

Uma Aplicação de Coloração em Grafos para a Tecnologia de Grupo

Ricardo Cezar Ferreira

Orientador: Prof. Dr. José Francisco Ferreira Ribeiro

Dissertação apresentada ao Instituto de Ciências Matemáticas de São Carlos, da Universidade de São Paulo, para obtenção do Título de Mestre em "Ciências de Computação e Matemática Computacional".

São Carlos

1996

"A educação é a caminhada
do homem da ignorância
excessivamente confiante
para a incerteza refletida".

Don Clark

A minha esposa,
aos meus pais e
meus irmãos.

Agradecimentos

Ao Prof. Dr. José Francisco Ferreira Ribeiro pela orientação dedicada, participativa, segura e objetiva.

Aos professores do ICMSC pelas lições recebidas que tanto contribuíram na minha formação acadêmica.

A minha esposa Neuza pelo companheirismo, compreensão, apoio e dedicação em todos os momentos.

A Universidade Estadual de Londrina pela oportunidade e apoio.

Aos funcionários do ICMSC que proporcionaram um ambiente de trabalho agradável.

A todos os amigos que compartilharam tantas dificuldades e alegrias.

Aos meus pais e irmãos que compreenderam a necessidade de minha ausência.

À CAPES e CNPq pelo apoio financeiro concedido.

Resumo

Nesta dissertação é apresentado um algoritmo de coloração em grafos para o projeto de células de manufatura em Tecnologia de Grupo. O algoritmo proposto computa as dissimilaridades entre as peças e estabelece uma partição de peças e máquinas por meio da construção e coloração de uma árvore de peso máximo em um número de cores igual ao número requerido de células. Este algoritmo permite a resolução eficiente de exemplos da literatura, apesar do caráter não polinomial do problema estudado. O programa correspondente foi escrito em Turbo-Pascal e está implantado em um microcomputador.

Abstract

In this thesis a graph colouring algorithm is presented for cellular manufacturing design in Group Technology. The proposed algorithm computes dissimilarities between parts and establishes a partition of parts and machines through the construction and colouring of a maximum spanning tree in a number of colours equals to the desired number of cells. This algorithm allows us to solve efficiently some examples of the literature, despite the non-polynomial nature of the problems studied. The corresponding program is written in Turbo-Pascal and runs on a micro-computer.

Índice

Introdução.....	01
Capítulo 1 O Problema da Decomposição de Sistemas de Manufatura.....	03
1 Tecnologia de grupo	03
2 Algumas Vantagens da Tecnologia de Grupo	06
3 Análise da eficiência da Tecnologia de Grupo.....	07
Capítulo 2 Revisão Bibliográfica.....	09
1 MV - Método Visual	10
2 MC - Método da Codificação	10
3 FM - Formulação Matricial	10
3.1 PFA - Production Flow Analysis	11
3.2 SCM - Similarity Coefficient Methods.....	11
3.3 SBA - Sorting Based Algorithms	13
3.4 CBM - Cost-Based Method	14
3.5 CIA - Cluster Identification Algorithm.....	15
4 FPM - Formulação Programação Matemática	15
4.1 MPM - Modelo da P-Mediana	16
4.2 MPMG - Modelo da P-Mediana Generalizado	17
4.3 MPQ - Modelo de Programação Quadrática	18
4.4 MPF - Modelo da Programação Fracionária	19
5 FG - Formulação por Grafos	21
5.1 AGG - Algoritmo Grafo-Gargalo	21
5.2 AGP - Algoritmo Grafo-Particionamento	22
5.3 AGC - Algoritmo Grafo-Coloração	23
6 SE - Sistemas Especialistas	24
7 Comentários	25
Capítulo 3 Introdução à Teoria de Grafos.....	26

1 Grafos	26
2 Coloração de Grafos	33
3 Algoritmos de Coloração	34
3.1 Algoritmo Guloso	35
3.2 Algoritmo de Ligações-Contrações	36
Capítulo 4 Um Algoritmo de Coloração para o Projeto de Células	
de Manufatura.....	38
1 Dados do Problema	38
2 Algoritmo	39
3 Matriz de Cargas de Trabalho	39
4 Computação das Dissimilaridades.....	40
5 Decomposição do Sistema de Manufatura	41
5.1 Escolha da Família de Maior Diâmetro.....	41
5.2 Construção e Coloração de uma Árvore de Peso Máximo	42
5.3 Obtenção das Famílias de Peças	43
5.4 Obtenção dos Grupos de Máquinas	44
6 Aplicações	46
7 Comentários.....	51
Conclusão.....	52
Referências Bibliográficas.....	54

Introdução

Em países industrializados como o Brasil, 75% da produção mecânica é realizada em fábricas destinadas à produção em lotes pequenos e médios, com a cadência de fabricação variando de 1 a 50 peças por lançamento [09]. Neste tipo de fábrica, a ordem de utilização dos centros de trabalho pelos lotes de peças é variável e além disto, fabrica-se um grande número de peças diferentes ou de variantes de uma mesma peça. Na terminologia da área [27], denomina-se este tipo de fabricação de produção intermitente do tipo “job shop”, enquanto que a produção em massa de peças padronizadas é denominada intermitente do tipo “flow shop”. Em um sistema de produção “flow shop”, as peças, as máquinas, a mão de obra, etc., podem ser total ou parcialmente instaladas em uma linha de produção fixa, o que não pode ser realizado no caso de um sistema de produção “job shop”. O método Kanban [42], implementado pela Toyota Motor é reconhecido como uma ferramenta eficaz e bem adaptada à gestão da produção do tipo “flow shop”.

Numerosos estudos e contribuições tem sido realizados na área de fabricação “job shop” ao longo dos últimos anos. De um lado, as contribuições estão relacionadas com o aprimoramento da maquinaria disponível, com o desenvolvimento de robôs, AGVs (“automated guided vehicles”) e os sistemas flexíveis de manufatura. De outro, a proposição de novos procedimentos de gestão, como a Tecnologia de Grupo [06] e as diferentes técnicas de administração da produção auxiliada por computador como a CIM (“computer integrated manufacturing”). O método MRP: Manufacturing Resource Planning [39], capaz de estabelecer um plano diretor de produção e um sequenciamento da produção ao mais tarde é uma das ferramentas mais utilizadas em planejamento e controle da produção do tipo “job shop” [09].

Uma das idéias de base da Tecnologia de Grupo consiste em decompor o sistema de fabricação em subsistemas ou células mais fáceis de gerenciar que o sistema global. Nesta técnica, as máquinas, as ferramentas, a mão de obra, etc., são reagrupadas e dedicadas à fabricação de subconjuntos de peças. Este procedimento tem por objetivo melhorar os índices de produtividade e eficiência da fábrica, na medida em que proporciona uma diminuição do percurso das peças, do tempo de regulagem das máquinas (“set up time”), etc. Grosso modo, trata-se de identificar as peças que sofrem as mesmas operações e de reuni-las de maneira a obter sub-fábricas independentes umas das outras. A

organização de uma fábrica em células de manufatura não é, entretanto, uma tarefa simples e pode exigir a resolução de um problema matemático bastante complexo [10], dado o caráter NP-completo do problema estudado [16].

Nesta dissertação, apresentamos o estudo que realizamos acerca da Tecnologia de Grupo e do Projeto de Células de Manufatura.

No primeiro capítulo são introduzidos os conceitos de Tecnologia de Grupo e de Células de Manufatura. Também são apresentadas as principais vantagens advindas da aplicação desta técnica juntamente com sua eficiência em sistemas de produção destinados à fabricação de lotes de pequeno e médio porte.

Uma revisão bibliográfica é apresentada no segundo capítulo. Nesta revisão é proposta uma classificação dos métodos de particionamento para o projeto de células, assim como uma apresentação sucinta dos principais métodos estudados.

O terceiro capítulo é dedicado à apresentação dos principais conceitos e problemas pertinentes à Teoria dos Grafos. Neste estudo, interessa-nos fundamentalmente as técnicas de coloração em grafos, tendo em vista a aplicabilidade das mesmas na resolução de problemas de particionamento.

No capítulo 4 é apresentado um estudo detalhado do método e da implementação computacional desenvolvida neste trabalho. Trata-se de um método para o projeto de células de manufatura baseado em coloração em grafos que a exemplo de [11], [12] e [41], apresenta bons resultados em termos de qualidade da solução obtida e de tempo de cálculo. O programa computacional foi escrito em linguagem Turbo-Pascal e testado utilizando-se um microcomputador IBM-PC compatível.

Na conclusão é apresentada uma reflexão acerca do trabalho realizado e as perspectivas que julgamos mais interessantes para a continuidade deste trabalho.

Capítulo 1

O Problema da Decomposição de Sistemas de Manufatura

Neste capítulo é apresentada uma filosofia de organização e produção industrial denominada Tecnologia de Grupo, destinada principalmente à aplicação em sistemas de manufatura dedicados à produção de lotes pequenos e médios.

1 Tecnologia de Grupo

Em um sistema de manufatura procura-se agilizar a produção, torná-la mais eficiente, sem perder a qualidade. Isto decorre do aumento da competitividade e do desenvolvimento de novas tecnologias e aperfeiçoamento das existentes. Na busca por maior eficiência, surgiu por volta de 1945 uma nova técnica de produção industrial denominada Tecnologia de Grupo (TG). Segundo Grayson [20], os primeiros relatórios sobre essa nova técnica são provenientes dos países da Europa Oriental, principalmente da antiga União Soviética, onde foi publicado, em 1959, por Mitrofanov [38], um dos primeiros livros sobre o assunto, intitulado “Scientific Principles of Group Technology”. Burbidge [06] e Ham [24] deram continuidade às suas pesquisas, realizando importantes trabalhos e publicações.

A idéia básica da Tecnologia de Grupo (TG) consiste em decompor o sistema global de manufatura em subsistemas mais fáceis de administrar que o sistema considerado em sua totalidade. Esta decomposição é feita procurando-se agrupar as peças que apresentam similaridades geométricas e/ou que se submetem às mesmas operações durante o processo de fabricação, e agrupar as máquinas que executam as operações necessárias sobre estas peças de maneira a obter sub-fábricas o mais independentes possíveis umas das outras.

Durante o desenvolvimento da indústria, principalmente nos países da Europa Ocidental, constatou-se que em certos setores da indústria seria mais vantajoso

trabalhar com lotes pequenos e médios, mesmo no caso de indústrias de grande porte. Com esta visão do sistema industrial, a Tecnologia de Grupo conquistou rapidamente o seu espaço no campo científico, onde muitos pesquisadores passaram a se dedicar ao estudo da técnica.

Uma decomposição de um sistema de manufatura em sub-fábricas completamente independentes está ilustrada na figura 1.1. Na nova configuração, o lote de peças 1 será processado em uma sub-fábrica independente, composta pelas máquinas 8,9 e 10; o lote 2 pelas máquinas 5,6 e 7 e o lote 3 pelas máquinas 1,2,3 e 4.

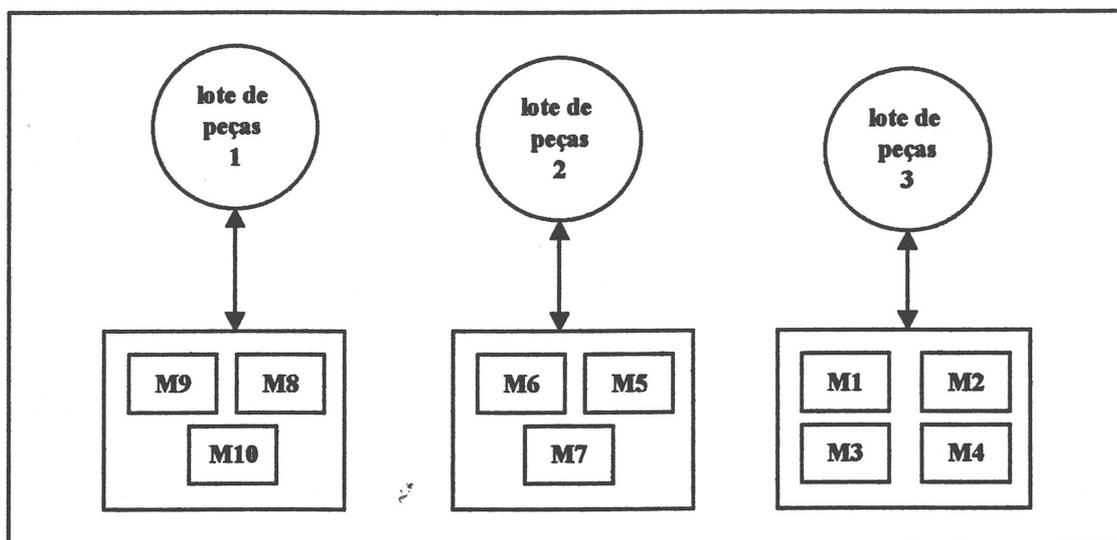


figura 1.1 - Decomposição de uma fábrica em três sub-fábricas independentes.

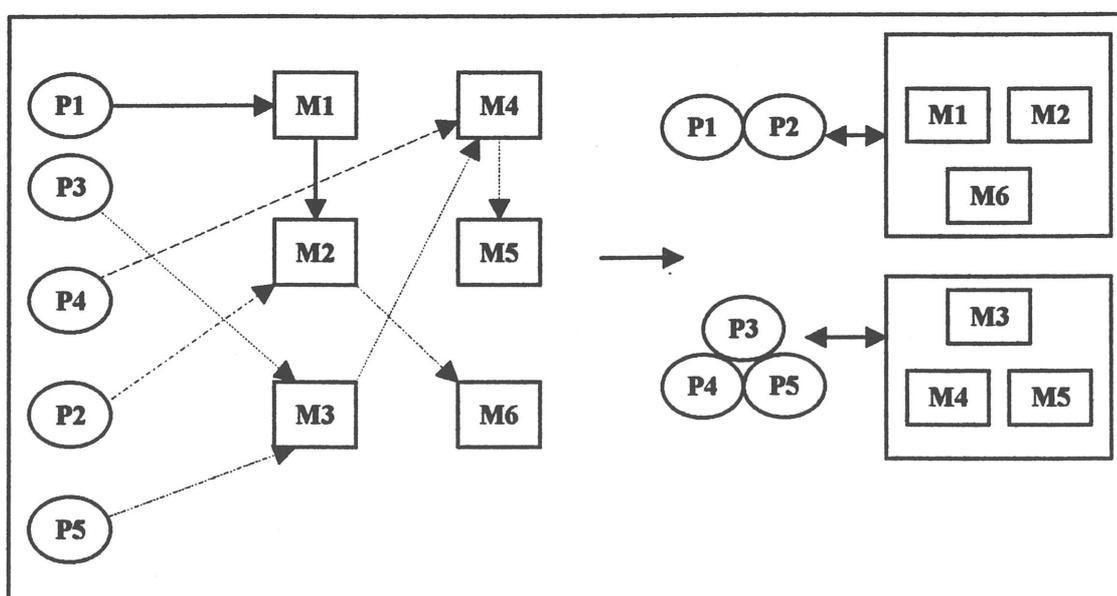


figura 1.2 - Decomposição de uma fábrica em duas sub-fábricas independentes.

Definição 1.1 Um conjunto de peças que apresentam similaridades de projeto e/ou fabricação é chamado de *família*.

Definição 1.2 Um conjunto de máquinas agrupadas para executar todas (ou a maior parte) as operações sobre as peças de uma determinada família é chamado de *grupo*.

