

## **RADIX ROC: UM ALGORITMO DE BAIXA COMPLEXIDADE PARA A FORMAÇÃO DE CÉLULA DE MANUFATURA**

### **RADIX ROC: A LOW COMPLEXITY ALGORITHM FOR THE CELL MANUFACTURING DESIGN**

Marcos Ricardo Rosa Georges\* E-mail: [marcos.georges@puc-campinas.edu.br](mailto:marcos.georges@puc-campinas.edu.br)

\*Pontifícia Universidade Católica de Campinas (PUC/Campinas), Campinas, SP

**Resumo:** O Sistema de Manufatura Celular tem sido amplamente utilizado na indústria como alternativa para o incremento da capacidade de competição. Os benefícios do uso da manufatura celular incluem o aumento da flexibilidade, redução do estoque em processo e redução do tempo total de processamento. O primeiro passo na adoção da manufatura celular nas indústrias é o projeto das células, que é feito a partir de uma análise na matriz de incidência máquina peça. Existem vários algoritmos para realizar a análise desta matriz em busca de agrupamentos que poderão dar origem as células de manufatura. Este artigo propõe um algoritmo de baixa complexidade computacional chamado RADIX ROC. Este algoritmo é baseado no clássico algoritmo ROC (Rank Order Clustering). Uma revisão da literatura sobre o assunto é realizada, assim como uma análise comparativa entre alguns algoritmos, para demonstrar a baixa complexidade do algoritmo proposto.

**Palavras-chave:** Célula de Manufatura. Tecnologia de Grupo. Formação de Célula. Análise de Grupo. RADIX ROC.

**Abstract:** The Cellular Manufacturing System has been widely used in industry as an alternative to increase the ability to compete. The benefits of the use of cellular manufacturing include increased flexibility, reduction of work in process and reducing the total processing time. The first step in the adoption of cellular manufacturing industries is in the design of the cell, which is made from an analysis of the incidence matrix. There are several algorithms to perform the analysis of this matrix to seek groups that may provide sources of manufacturing cells. This article proposes an algorithm of low computational complexity called RADIX ROC. This algorithm is based on the classic algorithm ROC (Rank Order Clustering). A comparative analysis of some algorithms is done at the end to demonstrate the low complexity of the algorithm and a review of the literature on the subject completes this work.

**Keywords:** Cellular Manufacturing Systems. Group Technology. Cell Formation. Cluster Analysis. RADIX ROC.

## **1 INTRODUÇÃO**

A manufatura celular e tecnologia de grupo estão amplamente difundidas na academia e na indústria. São metodologias de gestão que têm sido extensivamente objeto de estudo e aplicação no chão de fábrica nas indústrias do mundo inteiro.

A origem da tecnologia de grupo e da manufatura celular é atribuída aos trabalhos de Mitrofanov, intitulado “*Scientific Principles of Group Technology*” de 1961, e John L. Burbidge, intitulado “*Production Flow Analysis*” de 1963. Mas, foi em 1975 que a tecnologia de grupo ganhou projeção mundial com a publicação do livro

“*The Introduction of Group Technology*” de John L. Burbidge, considerada a primeira obra completa sobre o assunto que se tornou amplamente disponível nas bibliotecas das universidades e salas dos gerentes de produção.

No entanto, mesmo desfrutando de grande notoriedade e sendo estudado e aplicado há décadas, a manufatura celular e a tecnologia de grupo ainda hoje fornecem problemas capazes de instigar a mente de pesquisadores a produzirem novos conhecimentos neste campo do conhecimento que parece inesgotável.

Entre os problemas que ainda instigam a pesquisa de novos métodos de formação de célula de manufatura no contexto da tecnologia de grupo, Krushinsky e Goldengorin (2012), apontam para a busca por métodos de menor complexidade algorítmica, métodos exatos (não heurísticos) e métodos capazes de otimizar a formação de células de manufatura mediante um objetivo explícito, por exemplo, a minimização de movimentos entre células.

Motivado pela busca por métodos de formação de células de manufatura de baixa complexidade, este artigo propõe um algoritmo de formação de célula de manufatura de baixa complexidade chamado de *RADIX ROC*.

Logo, apresentar o algoritmo *RADIX ROC*, comparar sua complexidade algorítmica segundo a notação  $O(.)$  com outros algoritmos para formação de célula de manufatura e ilustrar sua aplicação através de um exemplo são os objetivos deste artigo.

Para cumprir estes objetivos, o artigo foi organizado em seções, sendo esta primeira, introdutória. Na seção 2, apresenta-se uma breve revisão teórica sobre tecnologia de grupo, manufatura celular e algoritmos de formação de célula de manufatura. Na seção 3, apresenta-se a metodologia utilizada; e na seção 4, apresenta-se o algoritmo *RADIX ROC* proposto neste artigo. Por fim, na seção 5, uma comparação entre o algoritmo *RADIX ROC* e outros algoritmos para formação de células de manufatura é feita para demonstrar sua baixa complexidade algorítmica e menor esforço computacional.

## **2 TECNOLOGIA DE GRUPO E MANUFATURA CELULAR**

O conceito de tecnologia de grupo e manufatura celular surgiu como alternativa ao sistema de produção em série e ao sistema de produção funcional,

caracterizando-se pela disposição de um conjunto de máquinas em células de manufatura.

Segundo Massote (2006), a Tecnologia de Grupo é um enfoque moderno aplicado ao estudo dos sistemas de manufatura que vem sendo utilizado por muitas indústrias, tais como *job shop* e *batch shop*. Propicia redução do ciclo de fabricação e de material em processo, menor movimentação de material e aumento da produtividade devido a simplificação do processo produtivo.

