

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA ELÉTRICA
DEPARTAMENTO DE ENGENHARIA DE SISTEMAS

UM MÉTODO HEURÍSTICO BASEADO EM GRAFOS PARA FORMAÇÃO DE CÉLULAS DE MANUFATURA EM TECNOLOGIA DE GRUPO

Este exemplar corresponde à redação final da tese
defendida por Débora Pretti Ronconi

provada pela Comissão

Julgadora em 21 / 5 / 93

Vinicius A. Armentano
Orientador

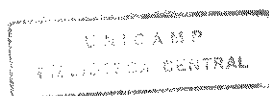
DÉBORA PRETTI RONCONI

Orientador:

Prof. Dr. Vinicius Amaral Armentano

Tese apresentada à Faculdade de Engenharia
Elétrica da Universidade Estadual de
Campinas - UNICAMP, como parte dos
requisitos exigidos para a obtenção do título
de MESTRE EM ENGENHARIA
ELÉTRICA.

- MAIO 1993 -



9658756

A todos aqueles que acreditam na importância
e no futuro da pesquisa no Brasil.

AGRADECIMENTOS

A todos que colaboraram na realização deste trabalho e em especial

- a Vinicius A. Armentano pela orientação dedicada e generosa;
- aos professores Aniltom S.Garcia, O.L. Agostinho e Tarcisio Faustini pelo incentivo inicial;
- a Silvana Pioli pelas intermináveis conversas sobre o papel de uma tese, e sua amizade constante;
- a Marília e Virginia pela convivência sensacional;
- a Regina Esther pelo apoio computacional e permanente interesse;
- a Zake Sabbag Neto pelas “super-dicas” na escrita desta tese e pela paciência para debater sobre produção;
- a Cintia, Franklina Maria e Lúcia pelas valiosas discussões sobre este trabalho;
- a Darli, Sandra, Rafael e Nishi pelo companheirismo;
- ao Conselho Nacional de Desenvolvimento Científico e Tecnológico pelo apoio financeiro;
- a Deus, pela coragem e determinação que me destes.

RESUMO

Tecnologia de Grupo (TG) é um conceito administrativo cujo objetivo é analisar e organizar peças e processos produtivos de acordo com similaridades de projeto e fabricação. O trabalho apresenta um algoritmo heurístico baseado na teoria de grafos, cujo objetivo é a formação de células de máquinas associadas a famílias de peças. O problema é representado por uma rede, que é seccionada através do algoritmo modificado de Gomory-Hu. Esta divisão em grupos objetiva formar células de manufatura com a menor interação possível. Este método é implementado computacionalmente e são realizadas comparações com algoritmos da literatura.

ABSTRACT

Group Technology (GT) is an administrative concept whose aim is to analyse and organize productive processes and parts according to the project and manufacturing similarities. This work presents a heuristic algorithm, based on the graph theory, whose purpose is to form machine cells associated with part-families. The problem is represented by a network, that is partitioned using a modified Gomory-Hu algorithm. This partition aims to form manufacturing cells with minimum intercellular interactions. The method is computationally implemented and comparisons are made with other algorithms from the literature.

ÍNDICE

Introdução.....	1
1. Tecnologia de Grupo	5
1.1. Introdução.....	5
1.2. Definições Básicas.....	8
1.3. Formação de Células.....	9
1.3.1. Sistemas de Classificação e Codificação.....	9
1.3.2. Análise de Fluxo de Produção.....	14
2. Estudo do Problema	20
2.1. Introdução.....	20
2.2. Descrição do Problema	20
2.3. Revisão Bibliográfica.....	22
2.4. Proposta do Trabalho	26
3. Descrição do Método.....	27
3.1. Introdução	27
3.2. Conceitos Preliminares	28
3.3. Algoritmo Principal.....	30

3.4. Algoritmo de Gomory-Hu Modificado	33
3.5. Exemplo de Aplicação do Algoritmo Principal	44
3.6. Complexidade do Algoritmo.....	48
3.7. Contra Exemplo	49
4. Apresentação e Análise dos Resultados	55
4.1. Introdução	55
4.2. Índices Avaliadores	56
4.2.1. Eficiência de Agrupamento	58
4.2.2. Eficácia de Agrupamento	59
4.3. Exemplos da Literatura	64
5. Comentários Finais.....	71
Apêndice I - Algoritmo MPM.....	73
Apêndice II - Estruturas de Dados da Implementação.....	89
Bibliografia	96

INTRODUÇÃO

Antes do novo ideário introduzido por Newcomen e Watt - inovações no processo produtivo - no fim do século XVIII e o advento da máquina a vapor, o próprio termo indústria referia-se apenas a qualidade de destreza e perícia. O modo de produção era basicamente artesanal e se dava em baixa escala, no caso de bens de consumo, em pequenas oficinas. Seus proprietários detinham o conhecimento do processo de fabricação e os meios de produção. Com as vendas recebiam a “paga” pelo seu trabalho (já que o lucro e a Teoria da Mais Valia só viriam a ser formuladas por Marx alguns anos mais tarde).

O crescimento populacional e a expansão do colonialismo europeu no mundo fez o mercado consumidor crescer muito e a demanda por manufaturados aumentar. O equilíbrio da “oferta e procura” só foi conseguido com o aprimoramento da tecnologia e dos equipamentos industriais, fazendo surgir o novo e poderoso ramo da atividade de manufatura.

Devido à grande oferta de mão-de-obra, ao baixo custo de produção e à pequena concorrência, não havia uma forte necessidade de se aumentar a eficiência dos sistemas produtivos. Com efeito, a procura pela otimização do rendimento do trabalho operário e do parque fabril em si, só teve início com as idéias do engenheiro Frederick Winslow Taylor, o precursor da chamada “Escola de Administração Científica”. Seu famoso estudo na Midvale Steel Co., um azeiteira dos Estados Unidos, fazia uma análise profunda de tempos e métodos de fabricação. Através de um rigoroso automatismo, implantou o que considerava os tempos e métodos “ótimos” de produção, extraindo um considerável aumento na produtividade da fábrica.

A partir daí, durante a primeira metade deste século, o comportamento da atividade industrial assistiu a grandes mudanças e aprimoramentos. As duas Grandes Guerras tiveram papel decisivo na aceleração do desenvolvimento de novas tecnologias e de novos métodos de produção. Logo após o fim da 1ª. Guerra Mundial, os países vencedores e principalmente os Estados Unidos, viveram um período de fausto e

prosperidade impulsionados pela vitalidade da economia, no afã de satisfazer as demandas crescentes dos mercados. Isto até o grande “crack” da Bolsa de Nova York em 1929, que desnudou a precariedade do sistema comercial mundial e a fragilidade camuflada em que se encontravam as finanças do mundo, em especial da Alemanha, países do Leste Europeu e Turquia, combalidos pelas pesadas dívidas da guerra. Com este cenário de fundo, foi fácil a subida ao poder de regimes autoritários de governo, os mais famosos na Alemanha e Itália. (Franco e Chacon 1980)

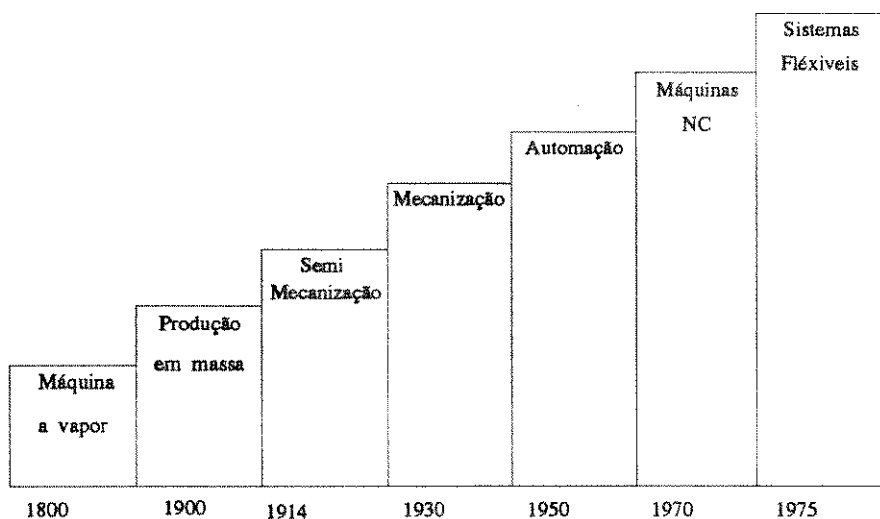
A detonação da 2a. Grande Guerra, mergulhou os países participantes numa corrida armamentista, impulsionando muito os setores de manufatura e siderurgia. Com o Plano Marshal e a reconstrução da Europa, o aprendizado de guerra foi fundamental na retomada do crescimento mundial. É nesta época que surgem as linhas de transferência na indústria, especializadas na produção em série e caracterizadas pela automação rígida (mecanização de processos físicos através de equipamentos que realizam operações repetitivas).

A crescente preocupação com o aumento da produtividade, faz surgir entre as décadas de 50-60 as primeiras máquinas de controle numérico (NC), que são o primeiro passo em direção aos atuais Sistemas Flexíveis de Manufatura (sistemas totalmente automatizados). Estes são ideais para a configuração de produção de hoje, fundamentada em lotes pequenos e médios. Buscam alta produtividade aliada à flexibilidade, que é a possibilidade de atender à variações na demanda e/ou no produto com rapidez e qualidade a um custo razoável.

Simultaneamente com esta atenção ao processo foram desenvolvidas algumas tecnologias de produto, tais como:

- a. Tecnologia de Grupo: técnica de agrupamento de máquinas e peças por semelhança de projeto e/ou fabricação.
- b. Técnicas Estatísticas e de Controle de Qualidade: métodos de avaliação e aperfeiçoamento do nível de qualidade da produção. Ex.: Controle de Qualidade Total (TQC), Análise de Experimentos, etc..
- c. Práticas de Redução de Estoques: técnicas para controle da produção. Ex.: “Kanban”, “Just-in-Time”.

Através da figura a seguir observa-se a evolução dos processos produtivos. (Agostinho 1991)



Desenvolvimento da Tecnologia de Produção

Dentro das premissas acima expostas, o trabalho visa auxiliar a implantação da técnica de produção - Tecnologia de Grupo. Esta técnica é estratégia de manufatura cujo objetivo é analisar e organizar peças e processos produtivos de acordo com similaridades de projeto e fabricação. Assim, formam-se grupos e famílias que podem ser utilizados para racionalizar processos de produção de pequenos e médios lotes, ou seja, pode se aplicar estratégias normalmente usadas para alta produção ou produção em massa para produção em média e pequena série (Sério 1990).

O trabalho apresenta um algoritmo heurístico cuja proposta é a formação de grupos de peças e máquinas. O problema é representado por uma rede de nós, que será seccionada através do algoritmo modificado de Gomory-Hu. Esta divisão em grupos objetiva formar células de manufatura com a menor interação possível.

Através de cinco capítulos, descreve-se a pesquisa realizada. No primeiro capítulo, apresenta-se um estudo resumido sobre Tecnologia de Grupo e suas formas clássicas de implantação. No segundo, o problema que será focado é descrito detalhadamente. No capítulo três, analisa-se o algoritmo heurístico para resolução do problema proposto. No

capítulo quatro são apresentados os resultados da implementação do algoritmo, e os índices de avaliação do trabalho. No quinto capítulo é realizada uma análise do trabalho, e são apresentadas propostas para pesquisas futuras.

Capítulo 1

TECNOLOGIA DE GRUPO

1.1. Introdução

Apesar de acreditar que se vive numa era de produção em massa, é interessante constatar que a maioria das indústrias, nos chamados países industrializados, trabalham com lotes pequenos e médios. Em 1965, uma pesquisa realizada nos Estados Unidos constatou que 75% de toda produção mecânica daquele país constava de lotes com menos de 50 peças.

Outro fato a ser relatado, é que nessas indústrias somente 5% do tempo total de produção são realmente gastos na máquina-ferramenta, enquanto que os outros 95% são gastos com movimentação de material e espera na máquinas. Entende-se como tempo total o tempo gasto desde a emissão da ordem de fabricação do componente até a sua entrega no armazém de produtos acabados.(Sério 1990)

Isto ocorre devido a maioria das indústrias mecânicas, que fabricam componentes em lotes pequenos e médios, utilizarem um sistema produtivo que praticamente não mudou desde do início do século XX. Este sistema de organização se justificava devido à composição anterior do mercado consumidor, essencialmente carente de produtos variados feitos em pouca quantidade. A figura 1.1 ilustra as características de fabricação das indústrias mecânicas.