A figura 1.2 mostra o agrupamento de peças em duas famílias e das máquinas em dois grupos, a saber:

família1= {P1,P2}

grupo1= {M1, M2, M6}

família2= {P3,P4,P5}

grupo2= {M3, M4, M5}

Às vezes pode não ser interessante fazer o agrupamento físico das máquinas e isto pode ocorrer, por exemplo, nos seguintes casos:

- o chão de fábrica não permite o rearranjo das máquinas;
- o custo para rearranjar as máquinas é alto;
- quando há mudanças frequentes na linha de produção;
- se as máquinas estão fixas e não se pode removê-las do lugar.

Quando uma destas situações ocorre, descarta-se então a implementação integral de todas as alterações físicas previstas em um plano de implantação da Tecnologia de Grupo. Pode-se, entretanto, fazer um uso lógico das informações contidas em um projeto de famílias e grupos, levando-se em conta peças que utilizam as mesmas máquinas, das máquinas que deveriam estar fisicamente mais próximas, etc.

Definição 1.3 O par formado por uma família de peças e um grupo de máquinas é chamado de *célula de manufatura*.

Nem sempre é possível obter as células de manufatura totalmente independentes umas das outras. Quando tal situação ocorre, existe pelo menos uma peça que requer operações em máquinas que não estão em sua célula de atribuição. Este problema pode ser resolvido da seguinte maneira:

- transportando a peça até a máquina requerida;
- duplicando as máquinas que estão sendo requeridas em mais de uma célula;
- parar de fabricar esta peça;
- em vez de produzi-la, pode ser mais viável passar a comprá-la.

Definição 1.4 O transporte de uma peça de uma célula de manufatura para outra célula é chamado de *movimento inter-células*.

Definição 1.5 A peça que necessita do movimento inter-células é chamada de *peça problema*.

Definição 1.6 As máquinas que executam movimentos inter-células são chamadas de *máquinas problema*.

Definição 1.7 A passagem de uma peça por uma máquina é chamada de *operação*.

Definição 1.8 A máquina que executa o maior número de operações em uma fábrica é chamada de *máquina gargalo*, levando-se em conta a disponibilidade de capacidade.

Quando a sobrecarga de trabalho sobre a máquina gargalo é muito alta, alguns cuidados devem ser tomados. O ideal seria duplicar esta máquina e caso isso não seja possível, deve-se frequentemente fazer revisões na mesma para evitar quebras que prejudicariam bastante a produção.

2 Algumas Vantagens da Tecnologia de Grupo

A implantação da TG no sistema de manufatura pode fornecer as seguintes vantagens [09]:

- as células de manufatura podem aumentar a produtividade da indústria, diminuindo assim o tempo de fabricação e conseqüentemente o tempo de entrega;

- o tempo perdido para o transporte das peças às máquinas e a decisão sobre a máquina efetiva para executar uma dada operação deixam de existir, uma vez que trabalha-se com células de manufatura. Em outras palavras, há diminuição do trabalho em processo;
- quando já se conhece a destinação de cada peça, o transporte é facilitado e conseqüentemente pode-se intensificar o trabalho;
- executando-se as operações sobre as peças nas máquinas mais adequadas, espera-se diminuir o número de horas de retrabalho e os refugos;
- as similaridades entre as peças a serem fabricadas em uma mesma célula de manufatura permitem uma redução do tempo de regulagem das máquinas, a utilização de um mesmo ferramental, uma diminuição do trajeto das peças pela fábrica, etc;
- pode-se ainda, reduzir a burocracia e melhorar o relacionamento humano entre os funcionários;

3 Análise da Eficiência da Tecnologia de Grupo

Existem dúvidas quanto à eficiência da decomposição de um sistema de manufatura automatizado. Alguns estudos [32] tem mostrado que a partição das máquinas em grupos pode limitar a flexibilidade do sistema. Entretanto, para muitas aplicações industriais, Hyer e Wemmerlöw [26] mostram que é muito difícil operar um sistema de manufatura de grande porte sem a decomposição do mesmo em células de manufatura. Existem pelo menos quatro fatores que sustentam esta tese, a saber:

1 - Volume de Informações : O volume de informações em um sistema de manufatura automatizado de grande porte é geralmente grande e pode ser muito caro efetivar o processamento das informações sem a decomposição do sistema.

2 - Sistema de Manuseio e Transporte de Materiais : Em um sistema de manufatura automatizado típico, para carregar o material podem ser utilizados “automated guided vehicles” (AGV - veículos guiados automaticamente) ou robôs. Cada um destes

carregadores (AGV ou robô) pode atender somente um número limitado de máquinas. Esta limitação é imposta por:

- número máximo de viagens que um AGV pode fazer por unidade de tempo;
- o caminho de orientação do AGV, por exemplo, um trilho, que não poderia ser interceptado por outro caminho de orientação na área de trabalho de AGV. Quando um AGV se aproxima de uma interseção, a sua velocidade deve diminuir, o que pode reduzir sua utilização.

3 - Exigências Tecnológicas : Algumas máquinas devem ser arranjadas fisicamente próximas devido a exigências tecnológicas, por exemplo: uma máquina de forjar e uma estação de tratamento por calor (aquecedor).

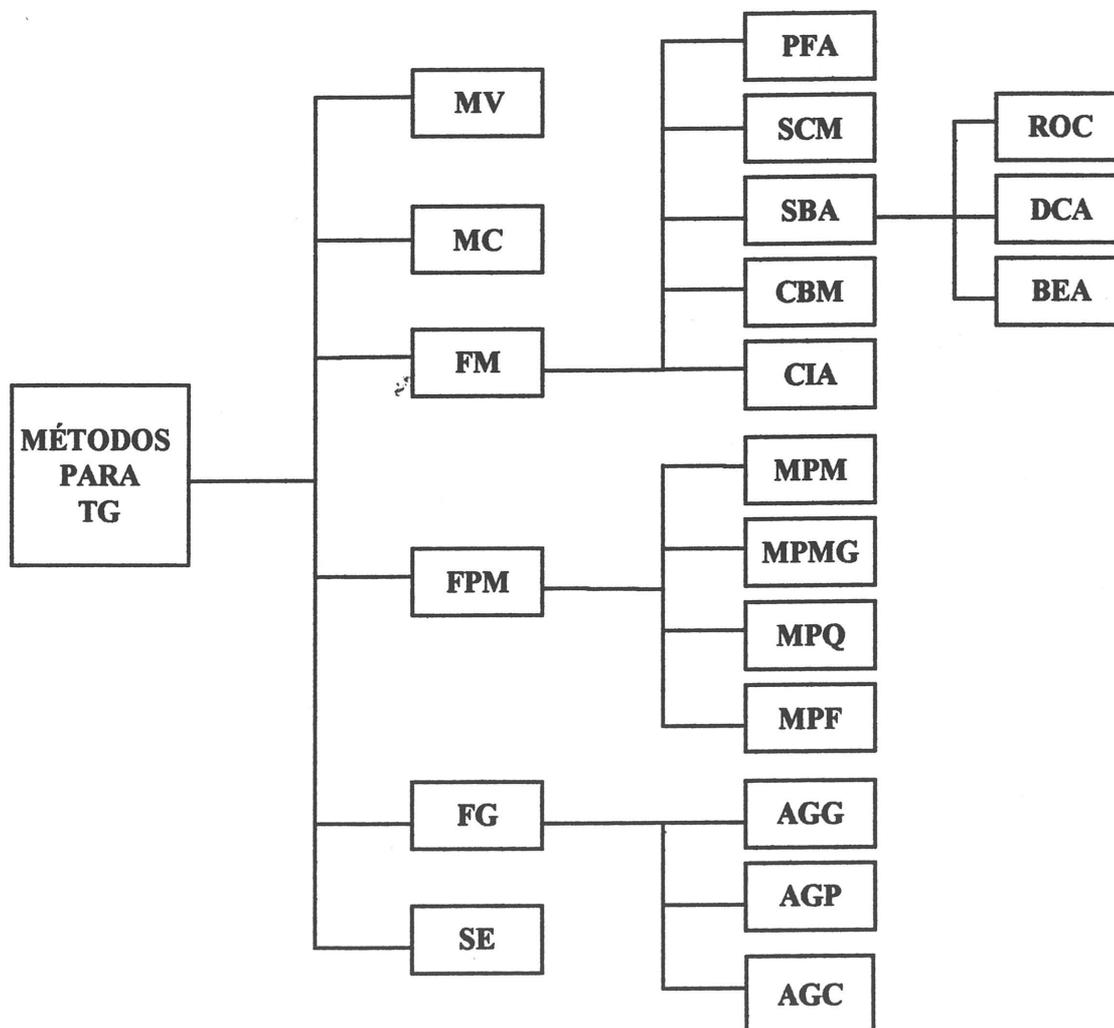
4 - Controle ou Administração : Na maior parte de um sistema automatizado de manufatura, o grau de automatização é maior que nos sistemas clássicos. A mão de obra humana, entretanto, será por um longo tempo ainda, uma parte importante destes novos sistemas de manufatura. Dada a dimensão limitada de cada grupo de máquinas, o sistema de manufatura decomposto em células é mais fácil de gerenciar que um sistema organizado por função (seção de tornos, seção de fresadores, etc.).

Ferreira e Resende [14] apresentam a aplicação da Tecnologia de Grupo em 3 indústrias paulistas e Stabile de Arruda [44] descreve os resultados obtidos em 33 empresas paulistas usuárias da Tecnologia de Grupo.

Capítulo 2

Revisão Bibliográfica

Numerosos algoritmos e modelos foram propostos nos últimos anos para realizar o projeto de células de manufatura. Neste capítulo, apresentamos algumas destas técnicas e, quando possível, as formulações matemáticas correspondentes.



O diagrama acima fornece uma classificação para os diferentes métodos propostos na literatura.

1 MV - Método Visual [32]

O método visual agrupa as peças a serem fabricadas considerando suas semelhanças geométricas. O julgamento destas semelhanças é feito por um profissional habilitado, de acordo com sua opinião pessoal. Assim, para cada projetista pode-se obter agrupamentos diferentes. Esta técnica é indicada somente em situações onde o número de peças a repartir em famílias seja reduzido.

2 MC - Método da Codificação [32]

O método da codificação agrupa as peças de acordo com as seguintes características:

- complexidade e forma geométrica;
- tipo do material;
- forma da matéria prima;
- precisão do acabamento das peças.

Usando um sistema de codificação, por exemplo: Optiz, Vuoso, Brisch, etc., [47] as peças recebem um código numérico, alfabético ou alfanumérico. Cada dígito deste código representa uma característica da peça e as famílias são formadas realizando-se uma comparação entre os dígitos dos seus respectivos códigos. Fu [15] considera que a implementação do método da codificação em um sistema de manufatura é um processo caro e difícil. Esta é provavelmente a razão pela qual muitas companhias decidem não implementar este método.

3 FM - Formulação Matricial [32]

Na formulação matricial, uma matriz de incidência máquina-peça é construída. A matriz de incidência $[a_{ij}]$ é constituída de elementos 0 ou 1, onde o inteiro 1 (0) indica que a máquina i é utilizada (não usada) para processar a peça j . Os algoritmos rearranjam as linhas e colunas da matriz de incidência de modo a obter os agrupamentos desejados.

Para resolver o modelo matricial, os seguintes métodos foram desenvolvidos:

- production flow analysis; (PFA)

- similarity coefficient methods; (SCM)
- sorting-based algorithms; (SBA)
- cost-based method; (CBM)
- cluster identification algorithm. (CIA)

3.1 PFA - Production Flow Analysis [47]

Este método é constituído de quatro passos, a saber:

- Passo 1** - Realização de uma análise do fluxo da fábrica, preocupando-se com os roteiros das peças, com o objetivo de obter um roteiro simples e eficiente. A matriz de incidência peça-máquina é resultado desta análise.
- Passo 2** - Tenta-se obter os agrupamentos rearranjando linhas e colunas da matriz de incidência.
- Passo 3** - É feita uma análise da sequência em que as peças usam as máquinas do grupo, determinando-se quais máquinas devem ficar próximas umas das outras a fim de se obter um fluxo o mais próximo possível de uma produção em linha.
- Passo 4** - A última etapa da PFA é a análise do ferramental, onde é realizada uma pesquisa com o objetivo de determinar a melhor sequência de fabricação de cada peça, minimizando-se assim o tempo de preparação das máquinas. Isto é feito dando preferência às peças que requerem o ferramental mais complexo.

3.2 SCM - Similarity Coefficient Methods [32]

Esta classe de métodos é baseada em um coeficiente de similaridade S_{ij} entre duas máquinas i, j . O coeficiente de similaridade pode ser calculado como segue:

$$S_{ij} = \frac{\sum_{k=1}^n d^1(a_{ik}, a_{jk})}{\sum_{k=1}^n d^2(a_{ik}, a_{jk})}$$

onde

$$d^1(a_{ik}, a_{jk}) = \begin{cases} 1 & \text{se } a_{ik} = a_{jk} = 1 \\ 0 & \text{caso contrário} \end{cases}$$

$$d^2(a_{ik}, a_{jk}) = \begin{cases} 0 & \text{se } a_{ik} = a_{jk} = 0 \\ 1 & \text{caso contrário} \end{cases}$$

n = número de peças.

Para realizar o projeto de células de manufatura usando SCM, todas as partições possíveis são geradas. Com base em um valor dado do coeficiente de similaridade entre as peças, as máquinas vão sendo aglomeradas e as células formadas.

Uma desvantagem do SCM é que este não evita a duplicação das máquinas. Assim, para aplicar esse método é interessante saber se haverá movimentos inter-células na solução. Os movimentos inter-células podem ser calculados da seguinte forma:

$$MIC(A) = \sum_{k=1}^n d^3(v_{ik}, v_{jk})$$

onde

A é uma matriz $m \times n$, constituída de elementos não negativos, onde m é o número de máquinas e n o número de peças. Na fórmula abaixo MC- t identifica as células formadas onde t varia de 1 até o número total de células.

$$v_{ik} = \begin{cases} 1 & \text{se } \sum_{i \in MC-t} a_{ik} \neq 0 \\ 0 & \text{caso contrário} \end{cases}$$

e

$$d^3(v_{ik}, v_{jk}) = \begin{cases} 1 & \text{se } v_{ik} = v_{jk} = 1 \\ 0 & \text{caso contrário} \end{cases}$$

Quando $d(v_{ik}, v_{jk}) = 1$, isto significa que a peça k requer operações em ambas as células de máquinas MC- i e MC- j e assim a peça k é uma peça problema.

Entre os métodos que utilizam um coeficiente de similaridade para efetuar o projeto de células de manufatura, podemos citar: Rajagopalan e Batra [40], Waghodekar e Sahu [48], Chandrasekharan e Rajagopalan [08], Kusiak [31], Wei e Gary [49], Gunasingh e Lashkari [22], Srinivasan et al [43], Gupta e Seifoddini [23], Ferreira

Ribeiro e Pradin [10]. Um estudo detalhado acerca dos coeficientes de similaridades é apresentado em Witte [50].

3.3 SBA - Sorting Based Algorithms [32]

Entre os métodos pertencentes a esta classe, podemos citar: ROC, DCA e BEA. King [28] desenvolveu o Rank Order Clustering (ROC). Este algoritmo incorpora os seguintes passos:

Passo 1 - Para cada linha da matriz de incidência, calcula-se um peso binário e um peso decimal equivalente.

Passo 2 - Arranja-se as linhas da matriz de incidência seguindo a ordem decrescente dos correspondentes pesos decimais.

Passo 3 - Repetir os passos acima para as colunas.