No início da década de 90, Kusiak (1990) publicou os resultados de uma pesquisa que enumerava e quantificava as vantagens da tecnologia de grupo quando comparado ao sistema de produção em série e ao sistema de produção funcional. Embora seja uma publicação relativamente antiga, os dados são válidos para demonstrar o impacto que esta tecnologia propicia em relação aos demais sistemas de manufatura:

- Redução do *lead time* de produção: de 20% a 80%;
- Redução do estoque em processo: acima de 88%;
- Redução do uso de ferramental: de 20% a 30%;
- Redução de tempo de *setup*: de 20% a 60%;
- Redução do retrabalho e desperdício de material: 15% a 75%;

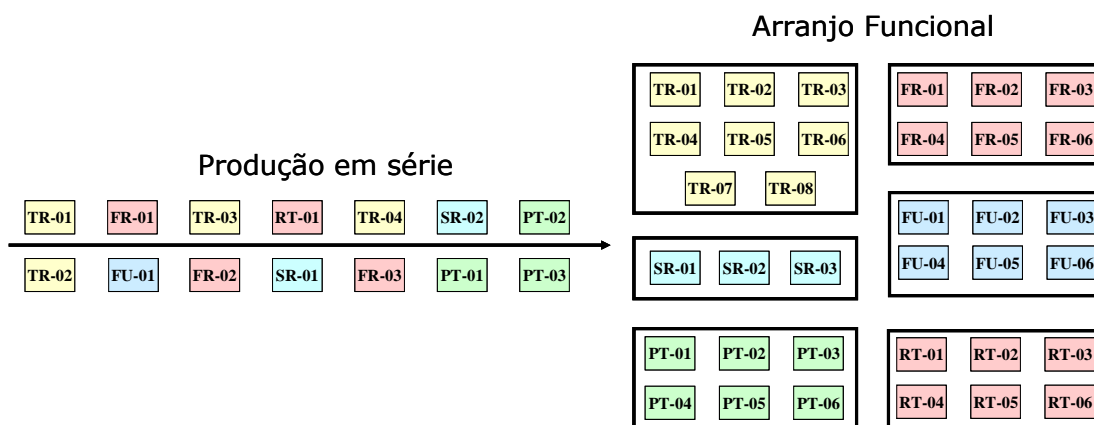
A idéia principal por traz da tecnologia de grupo é a organização do sistema de manufatura usando-se células. A essência das células de manufatura é formar um conjunto de máquinas que processam uma família específica de peças. As máquinas que compõem uma célula de manufatura são dispostas de forma seqüencial conforme as operações a serem executadas, tal como uma pequena linha de produção, mas para apenas uma peça ou um pequeno conjunto de peças semelhantes (famílias).

Assim, a manufatura celular concilia benefícios da produção em série com benefícios do arranjo físico funcional.

O sistema de produção em série é, sabidamente, capaz de produzir em grande escala, com tempo de movimentação mínimo entre as operações e perfazendo o caminho mais simples possível, como mostra a figura 1 a seguir. No sistema de produção em série, as máquinas são organizadas conforme a seqüência de operações necessárias para a fabricação do produto. No entanto, a produção em

série não é flexível e a quebra de uma máquina seria capaz de parar a linha de produção inteira.

**Figura 1** – Sistema de produção em série e arranjo físico funcional

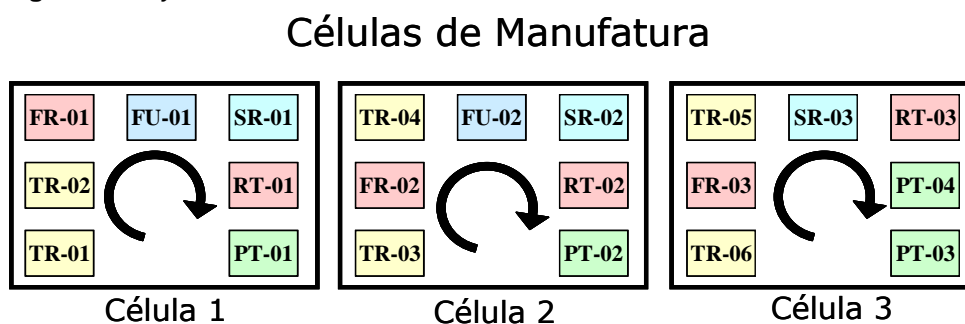


Fonte: O autor

Em oposição ao sistema de produção em série há o arranjo físico funcional, onde as máquinas são agrupadas por similaridade de funções, conforme ilustra a figura 1. Neste caso, para produzir uma peça qualquer, é a peça que se deslocar até as máquinas conforme a seqüência de operações definida. Este arranjo físico permite grande flexibilidade e robustez quando há quebra de máquinas, mas o tempo de movimentação é excessivo, assim como o fluxo das peças torna-se extremamente complexo.

A configuração de um *layout* em células de manufatura permite conciliar a simplicidade do fluxo e mínima movimentação de material da produção em série com a flexibilidade de produção e robustez a quebra do arranjo físico funcional. A figura 2 a seguir ilustra um *layout* em célula de manufatura.

**Figura 2** - *Layout* em células de manufatura



Fonte: Elaborado pelo autor

A opção pelo sistema de manufatura celular requer uma análise ampla do sistema produtivo. Sua implementação exige, em um primeiro momento, uma mudança de cultura e treinamento. Também é necessária uma análise cuidadosa da *interface* com outras áreas do sistema produtivo, como: ferramentaria, manutenção, almoxarifado, controle de qualidade e, especialmente, com as instalações físicas, pois, uma mudança de *layout* exigirá grande esforço na adequação da infraestrutura do chão de fábrica. Há, também, a necessidade de definição das células de manufatura, da disposição física das máquinas internas a cada célula, do dimensionamento do número de máquinas em cada célula e critérios para compartilhamento de máquina entre células, da definição do fluxo das peças e do seqüenciamento da produção nas células de manufatura (GEORGES, 2000).

Para se definir quantas células de manufatura existirão no sistema produtivo, bem como se definir quais máquinas e quantas máquinas cada célula terá, recorre-se aos chamados métodos de formação de célula de manufatura. São estes métodos usados na definição das células de manufatura o objeto de pesquisa deste trabalho, que propõe um algoritmo de baixa complexidade computacional para a formação de células de manufatura que é capaz de definir quais máquinas farão parte de uma célula a partir da análise da família de peças que cada máquina produz.

Para compreender melhor como os métodos de formação de células de manufatura funcionam, a seção 2.1 a seguir faz uma breve apresentação destes métodos.