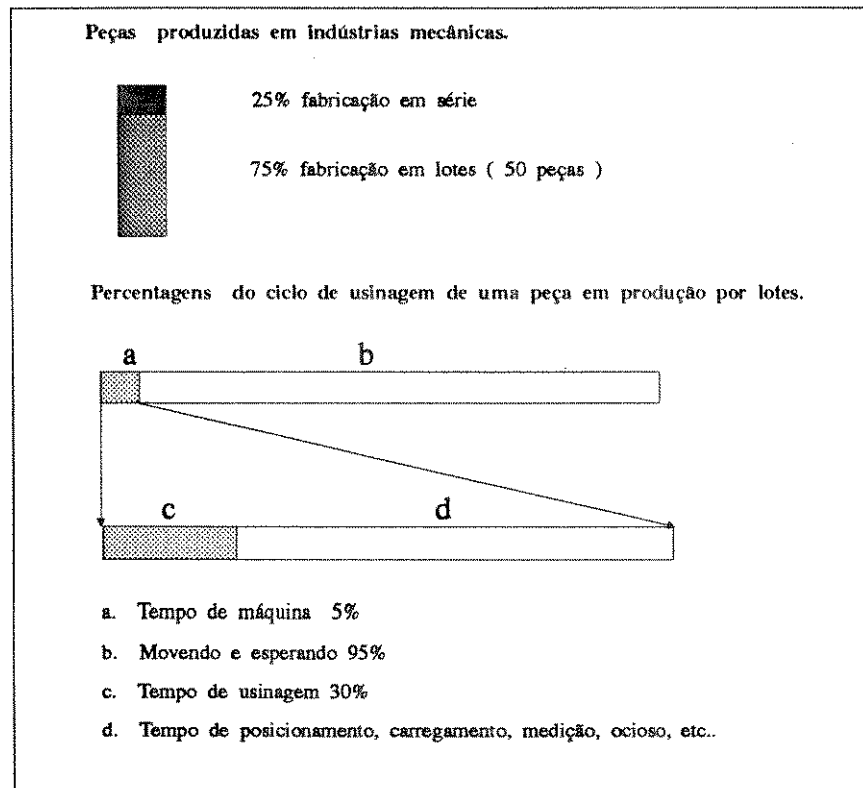


Figura 1.1. Características de fabricação

A evolução do mercado, com características substancialmente diferentes, provoca incompatibilidade com as estruturas atuais das empresas, dimensionadas para atender um perfil desatualizado de mercado consumidor. Entre as exigências do atual mercado tem-se:

- qualidade;
- maior diversificação de produtos;
- rapidez na produção.

Porém, a adaptação de uma estrutura convencional para uma estrutura mais flexível não é imediata. Na produção em massa, a utilização da automatização rígida, de processos de fabricação em linhas de fluxo, e o uso de máquinas especialistas, obtém resultados extremamente produtivos, porém para produção em lotes de pequenas e médias quantidades, são inaplicáveis. Os principais problemas de fabricação em lotes pequenos e médios, numa estrutura inadequada são:

- Em geral, a disposição das máquinas na fábrica atende a um arranjo físico funcional, isto é, todos os tornos são reunidos num setor de tornearia; todas as fresas num setor de fresagem e assim por diante. Uma peça típica deve

ser usinada em diversos setores de fabricação. Como normalmente não há uma organização sistemática na sequência de fabricação, peças muito diferentes são processadas em série. Logo, ocorrem grandes perdas de tempo na preparação das máquinas, dispositivos e ferramentas para cada peça.

- É difícil ter uma visão geral do desenvolvimento do trabalho. Peças que exigem muitas operações acabam se perdendo no sistema.
- Outra limitação é a flexibilidade do sistema. É quase impossível efetivar a execução acelerada de uma peça urgente, pois em geral o tempo de produção (“leadtime”) é muito grande.
- O estoque em processo é alto e portanto o investimento em capital de giro é alto.
- O Departamento de Projetos contém arquivos que podem conter dezenas de desenhos. Logo, quando aparece uma nova encomenda, pode ser mais fácil preparar um novo desenho, do que tentar localizar desenho de peça igual ou similar. Esta dificuldade de encontrar e recuperar dados tende a produzir um grande volume de duplicações, com perda de tempo e trabalho. A agilidade do setor de projetos é uma arma fundamental para a implantação eficiente da diversificação de produtos.

Devido a isso, a tendência atual é de aplicação de modernas técnicas administrativas no sentido de aumentar a produtividade e a flexibilidade, e conseqüentemente a competitividade (Moura 1987).

Tecnologia de Grupo (TG) ou Produção por Família de Peças é um conceito que se aplica na produção de pequenos e médios lotes. É baseado no agrupamento de peças em famílias, de acordo com certos atributos, e então se desenvolve o planejamento de todo o sistema produtivo para cada família.

O principal objetivo de sua aplicação é a utilização de métodos de alta produção para produção de baixa e média série. Utilizando estas técnicas, Tecnologia de Grupo conseguiu benefícios significativos, tais como: redução dos ciclos de fabricação, redução de material em processo, confiabilidade nos prazos de entrega, simplificação do planejamento e controle da produção, racionalização do projeto de produto e processos.

Via de regra a implantação da Tecnologia de Grupo é feita por etapas, de forma gradual, sendo esta também a forma mais prática dentro de um plano global (Sério

1990). E Burbidge (1975) mostra que o essencial para o sucesso da implantação da TG consiste apenas em:

- mudar para o “lay-out” em grupo;
- mudar o controle da produção.

Neste capítulo apresentam-se os conceitos gerais que envolvem a implantação de Tecnologia de Grupo .

1.2. Definições Básicas

Citam-se a seguir algumas definições importantes que aparecerão ao longo deste trabalho. (Sério 1990, Agostinho 1991)

a. Grupo: é a combinação do conjunto de operários, máquinas e/ou outras facilidades, arranjadas numa área reservada e projetada para completar um conjunto específico de peças. Os operários num grupo compartilham uma série comum de objetivos e metas de produção, em termos de listas de peças a serem completados em certa data. O número de operários num grupo deve ser o menor possível e adequado para obter coesão social.

b. Família: é um conjunto de peças, operações ou produtos com características similares produzidos por um grupo.

c. Célula: é um sistema de fabricação onde homens e máquinas estão arranjados em grupo de modo que todas as operações necessárias para a fabricação de uma família são executadas dentro da área delimitada pelo grupo.

d. Codificação: é um sistema para representar através de um conjunto de símbolos, as características básicas de uma peça, de modo a esta poder ser classificada.

e. Classificação: é o arranjo das peças dentro de conjuntos de acordo com algum princípio pelo qual são agrupadas.

1.3. Formação de Células

Para compor ou simplesmente conseguir formar as células de fabricação, faz-se necessária a identificação e formação de famílias de peças. Estas peças, no mínimo, devem ter requisitos de fabricação semelhantes. Além disso, deve-se considerar o número de peças de cada tipo e sua frequência de fabricação. Quanto maior a similaridade de requisitos de fabricação e a frequência de produção do lote, mais efetiva é a formação da família de peças para aplicações práticas de Tecnologia de Grupo, e na programação do sequenciamento e carregamento das máquinas.

O agrupamento de peças dentro de famílias é a chave da implantação da Tecnologia de Grupo. O problema que logo se apresenta é como as peças podem ser eficientemente agrupadas dentro dessas famílias. Existem três métodos clássicos para formação de famílias:

- Método Visual: Pessoa(s) com larga experiência sobre os componentes e os processos de fabricação dos mesmos, faz a formação de famílias e células de maneira subjetiva;
- Sistemas de Classificação e Codificação;
- Análise de Fluxo da Produção.

As duas últimas são técnicas sistematizadas e, a seguir são analisadas.

1.3.1. Sistemas de Classificação e Codificação

Nas empresas onde ocorre grande quantidade de peças, é conveniente codificar cada componente, de modo que seja possível identificar peças semelhantes pelas similaridades de seus códigos.

Um Sistema de Classificação e Codificação proporciona um meio efetivo para seleção de peças, baseado em parâmetros específicos do sistema, independente da origem

ou uso das peças. Um exemplo de uma peça codificada usando um Sistema de Classificação e Codificação é mostrado na figura 1.2 (Sério 1990).

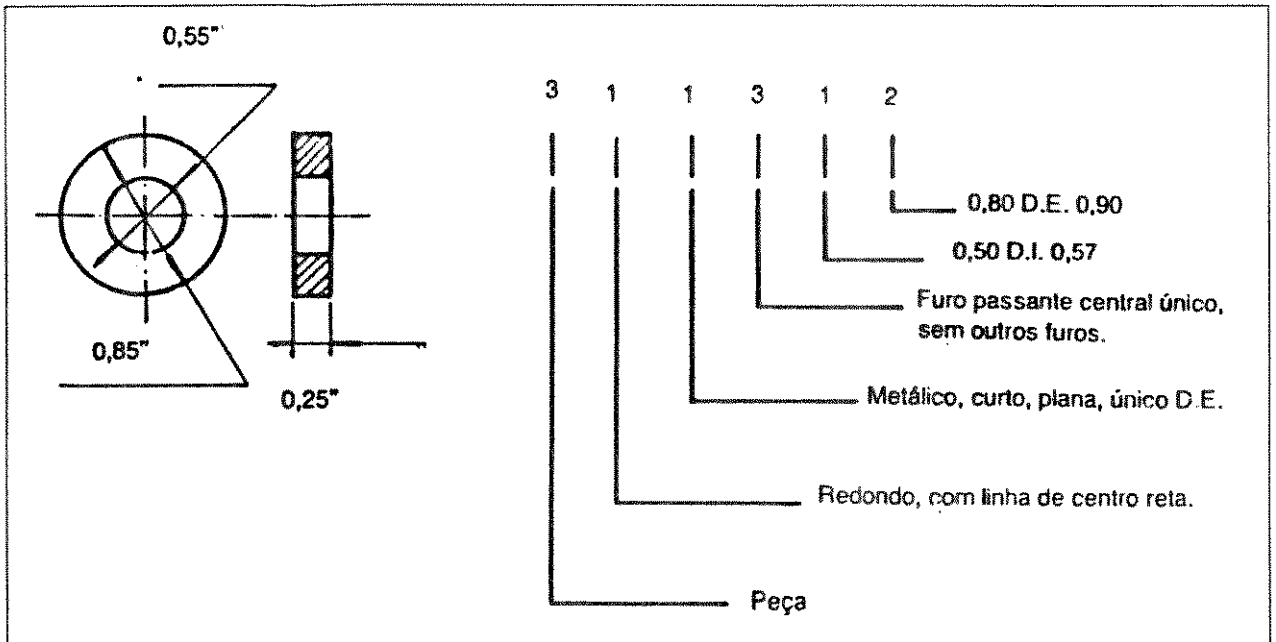


Figura 1.2. Exemplo de codificação - 6 dígitos monocódigo

A codificação e a classificação são problemas complexos. Embora muitos sistemas tenham sido desenvolvidos e melhorados, ainda não existe um sistema universalmente aceito. Uma vez que cada indústria possui condições e necessidades específicas, é importante pesquisar um sistema que possa ser adequado para os requerimentos específicos da indústria.

■ Codificação

A codificação dentro do processo de racionalização da produção, tem um papel organizador da informação que é necessário tanto para a planificação da produção como para a formação do arranjo físico da planta.

A codificação atinge as peças fabricadas, as operações requeridas, e o equipamento envolvido. Com isso forma-se um banco de dados para ser utilizado num programa de avaliação que tem como saída parâmetros para a administração da fábrica.

Existem três estruturas básicas de códigos para aplicações na Tecnologia de Grupo:

- a. Estrutura hierárquica (monocódigo)
- b. Estrutura do tipo dígito fixo (policódigo)
- c. Estrutura combinada (multicódigo)

Um código hierárquico é construído como um diagrama em árvore, onde cada dígito amplia a informação do dígito anterior, tornando assim cada dígito dependente do anterior. Este sistema proporciona uma análise extensiva das peças, dado que uma estrutura de código compacta pode conter uma enorme quantidade de informações. O tipo de dígito fixo, tem uma estrutura de código na qual cada posição de um dado dígito, representa uma informação independente e não diretamente relacionada à informação dada pelos outros dígitos. Para construir uma estrutura completa do tipo dígito fixo é necessário especificar todos os itens no código. Dessa forma, este tipo de sistema de codificação usualmente requer um grande número de dígitos. Na prática, a maioria dos sistemas empregam códigos híbridos, que são combinações de ambos os tipos, hierárquicos e de dígito fixo. Na seleção de um sistema de codificação é importante a manutenção de um balanceamento entre a quantidade de informação necessária e o número de dígitos requeridos para proporcionar esta informação. Cada dígito adicional aumenta o problema de manipulação, todavia necessita-se de um número suficiente de dígitos para codificar todas as características básicas da peça.

A utilização do código na formação de famílias, requer uma estratégia, que depende exclusivamente do sistema de produção tratado. Um estudo minucioso deve ser feito, na procura das características a serem consideradas para a classificação das peças. Por exemplo, se a codificação e a classificação forem orientadas para o projeto, as peças com formas semelhantes serão agrupadas; se forem orientadas pela utilização de máquinas, peças com roteiro de fabricação semelhantes serão agrupadas e assim por diante. Na maioria dos casos, cada sistema emprega uma combinação dessas características, o importante é que o sistema se adapte para satisfazer necessidades específicas da indústria.

■ Requisitos Básicos

Um Sistema de Classificação e Codificação bem projetado para a implantação de Tecnologia de Grupo, deve satisfazer vários requisitos básicos. O que se segue são os principais requisitos que este sistema deve possuir. (Agostinho 1991)

- **Abrangente:** o sistema deve abranger todos os itens existentes na produção e/ou comprados, e ser capaz de aceitar novos itens.
- **Mutuamente Exclusivo:** o sistema deve ter a capacidade de agrupar itens semelhantes enquanto exclui itens diferentes.
- **Baseado em Características Permanentes:** o sistema deve ser baseado em características visíveis permanentes e facilmente identificadas.
- **Específico Para as Necessidades do Usuário:** o sistema deve ser desenvolvido de forma a se adequar ao usuário e não o usuário a ele.
- **Adaptável à Futuras Mudanças:** o sistema deve ser adaptável às futuras expansões e mudanças tecnológicas.
- **Adaptável ao Processamento com Computador:** um Sistema de Classificação e Codificação. Pode ser funcional sem o uso de um computador, porém, é frequentemente desejável operar o sistema de forma informatizada, assim ele deve ser adaptável ao processamento computadorizado.
- **Aplicável em Toda a Empresa:** o sistema deve ser aplicável em todos os departamentos da empresa, incluindo projeto, planejamento e controle da produção, fabricação e administração

■ Características do Sistema de Classificação e Codificação

Após o estudo do Sistema de Classificação e Codificação para a formação de famílias, torna-se importante analisar os aspectos práticos que envolvem esta técnica. A tabela a seguir cita as principais vantagens e desvantagens do sistema (Sério 1990).