Passo 4 - Repetir os passos acima até que a posição das linhas e colunas da matriz não seja mais alterada.

O peso de cada linha i e coluna j é calculado como segue:

$$\text{linha } i: \sum_{k=1}^n a_{ik} 2^{n-k}, \text{ onde } n \text{ é o número de peças.}$$

$$\text{coluna } j: \sum_{k=1}^m a_{kj} 2^{m-k}, \text{ onde } m \text{ é o número de máquinas.}$$

Outro algoritmo desta classe é o Direct Cluster Algorithm (DCA), desenvolvido por Chan and Milner [07]. O DCA consiste dos seguintes passos:

Passo 1 - Determina-se o número total de "1" s em cada linha e coluna na matriz de incidência.

Passo 2 - Coloca-se as linhas em ordem decrescente do número total de "1" s.

Passo 3 - Coloca-se as colunas em ordem decrescente do número total de "1" s.

Passo 4 - Repetir os passos acima até que cada elemento não seja mais trocado de linha e coluna .

McCormick et al [36] desenvolveram o Bond-Energy Algorithm (BEA). O BEA procura obter uma estrutura bloco-diagonal para a matriz de incidência a_{ij} através da maximização de uma medida de eficácia. Esta medida pode ser definida por:

$$ME(A) = \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^n a_{ij} [a_{i,j-1} + a_{i,j+1} + a_{i-1,j} + a_{i+1,j}]$$

onde

A é uma matriz $n \times m$, constituída de elementos não negativos, n é o número de peças e m o número de máquinas.

O algoritmo BEA envolve os seguintes passos:

Passo 1 - Seja $i=1$

Selecionar uma das colunas arbitrariamente.

Passo 2 - Posicionar cada uma das $n - i$ colunas restantes, uma de cada vez, em cada uma das $i + 1$ posições e calcular cada contribuição das colunas por ME. Posicionar a coluna que tem a maior contribuição incrementada por ME na melhor localização, incrementar i de 1 e repetir até $i = n$.

Passo 3 - Quando todas as colunas tiverem sido posicionadas repetir o procedimento para as linhas.

3.4 CBM - Cost-Based Method [32]

Askin e Subramanian [04] desenvolveram um algoritmo de agrupamento que consiste dos seguintes custos de manufatura:

- custo fixo e variável das máquinas;
- tempo de preparação das máquinas (“set up”);
- custo do estoque;
- custo do trabalho em processo;

- custo de manuseio e transporte.

Este algoritmo é desenvolvido em três estágios. No primeiro estágio, as peças são classificadas usando um sistema de codificação. No segundo estágio, uma tentativa para desenvolver um possível agrupamento entre as peças, baseado em custos de manufatura, é realizado. No terceiro estágio, o arranjo físico das máquinas é estabelecido.

3.5 CIA - Cluster Identification Algorithm [32]

Kusiak e Chow [32] desenvolveram o algoritmo CIA. Este permite verificar a existência de um agrupamento sem movimento inter-células em uma matriz de incidência [peças x máquinas], desde que este exista. O algoritmo tem um tempo computacional baixo, de complexidade $O(2nm)$. O algoritmo de Al-Qattan [02] executa uma busca arborescente com o mesmo objetivo.

4 FPM - Formulação Programação Matemática [32]

Muitos dos modelos de programação matemática utilizados para o projeto de células de manufatura considera uma distância, ou dissimilaridade, d_{ij} medida entre duas peças i e j . Esta medida é uma função de valor real que em muitas implementações obedece aos seguintes axiomas [15]:

- reflexividade $d_{ii} = 0$;
- simetria $d_{ij} = d_{ji}$;
- desigualdade triangular $d_{ij} \leq d_{ip} + d_{pj}$.

As medidas de distâncias mais usadas são as seguintes:

1. Medida de distância de Minkowski [03]

$$d_{ij} = \left[\sum_{k=1}^n |a_{ik} - a_{jk}|^r \right]^{\frac{1}{r}}, \text{ onde } r \text{ é um número positivo e } n \text{ o número de}$$

peças.

Dois casos especiais desta medida são:

- medida da métrica absoluta para $r = 1$;
- medida da métrica absoluta para $r = 2$.

2. Medida de distância de Minkowski com peso [03]

$$d_{ij} = \left[\sum_{k=1}^n w_k |a_{ik} - a_{jk}|^r \right]^{\frac{1}{r}}, \text{ onde } w_k \text{ é o peso correspondente e } n \text{ o}$$

número de peças.

Dois casos especiais desta medida são:

- medida da métrica absoluta com peso para $r = 1$;
- medida da métrica absoluta com peso para $r = 2$.

3. Medida de distância de Hamming [34]

$$d_{ij} = \sum_{k=1}^n z(a_{ik}, a_{jk})$$

onde

n = número de peças.

$$z(a_{ik}, a_{jk}) = \begin{cases} 1 & \text{se } a_{ik} \neq a_{jk} \\ 0 & \text{caso contrário} \end{cases}$$

Os seguintes modelos de programação matemática foram propostos para auxiliar no projeto de células:

- modelo da p-mediana; (MPM)
- modelo da p-mediana generalizado; (MPMG)
- modelo da programação quadrática; (MPQ)
- modelo da programação fracionária. (MPF)

4.1 MPM - Modelo da P-Mediana [32]

Este modelo é usado para agrupar n peças em p famílias de peças. A fim de apresentá-lo, as seguintes variáveis são definidas:

n = número de peças

p = número de famílias de peças

d_{ij} = medida da dissimilaridade entre as peças i e j

$$x_{ij} = \begin{cases} 1 & \text{se a peça } i \text{ encontra-se na família} \\ & \text{formada em torno da peça } j \\ 0 & \text{caso contrário} \end{cases}$$

A função objetivo do modelo da p -mediana consiste em minimizar a soma total das dissimilaridades entre as peças reunidas em uma mesma família e a peça em torno da qual esta família foi formada.

$$(M1) \quad \min \sum_{i=1}^n \sum_{j=1}^n d_{ij} x_{ij}$$

$$\text{s.a.} \left\{ \begin{array}{l} \sum_{j=1}^n x_{ij} = 1, \text{ para todo } i=1,2,\dots,n \quad (01) \\ \sum_{j=1}^n x_{jj} = p \quad (02) \\ x_{ij} \leq x_{jj}, \text{ para todo } \begin{cases} i = 1,2,\dots,n \\ j = 1,2,\dots,n \end{cases} \quad (03) \\ x_{ij} = 0, 1 \text{ para todo } \begin{cases} i = 1,2,\dots,n \\ j = 1,2,\dots,n \end{cases} \quad (04) \end{array} \right.$$

(01) - assegura que cada peça pertença a uma só família.

(02) - especifica o número de famílias.

(03) - assegura que a peça i pertencerá a família formada em torno da peça j somente quando esta família existir.

(04) - fornece os valores possíveis para a variável x_{ij} .

4.2 MPMG - Modelo da P-Mediana Generalizado [32]

O modelo (M1) foi desenvolvido com base na existência de um único roteiro de fabricação. Para cada uma das peças, Kusiak [30] modificou (M1), de modo a considerar mais de um roteiro de fabricação para cada peça. Para formular o problema as seguintes notações são usadas:

n = número de peças

q = número de roteiros

F_k = conjunto de roteiros possíveis $k=1,2,\dots,q$

p = número requerido de famílias

d_{ij} = medida da dissimilaridade entre as peças i e j

$$x_{ij} = \begin{cases} 1 & \text{se a peça } i \text{ encontra-se na família} \\ & \text{formada em torno da peça } j \\ 0 & \text{caso contrário} \end{cases}$$

A função objetivo do modelo da p -mediana generalizado consiste em minimizar a soma total das dissimilaridades entre as peças reunidas em uma mesma família e a peça em torno da qual esta família foi formada.

$$(M2) \quad \min \sum_{i=1}^n \sum_{j=1}^n d_{ij} x_{ij}$$

$$\left. \begin{array}{l} \sum_{i \in F_k} \sum_{j \in F^k} x_{ij} = 1, \quad k = 1, 2, \dots, q \quad (05) \\ \sum_{j=1}^n x_{jj} = p \quad (06) \\ x_{ij} \leq x_{jj}, \quad \text{para todo } \begin{cases} i = 1, 2, \dots, n \\ j = 1, 2, \dots, n \end{cases} \quad (07) \\ x_{ij} = 0, 1, \quad \text{para todo } \begin{cases} i = 1, 2, \dots, n \\ j = 1, 2, \dots, n \end{cases} \quad (08) \end{array} \right\} \text{s.a.}$$

(05) - assegura que para cada peça i , somente um roteiro é utilizado.

(06)-(07)-(08) correspondem as restrições (02)-(03)-(04) respectivamente.

4.3 MPQ - Modelo de Programação Quadrática [32]

Para apresentar a formulação do modelo da programação quadrática as seguintes variáveis são definidas:

n = número de peças

p = número de família de peças

n_j = número de peças de cada família $j = 1, 2, \dots, p$

d_{ij} = dissimilaridade entre as peças i e j

$$x_{ij} = \begin{cases} 1 & \text{se a peça } i \text{ encontra-se na família} \\ & \text{formada em torno da peça } j \\ 0 & \text{caso contrário} \end{cases}$$

Esta formulação pode ser expressa na forma do seguinte modelo de programação matemática quadrática 0 - 1 :

$$(M3) \quad \min \sum_{i=1}^{n-1} \sum_{j=i+1}^n \sum_{k=1}^p d_{ij} x_{ik} x_{jk}$$

$$s.a \left\{ \begin{array}{l} \sum_{j=i+1}^n x_{ij} = 1, \text{ para todo } i = 1, 2, \dots, n-1 \quad (09) \\ \sum_{i=1}^{n-1} x_{ij} = n_j, \text{ para todo } j = i+1, 2, \dots, n \quad (10) \\ x_{ij} = 0, 1, \text{ para todo } \begin{cases} i = 1, 2, \dots, n-1 \\ j = i+1, 2, \dots, n \end{cases} \quad (11) \end{array} \right.$$

(09)- diz que cada peça é atribuída a uma só família.

(10)- impõe que cada família contenha exatamente n_j peças.

(11)- corresponde a restrição (04).

4.4 MPF - Modelo da Programação Fracionária [32]

Lashkari et al [33] estendeu o modelo (M1) ao modelo (M4) apresentado abaixo. Em sua formulação as seguintes variáveis são usadas:

n = número de peças

p = número de famílias de peças

d_{ij} = dissimilaridade entre as peças i e j

s_{ij} = similaridade entre as peças i e j

L = número mínimo de peças em cada família de peças

A função objetivo consiste em minimizar a razão da soma total das dissimilaridades e similaridades entre peças i e j .

$$(M4) \quad \min \frac{\sum_{k=1}^p \sum_{i=1}^{n-1} \sum_{j=i+1}^n d_{ij} \cdot x_{ik} \cdot x_{jk}}{\sum_{k=1}^p \sum_{i=1}^{n-1} \sum_{j=i+1}^n s_{ij} \cdot x_{ik} \cdot x_{jk}}$$

$$\text{s.a.} \left\{ \begin{array}{l} \sum_{k=1}^p x_{ik} = 1, \text{ para todo } i = 1, 2, \dots, n-1 \quad (11) \\ \sum_{i=1}^{n-1} x_{ik} \geq L, \text{ para todo } k = 1, 2, \dots, p \quad (12) \\ x_{ik} = 0, 1, \text{ para todo } \begin{cases} i = 1, 2, \dots, n-1 \\ k = 1, 2, \dots, p \end{cases} \quad (13) \end{array} \right.$$

(11)- corresponde a restrição (09).

(12)- diz que cada família consiste no mínimo de L peças.

(13)- corresponde a restrição (04).

Para resolver este modelo, Lashkari et al [33] aplicou a transformação introduzida por Glover e Woolsey [17], que consiste na substituição de cada termo $x_{ik} \cdot x_{jk}$ pela variável y_{ijk} . O novo modelo é mostrado a seguir:

$$(M5) \quad \min \frac{\sum_{k=1}^p \sum_{i=1}^{n-1} \sum_{j=i+1}^n d_{ij} y_{ijk}}{\sum_{k=1}^p \sum_{i=1}^{n-1} \sum_{j=i+1}^n s_{ij} y_{ijk}}$$

$$\text{s.a.} \left\{ \begin{array}{l} x_{ik} + x_{jk} - y_{ijk} < 1 \\ y_{ijk} < x_{ik} \\ y_{ijk} \leq x_{jk} \\ y_{ijk} = 0, 1, \text{ para todo } \begin{cases} i = 1, 2, \dots, n-1 \\ j = i+1, \dots, n \\ k = 1, 2, \dots, p \end{cases} \end{array} \right.$$

5 FG - Formulação por Grafos [32]

Três tipos de grafos, a saber: [05], [19] e [45], têm apresentado interesse especial na literatura para o modelamento e proposição de algoritmos para tratamento do problema do projeto de células de manufatura.

Grafo Bipartido: Em um grafo bipartido o conjunto de nós representa as peças e as máquinas.

Grafo de Transição: Em um grafo de transição uma peça é representada por um nó enquanto as máquinas são representadas por arcos.

Grafo limitado: Um grafo limitado consiste de uma hierarquia de grafos bipartidos. Em cada nível deste grafo, os nós representam máquinas ou peças.

Na formulação por grafos, a matriz de incidência binária [peças x máquinas] pode ser representada por um dos grafos acima. Três algoritmos baseados em teoria dos grafos são apresentados a seguir.

5.1 AGG - Algoritmo Grafo-Gargalo [35]

Seja:

N = número de nós no grafo

K = número máximo de nós no sub-grafo

O algoritmo Grafo-Gargalo [35] é constituído dos seguintes passos:

Passo 1 - Determinar o valor $m' = \left\lceil \frac{(N-1)}{K} \right\rceil > 1$

onde $\left\lceil \frac{(N-1)}{K} \right\rceil$ é o menor valor inteiro maior que $\frac{(N-1)}{K}$.

Passo 2 - Faça $m = m'$

Escolha m nós iniciais, um para cada sub-grafo.

Passo 3 - Determinar o nó comum de cada sub-grafo.

Passo 4 - Remover o nó comum do grafo. Adicionar o nó não-comum ao correspondente sub-grafo.

Passo 5 - Repita os passos 3 e 4 até todo nó ser atribuído.

O algoritmo de Lee et al [35] acima foi estendido em Vanneli e Kumar [46].

5.2 AGP - Algoritmo Grafo-Particionamento [41]

Este método decompõe um sistema de manufatura constituído de n peças e m máquinas a partir da obtenção de uma matriz de cargas de trabalho [peças x máquinas]. Representa-se esta matriz por um grafo, de modo que, o conjunto de nós representa as máquinas e as peças e o conjunto de arcos representa as ligações entre o conjunto de nós-máquinas e o conjunto de nós-peças, ou seja, um arco (i, j) indica que a máquina i é utilizada no processamento da peça j . O peso do arco (i, j) representa o tempo de processamento e/ou volume de produção. Feito isto, particiona-se o grafo em um número de subgrafos igual ao número desejado de células tal que cada subgrafo corresponda a uma célula. Esta partição é feita de modo que a soma dos pesos dos arcos que conectam os subgrafos seja mínima e além disso, o número de nós em cada subgrafo esteja limitado superiormente e inferiormente. Este método pode ser dividido em quatro passos, a saber:

Passo 1 - Obter a matriz de cargas de trabalho, $C = [\text{peças} \times \text{máquinas}]$, de dimensão $n \times m$, onde n é o número de peças e m é o número de máquinas. Os coeficientes da matriz $C = [c_{ij}]$ podem ser obtidos da seguinte maneira:

$$c[i, j] = \text{unidades}[i] * \sum_{k/\text{roteiro}[i, k] = j} \text{duração}[i, k]$$

$\text{unidade}[i] =$ número de unidades da peça $[i]$ a fabricar.