## **2.1 Métodos De Formação De Células De Manufatura**

O agrupamento de máquinas em uma célula deve ser feito através de algum método que identifique características comuns aos processos de fabricação das peças que permitem o reconhecimento das máquinas necessárias para a fabricação destas peças. Estes métodos são chamados de métodos de formação de células de manufatura.

Já se passaram mais de 40 anos desde o desenvolvimento dos pioneiros métodos para a formação de células de manufatura de Burbidge e Mitrofanov. Desde então, muitos métodos têm sido desenvolvidos e ainda hoje pode ser observado na literatura científica a proposta de novos métodos para este fim.

Segundo Ribeiro Filho (2000), os métodos para a formação células de manufatura podem ser classificados como orientados pelo projeto ou pela produção. Os métodos orientados pelo projeto agrupam partes baseando-se em características de seu projeto; os métodos orientados pela produção o fazem baseando-se nos processos requeridos para sua produção.

Já Offodile et al. (1994), em um estudo envolvendo diversos métodos, definiu três classes de métodos de agrupamento: por inspeção visual, por análise do processo de produção e por análise da codificação e classificação da peça.

O método de inspeção visual é muito dependente do conhecimento, da experiência e da capacidade de observação de quem faz a inspeção visual, e por isso é pouco documentado na literatura.

Já os métodos não visuais utilizam procedimentos específicos para a determinação das células. Estes procedimentos são executados sobre atributos e características identificadas nas peças e nas máquinas e devem ser suficientemente excludentes para a execução em computadores. O critério para o agrupamento deve ser rigorosamente estabelecido para evitar ambigüidades e a obtenção de resultados diferentes para um mesmo conjunto inicial de máquinas e peças.

Os métodos para a formação de células de manufatura utilizam ampla gama de informações que podem ser extraídas dos processos de fabricação, das características das peças, das características das máquinas, do fluxo da peça na fábrica, do roteiro de fabricação, entre outras. Não há uma padronização da informação requerida para se fazer o agrupamento, cada método pode requerer diferentes informações em diferentes formatos.

No entanto, observa-se claramente que a informação mais utilizada pelos métodos de formação de célula de manufatura é a chamada matriz de incidência máquina peça, como ilustra a figura 3 a seguir.

**Figura 3** - Matriz de incidência máquina peça

		PECAS									
		P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
MÁQUINAS	M1	0	1	0	1	1	0	0	0	0	0
	M2	1	0	0	0	0	0	0	1	1	0
	M3	0	0	1	0	0	0	1	0	0	0
	M4	0	0	0	0	0	1	0	1	1	0
	M5	0	0	0	0	0	0	1	0	0	1
	M6	0	0	1	0	0	0	1	0	0	1
	M7	0	1	0	0	1	0	0	0	0	0

A matriz de incidência máquina peça é uma matriz que suas linhas representam as máquinas presentes no sistema de manufatura e suas colunas representam as peças produzidas por este sistema. Cada valor da matriz é preenchido com “1” ou “0”. Se para fabricar uma peça é necessário usar uma máquina, o valor da matriz correspondente a esta peça e máquina assume valor “1”, caso contrário será “0”. Por exemplo, na matriz de incidência máquina peça apresentada na figura 3, para se fabricar a peça “2” (P2) são necessárias as máquinas “1” (M1) e “7” (M7).

Os métodos que utilizam a matriz de incidência máquina peça como informação central para a formação das células de manufatura são chamados de métodos de Análise de Grupo ou métodos de *cluster analysis*.

Muitas abordagens têm sido empregadas para, a partir da informação contida na matriz de incidência máquina peça, obter agrupamentos para a sugestão de células de manufatura.

Um pesquisa rápida na literatura fornece muitos métodos que recorrem aos algoritmos genéticos e a inteligência artificial como abordagens para a formação de célula de manufatura, como os trabalhos de Gonçalves Filho et al. (2004), Vaghefinezhad e Wong (2012), Gharbi et al. (2011) e Trindade e Ochi (2006).

Também é fácil encontrar algoritmos que utilizam programação matemática para encontrar agrupamentos. Segundo o estudo de Arora et al. (2011), foram identificados quase uma centena trabalhos que recorrem a programação linear, programação por objetivos e programação dinâmica como abordagens da programação matemática para a formação de células de manufatura.

Outras abordagens também são encontradas na literatura, como algoritmos baseados na Teoria dos Grafos (RIBEIRO, 2009), análise Presa Predador (BANERJEE & DAS, 2012), Simulações (LIN et al., 2010), e outros.

No entanto, a abordagem mais empregada é a chamada análise de grupo, que tem entre seus métodos o popular algoritmo *ROC (Rank Order Clustering)*, no qual se baseia o algoritmo de baixa complexidade apresentado neste artigo que é denominado *RADIX ROC*. Portanto, antes de apresentar o algoritmo aqui proposto, faz-se necessário o detalhamento dos métodos baseados em análise de grupo conforme a seção 2.2 a seguir, e o detalhamento do algoritmo *ROC* conforme a seção 2.3.

## 2.2 Método Baseado em Análise de Grupo

A principal característica que classifica os métodos em análise de grupo é a maneira de atuar sobre a matriz de incidência máquina peça (figura 3). O objetivo destes métodos é identificar um conjunto de máquinas e peças que possam ser agrupadas de modo que as peças de um grupo dependam somente das máquinas deste grupo e não dependam de outras máquinas podendo, assim, configurar-se em uma célula.

Esse agrupamento de máquinas e peças capaz de formar uma célula de manufatura é identificado pela Estrutura Bloco Diagonal da matriz de incidência máquina peça, conforme ilustra a figura 4 a seguir. Observe que a estrutura bloco diagonal da matriz de incidência produziu três agrupamentos distintos que podem formar três células de manufaturas.