VANTAGENS	DESvantagens
1. Padronização e recorrência de elementos de forma, família de peças e produtos.	1. Complexidade.
2. Especificação de equipamentos.	2. Grande número de dígitos.
3. Identificar elementos de forma e família de peças e de processos.	3. Necessário manipulação automatizada.
4. Agrupar similaridades de forma e processo.	4. Altos custos no desenvolvimento e/ou compra do sistema.
5. Possibilitar padronização e recorrência de processos de fabricação.	5. Necessita de treinamento de operadores.
6. Auxiliar no projeto de dispositivos e ferramental.	6. Necessidade de desenvolver códigos para cada indústria. Dificil utilização de códigos universais.
7. Possibilitar desenvolvimento de codificação integrada de forma (projeto), processo e equipamento.	

Tabela 1.1. Características de Aplicação dos Sistemas de Classificação e Codificação

Um Sistema de Classificação e Codificação é essencial para a exploração completa dos benefícios da Tecnologia de Grupo. Por outro lado, exige altos investimentos e o seu uso na formação de famílias e células não garante que, peças que usam as mesmas máquinas se agrupem na célula que produzirá esta família de peças (Burbidge 1975).

Deve-se compreender então, que a instalação desse sistema é apenas o começo, simplesmente um pré-requisito para as aplicações de Tecnologia de Grupo.

1.3.2. Análise de Fluxo de Produção

A Análise de Fluxo da Produção é uma técnica para analisar a seqüência de operações e o percurso da peça, através das máquinas e estações de trabalho dentro da fábrica. As peças com rotas e operações comuns são agrupadas e identificadas como uma família. Similarmente, as máquinas e estações de trabalho usadas para produzir as famílias de peças podem ser agrupadas para formar grupos de máquinas. Um exemplo de formação de famílias através deste método é mostrado na figura 1.3.

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

Figura 1.3.a. Matriz máquinas x peças antes do agrupamento

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

Figura 1.3.b. Matriz máquinas x peças após agrupamento

As únicas informações necessárias para a Análise de Fluxo da Produção estão contidas em:

a. Roteiros de Fabricação: fichas que registram seqüencialmente os processos de fabricação das peças. Devem ser únicas para cada peça, conter os tempos de operação em cada equipamento e principalmente devem ser confiáveis.

b. Lista de Equipamentos: contém as máquinas disponíveis (em uso) na fábrica. Deve ser o mais abrangente possível e de preferência codificada, de modo, que se dois equipamentos tiverem o mesmo código, qualquer trabalho alocado num deles deve poder ser feito igualmente no outro.

Através de uma análise progressiva das informações citadas acima, pode se encontrar grupos de máquinas e peças utilizando somente os recursos já existentes, como instalações, equipamentos e ferramentas.

Na prática, existem algumas desvantagens na aplicação desta técnica, devido à confiabilidade dos dados de produção e de processos existentes. Outra desvantagem refere-se à dificuldade de atualização dos agrupamentos quando se introduzem peças no sistema.

A vantagem principal consiste na sua aplicação imediata, permitindo que se obtenha resultados a curto prazo, gerando maior confiabilidade no programa geral de Tecnologia de Grupo.

■ Fases na Implantação da Análise de Fluxo da Produção

1º Estágio: Análise do Fluxo entre Departamentos

O principal objetivo desta análise é, através da seqüência de processos de cada peça, determinar a melhor divisão e organização dos departamentos da indústria, visando encontrar um sistema de fluxo de material simples e eficiente.

Os departamentos são unidades de processamento que realizam operações com características básicas diferentes, sendo que estes possuem sua própria lista de equipamentos. Entende-se como departamento: funilaria, corte, usinagem, forjaria, solda, montagem, etc..

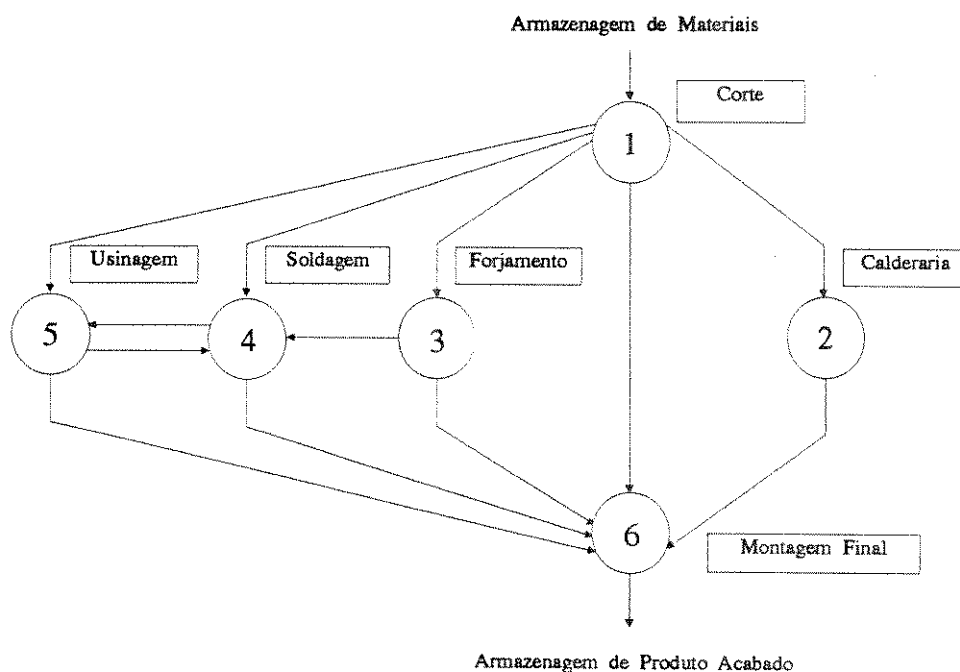


Figura 1.4. Fluxograma Básico Simplificado (Sério 1990).

Nesta análise, busca-se, dentro do possível, que cada peça seja fabricada dentro de um único departamento, que um tipo de equipamento exista somente em um local de processamento, que departamentos recebam materiais de um mínimo possível de fontes

e que os envie para o menor número possível de locais. Nem sempre se alcança os parâmetros definidos acima, porém estes servem como guia na busca do objetivo principal.

A figura 1.4 mostra um fluxograma básico com o roteiro de peças já simplificado. A simplificação do fluxo de fabricação é o primeiro nível do planejamento para a aplicação da Análise do Fluxo de Produção.

2º Estágio: Análise de Grupos

A Análise de Grupos consiste em dividir as peças alocadas em cada departamento (unidades de processamento) em grupos, de modo que cada família de peças seja totalmente processada dentro de um grupo. A forma mais usual de se realizar esta divisão é através de uma matriz de peças e máquinas como ilustra a figura 1.3.a. Esta matriz é manipulada até se obter células com a mínima interação possível (figura 1.3.b).

3º Estágio: Análise de Linha

O objetivo deste estágio é achar uma seqüência para as máquinas, que dará a solução mais próxima do arranjo em linha, dentro dos grupos.

Esta seqüência é obtida através da análise do roteiro das peças do grupo. Mesmo quando algumas peças utilizam máquinas numa seqüência diferente da maioria, ainda é possível simplificar a rota. A figura 1.7 ilustra um roteiro de produção otimizado. O objetivo é encontrar o arranjo físico mais simples de ser usado.

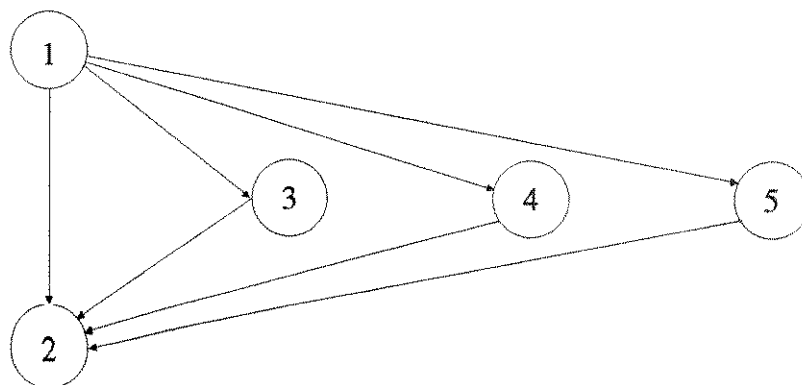


Figura 1.6 Análise de Linha na Célula

4º Estágio: Análise de Ferramental

A Análise de Ferramental consiste na última fase da Análise de Fluxo de Produção. Seu objetivo é encontrar, dentro das famílias, peças que utilizem o mesmo ferramental para serem fabricadas. Nesta etapa lança-se mão, novamente, das propriedades das matrizes para formação de famílias de ferramental (figura 1.6). Em seguida, deve-se procurar a seqüência otimizada para programá-la nas máquinas.

	1	2	3	4	5	6
1			x		x	
2		x	x			
3	x			x	x	
4		x	x		x	
5	x	x		x		x

Figura 1.6.a. Matriz ferramentas x peças antes do agrupamento

	1	4	6	3	2	5
5	x	x	x		x	
3	x	x				x
4				x	x	x
2				x	x	
1				x		x

Figura 1.6.b. Matriz ferramentas x peças após agrupamento

■ Características da Análise de Fluxo de Produção

Apresenta-se a seguir as principais vantagens e desvantagens do Sistema de Análise de Fluxo de Produção. (Sério 1990)

VANTAGENS	DESVANTAGENS
<ol style="list-style-type: none"> 1. Pouca exigência para implantação. 2. Serve de Bancos de Dados Tecnológicos. 3. Possibilita melhoria e padronização de fluxo. 4. Não exige altos custos para implantação. Apenas roteiros de fabricação e listas de equipamentos confiáveis. 5. Possibilita implantação imediata. 6. Simples para utilização, manuseio e interpretação dos usuários. 	<ol style="list-style-type: none"> 1. Limitado na geração da Tecnologia de Projeto. 2. Não favorece a integração entre o Projeto e a Fabricação. 3. Dificil aplicação e atualização para empresas com grande número de peças.

Tabela 1.1. Características de Aplicação da Análise de Fluxo de Produção

Nesta fase do estudo das técnicas clássicas de implantação da técnica de produção - Tecnologia de Grupo, obtém-se algumas conclusões:

A Análise de Fluxo de Produção é recomendada para:

- Obter resultados a curto prazo;
- Primeira fase de organização, padronização, geração de banco de dados para posterior implantação de um sistema de codificação, se este for necessário, desejado ou justificado.

Os Sistemas de Classificação e Codificação são recomendados para:

- Obter resultados a médio e a longo prazo (acima de 2 a 5 anos);
- Grande número de peças e diversificação de produtos;
- Desenvolver e padronizar banco de dados;
- Gerar estruturas integradas para projeto e fabricação visando futuras aplicações de tecnologias de maior nível de automação.

No capítulo seguinte, analisam-se formas de auxiliar a implantação de Tecnologia de Grupo através de métodos matemáticos.

Capítulo 2

ESTUDO DO PROBLEMA

2.1. Introdução

Neste capítulo, o problema da formação de células de manufatura, específicas para aplicação do conceito de Tecnologia de Grupo (TG), é detalhado. Este é descrito em função dos objetivos básicos de sua resolução e de sua complexidade. Uma revisão bibliográfica resumida é apresentada.

2.2. Descrição do Problema

Durante as últimas duas décadas, TG tem surgido como um importante princípio científico para o aumento da produtividade dos sistemas de manufatura. A aplicação desde conceito está fortemente ligado à formação de famílias de peças e correspondentes grupos de máquinas, ou seja, formação de células (Seifoddini e Wolfe 1986).

Cada célula, com diferentes máquinas, possui a capacidade de processar uma ou mais famílias de peças. Esta associação de famílias de peças e grupos de máquinas deve satisfazer os seguintes objetivos (Gupta e Seifoddini 1990):

- Maximizar a utilização das máquinas dentro das células.
- Minimizar a duplicação de máquinas em diferentes células.
- Minimizar o número de peças excepcionais (peças que são processadas em mais de uma célula).
- Minimizar o custo total de manuseio de material devido a movimentos intercelulares.

O problema de formação de células pode ser abordado utilizando a teoria de grafos, ou seja, representando os requerimentos de processamento das peças através de um grafo bipartido $G(V_m, V_p, E)$ onde V_m e V_p são os conjuntos de nós do grafo, que correspondem, respectivamente, ao conjunto de máquinas e peças (figuras 2.1.a e 2.1.b). E é o conjunto de arestas do grafo tal que:

(i) se existe uma aresta entre a máquina do nó i e a peça do nó j , a peça j requer processamento da máquina i .