$\text{duração}[i, k] =$ duração da operação k sobre a peça $[i]$.

$\text{roteiro}[i, k] =$ tipo de máquina utilizada para executar a operação k sobre a peça $[i]$.

Passo 2 - A matriz $C = [\text{peças} \times \text{máquinas}]$ gerada no passo 1 pode ser representada por um grafo $G = (V, A)$ onde V representa o conjunto de nós-peças e nós-máquinas e A é o conjunto das arestas que indicam as conexões existentes entre as máquinas e as peças.

Passo 3 - Enquanto o número desejado de subgrafos não for obtido faça:

- Particione o grafo fazendo um corte de um conjunto de arestas (conjunto-corte) tal que se obtenha dois subgrafos conectados pelo conjunto-corte. A soma dos pesos das arestas representa o movimento inter-celular. O objetivo é encontrar o conjunto-corte que possua a menor soma de pesos. Para isto, aplica-se o algoritmo de Gomory e Hu [18]. Se o número desejado de células foi obtido, fim do laço. Senão, particione cada um dos subgrafos obtidos até obter o número desejado de subgrafos, com o conjunto-corte que possua a menor soma de pesos;

Fim do enquanto.

Passo 4 - Obtenção das células de manufatura.

5.3 AGC - Algoritmo Grafo-Coloração [12]

Em uma primeira fase, esse método obtém a matriz de cargas de trabalho [peças x máquinas] e computa as dissimilaridades entre as peças a fabricar. Feito isto, calcula-se a dissimilaridade crítica que é igual ao valor da dissimilaridade média, constrói-se o grafo de dissimilaridades de maneira que existirá aresta entre dois nós que representam peças somente se a dissimilaridade entre essas peças for maior ou igual a dissimilaridade crítica e faz-se então a coloração do grafo, onde nós adjacentes recebem cores diferentes. Esta coloração é realizada procurando-se utilizar o menor número de cores possível. As peças que possuem as mesmas cores são aglomeradas formando-se as famílias de peças, de modo que o número de cores é igual ao número de famílias. Se o número de cores obtido é igual ao número desejado de células, o processo pára e então



são obtidos os grupos de máquinas: estas são atribuídas à família onde executam o maior número de operações. Se o número de cores é maior (menor) do que o número de células desejado aumenta-se (diminui-se) o valor da dissimilaridade crítica de uma unidade até que o número de cores encontrado seja igual ao número de células.

O algoritmo correspondente ao método proposto em [12] é constituído de quatro passos fundamentais:

Passo 1 - Determinação da matriz de cargas de trabalho.

Passo 2 - Computação das dissimilaridades entre as peças.

Passo 3 - Enquanto o número desejado de células não for obtido fazer:

- determinação da dissimilaridade crítica;
- construção do grafo de dissimilaridades;
- coloração do grafo de dissimilaridades (através do método de ligações-contrações [05], [45]);

Fim do enquanto.

Passo 4 - Obtenção dos grupos de máquinas.

6 SE - Sistemas Especialistas [32]

Kusiak [29], [31] aplica um sistema de arquitetura Tandem para tratar o problema da TG. O sistema proposto permite realizar o projeto de células de manufatura para um sistema de produção de grande porte em tempo computacional aceitável.

Este sistema considera roteiros de fabricação alternativos para as peças e avalia cada solução parcial gerada através de um algoritmo de partição selecionado.

O problema de agrupamento em um sistema de manufatura automatizado é formulado por Kusiak [29] da seguinte forma: determinar as células de máquinas e as famílias de peças e selecionar um AGV (transportador automático de material), com correspondente custo mínimo, levando-se em consideração as seguintes restrições:

Restrição 1 - O tempo de processamento disponível em cada máquina não é excedido.

Restrição 2 - O limitante superior sobre a frequência de viagens do transportador de material para cada célula não é excedido.

Restrição 3 - O número de máquinas em cada célula não excede o limite superior ou a dimensão de cada célula não é excedida.

Restrição 4 - Algumas máquinas devem ser incluídas nas mesmas células devido às necessidades tecnológicas.

Um exemplo numérico resolvido por este método pode ser encontrado em Kusiak [29].

7 Comentários

Neste capítulo, foram apresentados vários métodos e modelos para decompor um sistema de produção em células de manufatura. Estes métodos foram divididos em seis classes principais:

- método visual;
- método da codificação;
- formulação matricial;
- formulação por programação matemática;
- formulação por grafos;
- sistemas especialistas.

Em nosso trabalho é proposto, implementado e testado um método de formação de células de manufatura baseado em teoria dos grafos e mais especificamente em coloração em grafos. Os conceitos fundamentais de grafos e coloração, o método desenvolvido e os resultados obtidos são apresentados nos capítulos seguintes.

O grafo da figura 3.1 possui $V = \{ 1, 2, 3, 4, 5, 6, 7, 8 \}$, $|V| = 8$ e $A = \{ (1,2), (1,4), (1,5), (1,6), (1,8), (2,3), (2,4), (2,8), (3,4), (3,7), (4,5), (4,6), (4,7), (4,8), (5,6), (5,7), (5,8), (6,7), (6,8), (7,8) \}$

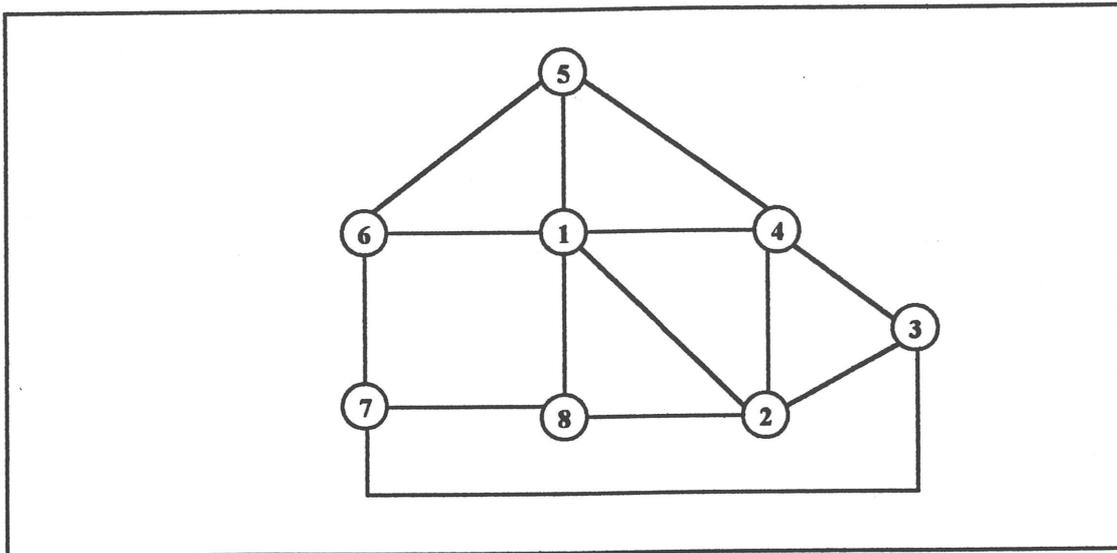


figura 3.1 - Representação geométrica de um grafo.

Definição 3.3 Uma sequência de vértices v_1, v_2, \dots, v_k tal que $(v_j, v_{j+1}) \in A$, $1 \leq j \leq k-1$, é denominado *caminho de* v_1 a v_k . Se todos os vértices do caminho v_1, v_2, \dots, v_k forem distintos, a sequência recebe o nome de *caminho simples ou elementar*.

Definição 3.4 Um *ciclo* é um caminho $v_1, v_2, \dots, v_k, v_{k+1}$ sendo $v_1 = v_{k+1}$ e $k \geq 3$. Se o caminho v_1, v_2, \dots, v_k for simples, o ciclo $v_1, v_2, \dots, v_k, v_{k+1}$ é denominado *simples ou elementar*.

No grafo da figura 3.1 o caminho 1,2,3,4,1 é um ciclo simples.

Definição 3.5 Um grafo $G(V,A)$ é denominado *conexo* quando existe caminho entre cada par de vértices de $G(V,A)$. Caso contrário $G(V,A)$ é *desconexo*.

Seja $G(V,A)$ um grafo, $a \in A$ uma aresta. Denota-se por $G - a$ o grafo obtido de G , pela exclusão da aresta a . Se v,w é um par de vértices não adjacentes em G , a notação $G + (v,w)$ representa o grafo obtido adicionando-se a aresta (v,w) .

Analogamente, seja $v \in V$ um vértice de G . O grafo $G - v$ denota aquele obtido de G pela remoção do vértice v . Observe que excluir um vértice implica em remover de G o vértice em questão e as arestas a ele incidentes. Da mesma forma, $G + w$ representa o grafo obtido adicionando-se a G o vértice w .

De um modo geral, se G é um grafo e S um conjunto de arestas ou vértices, $G - S$ e $G + S$ denotam, respectivamente, o grafo obtido de G pela exclusão e inclusão de S , respectivamente, veja figura 3.2.

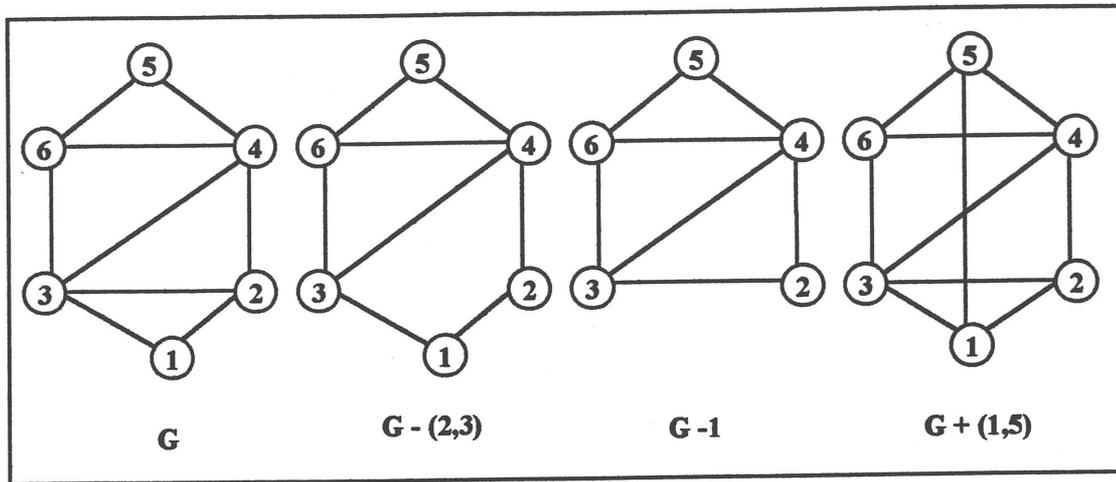


figura 3.2 - Soma e subtração de vértices e arestas.

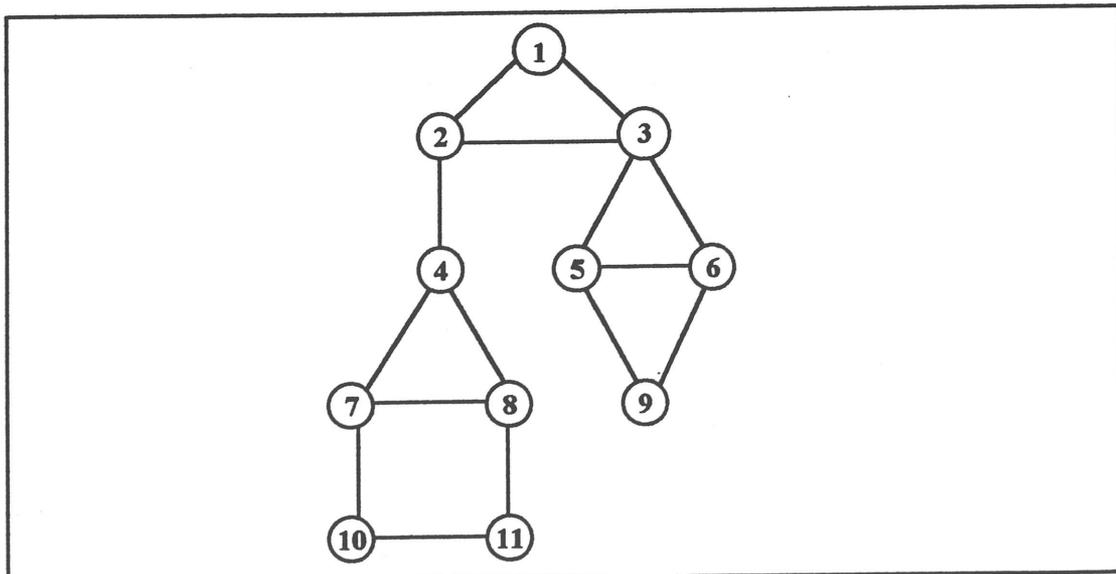


figura 3.3 - Articulação e ponte.

Definição 3.6 Um vértice v é denominado *articulação* quando sua remoção de G o desconecta, isto é, $G - v$ torna-se desconexo.

Definição 3.7 Uma aresta $a \in A$ é chamada *ponte* quando sua remoção de G o desconecta. Nesse caso, $G - a$ também torna-se desconexo.

Na figura 3.3 os vértices 2 e 3 são uma articulação, e a aresta (2,4) é uma ponte.

Definição 3.8 Um grafo $G(V,A)$ é chamado de *trivial* quando $|V| = 1$.

Definição 3.9 Seja $V' \subseteq V$, cuja remoção de G o desconecta ou o transforma no grafo trivial e para todo subconjunto próprio $V'' \subset V'$, $G - V''$ é conexo e não trivial. Diz-se que V' é um *corte de vértices* de G . Analogamente, um *corte de arestas* de G é um subconjunto $A' \subseteq A$, cuja remoção de G o desconecta, e para todo subconjunto próprio $A'' \subset A'$, $G - A''$ é conexo.