Figura 4 - Estrutura Bloco Diagonal

		PECAS									
M		P3	P7	P10	P5	P4	P2	P1	P6	P9	P8
Á	M3	1	1	0	0	0	0	0	0	0	0
Q	M5	0	1	1	0	0	0	0	0	0	0
U	M6	1	1	1	0	0	0	0	0	0	0
I	M1	0	0	0	1	1	1	0	0	0	0
N	M7	0	0	0	1	0	1	0	0	0	0
A	M2	0	0	0	0	0	0	1	0	1	1
S	M4	0	0	0	0	0	0	0	1	1	1

Os algoritmos de formação de célula de manufatura atuam na matriz de incidência máquina peça, permutando suas linhas e colunas a procura da estrutura bloco diagonal. O algoritmo mais conhecido para permutar a matriz de incidência baseado nesse critério é o *ROC*, acrônimo de *Rank Order Clustering*, que é detalhado na seção 2.3 a seguir.



## 2.3 Rank Order Clustering – Roc

O algoritmo *ROC* foi desenvolvido por King (1980) e publicado em um número especial do periódico *International Journal of Production Research* e tornou-se um dos algoritmos mais citados devida a sua simplicidade e bons resultados.

O algoritmo *ROC*, a grosso modo, converte cada linha em um número decimal, como se a linha fosse um número escrito em base binária, e depois ordena as linhas em ordem decrescente de cima para baixo. Em seguida ordena as colunas em ordem decrescente da esquerda para a direita utilizando o mesmo critério, transformando a coluna em um número decimal como se estivesse escrito em código binário. O algoritmo repete estas ordenações de linha e coluna até que a matriz não possa ser mais modificada, chegando-se a uma estrutura bloco diagonal como resultado final. O algoritmo *ROC* está descrito a seguir na figura 5.

**Figura 5 – Algoritmo ROC**

Considere a matriz de binária  $A$  com  $M$  linhas e  $P$  colunas.

Cada elemento da matriz  $A$  é chamado de  $a_{mp}$ , onde  $m$  indica a linha e  $p$  indica a coluna que se encontra o elemento na matriz de incidência.

Passo 1: Para cada linha  $m$  da matriz  $A$ , calcula-se o número decimal  $Lm$  equivalente através da formula:

$$Lm = \sum_{p=1}^P 2^{P-p} \cdot a_{pm} \quad (1)$$

Passo 2: Ordene as linhas da matriz  $A$  em ordem decrescente de  $Lm$  de cima para baixo. Chame esta matriz ordenada de  $A'$ .

Passo 3: Para cada coluna  $p$  da matriz  $A'$ , calcula-se o número decimal  $Cp$  equivalente através da formula:

$$Cp = \sum_{m=1}^M 2^{M-m} \cdot a_{pm} \quad (2)$$

Passo 4: Ordene as colunas da matriz  $A'$  em ordem decrescente de  $Cp$  da esquerda para a direita. Chame esta matriz ordenada de  $A''$ .

Passo 5: Se  $A = A''$ , vá para o passo 6, caso contrário, faça  $A = A''$  e volte ao passo 1.

Passo 6: Fim. A matriz ordenada  $A''$  apresentará a estrutura bloco diagonal.

**Fonte:** Adaptado pelo de King (1980)

Para ilustrar o algoritmo *ROC*, considere que a matriz  $A$  seja a matriz de incidência máquina peça apresentada na figura 3. Esta matriz possui sete linhas e dez colunas. Cada linha desta matriz é interpretada como um número escrito em notação binária e que deve ser convertido para um número em notação decimal conforme a equação 1 do passo 1. Esta conversão do número binário de uma dada linha em um número decimal  $Lm$  está exemplificada na figura 6 a seguir, que mostra

a conversão da primeira linha da matriz da figura 3 no número decimal **L1** igual a 352.

**Figura 6** – Conversão do número formado pela linha 1 em um único número decimal

$$L1 = 0.2^{10^{-1}} + 1.2^{10^{-2}} + 0.2^{10^{-3}} + 1.2^{10^{-4}} + 1.2^{10^{-5}} + 0.2^{10^{-6}} + \dots + 0.2^{10^{-10}}$$

$$L1 = 0 + 256 + 0 + 64 + 32 + 0 + 0 + 0 + 0 + 0 = 352$$

Fonte: Elaborado pelo autor

A figura 7 a seguir ilustra cada número decimal gerado por cada uma das linhas desta matriz de incidência conforme o passo 1 do algoritmo *ROC* mostrado na figura 5.

**Figura 7** – Ilustração do algoritmo *ROC*

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	<i>Lm</i>
<b>M1</b>	0	1	0	1	1	0	0	0	0	0	<b>352</b>
<b>M2</b>	1	0	0	0	0	0	0	1	1	0	<b>518</b>
<b>M3</b>	0	0	1	0	0	0	1	0	0	0	<b>136</b>
<b>M4</b>	1	0	0	0	0	1	0	1	1	0	<b>534</b>
<b>M5</b>	0	0	0	0	0	0	1	0	0	1	<b>9</b>
<b>M6</b>	0	0	1	0	0	0	1	0	0	1	<b>137</b>
<b>M7</b>	0	1	0	0	1	0	0	0	0	0	<b>288</b>

Fonte: Elaborado pelo autor

De posse dos números decimais associado a cada linha da matriz de incidência, ordenam-se estas linhas em ordem decrescente, conforme o passo 2 do algoritmo *ROC*. O resultado desta ordenação do passo 2 do algoritmo *ROC* é a matriz *A'* que está ilustrada na figura 8 a seguir.

**Figura 8** – Resultado do passo 2 do algoritmo *ROC*

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	<i>Lm</i>
<b>M4</b>	1	0	0	0	0	1	0	1	1	0	<b>534</b>
<b>M2</b>	1	0	0	0	0	0	0	1	1	0	<b>518</b>
<b>M1</b>	0	1	0	1	1	0	0	0	0	0	<b>352</b>
<b>M7</b>	0	1	0	0	1	0	0	0	0	0	<b>288</b>
<b>M6</b>	0	0	1	0	0	0	1	0	0	1	<b>137</b>
<b>M3</b>	0	0	1	0	0	0	1	0	0	0	<b>136</b>
<b>M5</b>	0	0	0	0	0	0	1	0	0	1	<b>9</b>

Fonte: Elaborado pelo autor

Concluído o passo 2, inicia-se o passo 3 do algoritmo *ROC* que converte cada coluna da matriz *A'* da figura 8 em um número decimal **Cp** e, então, a matriz é ordenada de forma decrescente a partir deste número **Cp**. A figura 9 ilustra a esquerda o número **Cp** obtido para cada coluna e a direita a matriz *A''* obtida após a ordenação das colunas em ordem decrescente de **Cp**.