(ii) se não existe uma aresta entre a máquina do nó i e a peça do nó j , a peça j não requer processamento da máquina i . (King e Nakornchai 1982)

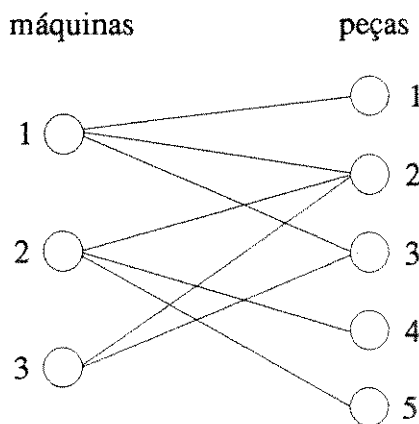


Figura 2.1.a. Grafo bipartido conexo

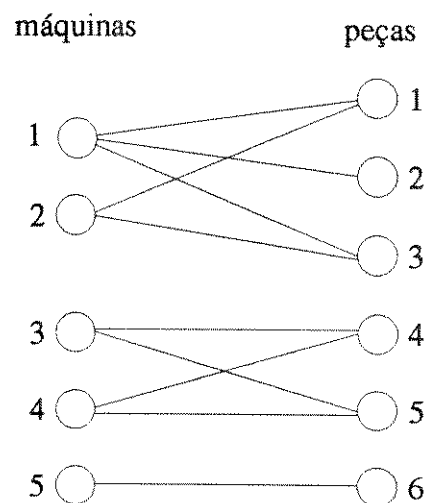


Figura 2.1.b. Grafo bipartido desconexo

Logo, o problema de formação de m células é equivalente ao problema de particionar um grafo $G(V, E)$ em m subgrafos, de tal modo que cada subgrafo possua no máximo k nós e o somatório dos pesos das arestas que conectam subgrafos diferentes seja mínimo. Este problema é reconhecido como NP-completo (Garey e Johnson 1979 - Particionamento de Grafos - pg. 209). Porém, considerando que este grafo possua grupos

naturais (subgrafos desconexos - figura 2.1.b), estes podem ser obtidos através de algoritmos polinomiais de complexidade $O(V + E)$ (Baase 1989).

2.3. Revisão Bibliográfica

Devido à dificuldade de resolução do problema, vários métodos foram desenvolvidos no sentido de alcançar os objetivos básicos de TG de forma eficiente. Boas revisões bibliográficas podem ser encontradas em King e Nakornchai (1982) e Waghodekar e Sahu (1984).

Burbidge (1975) descreve o uso da Análise de Fluxo de Produção (AFP) para a formação de células para TG. Esta técnica se concentra na análise da seqüência de operações para cada peça. A parte central da AFP é a matriz de máquinas-peças, com dados "0" ou "1", indicando que máquina é necessária para produção de cada peça, tal que:

$$\left\{ \begin{array}{l} a_{ij} = 1 \text{ ,se a máquina } i \text{ é necessária na produção da peça } j; \\ a_{ij} = 0 \text{ ,se a máquina } i \text{ não é necessária na produção da peça } j; \end{array} \right.$$

O objetivo do método é a diagonalização da matriz para obtenção de famílias de peças e grupos de máquinas. Esta técnica é suficientemente precisa para a indústria rearranjar seu chão-de-fábrica em células independentes. Baseado neste conceito diversos algoritmos foram desenvolvidos (King 1980, King e Nakornchai 1982, Vannelli e Kumar 1986).

Dentre as linhas de pesquisa desta área, o Agrupamento Baseado em Vetores caracteriza-se pelo rearranjo seqüencial das colunas e linhas da matriz máquinas-peças, de acordo com um critério, até que se obtenha a diagonalização da matriz. McCormick *et al.* (1972) desenvolveram uma técnica de agrupamento nesta linha, denominada Análise de Vínculo de Energia ("Bond Energy Analysis"). BEA busca a formação de blocos diagonais através da maximização do vínculo de energia entre linhas e colunas adjacentes da matriz. Vínculo de energia entre a linha x e a linha y é a soma dos produtos de

cada elemento da linha x com cada elemento da linha y. Este método inicia com a seleção arbitrária de uma linha (coluna), e esta é realocada próxima a linha (coluna) de maior contribuição de energia. Este processo é repetido para todas as linhas e colunas. BEA é aplicável para todas as dimensões de problema, porém sua solução depende da escolha da linha ou coluna inicial. King (1980) propôs o método Agrupamento por Ordem de Posição (“Rank Order Clustering”). ROC é uma técnica conhecida de agrupamento que objetiva a criação de blocos diagonais através de repetidas realocações das linhas e colunas da matriz de acordo com os valores binários das mesmas. O algoritmo inicialmente ordena as linhas em ordem decrescente e posteriormente as colunas. ROC é fácil de aplicar e tem como vantagem o fato de considerar o problema das máquinas-gargalo (máquinas necessárias em mais de uma célula). Porém este método possui algumas desvantagens. Primeiro, a qualidade da solução está fortemente dependente da disposição inicial da matriz. Segundo, os valores binários, utilizados como critério de agrupamento, restringem a dimensão dos problemas que o método tem capacidade de lidar. Este algoritmo tem complexidade de ordem cúbica. Uma versão revisada, conhecida como ROC2, foi desenvolvida por King e Nakornchai (1982), para superar estas limitações e aumentar a eficiência computacional do método. ROC2 implementou técnicas computacionais mais adequadas, de forma a reduzir a complexidade do método. Uma comparação interessante entre métodos que seguem esta linha é apresentada por Chu e Tsai (1990).

A utilização de técnicas que seguem a linha de Agrupamento Hierárquico tem sido amplamente investigada (McAuley 1972, Seifoddini e Wolfe 1986, Seifoddini 1990, Gupta e Seifoddini 1990). Esta linha se caracteriza pelo agrupamento acumulativo, ou seja, se dois componentes são agrupados num estágio do algoritmo, estes não serão separados nos próximos estágios. O processo de agrupamento baseia-se no cálculo de um coeficiente de similaridade entre pares de máquinas ou peças. Diversas medidas de similaridade foram propostas nestas últimas duas décadas. McAuley (1972) sugeriu a utilização do coeficiente de Jaccard para a formação de células, que é definido para duas máquinas como o número de peças que visitam ambas as máquinas dividido pelo número total de peças que visitam cada máquina. Também foi estabelecida uma estratégia de agrupamento, conhecida como agrupamento de ligação única (“Single Linkage Clustering”), que inicialmente forma células com máquinas que possuam o maior coeficiente de similaridade e nos estágios seguintes estabelece valores limitantes de coeficiente de similaridade para agrupamentos de máquinas ou grupos, ou seja, se a máquina x possui um coeficiente de similaridade, em relação à máquina y, maior que o valor limitante deste estágio, estas podem ser agrupadas. A estratégia de McAuley apresenta como

particularidade o tratamento dado aos valores de coeficiente de similaridade que o novo grupo terá. O SLC analisa os coeficientes dos componentes do novo grupo em relação a um outro e designa o maior coeficiente dentre eles para interligar o novo grupo com o grupo selecionado (figuras 2.2.a e 2.2.b). Gupta e Seifoddini (1990) apresentam uma análise sobre estratégias de agrupamento, além de proporem um coeficiente de similaridade que incorpora o volume da produção, a seqüência de operações e os tempos de processamento. A desvantagem desta linha é o uso de valores limitantes, que são determinados, na maioria dos casos, de forma arbitrária. Uma comparação entre esta linha de pesquisa e a Análise de Fluxo de Produção é realizada por Seifoddini (1990).

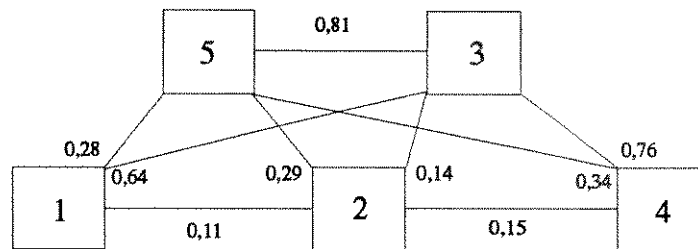


Figura 2.2.a. Diagrama de blocos representando as interações entre as máquinas através dos coeficientes de similaridade (antes do agrupamento da máquina 5 e 3).

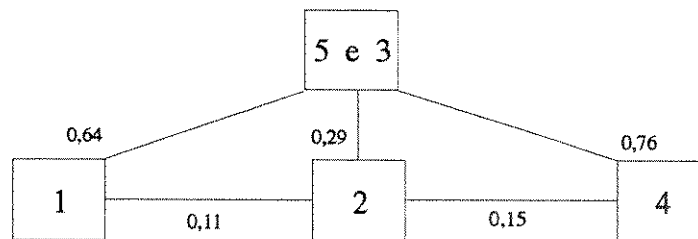


Figura 2.2.b. Diagrama de blocos modificado (depois do agrupamento da máquina 5 e 3).

A linha do Agrupamento Não-Hierárquico caracteriza-se pela possibilidade de realocação dos componentes de um grupo, num estágio posterior ao de sua formação. Chandrasekharan e Rajagopalan (1986, 1987) desenvolveram algoritmos baseados nesta filosofia. O problema é formulado como um grafo bipartido, conforme sugerido em King e Nakornchai (1982), e a este são aplicadas algumas técnicas clássicas de “clustering” com algumas modificações. Chandrasekharan e Rajagopalan (1986, 1989) também apresentaram contribuições interessantes para o estudo do problema, dentre estas: um limitante superior para o número de células, coeficientes avaliadores de soluções e uma análise da capacidade de diagonalização de matrizes de TG.

Rajagopalan e Batra (1975) desenvolveram um método baseado na teoria de grafos para a resolução deste problema. Os nós representam as máquinas e os pesos das arestas são coeficientes de similaridade de Jaccard. O primeiro passo é encontrar os possíveis cliques do grafo. Cliques, para esta análise, significa um grupo de nós, onde cada nó se interliga com um coeficiente de similaridade maior que T , que é estabelecido a partir da análise dos dados iniciais. Os cliques serão as células iniciais. Forma-se então, um novo grafo onde os nós são as células e os pesos dos arcos representam os movimentos intercelulares. A partição deste grupo é realizada através do algoritmo de Kernighan-Lin (1970) adaptado para considerar um limitante superior para o tamanho das células. Como o número de cliques varia exponencialmente com o número de nós (Moon e Moser 1965), o uso desta técnica é aceitável somente para problemas pequenos (King e Nakornchai 1982). Kumar *et al.* (1986) modelaram este problema como um problema de k -decomposição ótima de um grafo. Esta decomposição objetiva obter k subgrafos desconexos onde cada nó apareça somente uma vez em cada subgrafo e que os pesos das arestas que os interliga seja mínimo. É apresentada uma formulação de designação quadrática 0-1 equivalente a k -decomposição e baseado nesta formulação elaborou-se uma heurística que obtém limitantes para a solução ótima. Vannelli e Kumar (1986) desenvolveram uma heurística, também baseada em grafos, para a minimização das máquinas-gargalo (máquina necessária em mais de uma célula), além de considerarem as opções de duplicação de máquinas e/ou subcontratação de peças. Este método procura encontrar os cortes mínimos num grafo bipartido.

Este problema também foi formulado por Kusiak (1987) através de programação inteira 0-1, utilizando como critério a maximização da soma das similaridades entre peças. Ventura *et al* (1990), utilizam a relaxação Lagrangeana para obter um limitante superior para a formulação inteira 0-1 do problema de agrupamento. O problema lagrangeano é decomposto em uma rede linear e um conjunto de problemas da mochila. Estes subproblemas podem ser resolvidos pela exploração de suas estruturas especiais. O algoritmo do subgradiente, com algumas estratégias, é utilizado para minimizar a função dual.

Além de todas estas linhas tradicionais existem outras heurísticas que utilizam algumas das idéias básicas citadas acima. Harhalakis *et al.* (1990) desenvolveram uma técnica interessante que busca minimizar o Tráfego Normalizado Intercelular através de uma estratégia de agrupamento. Este coeficiente de similaridade considera o volume e o roteiro de produção. Após o agrupamento básico, o método utiliza uma estratégia de

refinamento que permite a realocação de uma máquina de acordo com o tráfego desta com as células existentes. O algoritmo apresenta uma filosofia simples e de fácil implementação computacional.

2.4. Proposta do Trabalho

O método descrito neste trabalho é baseado na teoria de grafos, e tem como objetivo o particionamento de uma rede capacitada em subgrafos (células) com a mínima iteração possível. Os vértices se dividem em dois grupos V_m e V_p , representando respectivamente as máquinas e peças. Os pesos dos arcos representam o tempo de processamento das peças nas máquinas. Esta formulação apresenta como vantagens:

- (i) o manuseio de dados diferentes de "0" ou "1", que podem representar de forma mais efetiva a relação de dependência entre máquina e peça, e
- (ii) não utilizar valores arbitrários como parâmetros na formação de células.

A idéia básica deste método foi sugerida por Vohra *et al* (1990), no entanto não foi realizada implementação computacional. A proposta deste trabalho é analisar seu desempenho através de critérios que tentam considerar os objetivos básicos de TG. O método é detalhado no capítulo 3.