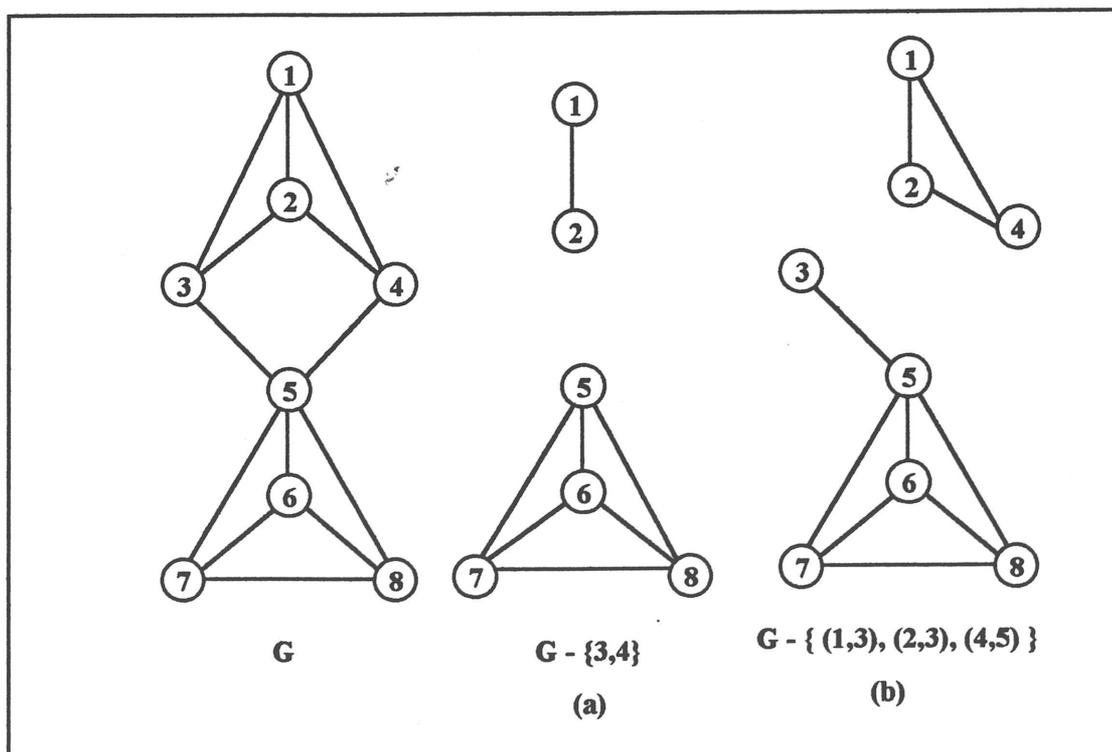


figura 3.4 - Corte de vértices e arestas.

Na figura 3.4 o subconjunto $\{3,4\}$ é um corte de vértices, pois sua remoção desconecta o grafo G . A figura 3.4.(a) ilustra o caso. Em relação a este mesmo grafo, o subconjunto de arestas $\{(1,3), (2,3), (4,5)\}$ é um corte de arestas, porque removendo-o de

G produz-se o grafo desconexo figura 4(b). Observe também que removendo de G o subconjunto de vértices $\{3,4,7\}$, G é desconectado. Contudo, $\{3,4,7\}$ não é um corte de vértices, pois contém propriamente o corte $\{3,4\}$.

Definição 3.10 Diz-se que um grafo é *completo* quando existe uma aresta entre cada par de seus vértices.

Definição 3.11 Um grafo $G(V,A)$ é *bipartido* quando o seu conjunto de vértices V puder ser particionado em dois subconjuntos V_1, V_2 tais que toda aresta de G une um vértice de V_1 a outro de V_2 . Um grafo *bipartido completo* possui uma aresta para cada par de vértices v_1, v_2 sendo $v_1 \in V_1$ e $v_2 \in V_2$.

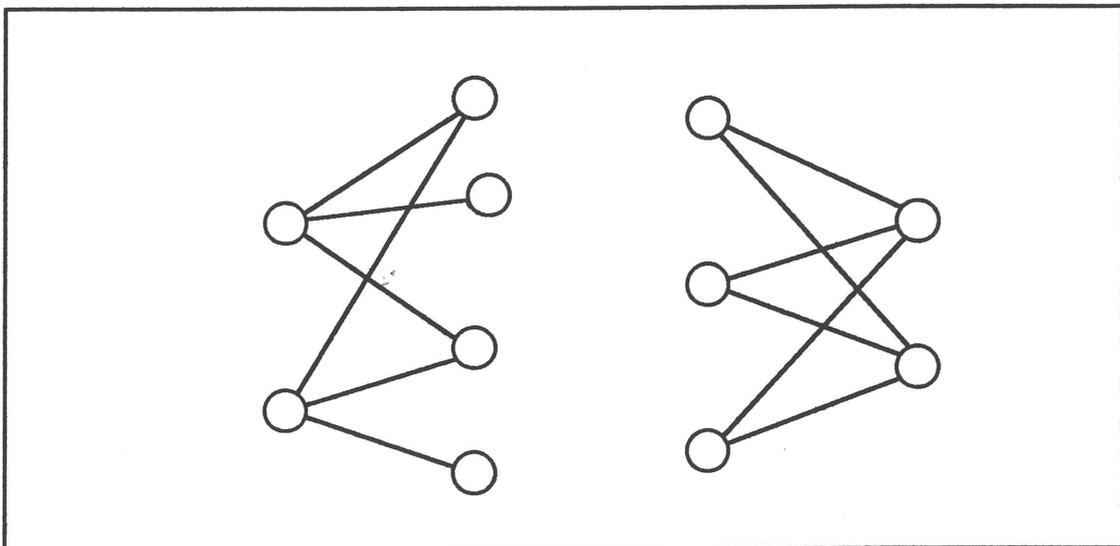


figura 3.5 - Grafo bipartido e bipartido completo.

A figura 3.5 ilustra dois grafos bipartidos, mas somente o da direita é um grafo bipartido completo.

Definição 3.12 Sejam $G_2(V_2, A_2)$ e $G_1(V_1, A_1)$ dois grafos. Se $V_2 \subseteq V_1$ e $A_2 \subseteq A_1$ dizemos que $G_2(V_2, A_2)$ é um *subgrafo* do grafo $G_1(V_1, A_1)$.

Um nome especial é dado aos subgrafos $G_2(V_2, A_2)$ de um grafo $G_1(V_1, A_1)$, a saber, *subgrafos induzidos*, se $G_2(V_2, A_2)$ satisfizer a seguinte propriedade: toda aresta (v, w) de $G_1(V_1, A_1)$ é aresta de $G_2(V_2, A_2)$ onde $v, w \in V_2$, ou seja, se $v, w \in V_2$ e a aresta $(v, w) \in A_1$ então $(v, w) \in A_2$, veja figura 3.6.

Um subgrafo *gerador* de um grafo $G_1(V_1, A_1)$ é um subgrafo $G_2(V_2, A_2)$ tal que $V_2 = V_1$.

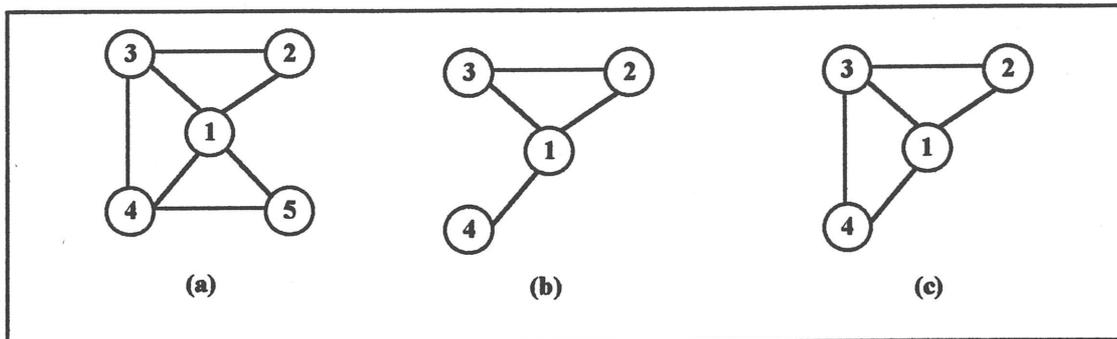


figura 3.6 - Subgrafo gerador e induzido.

Os grafos da figura 3.6(b) e 3.6(c) são ambos subgrafos da figura 3.6(a). Mas somente o da figura 3.6(c) é induzido.

Definição 3.12 Uma *árvore* é um grafo conexo e sem ciclos. Um conjunto de árvores é uma *floresta*.

Teorema 1: Toda árvore com n vértices possui $n-1$ arestas.

Teorema 2: Um grafo $G(V, A)$ é uma árvore se e somente se existir um único caminho entre cada par de vértices de $G(V, A)$.

Teorema 3: Seja $G(V, A)$ um grafo. As seguintes afirmações são equivalentes:

- a- $G(V, A)$ é uma árvore
- b- $G(V, A)$ é conexo e $|A|$ é mínima
- c- $G(V, A)$ não possui ciclos e $|A| = |V| - 1$
- d- $G(V, A)$ é conexo e $|A| = |V| - 1$
- e- $G(V, A)$ não possui ciclos e para $\forall i, j \in V$ a adição da aresta $(i, j) \notin A$ produz um grafo com exatamente um ciclo.

Teorema 4: *Todo grafo conexo $G(V,A)$ possui uma árvore geradora.*

Definição 3.14 Diz-se que um grafo $G(V,A)$ é *valorado* quando a cada aresta de A está associado um peso. O significado deste peso depende de cada problema específico, podendo corresponder a distâncias, fluxos, etc.

Definição 3.15 Um grafo $G(V,A)$ é dito *orientado* quando os elementos de A são ordenados.

A figura 3.7, abaixo, mostra um grafo valorado e orientado. Neste grafo, $V = \{1, 2, 3, 4\}$, $|V| = 4$, $A = \{(1,2), (1,3), (2,3), (2,4), (3,4)\}$ com pesos respectivamente iguais a 3, 2, 5, 1 e 4.

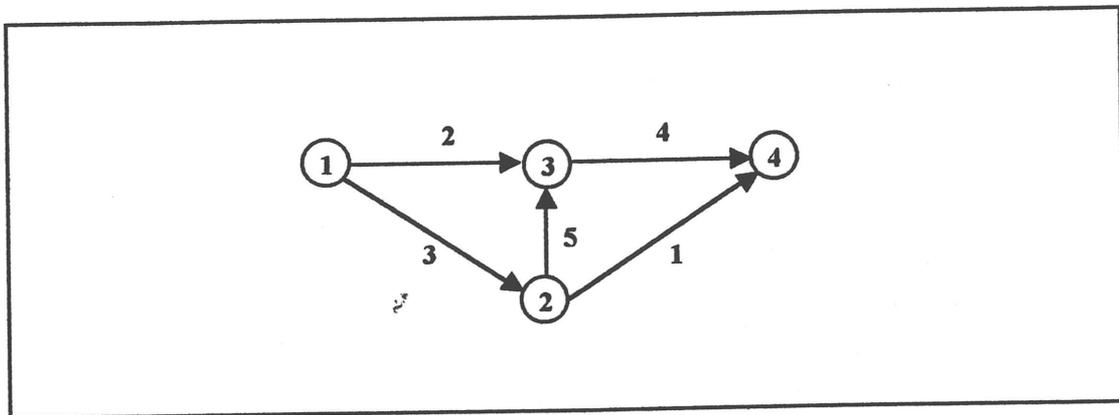


figura 3.7 - Grafo valorado e orientado

Definição 3.16 Uma árvore é chamada de *árvore de peso máximo (mínimo)* quando for um subgrafo gerador de um grafo $G(V,A)$ e a soma dos pesos pertencentes às suas arestas for máxima (mínima).

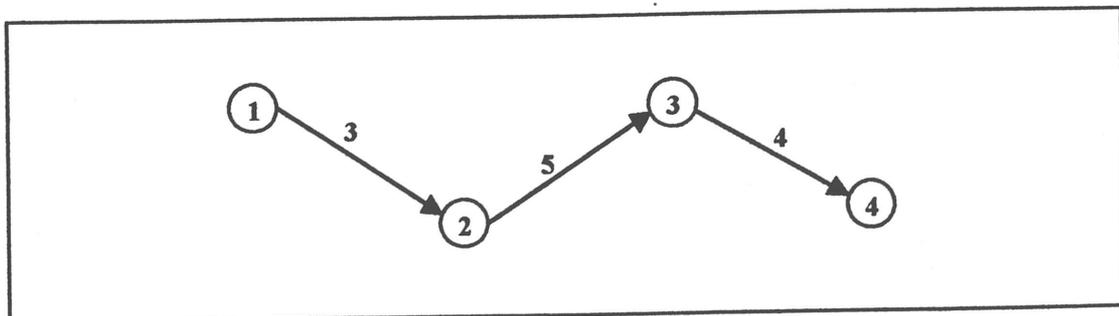


figura 3.8 - Uma árvore de peso máximo.

A figura 3.8 fornece a árvore de peso máximo associada ao grafo da figura 3.7.

Definição 3.17 Seja $G(V,A)$ um grafo. Dizemos que um subgrafo de $G(V,A)$ é uma *clique* se este for completo.

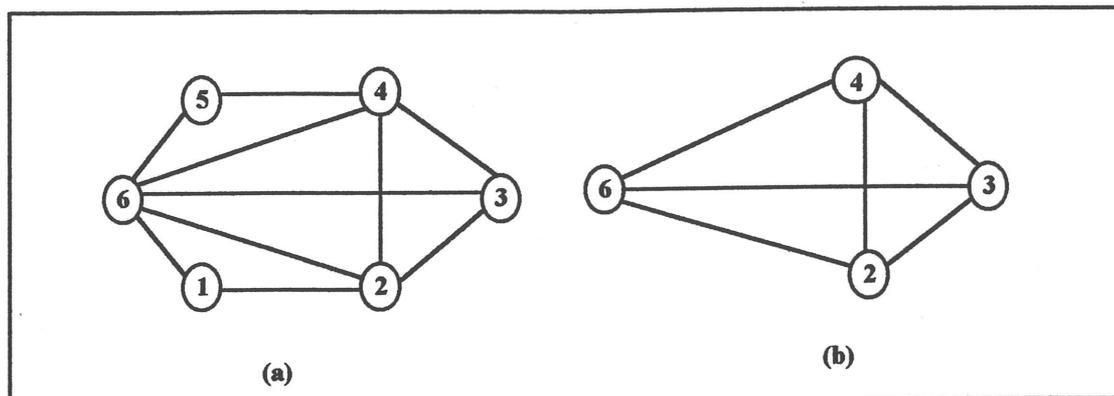


figura 3.9 - Representação de uma clique.

O subgrafo ilustrado na figura 3.9(b) é uma clique do grafo 3.9(a).

Definição 3.18 Seja $G(V,A)$ um grafo. Diz-se que um subgrafo induzido de $G(V,A)$ é um subconjunto *independente de vértices*, se este for totalmente desconexo, isto é, não há arestas entre qualquer par de vértices.

Na figura 3.9 (a) o conjunto $\{1,3,5\}$ é independente de vértices.

O tamanho de uma clique ou conjunto independente de vértices é igual à cardinalidade de seu conjunto de vértices.

Definição 3.19 Em um grafo $G(V,A)$, define-se *grau* de um vértice $v \in V$, denotado por $\text{grau}(v)$, como sendo o número de vértices adjacentes a v .

2 Coloração de Grafos

Considere um grafo $G(V,A)$ e um conjunto de cores $C = \{c_i : i \in \mathbb{N}\}$. Uma coloração de $G(V,A)$ é uma atribuição de alguma cor de C para cada vértice de V , de tal modo que a dois vértices adjacentes sejam atribuídas cores diferentes.

Definição 3.20 Uma *coloração* de G é uma função $f: V \rightarrow C$ tal que para cada par de vértices $v, w \in V$ tem-se $(v, w) \in A \Rightarrow f(v) \neq f(w)$.

Uma k -coloração de G é uma coloração que utiliza um total de k cores. Diz-se então que G é k -colorível.

Definição 3.21 Denomina-se *número cromático* de um grafo G e denota-se por $\chi(G)$ ao menor número de cores k , para o qual existe uma k -coloração de G .

É muito fácil colorir um grafo de n vértices: basta utilizar n cores, uma para cada vértice. Porém, o problema de se determinar um procedimento eficiente para realizar a coloração de um grafo com um número mínimo de cores é bastante complexo. Até o presente momento, não se conhece algoritmo eficiente [45] para a determinação do número cromático de um grafo. Contudo, o caso especial de bicoloração, pode ser resolvido de forma simples.

Teorema 5: Um grafo $G(V, A)$ é bicromático se e somente se for bipartido [05], [45].