**Figura 9** – Terceiro e quarto passo do algoritmo ROC

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
<b>M4</b>	1	0	0	0	0	1	0	1	1	0
<b>M2</b>	1	0	0	0	0	0	0	1	1	0
<b>M1</b>	0	1	0	1	1	0	0	0	0	0
<b>M7</b>	0	1	0	0	1	0	0	0	0	0
<b>M6</b>	0	0	1	0	0	0	1	0	0	1
<b>M3</b>	0	0	1	0	0	0	1	0	0	0
<b>M5</b>	0	0	0	0	0	0	1	0	0	1
<b>Cp</b>	96	24	6	16	24	64	7	96	96	5

	P1	P8	P9	P6	P5	P2	P4	P7	P3	P10
<b>M4</b>	1	1	1	1	0	0	0	0	0	0
<b>M2</b>	1	1	1	0	0	0	0	0	0	0
<b>M1</b>	0	0	0	0	1	1	1	0	0	0
<b>M7</b>	0	0	0	0	1	1	0	0	0	0
<b>M6</b>	0	0	0	0	0	0	0	1	1	1
<b>M3</b>	0	0	0	0	0	0	0	1	1	0
<b>M5</b>	0	0	0	0	0	0	0	1	0	1
<b>Cp</b>	96	96	96	64	24	24	16	7	6	5

Fonte: Elaborado pelo autor

O quinto passo do algoritmo compara a matriz ordenada  $A''$  com a matriz original  $A$  e, nesta primeira iteração, estas matrizes são diferentes. Então, o algoritmo ROC retorna ao passo 1, mas agora irá usar a matriz  $A''$  como sendo a matriz  $A$  que será novamente ordenada.

Executando novamente os passos de 1 a 4 e fazendo a comparação do passo 5, observa-se que a nova matriz ordenada  $A''$  é exatamente igual a matriz  $A$  que foi usada no começo do passo 1 da segunda iteração, encerrando o algoritmo com apenas duas iterações e revelando a estrutura bloco diagonal na matriz ordenada  $A''$ .

Observa-se que, embora a matriz  $A''$  resultante do algoritmo ROC exibida na figura 9 a direita não é exatamente igual a matriz da figura 4. No entanto, as estruturas bloco diagonal destas matrizes formam as mesmas células de manufatura, em igual número e igual composição de máquina e família de peças.

### 2.3.1 Limitações do Roc

A primeira limitação que surge ao usar o algoritmo ROC é a capacidade em transformar as linhas ou colunas da matriz em um número inteiro, pois, na maioria dos computadores o número inteiro é representado por, no máximo, pelo algarismo  $2^{48} - 1$ , assim a máxima dimensão da matriz que o ROC trabalha é de 47 linhas ou colunas (SING & RAJAMANI, 1996).

Outra limitação do ROC é não haver garantias que se chegará a uma solução final satisfatória na forma de estrutura bloco diagonal quando não houver uma estrutura bloco diagonal totalmente excludente em si. Em muitos casos, a estrutura bloco diagonal é praticamente perfeita, com exceção de um ou poucos elemento que pertence a dois ou mais blocos. Este único elemento é suficiente para o algoritmo fornecer um resultado que embaralha os dois blocos ao invés de revelar

dois blocos quase excludentes entre si, exceto um elemento. Ou seja, o algoritmo *ROC* não é um método exato e sim uma heurística, e não garante a obtenção de uma solução final satisfatória como afirmam Krushinsky e Goldengorin (2012).

O *ROC* também tem uma limitação em não explicitar outras condições de parada deste algoritmo, podendo o método entrar em um '*loop* infinito' ao encontrar uma matriz que, após mais de um ordenamento, torna-se idêntica a uma matriz utilizada anteriormente, provocando um '*loop* infinito', já que a condição de parada compara apenas a matriz atual com a da iteração anterior.

Também se nota uma ineficiência deste algoritmo no que se refere ao uso de espaço na memória do computador. Após cada ordenação dos números decimais originados das linhas ou colunas é necessário que estas linhas ou colunas sejam dispostas na ordem definida pela ordenação, para isto, é necessária uma matriz auxiliar com as mesmas dimensões que a matriz original, pois são reproduzidas nesta matriz as linhas ou colunas na ordem de leitura do cursor definida pela ordenação.

Outra limitação do algoritmo *ROC* é sua alta complexidade algorítmica que resulta num grande esforço computacional para matrizes de dimensões grandes. A complexidade algorítmica é medida através da notação  $O(\cdot)$ , que indica uma expressão em função do número de entradas, e representa a quantidade de operações elementares que será necessário para executar o algoritmo (MARUOKA, 2011).

O algoritmo *ROC*, segundo Sing & Rajamani (1996), possui complexidade algorítmica da ordem de  $O(2.n^3.\log n)$ , o que resulta em grande esforço quando a ordem de grandeza da matriz de incidência máquina peça é alta.

É conveniente citar que estas limitações motivaram o surgimento de modificações no *ROC*, são elas: *ROC 2* desenvolvido por King e Nakornchai em 1982 e *MODROC* desenvolvido por Chandrasekaran e Rajagopalan em 1986 (SING & RAJAMANI, 1996).

### 3 METODOLOGIA

Este artigo foi escrito a partir de resultados de uma pesquisa aplicada com propósito de desenvolver e descrever um algoritmo de formação de célula de

manufatura que fosse eficiente em relação a alguns métodos bem conhecidos já publicados na literatura.

O desenvolvimento do algoritmo proposto foi baseado no algoritmo *ROC*, mas sem a necessidade de transformar o número binário em decimal, ordenando-os diretamente a partir do número binário. Esta ordenação de uma seqüência de número através do código binário é conhecida como *Radix Sort*. Desta fusão, surge o algoritmo e o nome *RADIX ROC*.