Capítulo 3

DESCRIÇÃO DO MÉTODO

3.1. Introdução

Na manufatura celular a situação ideal consiste de células independentes, mas a total segregação das máquinas e peças em células sem nenhuma interação entre elas, é um objetivo muito difícil de se obter. A razão é a ocorrência de peças-gargalo ou de máquinas-gargalo. Uma máquina-gargalo é aquela que é necessária em mais de uma célula, enquanto que uma peça-gargalo é aquela que necessita ser processada em mais de uma célula. Conseqüentemente, a questão é reduzir estas interações ao mínimo. O grau de interação pode ser quantificado pelo cálculo da quantidade ou porcentagem do tempo total de processamento que é gasto fora das células originais. Esta medida de interação entre células representa, portanto, a função objetivo.

Neste capítulo, será apresentado um algoritmo de redes que forma células de máquinas e famílias de peças com o objetivo de minimizar os movimentos intercelulares. O algoritmo proposto é analisado em termos de seus passos básicos e de suas limitações. Um exemplo ilustra sua utilização. Os conceitos básicos para a compreensão deste capítulo estão descritos na seção a seguir.

3.2. Conceitos Preliminares

Dado que nas seções seguintes descreve-se um algoritmo baseado em redes, faz-se necessário apresentar algumas definições sobre o assunto (Chen 1990).

Grafo: um grafo $G(V,E)$ é um conjunto finito não-vazio de elementos denominados nós, e um conjunto E de pares não ordenados na forma (i,j) , $i,j \in V$, chamados de arestas (figura 3.1).

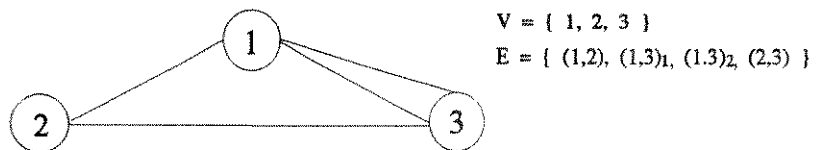


Figura 3.1. Representação de um grafo.

Uma aresta de um grafo é dita ser direcionada (ou orientada) se existir um sentido de direção atribuído a mesma, de modo que um nó é considerado o ponto de origem e o outro o ponto de destino. Um grafo direcionado é aquele em que todas as arestas são direcionadas. Uma aresta direcionada pode ser denominada de arco.

Um grafo não direcionado pode ser transformado em um grafo direcionado através da substituição de cada aresta não direcionada por duas arestas direcionadas conforme figura 3.2.a e 3.2.b.

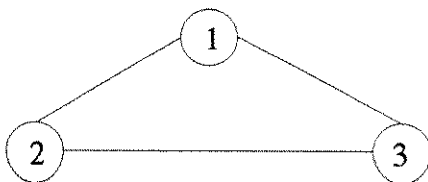


Figura 3.2.a. Grafo não direcionado.

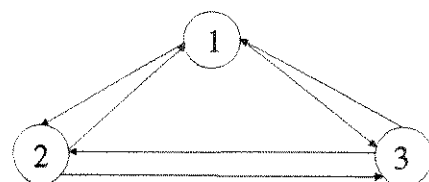


Figura 3.2.b. Grafo direcionado equivalente.

Grafo conexo: um grafo é conexo se existir uma seqüência de arestas (ou arcos) conectando cada par de nós.

Rede: uma rede é um grafo $G(V,E)$ onde a cada aresta (ou arco) (i,j) se associa uma capacidade $c(i,j)$ e um fluxo $f(i,j)$. Uma rede é representada pela quádrupla $G(V,E,c,f)$.

Corte: Para dois nós distintos s e t de um grafo direcionado $G(V,E)$, um corte s - t é o conjunto de arcos (W,\bar{W}) de G com $s \in W$ e $t \in \bar{W}$, onde $W \subset V$ e $\bar{W} = V - W$. A capacidade de um corte s - t (W,\bar{W}) numa rede $G(V,E,c,f)$ é definida por:

$$c(W,\bar{W}) = \sum_{(x,y) \in (W,\bar{W})} c(x,y)$$

Corte mínimo: é o corte que possui o menor valor dentre todos os cortes que separam dois nós distintos s (origem) e t (destino).

Rede Condensada: Seja $G(V,E,c,f)$ uma rede com n nós. Selecione arbitrariamente dois nós x_1 e x_2 como origem e destino, respectivamente, e localize o corte mínimo (X,\bar{X}) entre eles, com $x_1 \in X$ e $x_2 \in \bar{X}$, como ilustrado na figura 3.3. Quando todos os elementos de \bar{X} são condensados em um único nó, a rede resultante é denominada rede condensada. A capacidade de uma aresta na rede condensada é a soma das capacidades das arestas que ligam um nó de X aos nó de \bar{X} (figura 3.3.b).

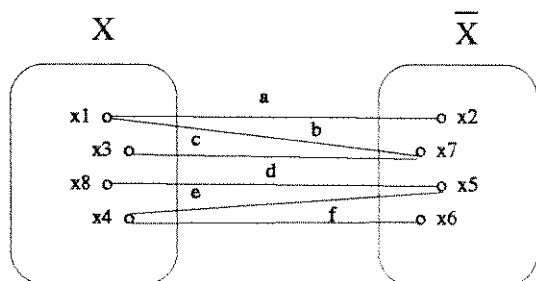


Figura 3.3.a. Representação do corte mínimo.

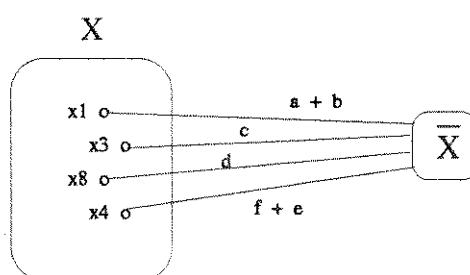


Figura 3.3.b. Rede condensada.

Ciclo: é uma seqüência de arestas (ou arcos) conectando um nó a ele mesmo. Na figura 3.2.b os arcos $(1,3)$, $(3,2)$ e $(2,1)$ definem um ciclo.

Floresta: uma floresta $G_s(V_s,E_s)$ é um subgrafo do grafo $G(V,E)$ onde $V_s = V$, $E_s \subset E$ e não contém ciclos.

Árvore: um grafo com n nós é uma árvore se e somente se qualquer das seguintes propriedades for verdadeira (figura 3.4):

- G tem $n-1$ arestas e não contém ciclos.
- G é conexo mas perde esta propriedade se qualquer das arestas for removida.
- existe um único caminho entre quaisquer dois nós.

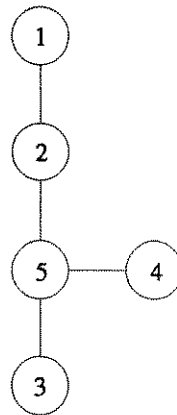


Figura 3.4. Exemplo de árvore.

3.3. Algoritmo Principal

O algoritmo consiste de três passos. O primeiro passo cria uma matriz relacionando máquinas e peças. O segundo constrói uma rede de nós a partir da matriz. E o terceiro passo cria as células de máquinas e famílias de peças através do particionamento da rede utilizando o algoritmo de Gomory-Hu modificado. Uma explicação detalhada dos três passos é dada a seguir.

Passo 1: Formação da Matriz

O tempo de processamento de cada tipo de peça em cada máquina é utilizado para gerar a matriz X de máquinas-peças. Nesta matriz, cada máquina é individualmente identificada por uma única linha e cada tipo de peça por uma coluna. A matriz de dimensão $m \times n$ é representada por:

$$X = [x_{ij}] , \quad i = 1,2,\dots,m; j = 1,2,\dots,n,$$

onde:

x_{ij} = tempo de processamento da peça do tipo j na máquina i

m = número total de máquinas

n = número total de tipos de peças

Passo 2: Construção da Rede

A matriz máquinas-peças gerada no passo 1 pode ser modelada como uma rede capacitada não direcionada G com V nós e E arestas, tal que $G = (V,E)$, onde V é o número total de máquinas e tipos de peças ($m + n$), e E é o número total de operações das peças nas máquinas. A rede tem cada nó-máquina conectado somente a nós-peças e vice-versa. O peso de cada aresta conectando um nó-máquina a um nó-peça representa o tempo de processamento que o tipo de peça requer da referida máquina.

A rede construída deve ser particionada em subredes para formar células de manufatura com o mínimo de movimentos intercelulares.

Passo 3: Particionamento da Rede

O particionamento de uma rede pode ser feito pelo corte de um conjunto de arestas (conjunto-corte) tal que dois conjuntos desconexos de nós e arestas sejam formados. A soma dos pesos das arestas, que compõem o conjunto-corte, representa o movimento intercelular. O objetivo é encontrar o conjunto-corte que possua a menor somatória de pesos.

Logo, para encontrar o ponto de particionamento da rede que possua o menor valor de corte é necessário que se encontre o corte mínimo entre cada par de nós da rede G , denominado conjunto de cortes mínimos de G . Obtém-se do Teorema do Fluxo Máximo - Corte Mínimo (Chen 1990) que o problema de encontrar o corte mínimo entre dois nós é equivalente ao problema de encontrar o fluxo máximo entre os mesmos. Partindo desta premissa, pode ser utilizado repetidamente o algoritmo MPM (Malhortra *et al.* 1978, Chen 1990), que encontra o fluxo máximo entre um par de nós, para localizar o corte mínimo entre eles (Apêndice 1). Porém, para a formação do conjunto de cortes mínimos da rede G é necessário, em princípio, um total de $(m+n).(m+n-1)/2$ aplicações do algoritmo MPM, onde $(m+n)$ é o número de nós do grafo. A seguir será

visto que o número de aplicações do algoritmo MPM pode ser substancialmente reduzido. Para tal, é necessário a seguinte definição.

Definição: Considere duas rede G_1 e G_2 não direcionadas com o mesmo número de nós. G_1 e G_2 são ditas equivalentes se os conjuntos de cortes mínimos de G_1 e G_2 são iguais.

O resultado a seguir é enunciado como um corolário em Chen (1990), e aqui é apresentado como teorema.

Teorema: *Uma rede não direcionada de $(m+n)$ nós é equivalente a uma árvore e conseqüentemente existem na máximo $m+n-1$ valores distintos de corte mínimo.*

Gomory e Hu (1961) sugeriram um procedimento para construção desta árvore equivalente que envolve a solução sucessiva de $(m+n-1)$ problemas de fluxo máximo - corte mínimo. Além disso, vários destes problemas são resolvidos em redes menores que a rede original. A árvore equivalente é usualmente chamada de árvore corte. Qualquer ramo da árvore corte que for retirado separa os nós da rede em dois subconjuntos disjuntos de nós, N_c e $\overline{N_c}$, conectados por um subconjunto de arestas A_c (conjunto corte). O peso total das arestas no subconjunto A_c é o valor de corte. Os subconjuntos N_c e $\overline{N_c}$, representam duas células com interação igual ao valor do corte mínimo. Cada um dos subconjuntos resultantes pode ser decomposto em mais dois. A iteração pode ser repetida até que o número requerido de células seja atingido. Este procedimento é heurístico e, portanto, não garante que uma solução ótima seja encontrada.

Durante o particionamento da rede é observado o número de elementos (máquinas) dentro de cada célula. Caso a célula gerada por um dos cortes da árvore, possua menos elementos do que o mínimo que deve ser definido na descrição do problema, o local de separação deve ser modificado de forma a se respeitar esta restrição. O algoritmo aqui proposto não tem a capacidade de controlar o número máximo de elementos em uma célula. É importante ressaltar que o problema de particionamento de redes em subredes (com ou sem número máximo de nós) com mínima interação é um problema NP-completo (Garey e Johnson 1979 - Particionamento de Grafos - pg. 209). Isto indica que é improvável resolver problemas de grande porte através de algoritmos ótimos em tempo computacional razoável.

Para melhorar a eficiência computacional, o algoritmo de Gomory-Hu é modificado para identificar os cortes mínimos somente entre os nós-máquina. Isto é válido dado que o objetivo principal é formar células de máquinas associadas a famílias de peças, não tendo significado separar as peças das máquinas. Como resultado tem-se a redução do número de iterações de $(m+n-1)$ para $(m-1)$, onde m é o número de máquinas. Como na prática, o número de peças é geralmente muito maior que o número de máquinas, a economia de tempo computacional é significativa. Na próxima seção descreve-se detalhadamente o algoritmo Gomory-Hu com estas alterações.

3.4. Algoritmo de Gomory-Hu Modificado

O objetivo desta seção é apresentar os requisitos básicos para a utilização do algoritmo Gomory-Hu. Inicialmente expõe-se um teorema que serve como base para o algoritmo. A seguir detalha-se as suas principais etapas. Um exemplo ilustra seu funcionamento.

■ Teorema das Redes Condensadas

Este teorema é fundamental para o procedimento desenvolvido por Gomory e Hu (1961). Através dele garante-se a validade de se lidar com redes condensadas ao invés da rede original, que é uma das principais idéias e vantagens do algoritmo, como será apresentado no próximo tópico.

Teorema: *O valor do fluxo máximo entre quaisquer dois nós de uma rede condensada é igual ao valor do fluxo máximo do correspondente par de nós na rede original.*

Prova: Seja x_1 e x_2 dois nós quaisquer de uma rede não direcionada $G(V,E,c,f)$, e seja (X,\bar{X}) o corte mínimo entre x_1 e x_2 . Seja $G^*(V^*,E^*,c^*,f^*)$ uma rede condensada obtida de G pela condensação dos nós de \bar{X} em um único nó. Escolha dois nós quaisquer x_3 e $x_4 \in X$, em G^* e seja (Y,\bar{Y}) o corte mínimo entre eles na rede original G .