Dem: Se $G(V, A)$ é bipartido podemos separar V em dois conjuntos $V_1, V_2 \subseteq V$, tal que, $V_1 \cup V_2 = V$. Atribua a cor c_1 aos vértices de V_1 e a cor c_2 aos de V_2 , portanto $G(V, A)$ é bicromático. Reciprocamente se G é bicromático, considere uma bicoloração de $G(V, A)$, com cores c_1 e c_2 . Sejam V_1 e V_2 os subconjuntos de vértices que possuem as cores c_1 e c_2 , respectivamente. Então V_1 e V_2 biparticionam $G(V, A)$.

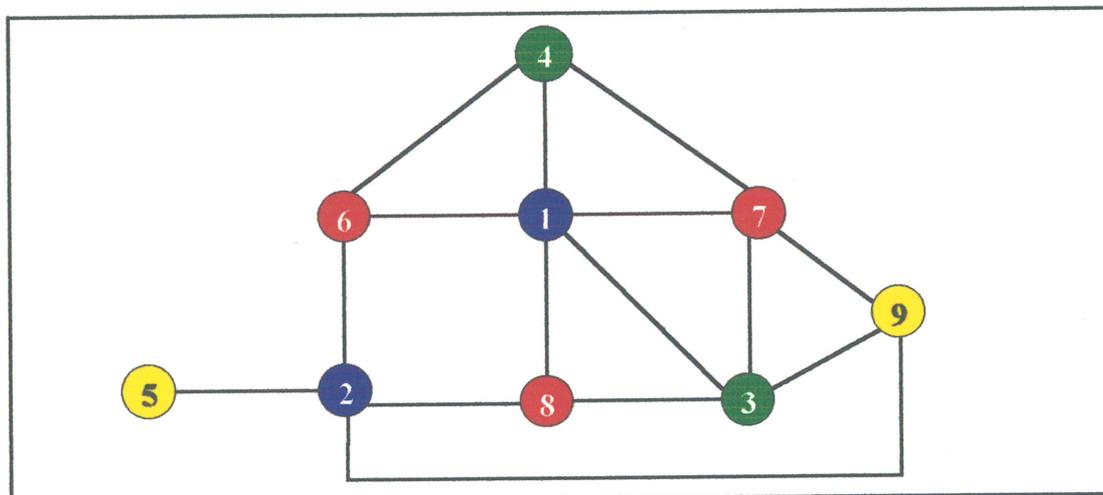
Os conceitos de coloração, clique e conjunto independente de vértices estão naturalmente relacionados. De fato, como são necessárias p cores para colorir os $p = 1, \dots, k$ vértices de uma clique de tamanho k , conclui-se que o número cromático de um grafo G é maior ou igual que o tamanho da maior clique de G . Considere agora uma k -coloração de $G(V, A)$. Sejam V_1, V_2, \dots, V_k os subconjuntos de V onde os vértices de V_1, V_2, \dots, V_k são coloridos com as cores c_1, c_2, \dots, c_k respectivamente, assim $\bigcup V_i = V$, $i = 1, 2, \dots, k$, e cada V_i é um conjunto independente de vértices. Então o problema de determinar uma coloração mínima de G pode ser formulado em termos de particionar V em um número mínimo de conjuntos independentes de vértices.

3 Algoritmos de Coloração

Neste item, apresentamos dois procedimentos clássicos descritos na literatura para a coloração em grafos: o procedimento guloso de Prim [01], [19] e o procedimento por ligações-contrações [05], [45]. O primeiro é um procedimento polinomial sem garantia de otimalidade para a solução obtida. O segundo tem complexidade avaliada em $O(2^n)$, mas garante a obtenção da solução ótima.

3.1 Algoritmo Guloso [01], [19]

O algoritmo guloso para coloração em grafos é baseado no grau dos nós, procurando-se colorir em primeira instância os nós de maior grau.



figuras 3.10 - Coloração utilizando o algoritmo guloso.

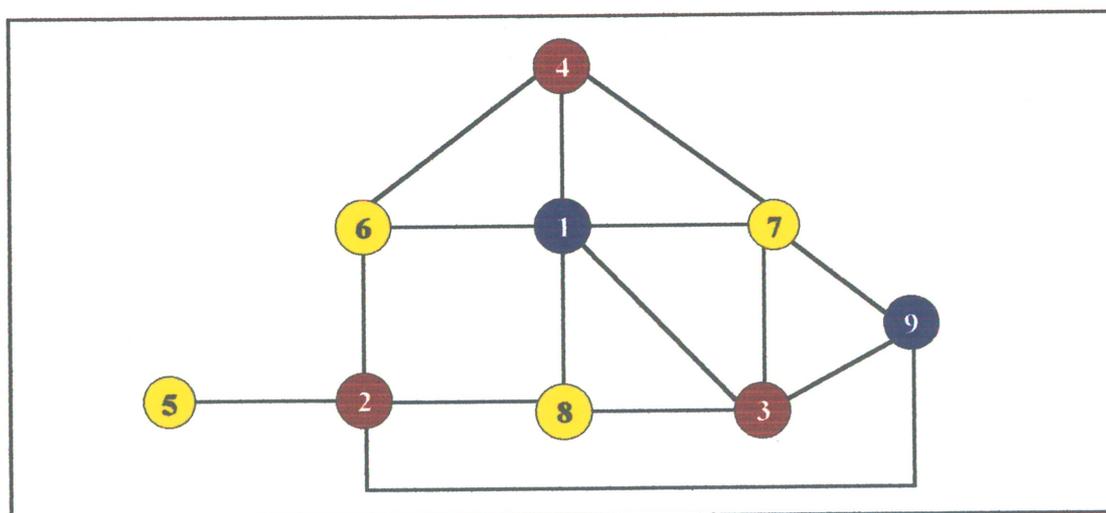


figura 3.11 - Coloração ótima.

A utilização do algoritmo guloso para colorir os nós do grafo da figura 3.10 fornece uma solução com 4 cores.

A solução ótima figura 3.11 possui 3 cores.

Algoritmo guloso

Dado $G(V,A)$

Enquanto existir nó a ser colorido fazer:

- colorir o nó de maior grau (em caso de empate, colorir o nó de menor rótulo);

Fim Enquanto.

3.2 Algoritmo de Ligações-Contrações [05], [45]

O algoritmo de ligações-contrações é baseado na transformação estrutural de um grafo $G(V,A)$ em um grafo $G'(V',A')$ de modo que se possa resolver um problema P' em G' correspondente à resolução de um problema P em G . Duas transformações estruturais são bastante comuns:

- Ligação: Introdução de uma aresta $(i,j) \notin A$ em $G(V,A)$
- Contração: Fusão de dois nós não adjacentes $i,j \in V$ em um só nó

Seja o grafo $G(V,A)$ tal que existem 2 vértices não adjacentes $i, j \in V$. Caso contrário, G seria completo e o número mínimo de cores necessárias para colorí-lo seria igual a $|V|$. Seja G_1 o grafo obtido de G pela ligação dos nós i, j e G_2 o grafo obtido de G pela contração dos vértices i e j . Repete-se o processo nos grafos G_1, G_2 obtendo novos subgrafos. Este procedimento continua até que todos os subgrafos sejam uma clique. Faz-se então a coloração destes obtendo o número de cores desejado. A figura 3.12 ilustra esse procedimento.

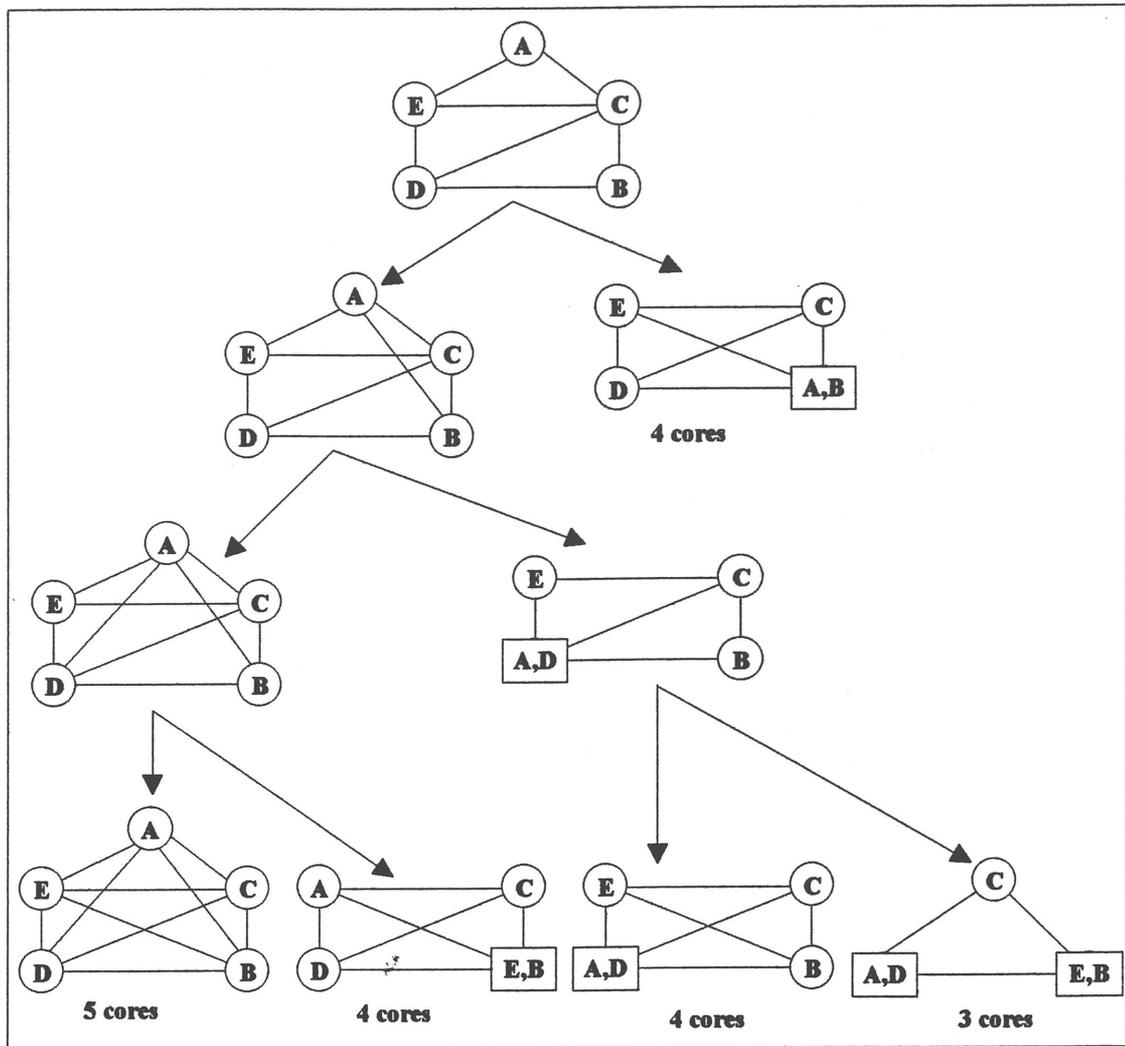


figura 3.12 - Aplicação do algoritmo de ligações e contrações.

Capítulo 4

Um Algoritmo de Coloração para o Projeto de Células de Manufatura

Neste capítulo apresentamos um estudo detalhado do método para decomposição de sistemas de produção em células de manufatura desenvolvido com base na bicoloração de uma árvore de peso máximo.

1 Dados do Problema

Para a implementação do método de particionamento em células proposto e a subsequente programação das tarefas a executar é necessário colher do sistema de produção os seguintes dados:

- os roteiros de fabricação das peças;
- a duração das operações feitas pelas máquinas sobre as peças;
- quantidade a ser produzida de cada peça.

peças	unidades	roteiro	duração
1	1	2 10 11 12	1 1 1 1
2	1	3 5 8 13 15	1 1 1 1 1
3	1	1 6 9 14	1 1 1 1
4	1	1 4 9 14	1 1 1 1
5	1	3 5 8 13 15	1 1 1 1 1
6	1	1 4 6 9 14	1 1 1 1 1
7	1	7 10 11 12	1 1 1 1
8	1	3 5 8 13 15	1 1 1 1 1
9	1	4 6 9 14	1 1 1 1
10	1	2 7 10 11 12	1 1 1 1 1

tabela 4.1

Para ilustrar a apresentação do método passo a passo é utilizado o exemplo mostrado na tabela 4.1 devido a Chan e Milner [07]. Esta tabela fornece os roteiros de fabricação e duração das peças de um sistema de produção composto de 10 peças e 15 máquinas e deve-se particioná-lo em 3 células de manufatura.

2 Algoritmo

O algoritmo correspondente ao método proposto é constituído de quatro passos fundamentais:

Passo 1 - Determinação da matriz de cargas de trabalho.

Passo 2 - Computação das dissimilaridades entre as peças.

Passo 3 - Enquanto o número desejado de células não for obtido fazer:

- identificar a família de peças com o maior diâmetro;
- construir uma árvore de peso máximo com os membros desta família;
- efetuar a bicoloração da árvore construída no item anterior;
- obter as famílias de peças.

Fim Enquanto.

Passo 4 - Obtenção dos grupos de máquinas.

Nos itens 3, 4, 5 e 6, a seguir, os passos fundamentais do algoritmo acima são explicados com o auxílio do exemplo ilustrativo.

3 Matriz de Cargas de Trabalho

A partir dos dados iniciais obtém-se uma matriz [peças x máquinas], que fornece o tempo total que cada peça gasta na passagem em cada máquina do seu roteiro. Esta matriz é chamada de matriz de cargas de trabalho e seus coeficientes são calculados da seguinte maneira:

$$\text{Carga}[i,j] = \text{unidade}[i] * \sum_{k/\text{roteiro}[i,k]=j} \text{duração}[i,k]$$

onde

$\text{unidade}[i]$ = número de unidades da peça[i] a fabricar.

duração[i,k] = duração da operação k sobre a peça[i].

roteiro[i,k] = tipo de máquina utilizada para executar a operação k sobre a peça[i].

A matriz de cargas de trabalho [peças x máquinas] correspondente ao sistema de produção mostrado na tabela 4.1 é fornecida pela tabela 4.2.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0	1	0	0	0	0	0	0	0	1	1	1	0	0	0
2	0	0	1	0	1	0	0	1	0	0	0	0	1	0	1
3	1	0	0	0	0	1	0	0	1	0	0	0	0	1	0
4	1	0	0	1	0	0	0	0	1	0	0	0	0	1	0
5	0	0	1	0	1	0	0	1	0	0	0	0	1	0	1
6	1	0	0	1	0	1	0	0	1	0	0	0	0	1	0
7	0	0	0	0	0	0	1	0	0	1	1	1	0	0	0
8	0	0	1	0	1	0	0	1	0	0	0	0	1	0	1
9	0	0	0	1	0	1	0	0	1	0	0	0	0	1	0
10	0	1	0	0	0	0	1	0	0	1	1	1	0	0	0

tabela 4.2

4 Computação das Dissimilaridades

O método de computação das dissimilaridades entre peças adotado no método proposto leva em consideração as diferenças existentes entre os roteiros de fabricação das peças. Dadas as peças P_i e P_j definidas pelos vetores 0/1 abaixo:

$$P_i = [a_{i1}, a_{i2}, \dots, a_{ik}, \dots, a_{im}]$$

$$P_j = [a_{j1}, a_{j2}, \dots, a_{jk}, \dots, a_{jm}]$$

onde

m = é o número de máquinas

$$a_{ik} = \begin{cases} 1 & \text{se } a_{ik} \neq 0 \\ 0 & \text{caso contrário} \end{cases}$$

Obs: quando $a_{ik} \neq 0$ a peça[i] sofre uma operação na máquina k.