A descrição do algoritmo é feita em detalhes na seção 4 a seguir. Por fim, com base num levantamento na literatura científica, selecionaram-se alguns métodos para a formação de célula de manufatura com suas respectivas complexidades algorítmicas para proceder a uma análise comparativa entre os métodos através da notação  $O(.)$ .

#### 4 O ALGORITMO *RADIX ROC*

Este algoritmo para a formação de célula de manufatura é baseado no algoritmo de ordenação chamado *Radix Sort*, e possui algumas vantagens em relação aos algoritmos mais conhecidos de ordenação no que refere-se a complexidade algorítmica e esforço computacional.

O *Radix Sort* é um algoritmo de ordenação utilizado em computadores que permitem acessar o código binário que representa os caracteres dos números que compõem a seqüência a ser ordenada (MANBER, 1989).

Cormen et al. (2009) afirma que o *Radix Sort* é um algoritmo de ordenação rápido e estável que pode ser usado para ordenar itens que estão identificados por chaves únicas e que cada chave é uma cadeia de caracteres ou número. Nos computadores, estas chaves são os números binários usados para representar todo caractere a partir de um conjunto de dados binários.

Embora os sistemas operacionais que permitem o acesso ao código binário não sejam convencionais, não será preciso este acesso para implementar o algoritmo, pois a idéia básica do *Radix Sort* será aplicada na matriz de incidência e não no código binário.

A grande vantagem desse algoritmo é que ele ordena em tempo linear, ou seja,  $O(n)$ , qualquer outro algoritmo de ordenação que não goza do artifício de acessar o código binário ordena, necessariamente, em pelo menos  $O(n \cdot \log n)$  operações (MANBER, 1989).

A princípio, o objetivo era aumentar a eficiência do algoritmo *ROC* (*Rank Order Clustering*), ao usar o algoritmo *Radix Sort* para fazer os ordenamentos dos números decimais exigidos pelo *ROC*.

Mas, ao invés de usar o *Radix Sort* conjuntamente com o *ROC* para ordenar os números decimais, se pensou na hipótese de eliminar a conversão das linhas e colunas em número decimais e usar o princípio do *Radix Sort* para ordenar diretamente as linhas e colunas como se fosse o código binário da seqüência numérica a ser ordenada.

Assim, surgiu a idéia deste novo método de formação de célula de manufatura, chamado *Radix ROC*, que elimina a etapa de conversão das linhas e colunas da matriz de incidência em números decimais para se fazer a ordenação, e faz a ordenação das linhas e das colunas diretamente dos valores binários da matriz de incidência, reduzindo a complexidade algorítmica em relação ao *ROC* original.

O algoritmo *Radix ROC* está detalhado na figura 10 a seguir.

#### **4.1 Complexidade Algorítmica e Limitações do *Radix Roc***

O método percorre duas vezes  $M$  linhas de dimensão  $P$ , e percorre duas vezes  $P$  colunas de dimensão  $M$ . Portanto, a complexidade algorítmica do *Radix ROC* é  $O(2.M.P + 2.P.M)$ , o que equivale a  $O(4.M.P)$ , ou simplesmente  $O(4n^2)$  quando consideramos  $M$  e  $P$  números de mesma ordem de grandeza.

### Figura 10 – Algoritmo *RADIX ROC*

Considere a matriz de binária  $A$  com  $M$  linhas e  $P$  colunas.

Cada elemento da matriz  $A$  é chamado de  $a_{mp}$ , onde  $m$  indica a linha e  $p$  indica a coluna que se encontra o elemento na matriz de incidência.

Passo 1: para  $i$  variando de  $m$  até  $I$ , faça:

(a) Percorra a linha  $i$  da matriz  $A$ . Ao encontrar um elemento  $a_{ij}$  desta linha que seja igual a “1”, transcreva esta coluna inteira da matriz  $A$  na matriz auxiliar  $A'$  conforme a ordem que foi encontrada, colocando como primeira coluna da matriz auxiliar  $A'$  aquela coluna que teve o elemento  $a_{ij} = 1$  encontrado em primeiro lugar.

(b) Percorra novamente a linha  $i$  da matriz  $A$ . Ao encontrar um elemento  $a_{ij}$  desta linha que seja igual a “0”, transcreva esta coluna inteira da matriz  $A$  na matriz auxiliar  $A'$ , imediatamente após as colunas transcritas do passo anterior, conforme a ordem que foi encontrada, colocando como primeira coluna da matriz auxiliar  $A'$  imediatamente após a última coluna transcrita do passo anterior, aquela coluna que teve o elemento  $a_{ij} = 0$  encontrado em primeiro lugar.

Passo 2: para  $j$  variando de  $p$  até  $I$ , faça:

(a) Percorra a coluna  $j$  da matriz  $A'$ . Ao encontrar um elemento  $a_{ij}$  desta coluna que seja igual a “1”, transcreva esta linha inteira da matriz  $A'$  na matriz auxiliar  $A''$  conforme a ordem que foi encontrada, colocando como primeira linha da matriz auxiliar  $A''$  aquela linha que teve o elemento  $a_{ij} = 1$  encontrado em primeiro lugar.

(b) Percorra novamente a coluna  $j$  da matriz  $A'$ . Ao encontrar um elemento  $a_{ij}$  desta coluna que seja igual a “0”, transcreva esta linha inteira da matriz  $A'$  na matriz auxiliar  $A''$ , imediatamente após as linhas transcritas do passo anterior, conforme a ordem que foi encontrada, colocando como primeira linha da matriz auxiliar  $A''$  imediatamente após a última linha transcrita do passo anterior, aquela linha que teve o elemento  $a_{ij} = 0$  encontrado em primeiro lugar.

Passo 3: Se  $A = A''$ , vá para o passo 4, caso contrário faça  $A = A''$  e volte ao passo 2.

Passo 4: Fim. A matriz  $A''$  apresentará a estrutura bloco diagonal.