Defindo-se:

$$\begin{aligned}
 A &= X \cap Y & \bar{A} &= X \cap \bar{Y} \\
 B &= \bar{X} \cap Y & \bar{B} &= \bar{X} \cap \bar{Y}
 \end{aligned}$$

onde \bar{A} denota o complemento de A em X, e \bar{B} denota o complemento de B em \bar{X} , como representa a figura 3.5. Sem perda de generalidade assume-se que $x_1 \in A$, $x_3 \in A$ e $x_4 \in \bar{A}$. Existem dois casos distintos a serem analisados:

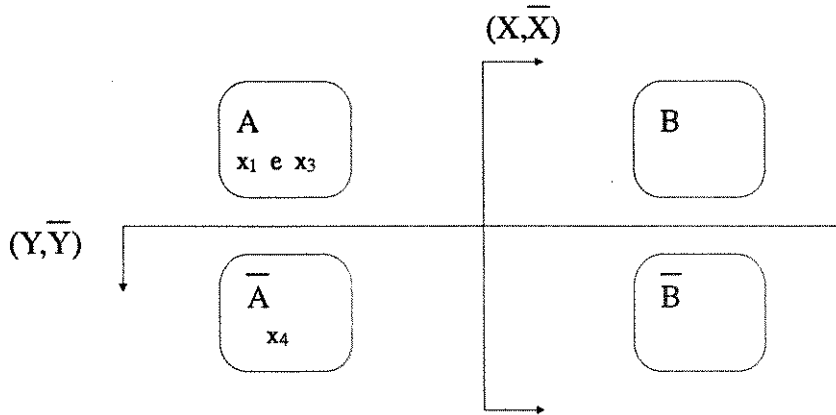


Figura 3.5. Representação simbólica da rede G com os cortes mínimos entre x_1 e x_2 , e entre x_3 e x_4 .

Caso 1: $x_2 \in B$. Então, obtém-se as seguintes expansões:

$$\begin{aligned}
 c(X, \bar{X}) &= c(A \cup \bar{A}, B \cup \bar{B}) = c(A, B) + c(A, \bar{B}) + c(\bar{A}, B) + c(\bar{A}, \bar{B}) \\
 c(Y, \bar{Y}) &= c(A \cup B, \bar{A} \cup \bar{B}) = c(A, \bar{A}) + c(A, \bar{B}) + c(B, \bar{A}) + c(B, \bar{B})
 \end{aligned}$$

Da figura 3.5 observa-se que $(A \cup B \cup \bar{B}, \bar{A})$ é um corte entre x_3 e x_4 e desde que (Y, \bar{Y}) é o corte mínimo entre eles, segue que:

$$c(Y, \bar{Y}) - c(A \cup B \cup \bar{B}, \bar{A}) = c(A, \bar{B}) + c(B, \bar{B}) - c(\bar{B}, \bar{A}) \leq 0 \quad (1)$$

Da mesma forma, desde que $(A \cup \bar{A} \cup \bar{B}, B)$ é um corte entre x_1 e x_2 , e (X, \bar{X}) é o corte mínimo entre eles, tem-se que:

$$c(X, \bar{X}) - c(A \cup \bar{A} \cup \bar{B}, B) = c(A, \bar{B}) + c(\bar{A}, \bar{B}) - c(\bar{B}, B) \leq 0 \quad (2)$$

Como G é uma rede não direcionada, tem-se que:

$$c(B, \bar{B}) = c(\bar{B}, B) \quad \text{e} \quad c(\bar{B}, \bar{A}) = c(\bar{A}, \bar{B}) \quad (3)$$

Adicionando as equações (1) e (2) e usando (3), obtém-se então:

$$c(\bar{B}, A) = c(A, \bar{B}) \leq 0$$

Como as capacidades das arestas são não negativas, segue-se que:

$$c(\bar{B}, A) = c(A, \bar{B}) = 0$$

Portanto, a soma de (1) e (2) resulta em:

$$[c(Y, \bar{Y}) - c(A \cup B \cup \bar{B}, \bar{A})] + [c(X, \bar{X}) - c(A \cup \bar{A} \cup \bar{B}, B)] = 0$$

Isto mostra que se um dos termos dentro dos colchetes é negativo, o outro tem que ser positivo, violando (1) e (2). Então, os termos devem ser identicamente nulos, ou seja:

$$c(Y, \bar{Y}) = c(A \cup B \cup \bar{B}, \bar{A})$$

$$c(X, \bar{X}) = c(A \cup \bar{A} \cup \bar{B}, B)$$

Em outras palavras, se (Y, \bar{Y}) é o mínimo corte entre x_3 e x_4 , então $(A \cup B \cup \bar{B}, \bar{A})$ é também o corte mínimo entre eles. Logo o corte que separa x_3 e x_4 não cruza o conjunto $B \cup \bar{B}$. Conseqüentemente, condensando \bar{X} em um único nó não afeta o valor do fluxo máximo entre x_3 e x_4 , e pode-se utilizar a rede G^* condensada para o cálculo deste fluxo.

Caso 2: $x_2 \in \bar{B}$. Seguindo argumento similar ao utilizado no caso 1, pode ser mostrado que $(A, \bar{A} \cup B \cup \bar{B})$ é também um corte mínimo entre x_3 e x_4 , e que condensando

\bar{X} em um único nó não afeta o valor do fluxo máximo entre eles. Isto completa a prova do teorema.

■ **Etapas do Algoritmo Gomory-Hu Modificado**

Passo 1: Faça $i=1$. Selecione dois nós-máquina x_1 e y_1 arbitrariamente em $G(V,E,c,f)$ e localize um corte mínimo (X_1,X_2) onde $X_2 = V - X_1$, $x_1 \in X_1$ e $y_1 \in X_2$. Represente os conjuntos de nós X_1 e X_2 por dois nós e conecte-os por uma aresta de capacidade $c(X_1,X_2)$, como na figura 3.6 abaixo. A árvore resultante é denominada árvore corte T_1 .

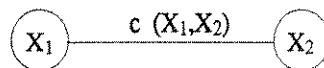


Figura 3.6. Representação simbólica do conjunto de nós X_1 e X_2 conectados por um arco de capacidade $c(X_1, X_2)$.

Passo 2: Selecione um conjunto de nós X_k ($1 \leq k \leq i+1$), com pelo menos dois elementos, e escolha dois nós-máquina y e z arbitrariamente dentro do conjunto X_k . Removendo o nó X_k e todas as suas arestas incidentes de T_i , resulta uma floresta contendo sub-árvores. Retorne à rede original G e forme uma nova rede $G^*(V^*, E^*, c^*, f^*)$ condensando os nós de cada sub-árvore. Observa-se no exemplo da figura 3.7, que com a retirada de X_k obtém-se três sub-árvores.

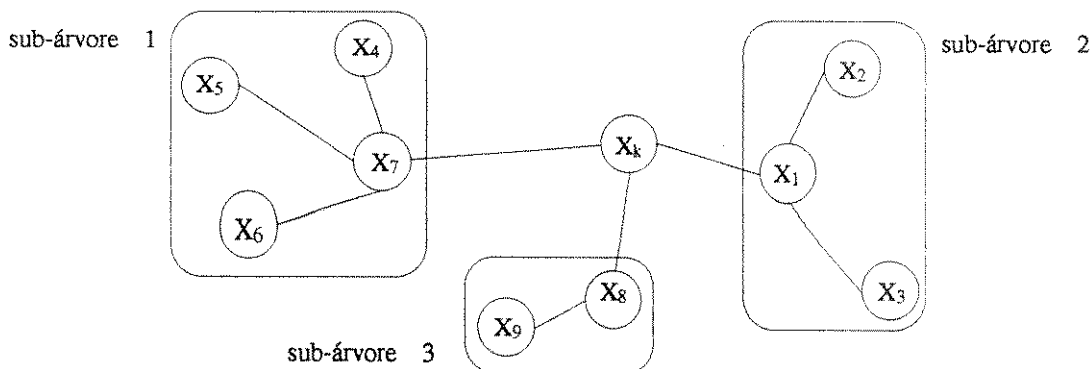


Figura 3.7. Árvore corte T_i com sub-árvores referentes à X_k .

Passo 3: Localize o corte mínimo entre y e z em G^* , que particione o conjunto de nós X_k em X_{k1} e X_{k2} com $y \in X_{k1}$ e $z \in X_{k2}$. Conecte X_{k1} e X_{k2} por uma aresta de capacidade $c^*(y,z)$. Os outros nós da árvore T_i são ligados a X_{k1} se com a retirada dos arcos que pertencem ao corte, estes se localizarem na parte X_{k1} do corte (y,z), e são ligados a X_{k2} no caso contrário (figura 3.8). A nova árvore é denominada T_{i+1} .

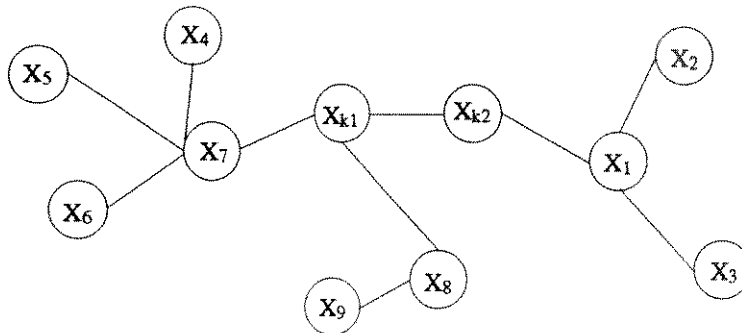


Figura 3.8. Nova árvore corte T_{i+1} .

Passo 4: Faça $X_k = X_{k1}$ e $X_{i+2} = X_{k2}$. Faça $i = i + 1$.

Passo 5: Se $i = m - 1$, pare. Senão, retorne ao passo 2.

■ Exemplo 3.1. Aplicação do Algoritmo Gomory-Hu Modificado

Seja a rede original $G(V,E,c,f)$ da figura 3.9 na qual os nós de 1 a 7 representam as peças, os de 8 a 14 as máquinas e os tempos de processamento denotam a capacidade das arestas. O objetivo do exemplo é a construção da árvore corte desta rede.

Passo 1: Faça $i = 1$, $x_1 = 11$ e $y_1 = 14$. Localize o corte mínimo entre os nós-máquina 11 e 14 utilizando o algoritmo MPM (apêndice 1). O local do corte está indicado na figura 3.9. A árvore corte inicial, T_1 é ilustrada na figura 3.9. Na tabela adjacente a árvore apresentam-se os componentes de cada conjunto.

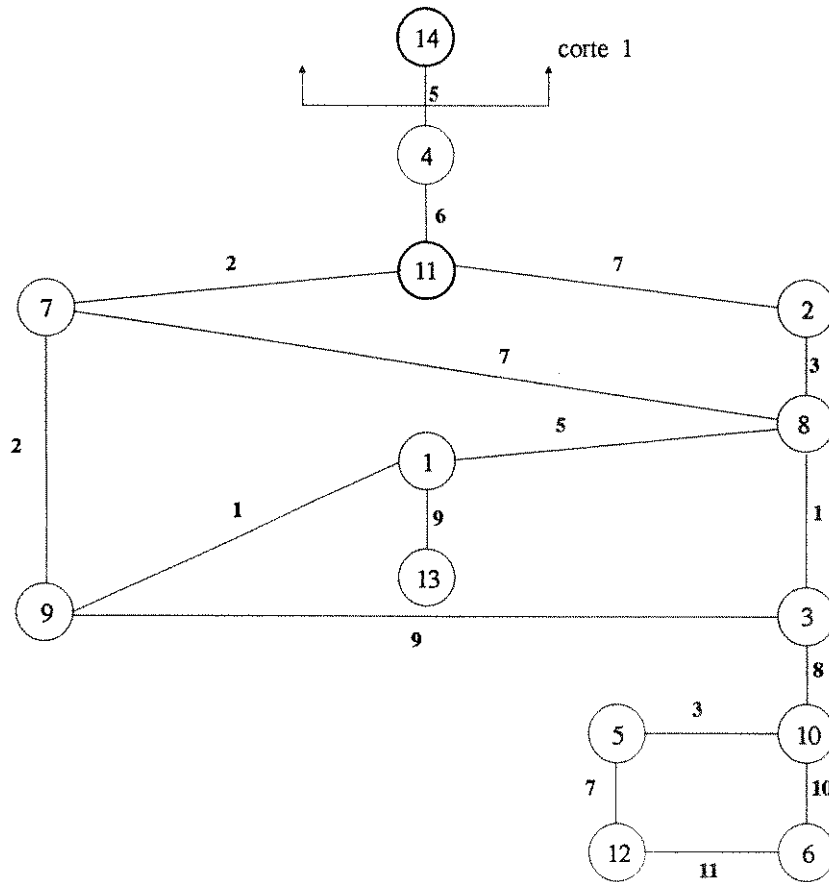


Figura 3.9. Rede original G (Vohra 1990).

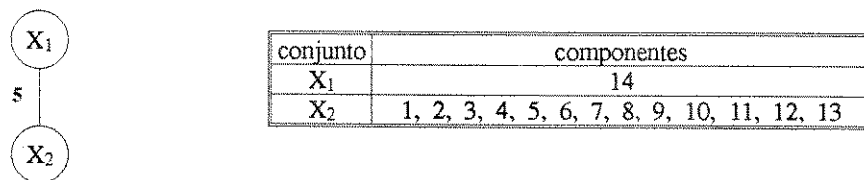


Figura 3.10. Árvore corte T_1 e seus conjuntos de nós.