Denota-se por $d(P_i, P_j)$ a dissimilaridade entre as peças P_i e P_j e esta é dada por:

$$d(P_i, P_j) = \sum_{k=1}^m \Theta(a_{ik}, a_{jk})$$

onde

$$\Theta(a_{ik}, a_{jk}) = \begin{cases} 1 & \text{se } a_{ik} \neq a_{jk} \\ 0 & \text{caso contrário} \end{cases}$$

A matriz de dissimilaridades para o exemplo utilizado é dada abaixo pela tabela 4.3.

	1	2	3	4	5	6	7	8	9	10
1	0	9	8	8	9	9	1	9	8	1
2	9	0	9	9	0	10	10	0	9	10
3	8	9	0	2	9	1	9	9	2	9
4	8	9	2	0	9	1	9	9	2	9
5	9	0	9	9	0	10	10	0	9	10
6	9	10	1	1	10	0	10	10	1	10
7	1	10	9	9	10	10	0	10	9	0
8	9	0	9	9	0	10	10	0	9	10
9	8	9	2	2	9	1	9	9	0	9
10	1	10	9	9	10	10	0	10	9	0

tabela 4.3

5 Decomposição do Sistema de Manufatura

Após a obtenção das matrizes de cargas de trabalho e de dissimilaridades, é realizada a decomposição do sistema de manufatura da seguinte maneira:

- identificação da família de peças de maior diâmetro;
- construção e coloração de uma árvore de peso máximo;
- obtenção das famílias de peças;
- obtenção dos grupos de máquinas.

5.1 Escolha da Família de Maior Diâmetro

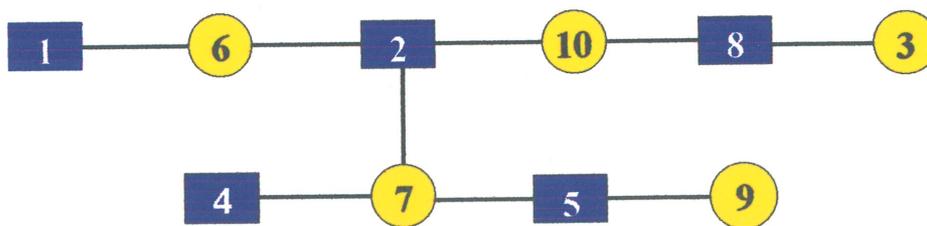
Seja V o conjunto das peças e $D = [d_{ij}]$ uma matriz de dissimilaridades entre as peças $i, j \in V$, com $d_{ij} \geq 0$ e $d_{ii} = 0$. Seja $P = \{C_1, C_2, \dots, C_q\}$ uma partição de V em q famílias. O diâmetro de uma família é igual à maior dissimilaridade intra-famílias, ou

seja, $\delta(C_k) = \{ \max(d_{ij}); i, j \in C_k \}$ e o diâmetro de uma partição é igual ao maior diâmetro das famílias, ou seja, $\Delta(P) = \{ \max \delta(C_k) \}; k = 1, 2, \dots, q$.

5.2 Construção e Coloração de uma Árvore de Peso Máximo

O algoritmo utilizado no método proposto para construir e colorir uma árvore de peso máximo é o algoritmo guloso de Prim [19]: inicia-se a construção da árvore de peso máximo por qualquer peça, atribuindo-lhe uma cor qualquer. A peça que possui o maior valor de dissimilaridade com a peça escolhida inicialmente será a próxima a ser inserida na árvore, e receberá uma cor diferente da cor atribuída à peça inicial. O processo continua escolhendo a peça que possui a maior dissimilaridade entre a peça inicial e a segunda peça escolhida e assim sucessivamente. A coloração da árvore é feita de maneira que as peças adjacentes possuem cores diferentes e as não adjacentes recebem a mesma cor. No final do processo a árvore de peso máximo estará colorida com duas cores apenas, de modo que as peças que possuírem a mesma cor serão aglomeradas, formando duas famílias de peças. O resultado obtido através deste algoritmo é ótimo, uma vez que o problema de obtenção de uma árvore de peso máximo é um matróide [19].

Uma árvore de peso máximo para o exemplo proposto está construída abaixo:



Esta árvore de peso máximo não é única: observe que a peça 9 poderia receber qualquer uma das duas cores, uma vez que esta possui uma dissimilaridade com a peça 10 igual a dissimilaridade que esta possui com a peça 5.

O teorema 6 proposto por Guénoche, demonstrado abaixo, garante que a bipartição obtida pela bicoloração de uma árvore de peso máximo é de diâmetro mínimo. Este resultado é a base sobre a qual podemos desenvolver um método de partição de sistemas de manufatura que possibilita a aglomeração de peças semelhantes em uma mesma célula de manufatura.

Teorema 6: *A partição em duas classes obtidas pela bicoloração de uma árvore de peso máximo é de diâmetro mínimo [21].*

Dem: Seja $P = \{C_1, C_2\}$ a partição obtida pela bicoloração de uma árvore de peso máximo A . Seja (i, j) uma aresta tal que $\delta(i, j) =$ maior diâmetro intra-classe. Esta aresta não pode pertencer a A uma vez que i e j estão na mesma classe e que as arestas da árvore de peso máximo, por construção das classes ligam um nó de C_1 a um nó de C_2 . Seja $U = \{i = i_1, i_2, \dots, i_p = j\}$ o caminho em A entre i e j e $T = \{i_1, i_2, \dots, i_p\}$. Por construção das classes, T tem um número ímpar de nós. A inserção de (i, j) em U forma um ciclo de comprimento ímpar. Disto se deduz que para toda bipartição de T existe uma aresta $\delta(i_n, i_{n+1}) \geq \delta(i, j)$. De onde conclui que toda bipartição de T tem diâmetro superior ou igual a $\delta(i, j)$ e portanto a bipartição de P tem diâmetro mínimo.

5.3 Obtenção das Famílias de Peças

O método é iterativo e na primeira iteração as peças são particionadas em duas famílias através da bicoloração da árvore de peso máximo. Caso o número desejado de famílias seja igual a 2, o processo pára e as famílias são apresentadas. Caso contrário, inicia-se a segunda iteração escolhendo-se a família de maior diâmetro: constrói-se então uma árvore de peso máximo para esta família, particionando-a em duas famílias de peças. Assim, obtém-se três famílias e se este é o número de famílias desejado, o processo pára e as famílias são apresentadas. Se este não for o caso, o processo continua até que o número desejado de famílias seja obtido.

As peças que no processo de construção da árvore de peso máximo puderem ser atribuídas a duas famílias distintas por apresentar empate no valor de dissimilaridade, não são atribuídas a nenhuma delas em um primeiro momento. Após ter-se obtido todas as famílias de peças desejadas, faz-se uma varredura, onde as peças que ainda não pertencem a nenhuma família, são então atribuídas à família que possuir a menor dissimilaridade com estas.

Para o exemplo proposto, tem-se na primeira iteração as seguintes famílias:

$$\text{família1} = \{2, 5, 8\}$$

$$\text{família2} = \{1, 3, 4, 6, 7, 10\}$$

Nesse exemplo a peça 9 não será atribuída inicialmente a nenhuma família pois esta poderia estar em qualquer delas.

A família1 tem diâmetro 0, a família2 tem diâmetro 9. Escolhe-se, então, a família2. Constrói-se a árvore de peso máximo para esta família particionando-a da seguinte maneira:

$$\text{família2.1} = \{1, 7, 10\}$$

$$\text{família2.2} = \{3, 4, 6\}$$

A peça 9 será atribuída a família2.2, pois são com as peças atribuídas a esta família que a peça 9 possui a menor dissimilaridade (igual a zero).

As três famílias obtidas são:

$$\text{família1} = \{2, 5, 8\}$$

$$\text{família2} = \{1, 7, 10\}$$

$$\text{família3} = \{3, 4, 6, 9\}$$

5.4 Obtenção dos Grupos de Máquinas

As máquinas disponíveis na fábrica são atribuídas às famílias de peças onde executam o maior número de operações de modo a definir-se células de manufatura com o menor número de movimentos inter-células possível.

máquinas	família1	família2	família3
1	0	0	3
2	2	0	0
3	0	3	0
4	0	0	3
5	0	3	0
6	0	0	3
7	2	0	0
8	0	3	0
9	0	0	4
10	3	0	0
11	3	0	0
12	3	0	0
13	0	3	0
14	0	0	4
15	0	3	0

tabela 4.6

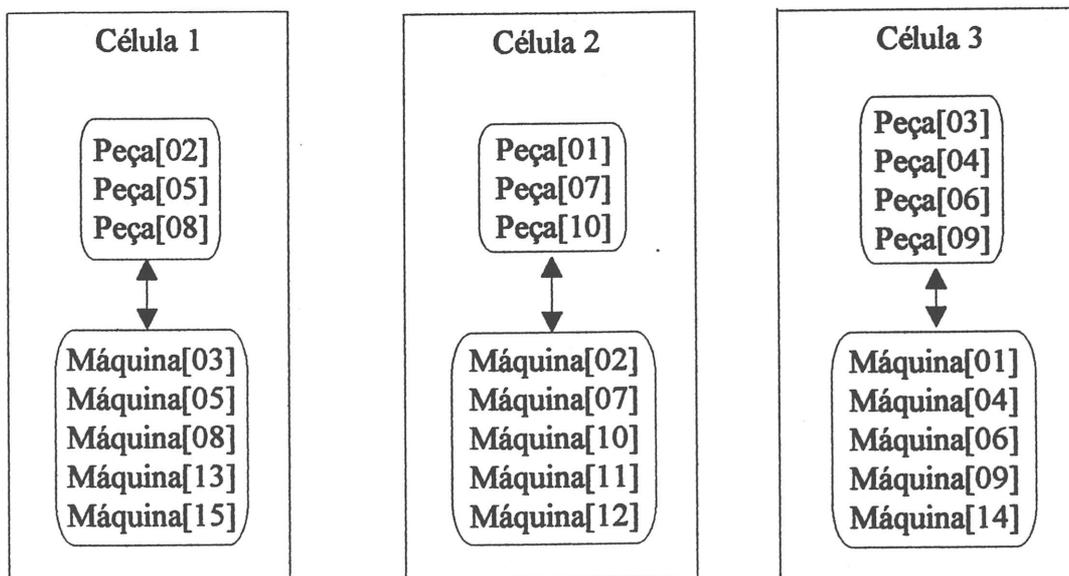
	3	5	8	13	15	2	7	10	11	12	1	4	6	9	14
2	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
5	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
8	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	1	0	1	1	1	0	0	0	0	0
7	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0
10	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1
4	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1
6	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
9	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

tabela 4.7

A tabela 4.6, mostra a contagem do número de operações executadas pelas máquinas 1,2,...,15 em cada uma das famílias formadas no processo de coloração.

A tabela 4.7 [peças x máquinas] apresenta as células de manufatura.

Com este resultado o sistema de produção está particionado em três células de manufatura, que são dadas abaixo:



A solução obtida não apresenta movimentos inter-células, ou seja, os agrupamentos são independentes. Esse resultado coincide com o resultado apresentado na literatura [07].

6 Aplicações

Efetuamos algumas aplicações do algoritmo proposto, com o objetivo de testar a sua eficiência, comparando com resultados encontrados na literatura.

1º) Kusiak [32] particiona um sistema de produção formado por cinco peças e quatro máquinas, em duas células de manufatura sem nenhum movimento inter-células. Esse sistema é fornecido na tabela 4.8.

Aplicando o algoritmo proposto no sistema acima, obtemos o mesmo resultado proposto pelo autor, que se encontra na tabela 4.9.

peças	unidades	roteiro		duração	
1	1	2	4	1	1
2	1	1	3	1	1
3	1	2	4	1	1
4	1	1	3	1	1
5	1	1		1	

tabela 4.8

	2	4	1	3
1	1	1	0	0
3	1	1	0	0
2	0	0	1	1
4	0	0	1	1
5	0	0	1	0

tabela 4.9

2º) Waghodekar and Sahu [48] decompõem um sistema de produção, constituído de sete peças e cinco máquinas, em duas células de manufatura, com dois movimentos inter-células. Esse sistema é fornecido na tabela 4.10:

peças	unidades	roteiro	duração
1	1	3 4	1 1
2	1	1 3	1 1
3	1	2 4 5	1 1 1
4	1	1 3 5	1 1 1
5	1	2 5	1 1
6	1	4 5	1 1
7	1	1 3	1 1

tabela 4.10

Utilizando o algoritmo proposto no sistema acima obtemos o mesmo resultado proposto pelo autor, que se encontra na tabela 4.11. Esta solução apresenta 2 movimentos inter-células.

	1	3	2	4	5
1	0	1	0	1	0
2	1	1	0	0	0
4	1	1	0	0	1
7	1	1	0	0	0
3	0	0	1	1	1
5	0	0	1	0	1
6	0	0	0	1	1

tabela 4.11

3º) Meguelati [37], resolve um exemplo constituído de nove peças e doze máquinas, particionando-o em duas células de manufatura. A tabela 4.12 fornece o roteiro de fabricação das peças desse sistema.

peças	unidades	roteiro	duração
1	1	1 2 6 8 11	1.0 2.0 2.0 3.0 1.0
2	1	1 2 4 6 8	2.0 3.0 2.0 1.0 2.0
3	1	1 4 6 8	3.0 1.0 4.0 1.0
4	1	3 7 9 10	0.8 2.0 3.0 3.0
5	1	2 4 8 11	2.0 2.0 3.0 1.0 3.0
6	1	1 2 4 8 11	2.0 2.0 3.0 1.0 3.0
7	1	5 10 12	1.0 2.0 3.0
8	1	3 5 7	3.3 1.0 2.0
9	1	1 3 5 9 12	2.0 1.7 5.0 2.0 1.0

tabela 4.12

A solução apresentada por Meguelati [37] possui quatro movimentos inter-células e duas células de dimensão 5 x 6 e 4 x 6. O algoritmo proposto forneceu uma solução com dois movimentos inter-células e duas células de mesma dimensão mostrada na tabela 4.13. Pode-se, portanto, concluir que o algoritmo proposto foi mais eficiente para resolver este exemplo que o algoritmo de [37].

	3	5	7	9	10	12	1	2	4	6	8	11
4	0.8	0.0	2.0	3.0	3.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	1.0	0.0	0.0	2.0	3.0	0.0	0.0	0.0	0.0	0.0	0.0
8	3.3	1.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	1.7	5.0	0.0	2.0	0.0	1.0	2.0	0.0	0.0	0.0	0.0	0.0
1	0.0	0.0	0.0	0.0	0.0	0.0	1.0	2.0	0.0	2.0	3.0	1.0
2	0.0	0.0	0.0	0.0	0.0	0.0	2.0	3.0	2.0	1.0	2.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	3.0	0.0	1.0	4.0	1.0	0.0
5	0.0	0.0	0.0	0.0	2.0	0.0	0.0	1.0	4.0	0.0	3.0	3.0
6	0.0	0.0	0.0	0.0	0.0	0.0	2.0	2.0	3.0	0.0	1.0	3.0

tabela 4.13

4^o) Harhalakis et al [25], decompõem um sistema de produção formado por vinte peças e vinte máquinas em cinco células de manufatura com quatorze movimentos inter-células. Esse sistema de produção é fornecido na tabela 4.16. O algoritmo proposto forneceu o mesmo resultado. Este resultado está na tabela 4.17.