Fonte: Elaborado pelo autor

## 5 ANÁLISE COMPARATIVA

Nesta seção será comparado o esforço computacional e complexidade algorítmica do *RADIX ROC* com alguns dos métodos de formação de célula de manufatura mais conhecidos da literatura, são eles:

- *ROC (Rank Order Clustering)*, publicado por King (1980);
- *BEA (Bond-energy Algorithm)*, desenvolvido por McCormick Jr. et al. (1972);
- *CIA (Cluster Identification Algorithm)*, desenvolvido por Kusiak e Chow (1987);
- *SLC (Single Linkage Clustering)*, apresentado por Vakharia & Wemmerlo (1995);
- Método baseado na Teoria do Grafo (Kummar et al., 1986).

Para simplificar, assume-se que o número de máquinas seja da mesma ordem de grandeza do número de peças, assim ao invés de se obter a complexidade algorítmica em termos de *M* e *P*, obtém-se em termos de *n*.

A diferença entre o número de operações elementares requerido por cada método é cada vez mais evidente quanto maior for o parâmetro de entrada (o valor de *n*). Para demonstrar esse efeito, é apresentada a tabela 1, a seguir, que compara o número de operações necessária por cada método em função do número de entrada *n*. O valor da entrada *n* irá variar de 10 até 10.000, sendo este maior caso equivalente a uma matriz de incidência da ordem de 100 máquinas e 100 peças.

Observe que uma entrada *n* da ordem de 10.000 não é exagero, pois segundo um extensivo estudo feito por Yin e Yasuda (2005), onde foram pesquisados 70 casos de aplicação dos métodos de formação de células de manufatura, as matrizes de incidência variavam de 4x5 (menor caso) até 40x100 (maior caso), esta última contendo 10 agrupamentos.

A tabela 1 a seguir mostra o número de operações requeridas para executar cada um dos métodos vistos até aqui, para diferentes valores de entrada. As complexidades algorítmicas dos métodos contidos nesta tabela foram extraídos de Sing & Rajamani (1996).

**Tabela 1** – Comparação da complexidade algorítmica para diferentes valores de entrada

<b>Número de operações</b>					
<b>Método</b>	<b>Complexidade</b>	<b>n = 10</b>	<b>n = 100</b>	<b>n = 1000</b>	<b>n = 10000</b>
<i>ROC</i>	$2 n^3 \cdot \log n$	4605	$9,21 \times 10^6$	$1,38 \times 10^{10}$	$1,84 \times 10^{13}$
<i>BEA</i>	$2 \cdot n^3$	2000	$2 \times 10^6$	$2 \times 10^9$	$2 \times 10^{12}$
<i>CIA</i>	$2 \cdot n^2$	200	$2 \times 10^4$	$2 \times 10^6$	$2 \times 10^8$
<i>RADIX ROC</i>	$4 \cdot n^2$	400	$4 \times 10^4$	$4 \times 10^6$	$4 \times 10^8$
<i>SLC</i>	$n^2 / 2 + n^3 \cdot \log(n^2 / 2)$	3962	$8,52 \times 10^6$	$1,31 \times 10^{10}$	$1,77 \times 10^{13}$
Grafo	$2 \cdot n + n^2$	120	$10,2 \times 10^3$	$10^6$	$10^8$

**Fonte:** Elaborado pelo autor

Para se obter uma maior evidência da diferença de esforço computacional requerido por cada um desses métodos, será feita uma comparação no tempo de execução teórico em um computador cuja capacidade de processamento seja de 2 milhões de operações elementares por segundo. A tabela 2 a seguir mostra a grande diferença que existe entre os métodos com relação ao tempo de processamento.



**Tabela 2** – Comparação do tempo de execução dos métodos

<b>Tempo gasto em um Computador ( 2 x 10<sup>6</sup> operações por segundo )</b>			
<b>Método</b>	<b>n = 100</b>	<b>n = 1000</b>	<b>n = 10000</b>
<i>ROC</i>	4,6 seg.	115 min.	≈106 dias
<i>BEA</i>	1 seg.	16,67 min.	≈ 12 dias
<i>CIA</i>	0,01 seg.	1 seg.	100 seg.
<i>RADIX ROC</i>	0,02 seg.	2 seg.	200 seg.
<i>SLC</i>	4,26 seg.	109 min.	≈ 102 dias
Grafos	0,0051 seg.	0,5 seg.	50 seg.

**Fonte:** Elaborado pelo autor

É incrível como alguns métodos demoram segundos enquanto outros levam centenas de dias.

Outros critérios também são de extrema importância para a comparação entre métodos, como a utilização de espaço em memória, facilidade de programação, utilização de linguagem recursiva ou iterativa, entre outros.

Entre os métodos pesquisados e sob os critérios definidos acima, recomenda-se o uso do *CIA*, *RADIX ROC* e do baseado em grafos, pois possuem a menor complexidade algorítmica, como mostrado na tabela 1.

É interessante relatar que o *CIA* e o método baseado em grafos são métodos exatos e não heurísticas, conforme Krushinsky e Goldengorin (2012).

## **6 COMENTÁRIOS FINAIS**

Entretanto, é relevante citar que os algoritmos apenas utilizam a informação contida na matriz de incidência, o que os tornam incapazes de distinguir uma máquina rígida de uma máquina flexível e qualquer outra característica não contida na matriz de incidência e atribuída ao sistema, prejudicando uma eficaz implementação da Tecnologia de Grupo na organização industrial.

Esses critérios de comparação não levam em consideração fatores conhecidos da engenharia e cruciais à determinação da célula, como problemas com temperatura, ruídos, vibrações, ergonomia e outros.

Os algoritmos são apenas uma sugestão inicial, e o agrupamento definitivo deve considerar diversas questões excluídas aqui.

Mas, nas organizações industriais com quantidade grande de peças e máquina, é impraticável a aplicação da Tecnologia de Grupo sem o auxílio de um

algoritmo, porém a sua utilização deve ser acompanhada de análise posterior para a confirmação das células determinadas pelo algoritmo.

Por enquanto, este método não passa de uma heurística, pois sua prova de correteza ainda não foi feita, embora os testes feitos até agora indiquem que funcionará para qualquer matriz e revelou-se não seguro quando se trata de matrizes com uma grande diferença de dimensão entre linhas e colunas, ou seja  $M \gg P$  ou  $P \gg M$ .

