ponto de vista computacional, *versus* as dimensões dos problemas (matriz de incidência). Como proposta para estudos futuros, aconselhamos a busca em determinar com quais tipos de problemas os algoritmos estudados tendem a falhar ou não propiciam soluções satisfatórias.

Referências

BOCTOR, F. F. A linear formulation of the machine-part cell formation problem. *International Journal of Production Research*, Londres, v. 29, p. 343-56, 1991.

BURBIDGE, J. L. Production flow analysis. *The Production Engineer*, v. 50, n. 4-5, p. 139-152, 1971.

CHAN, H. M.; MILNER, D. A., Direct clustering algorithm for group formation in cellular manufacture. *Journal of Manufacturing Systems*, Dearborn, v. 1, n. 1, p. 65-74, 1982.

CHANDRASEKHARAN, M. P.; RAJAGOPALAN, R. Zodiac: an algorithm for concurrent formation of part-families and machine-cells. *International Journal of Production Research*, Londres, v. 25, n. 6, p. 835-850, 1987.

CHANDRASEKHARAN, M. P.; RAJAGOPALAN, R., Modroc: an extension of rank order clustering for group technology. *International Journal of Production Research*, Londres, v. 24, n. 5, p. 1.221-1.233, 1986.

CHU, C. H. Clustering analysis in manufacturing cellular formation. *Omega: International Journal of Management Sciences*, v. 17, n. 3, p. 289-295, 1989.

DAGLI, C. H.; HUGGAHALLI, R. A neural network approach to group technology. In: WANG. J.; TAKEFUJI, Y. *Neural networks in design and manufacturing*. 1. ed. River Edge: World Scientific Publishing, 1993. p. 1-55.

DAGLI, C. H.; HUGGAHALLI, R. Neural network approach to group technology. In: SHARDA R.; CHEUNG, J. Y.; COCHRAN, W. J. (Ed.). *Knowledge-based systems and neural networks: techniques and applications*. 1. ed. Nova York: Elsevier, 1991. p. 213-228.

GREENE, T. J.; SADOWSKI, R. Cellular manufacturing control. *Journal of Manufacturing Systems*, v. 12, n. 2, p. 137-146, 1983.

GROOVER, M. *Automation, production systems and computer integrated manufacturing*. 2. ed. Nova Jersey: Prentice Hall, 1987.

HARHALAKIS, G.; NAGI, R.; PROTH, J. M. An efficient heuristic in manufacturing cell formation for group technology applications. *International Journal of Production Research*, Londres, v. 28, n. 1, p. 185-198, 1990.

KAPARTHI, S.; SURESH, N. C. Machine-component cell formation in group technology: a neural network approach. *International Journal of Production Research*, Londres, v. 30, n. 6, p. 1.353-1.367, 1992.

KING, J. R. Machine-component grouping in production flow analysis: an approach using a rank order clustering algorithm. *International Journal of Production Research*, Londres, v. 18, n. 2, p. 213-232, 1980.

KING, J. R.; NAKORNCHAI, V. Machine-component group formation in group technology: review and extension. *International Journal of Production Research*, Londres, v. 20, n. 2, p. 117-133, 1982.

KUSIAK, A. *Intelligent manufacturing systems*. 1. ed. Englewood Cliffs: Prentice Hall, 1990.

KUSIAK, A. The generalized group technology concept. *International Journal of Production Research*, Londres, v. 25, n. 4, p. 561-569, 1987.

KUSIAK, A.; CHOW, W. S. Efficient solving of the group technology problem. *Journal of Manufacturing Systems*, v. 6, n. 2, p. 117-124, 1987.

MALAKOOTI, B.; YANG, Z. A variable-parameter unsupervised learning clustering neural network approach with application to machine-part group formation. *International Journal of Production Research*, Londres, v. 33, n. 9, p. 2.395-2.413, 1995.

MCAULEY, J. Machine grouping for efficient production. *The Production Engineer*, v. 51, n. 2, p. 53-57, 1972.

MOON, Y. Interactive activation and competition model for machine-part family formation in Group technology. *Proceedings of the International Journal Conference on Neural Networks*, Washington, v. 2, p. 667-670, 1990.

MURTHY, V. R.; SRINIVASAN, G. Fractional cell formation in group technology. *International Journal of Production Research*, Londres, v. 33, n. 5, p. 1.323-1.337, 1995.

SEIFODDINI, H. A note on the similarity coefficient method and the problem of improper machine assignment in group technology applications. *International Journal of Production Research*, Londres, v. 27, n. 7, p. 1.161-1.165, 1989.

SRINIVASAN, G.; NARENDRAN, T. T. Grafics: a nonhierarchical clustering algorithm for group technology. *International Journal of Production Research*, Londres, v. 29, n. 3, p. 463-478, 1991.

STEUDEL; H. J. BALLAKUR, A. A dynamic programming based heuristic for machine grouping in manufacturing cell formation. *Computers and Industrial Engineering*, Tarrytown, v. 12, n. 3, p. 215-222, 1987.

VOHRA, T. et al. A network approach to cell formation in cellular manufacturing. *International Journal of Production Research*, Londres, v. 28, p. 2.075-2.084, 1990.

WEMMERLOV, U.; HYER, N. L. Cellular manufacturing in the U. S. industry: a survey of users. *International Journal of Production Research*, Londres, v. 27, n. 9, p. 1.511-1.530, 1989.

Para referenciar este texto

MASSOTE, A. A. Algoritmos de tecnologia de grupo para projetos de células de manufatura. *Exacta*, São Paulo, v. 4, n. especial, p. 31-44, 25 nov. 2006.