Passo 2: Faça $X_k = X_2$. Retirando X_2 da árvore T_1 , obtém-se somente uma sub-árvore. Retornando à árvore original G, condensa-se esta sub-árvore em um único nó denominado $W_1 = \{ X_1 \}$ (figura 3.11). Escolha $y = 8$ e $z = 9$ em X_2 .

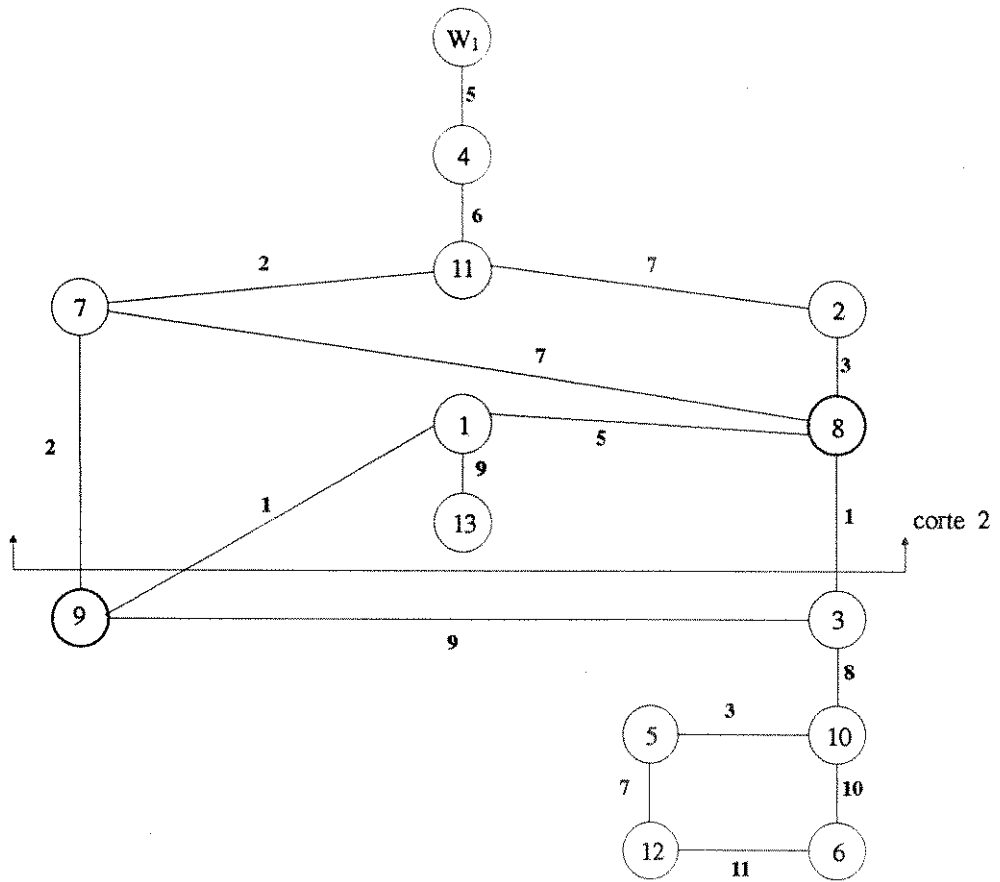
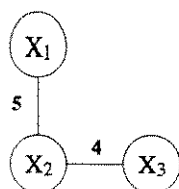


Figura 3.11. Rede condensada G_1^* .

Passo 3: Localize o corte mínimo entre os nós-máquina 8 e 9 em G_1^* . O local de corte está indicado na figura 3.11. Este corte particiona X_2 em dois subconjuntos, gerando o novo conjunto X_3 . Conecte X_2 e X_3 através de uma aresta de capacidade igual ao valor do corte (figura 3.12). Observe que o conjunto X_1 fica conectado a X_2 pois ambos estão do mesmo lado do corte.



conjunto	componentes
X_1	14
X_2	1, 2, 4, 7, 8, 11, 13
X_3	3, 5, 6, 9, 10, 12

Figura 3.12. Árvore corte T_2 e seus conjuntos de nós.

Passo 4: $i = 2$

Passo 5: $i = 6$? Não.

Passo 2: Faça $X_k = X_2$. Retirando X_2 da árvore T_2 , obtém-se duas sub-árvores $W_1 = \{ X_1 \}$ e $W_2 = \{ X_3 \}$. A árvore condensada obtida é ilustrada na figura 3.13. Escolha $y = 8$ e $z = 11$ em X_2 .

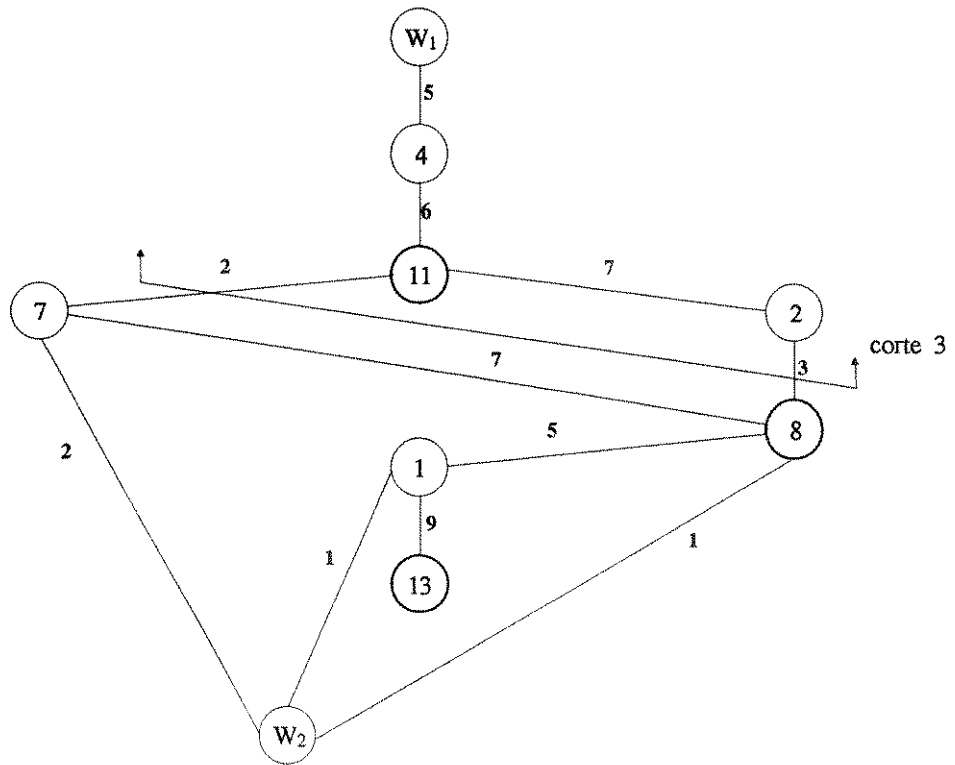


Figura 3.13. Rede condensada G_2^* .

Passo 3: Localize o corte mínimo entre os nós-máquina 8 e 11 em G_2^* . O local de corte está indicado na figura 3.13. Este corte particiona X_2 em dois subconjuntos, gerando o novo conjunto X_4 . Conecte o novo conjunto conforme figura 3.14.

Passo 4.: $i = 3$

Passo 5: $i = 6$? Não.

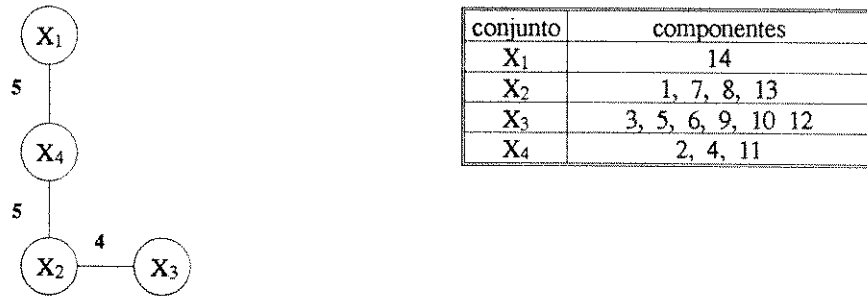


Figura 3.14. Árvore corte T_3 e seus conjuntos de nós.

Passo 2: Faça $X_k = X_2$. Retirando X_2 da árvore T_3 , obtém-se duas sub-árvores $W_1 = \{ X_1, X_4 \}$ e $W_2 = \{ X_3 \}$. A árvore condensada obtida é ilustrada na figura 3.15. Escolha $y = 8$ e $z = 13$ em X_2 .

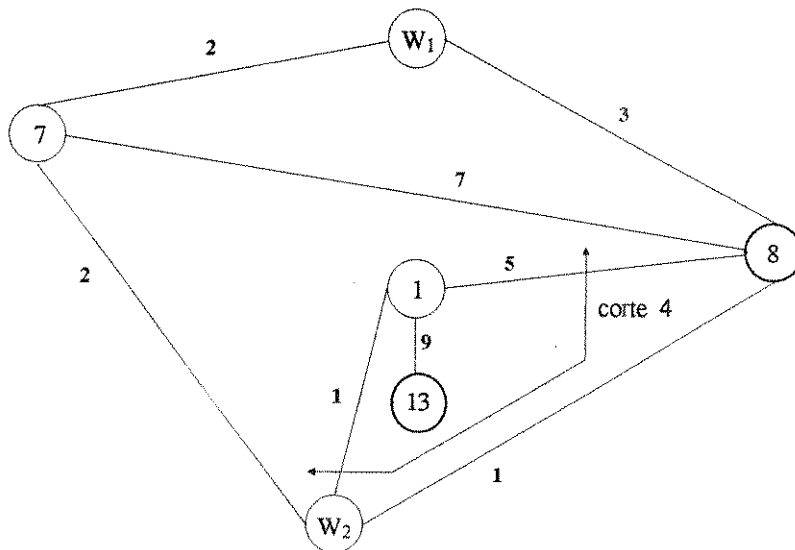


Figura 3.15. Rede condensada G_3^* .

Passo 3: Localize o corte mínimo entre os nós-máquina 8 e 13 em G_3^* . O local de corte está indicado na figura 3.15. Este corte particiona X_2 em dois subconjuntos, gerando o novo conjunto X_5 . Conecte o novo conjunto conforme figura 3.16. Observe que o conjunto X_5 fica conectado somente a X_2 .

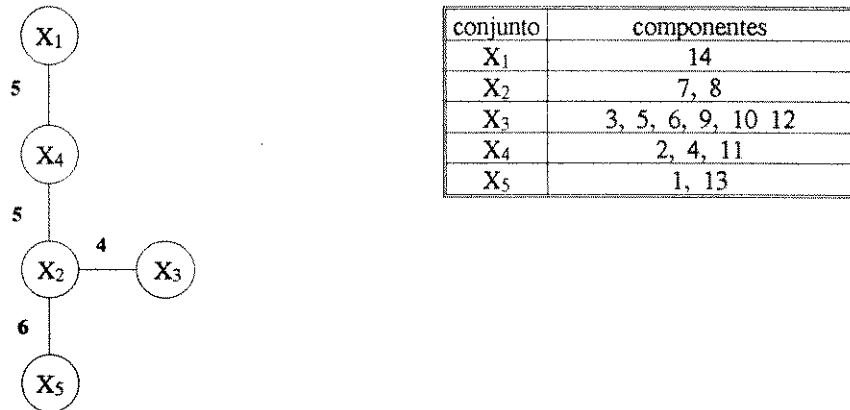


Figura 3.16. Árvore corte T_4 e seus conjuntos de nós.

Passo 4: $i = 4$

Passo 5: $i = 6$? Não.

Passo 2: Faça $X_k = X_3$. Retirando X_3 da árvore T_4 , obtém-se uma sub-árvore $W_1 = \{ X_1, X_2, X_4, X_5 \}$ conforme figura 3.17. Escolha $y = 12$ e $z = 10$ em X_3 .

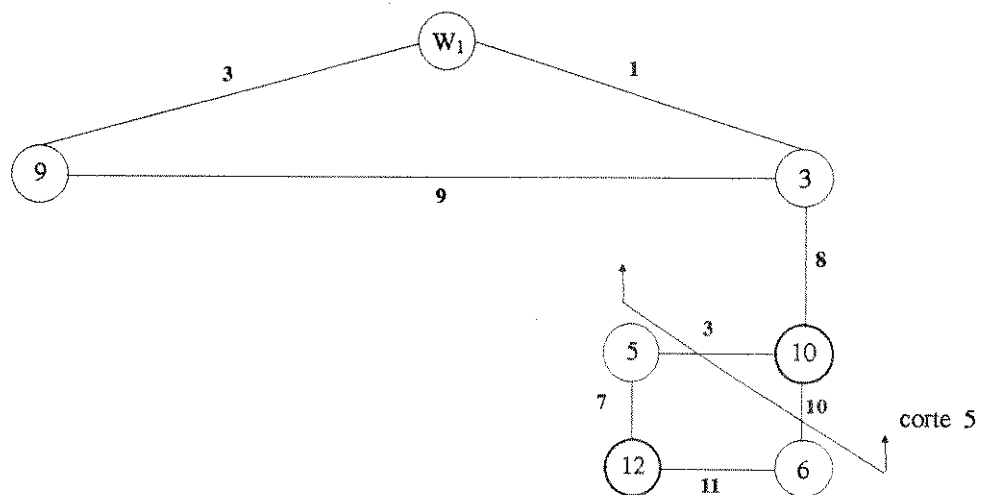


Figura 3.17. Rede condensada G_4^* .