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

tabela 4.16

	4 6 7 13 15	1 9 12 18	2 3 10 11	8 19 20	5 14 16 17
5	1 3 4 0 2	0 0 0 0	0 0 0 0	0 0 0	0 0 0 0
8	5 0 3 2 1	0 4 0 0	0 0 0 0	0 0 0	0 0 0 0
13	0 1 2 0 3	0 0 0 0	0 0 0 0	0 0 0	0 0 0 4
16	0 3 2 0 1	0 0 0 0	0 0 0 0	0 4 0	0 0 0 0
1	0 0 0 0 0	2 3 1 4	0 0 0 0	0 0 5	0 0 0 0
9	0 0 0 0 0	4 2 5 1	0 0 0 3	0 0 0	0 0 0 0
12	0 0 0 0 0	5 1 4 2	0 0 0 0	0 0 0	3 0 0 0
17	0 0 0 0 0	2 1 3 0	0 0 0 0	0 0 0	0 0 0 0
20	0 0 0 0 0	3 0 4 1	0 0 2 0	0 0 0	0 0 0 0
2	0 0 0 0 0	0 0 0 0	3 2 0 1	0 0 0	0 0 0 0
4	0 0 0 0 0	0 0 0 0	3 1 4 2	0 0 0	0 0 0 0
11	0 0 0 0 0	3 0 0 0	4 0 2 0	1 0 0	0 2 0 0
14	0 0 0 0 0	0 0 0 0	2 1 0 3	0 0 0	0 0 0 0
19	0 0 0 0 0	0 0 0 0	2 1 0 3	0 0 0	4 0 0 0
3	0 0 0 0 0	0 0 0 0	0 0 0 0	1 3 2	0 0 0 0
10	0 0 0 0 0	0 0 0 0	0 0 0 0	3 1 2	0 0 0 0
18	0 0 0 0 0	0 0 0 0	0 0 4 0	1 2 3	0 0 0 0
6	0 0 0 0 0	0 0 0 0	0 0 0 1	0 0 0	5 2 3 4
7	0 0 0 0 0	0 0 0 0	0 0 0 0	0 0 0	1 0 2 3
15	0 0 0 1 0	0 0 0 0	0 0 0 0	0 0 0	0 2 3 4

tabela 4.17

7 Comentários

O algoritmo para auxiliar no projeto de células de manufatura baseado na bicoloração de uma árvore de peso máximo estudado neste capítulo apresentou resultados de boa qualidade nos testes realizados com exemplos extraídos da literatura. O tempo computacional foi irrelevante (menos de um segundo) para todos os exemplos testados. O programa computacional correspondente ao algoritmo proposto foi escrito em Turbo-Pascal e está implantado em um micro-computador IBM-PC 386 - 40 MHZ.

Para o exemplo proposto em [37], foi obtida uma solução de melhor qualidade, com dois movimentos inter-células em lugar dos quatro movimentos originais. As duas células apresentam a mesma dimensão.

Conclusão

Nesta dissertação foram apresentados os principais conceitos da Tecnologia de Grupo e do Projeto de Células de Manufatura. Realizou-se um estudo do interesse que a aplicação desta técnica tem para aumentar a produtividade das fábricas que operam na base da produção em lotes pequenos e médios. Estas fábricas são responsáveis por 75% da produção mecânica em países industrializados como o Brasil e o objetivo de aumentar a eficiência das mesmas a partir do emprego de novas formas da organização é de grande relevância.

A revisão bibliográfica realizada permitiu o levantamento de diversos modelos matemáticos e algoritmos para a resolução do problema de particionamento de um sistema de produção em células de manufatura. Um grande número de métodos utilizam a computação de um coeficiente de similaridade entre as peças fabricadas e em seguida fazem uso de técnicas simples de rearranjo de linhas e colunas de uma matriz ou de técnicas mais complexas como a programação inteira, os sistemas especialistas e a teoria dos grafos.

Com o objetivo de desenvolver um algoritmo original para o projeto de células de manufatura baseado em teoria dos grafos, empreendemos um estudo aprofundado da mesma e, em particular, de conceito de coloração em grafos e das técnicas disponíveis para efetuar uma coloração de cardinalidade mínima. Em seguida, escolhemos dois dos métodos descritos na literatura [11], [12] para a organização de uma fábrica em células desenvolvido com base na coloração de nós de um grafo e realizamos um estudo detalhado dos algoritmos e das implementações propostas.

O resultado destes estudos e implementações levou-nos a propor um método original para a realização do projeto de células de manufatura com base no Teorema proposto por Guénoche [21]. Este método identifica iterativamente as famílias de produtos de diâmetro máximo e efetua uma bicoloração das peças pertencentes a estas até que o número desejado de células seja obtido. A obtenção das árvores de peso máximo e a bicoloração das mesmas é feita com o auxílio do algoritmo guloso de Prim [19] e o resultado obtido tem a otimalidade garantida, uma vez que este problema está inserido na classe dos matróides [19].

O programa computacional correspondente ao método proposto foi escrito em linguagem Turbo-Pascal e está implantado em um microcomputador IBM-PC 386-40 MHZ.

Os cinco testes realizados com exemplos extraídos da literatura mostraram a boa qualidade das soluções obtidas: no caso do exemplo proposto em [37], diminui-se 2 movimentos inter-células da solução existente, conservando-se a dimensão das células e para os outros exemplos testados obteve-se a solução original proposta pelos autores. O tempo computacional exigido para a resolução dos problemas foi irrelevante (inferior a um segundo) para todos os exemplos considerados.

O método desenvolvido, o programa computacional codificado e os testes realizados proporcionaram a submissão e aceitação deste trabalho para publicação em um congresso nacional [13].

Entre as tarefas a executar, como continuidade do trabalho apresentado nesta dissertação, destacam-se:

- realização de testes computacionais para exemplos reais de médio e grande porte com o objetivo de avaliar o tempo de cálculo, a utilização de memória e a qualidade das soluções obtidas;
- continuação do levantamento bibliográfico realizado, consultando-se outras áreas que também têm se dedicado ao estudo do projeto de células de manufatura, tais como: conjuntos nebulosos e algoritmos genéticos;
- implementação do método proposto nesta dissertação na forma de um sistema especialista, fazendo-se uso de conhecimentos específicos de difícil modelamento;
- implementação de um programa computacional com processamento em paralelo, capaz de resolver um dos modelos matemáticos de programação inteira estudados no capítulo 2 em um tempo de processamento aceitável.

Referências Bibliográficas

- [01] - AHO, A., HOPCROFT, J., ULLMAN, J., (1983), Data structures and algorithms, Addison-Wesley.
- [02] - AL-QATTAN, (1990), Designing flexible manufacturing cells using a branch and bound method, Int. J. Prod. Res., 28, 2, p. 325-336.
- [03] - ARTHANARI, T. S., DODGE, Y., (1981), Mathematical Programming in Statistics, New York, NY, Wiley.
- [04] - ASKIN, R., SUBRAMANIAN, S., (1987), A cost-based heuristic for Group Technology configuration, Int. J. Prod. Res., 25, 1, p.101-114.
- [05] - BERGE, C., (1970), Graphes, Gauthier-Villars.
- [06] - BURBIDGE, J. L., (1975), The introduction of Group Technology - John Willey.
- [07] - CHAN, H M., MILNER, (1981), Direct clustering algorithm for group formation in cellular manufacturing, J. Manuf. Syst. vol.1 ,nº 1, p. 65-74.
- [08] - CHANDRASEKHARAN, M. P., RAJAGOPALAN, R., (1987), ZODIAC: an algorithm for concurrent formation of part-families and machine cells, Int. J. Prod. Res., 25, nº 6, p. 835-850.
- [09] - FERREIRA RIBEIRO, J. F., (1991), Constitution d'Ilots et Ordonnancement dans un Atelier de Production, Tese de Doutorado LAAS/INSA, Toulouse, França.
- [10] - FERREIRA RIBEIRO, J. F., PRADIN, B., (1993), A methodology for cellular manufacturing design, Int. J. Prod. Res., 31,1, p. 235-250.
- [11] - FERREIRA RIBEIRO, J. F., RIBEIRO, C. M., (1993), Um algoritmo de coloração de grafos para o projeto de células de manufatura, XIV CILAMCE, São Paulo, p. 898-903.

- [12] - FERREIRA RIBEIRO, J. F., ALVES, P. M. R., (1994), Projeto de Células de Manufatura por meio de Ligações-Contrações CILAMCE, Belo Horizonte (MG), p. 1470-1476.
- [13] - FERREIRA R. C., FERREIRA RIBEIRO, J. F., (1995), Coloração em Grafos: uma aplicação em Tecnologia de Grupo, XV ENEGEP/IIICIE, p. 1142-1147.
- [14] - FERREIRA, M. S., RESENDE, M. O., (1995), Um exame à prática do controle de produção em células de manufatura, XV ENEGEP/IIICIE, p. 1579-1583.
- [15] - FU, K. S., (1980), Recent developments in pattern recognition, IEE Trans. Comput., vol. C-29, nº 10, p. 845-854.
- [16] - GAREY, M. R., JOHNSON, D. S., (1979), Computers and intractability: a guide to the theory of no-completeness, Freeman.
- [17] - GLOVER, F., WOOLSEY, E., (1974), Converting the 0-1 polynomial problem to a 0-1 linear programming, Oper. Res., vol. 22, p. 180-182.
- [18] - GOMORY, R. E., HÜ, T. C., (1961), Multi-terminal Network Flows, SIAM Journal of Applied Mathematics, 9, 4, p. 551-570.
- [19] - GONDRAN, M., MINOUX, M., (1985), Graphes et algorithmes, Eyrolles.
- [20] - GRAYSON, T. J., (1971), Group Technology - a brief international appraisal . In Peklenik, J., - Advances in manufacturing systems research and development. s.l., s.e., Paper nº5. p. 51-61.
- [21] - GUÉNOCHE, A., (1989), Partitions with minimum diameter, 2nd Conf., IFCS, Charlottesville, USA.
- [22] - GUNASINGH, K. R., LASHKARI, R. S., (1989), Machine grouping problem in cellular manufacturing systems - an integer programming approach, Int. J. Prod. Res., 27, 9, p. 1465-1473.

- [23] - GUPTA, T., SEIFODDINI, H., (1990), Production data based similarity coefficient for machine-component grouping decisions in the design of a cellular manufacturing system, *Int. J. Prod. Res.*, 28, 7, p. 1247-1269.
- [24] - HAM, I., HITOMI, K., YOSHIDA, T., (1985), *Group Technology*, Boston. MA, Kluwer- Nijhoff Publishing.
- [25] - HARHALAKIS, G., NAGI, R., PROTH, J. M., (1990), An efficient heuristic in manufacturing cell formation for Group Technology applications, *Int. J. Prod. Res.*, 28, 1, p. 185-198.
- [26] - HYER, N. L., and WEMMERLÖW, U., (1989), Group Technology in the US manufacturing industry: a survey of current practices, *Int. J. Prod. Res.*, 27,8, p. 1287-1304.
- [27] - JOHNSON, L, A., MONTGOMORY, D. C., (1974), *Operations Research in production planning, scheduling and inventory control*, John Wiley & Sons.
- [28] - KING, J. B., (1980), Machine-component grouping formation in Group Technology: review and extension, *Int. J. Prod.Res.*, 25,12, p. 1715-1728.
- [29] - KUSIAK, A., (1987), Artificial intelligence and operations research in flexible manufacturing system, *Information Syst. Operat. Res.*, (INFOR), vol. 25, p. 2-12.
- [30] - KUSIAK, A., (1987), The generalized Group Technology concept, *Int. J. Prod. Res.*, 25,4, p. 561-569.
- [31] - KUSIAK, A., (1988), EXGT-S: A knowledge based system for Group Technology, *Int. J. Prod. Res.*, 26, 5, p. 887-904.
- [32] - KUSIAK, A., CHOW, W. S., (1988), Decomposition of manufacturing systems, *IEEE Journal of Robotics and Automation*, 4, n° 5, p. 457-471.
- [33] - LASHKARI, R. S., DUTTA, S. P. NADOLI, G., (1987), Part family formation in flexible manufacturing systems- An integer programming approach, in A.

- Kusiak, Ed., *Modern Production Management Systems*, Amsterdam, The Netherlands: North Holland, p. 627-635.
- [34] - LEE, R. C. T., (1981), *Clustering analysis and its applications*, in J. T. Tou, Ed. *Advances in Information Systems Science*, New York, USA.
- [35] - LEE, J. L., VOGT., W. G. MICKLE., M. H., (1982), *Calculation of shortest paths by optimal decomposition*, *IEEE Trans. Syst. Man. Cybern.* Vol. SMC-12 p. 410-415.
- [36] - McCORMICK, W. T. , SCHWEITZER, P. J., WHITE, T. W., (1972), *Problem decomposition and data reorganization by cluster technique*, *Oper. Res.*, vol. 20, n° 20, p. 993-1009.
- [37] - MEGUELATI, S., (1988), *Méthodes de classification pour la constitution d'îlots de fabrication et l'ordonnancement*, Tese de Doutorado LAAS/INSA, Toulouse, France.
- [38] - MITROFANOV, S.P., (1959), *The scientific principles of Group Technology*, National Lending. Library London.
- [39] - ORLICKY, J., (1965), *Material Requirements Planning*, Librairie Lavoisier.
- [40] - RAJAGOPALAN, R., BATRA, J. L., (1975), *Design of cellular production systems - a graph theoretic approach*, *Int. J. Prod.Res.*, *Int. J. Prod. Res.*, 13, n° 6, p. 567-579.
- [41] - RONCONI, D. P., ARMENTANO, V. A., (1993), *Um método heurístico baseado em grafos para formação de células de manufatura*, *cadernos DEP-UFSCar*, 21, p. 24-28.
- [42] - SCHONBERGER, R. J., (1982), *Japanese manufacturing techniques*, The Free Press.
- [43] - SRINIVASAN, G., NARENDAN, T. T., MAHAVEDAN, B., (1990), *An assignment model for the part-families problem in Group Technology*, *Int. J. Prod. Res.*, 28, 1, p. 145-152.

- [44] - STABILE DE ARRUDA, P. E., VILA F. E. G. , (1994), Levantamento do estágio atual de implantação da Tecnologia de Grupo e Células de Manufatura do Estado de São Paulo, XV ENEGEP/IIICIE, p. 1559-1562.
- [45] - SZWARCFITER, J.L., (1983), Grafos e Algoritmos Computacionais, Campus.
- [46] - VANELLI, A. KUMAR, K. R., (1986), A method for finding minimal bottleneck cells for grouping part-machine families, Int. J. Prod. Res., vol. 24, n° 2, p. 387-400.
- [47] - VILA F. E. G. , (1982), Introdução à TG: um novo enfoque em sistemas de produção, Tese de Mestrado, USP-EESC-DEM, São Carlos (SP).
- [48] - WAGHODEKAR, P. H., SAHU, S., (1984), Machine-component cell formation in Group Technology: MACE, Int. J. Prod. Res., 22, 6, p. 937-948.
- [49] - WEI, J. C., GARY, M. K., (1989), Commonality analysis: a linear cell clustering algorithm for Group Technology, Int. J. Prod. Res., 27, n° 12, p. 2053-2062.
- [50] - WITTE, J., (1980), The use of similarity coefficients in production flow analysis, Int. J. Prod. Res., 18, 4, p. 503-514.