## REFERÊNCIAS

- ARORA, P.K.; HALEEM, A.; SINGH, M. K. Cell Formation Techniques – A Study. In: **International Journal of Engineering Science and Technology (IJEST)**. v.3, n.2, p.1178-1181, fev. 2011. <http://dx.doi.org/10.1016/j.asoc.2011.07.021>
- BANERJEE, I.; DAS, P. Group technology based adaptive cell formation using predator–prey genetic algorithm. In: **Applied Soft Computing**, v.12, n.1, p. 559–572, January, 2012 <http://dx.doi.org/10.1080/00207549208942951>
- BURBIDGE, J.L. Change to Group Technology: Process Organization is Obsolete, **International Journal of Production Research**, v.30, n.5, p. 1209-1219, 1992.
- CORMEN, T.H.; LEISERSON, C.E.; RIVEST, R.L.; STEIN, C. **Introduction to Algorithms**, 3. ed., MIT Press, Boston, 2009.
- GEORGES, M.R.R. Desenvolvimento de um algoritmo de baixa complexidade para a formação de célula de manufatura. In: SIMPOSIO DE ENGENHARIA DE PRODUÇÃO, 7., 2000, Bauru - SP. **Anais...** 2000.
- GHARBI, A.; AL-AHMARI, A.M.; MSAKNI, M.K.; AL-KHALEFAH. H.. A New Genetic Algorithm for Designing Cellular Manufacturing Systems with Labor and Tools Issues. **Applied Mechanics and Materials**, n. 4307, p. 110-116, out.2011.
- GONÇALVES FILHO, E.V.; GORGULHO JÚNIOR, J. H. C.; ARGOUD, A.R.T.T. Algoritmo genético para formação de células de fabricação. In. **ENEGEP** , 24., 2004. **Anais....** Florianópolis, SC, Brasil, 2004.
- KING, J.R. Machine-component grouping in production flow analysis: an approach using a rank order clustering algorithm. **International Journal of Production Research** . v. 18, n. 2, 1980.
- KRUSHINSKY, D.; GOLDENGORIN, B. An exact model for cell formation in group technology. In: **Computational Management Science**, v.9, n. 3, p 323-338, Springer-Verlag, August 2012.
- KUMAR, K.R.; KUSIAK, A.; VANNELLI, A. Grouping of parts and components in flexible manufacturing systems. In: **European Journal of Operational Research**. v. 24, n. 3, p. 387–397, March, 1986. [http://dx.doi.org/10.1016/0377-2217\(86\)90032-9](http://dx.doi.org/10.1016/0377-2217(86)90032-9)

KUSIAK, A. **Intelligent manufacturing systems**. New Jersey: Prentice-Hall, 1990. [http://dx.doi.org/10.1016/0278-6125\(87\)90035-5](http://dx.doi.org/10.1016/0278-6125(87)90035-5)

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

LIN, SHIH-WEI; YING, KUO-CHING; LEE, ZNE-JUNG. Part-machine cell formation in group technology using a simulated annealing-based meta-heuristic. In: **International Journal of Production Research**. v. 48, n.12, 2010.

MARUOKA, A. **Concise guide to computational theory**. Springer-Verlag, London, 2011. <http://dx.doi.org/10.1007/978-0-85729-535-4>

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

MCCORMICK JR, W.T.; SCHWEITZER, P.J.; WHITE, T.W. Problem Decomposition and Data Reorganization by a Clustering Technique. **Operations Research** v. 20, n. 5, p. 993-1009, Sep./Oct. 1972.

OFFODILE, O. F.; MEHREZ, A.; GRZNAR, J. Cellular manufacturing: A taxonomic review framework. In: **Journal of Manufacturing Systems**, Volume 13, Issue 3, pp. 196–220, 1994. [http://dx.doi.org/10.1016/0278-6125\(94\)90005-1](http://dx.doi.org/10.1016/0278-6125(94)90005-1)

RIBEIRO FILHO, G. **Melhoramentos no algoritmo genético construtivo e novas aplicações em problemas de agrupamento**. Tese Doutorado. São José dos Campos: INPE, 2000.

RIBEIRO, J.F.F. Manufacturing cells formation based on graph theory. **Computers & Industrial Engineering**, p. 658 – 662, July, 2009

SING., N., RAJAMANI, D. **Cellular Manufacturing Systems: desing, Planning and Control**, Chapman & Hall, Londres, 1996.

TRINDADE, A. R.; OCHI, L. S. Um algoritmo evolutivo híbrido para a formação de células de manufatura em sistemas de produção. **Revista Pesquisa Operacional**, v..26, n.2, p. 255-294, 2006.  
Disponível em <[http://www.scielo.br/scielo.php?script=sci\\_arttext&pid=S0101-74382006000200005&lng=en&nrm=iso](http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0101-74382006000200005&lng=en&nrm=iso)>.

UDI MANBER. **Introduction to algorithms: a creative approach** , Addison-Wesley, New York, 1989.

VAGHEFINEZHAD, S.; WONG, K.Y.. **A Genetic Algorithm Approach for Solving Group Technology Problem with Process Plan Flexibility**. In: ukxim, pp.52-58, 2012 UKSim 14th International Conference on Modelling and Simulation, 2012

VAKHARIA, A. J.; WEMMERLOV, U. A comparative investigation of hierarchical clustering techniques and dissimilarity measures applied to the cell formation problem. **Journal of Operations Management**, 13, 117–138, 1995. [http://dx.doi.org/10.1016/0272-6963\(95\)00017-M](http://dx.doi.org/10.1016/0272-6963(95)00017-M)

YIN, YONG; YASUDA, K. Similarity coefficient methods applied to the cell formation problem: a comparative investigation. **Computers & Industrial Engineering**, v. 48, p. 471–489, 2005. <http://dx.doi.org/10.1016/j.cie.2003.01.001>



Artigo recebido em 25/05/2012 e aceito para publicação em 26/11/2014  
DOI: <http://dx.doi.org/10.14488/1676-1901.v14i4.1306>