Passo 3: Localize o corte mínimo entre os nós-máquina 12 e 10 em G_4^* . O local de corte está indicado na figura 3.17. Este corte particiona X_3 em dois subconjuntos, gerando o novo conjunto X_6 . Conecte o novo conjunto conforme figura 3.18.

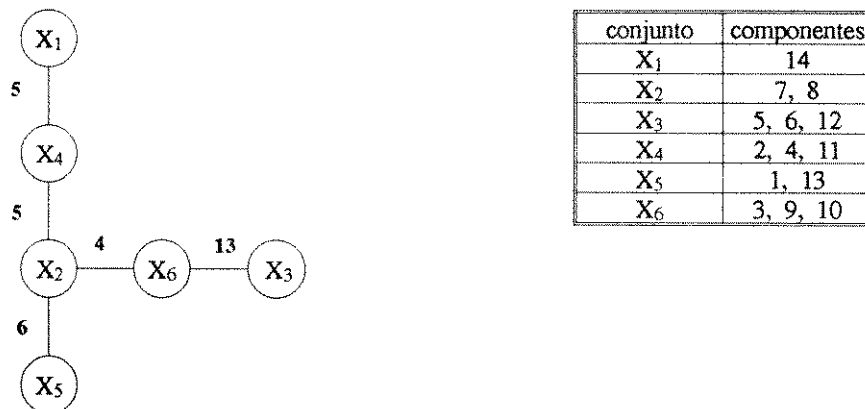


Figura 3.18. Árvore corte T_5 e seus conjuntos de nós.

Passo 4: $i = 5$

Passo 5: $i = 6$? Não.

Passo 2: Faça $X_k = X_6$. Retirando X_6 da árvore T_5 , obtém-se duas sub-árvores $W_1 = \{ X_1, X_2, X_4, X_5 \}$ e $W_2 = \{ X_3 \}$, conforme figura 3.19. Escolha $y = 9$ e $z = 10$ em X_6 .

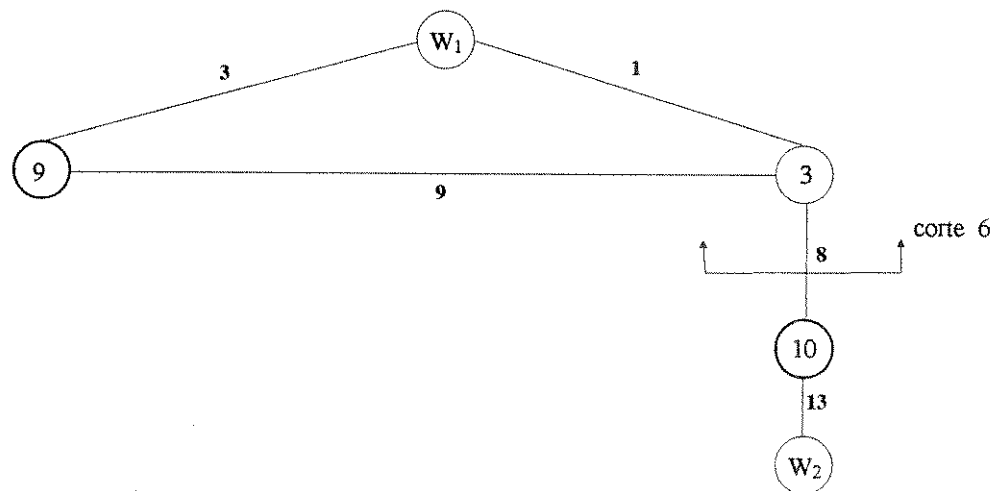
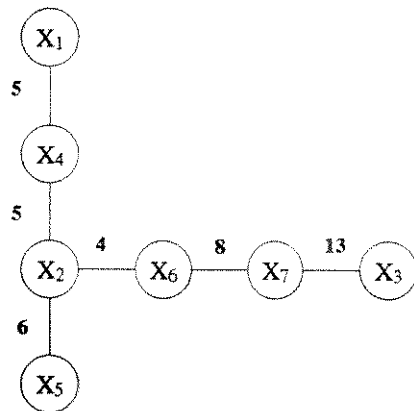


Figura 3.19. Rede condensada G_5^* .

Passo 3: Localize o corte mínimo entre os nós-máquina 9 e 10 em G_5^* . O local de corte está indicado na figura 3.19. Este corte particiona X_6 em dois subconjuntos, gerando o novo conjunto X_7 . Conecte o novo conjunto conforme figura 3.20.



conjunto	componentes
X ₁	14
X ₂	7, 8
X ₃	5, 6, 12
X ₄	2, 4, 11
X ₅	1, 13
X ₆	3, 9
X ₇	10

Figura 3.20. Árvore corte final T_6 com seus conjuntos de nós.

Passo 4: $i = 6$

Passo 5: $i = 6$? Sim, pare.

3.5. Exemplo de Aplicação do Algoritmo Principal

O exemplo seguinte foi selecionado para elucidar os passos do algoritmo principal. Considere uma matriz X , 7×7 , representando 7 tipos de peças ($n=1, \dots, 7$), e 7 máquinas ($m=8, \dots, 14$) como mostrado na figura 3.21. Os elementos da matriz, x_{ij} , denotam o tempo de processamento da peça do tipo j na máquina i . Esta matriz é gerada no passo 1 do algoritmo principal descrito na seção 3.1. O objetivo da resolução deste problema é encontrar três células com a menor iteração possível entre elas. Define-se também que as células devem ter no mínimo dois elementos (duas máquinas).

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

Figura 3.21. Matriz 7 x 7 de máquinas x peças antes do agrupamento.

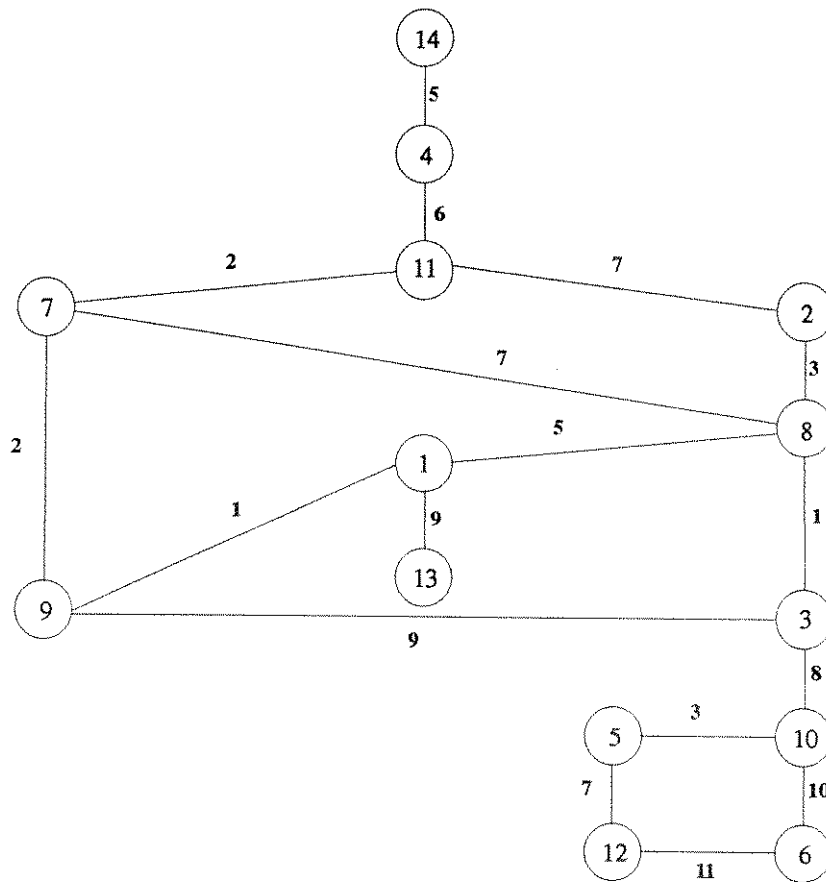


Figura 3.22. Rede original obtida da matriz da figura 3.21.

O passo 2 do algoritmo é então utilizado na matriz para construção da rede, conforme figura 3.22. Nesta rede, todas as máquinas e tipos de peças são representados como nós e os tempos de processamento são representados como as capacidades das

arestas. Note que nesta rede todo nó-máquina é conectado somente a nós-peça, e vice-versa.

No passo 3, os cortes entre nós da rede não direcionada G da figura 3.22 são calculados através do algoritmo de Gomory-Hu modificado. A árvore corte gerada neste passo é mostrada na figura 3.23. O procedimento de geração desta árvore está detalhado no exemplo da seção 3.3.

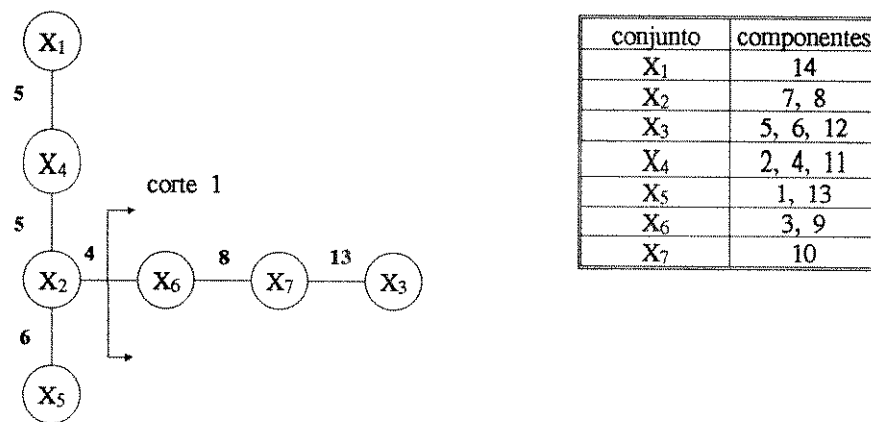


Figura 3.23. Árvore corte e seus conjuntos de nós.

A rede é particionada seccionando o ramo que possui menor valor de corte. Na figura 3.23, o menor valor é 4, entre X_2 e X_6 . Este corte forma duas subredes compostas dos seguintes nós:

- célula 1: { 1, 2, 4, 7, 8, 11, 13, 14 }
- célula 2: { 3, 5, 6, 9, 10, 12 }

Neste ponto a função objetivo possui o valor deste corte:

- função objetivo = 4

Estas duas células (subredes) podem agora ser decompostas em outras células menores. Para isso, é necessário atualizar os valores de corte da árvore. Para tal, aplica-se novamente o algoritmo de Gomory-Hu na rede resultante da retirada das arestas do conjunto-corte de G . A nova árvore corte obtida desta operação está ilustrada na figura 3.24.

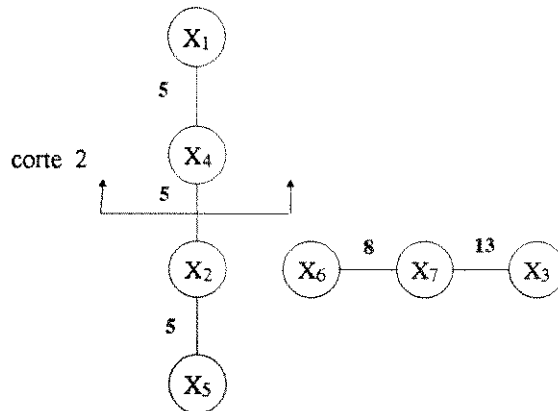


Figura 3.24. Árvore corte após a primeira iteração.

Considerando os novos valores de corte (figura 3.24) seleciona-se o ramo entre X_2 e X_4 para ser seccionado. Observa-se neste caso um empate de valores de corte, porém somente o corte entre X_2 e X_4 obtém o número mínimo de elementos numa célula. O valor de corte obtido é 5. Com a formação desta nova célula (figura 3.25) tem-se:

- célula 1: { 1, 7, 8, 13 }
- célula 2: { 3, 5, 6, 9, 10, 12 }
- célula 3: { 2, 4, 11, 14 }

Atualização da função objetivo:

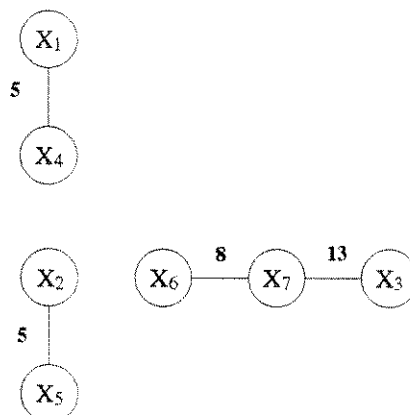


Figura 3.25. Árvore corte após a segunda iteração.

função objetivo = 4 + 5 = 9

Dado que o número desejado de células é três, o algoritmo alcançou seu objetivo. A matriz resultante é apresentada na figura 3.26.

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

Figura 3.26. Matriz 7 x 7 de máquinas x peças após agrupamento.

3.6. Complexidade do Algoritmo

A complexidade computacional de um algoritmo está relacionada com o esforço computacional requerido para sua resolução. Nesta seção analisa-se a complexidade do algoritmo proposto, que pode ser descrita como:

$$\begin{array}{c} \text{complexidade} \\ \text{do algoritmo} \\ \text{proposto} \end{array} = \begin{array}{c} \text{número} \\ \text{de} \\ \text{células} = NC \end{array} \times \begin{array}{c} \text{complexidade} \\ \text{do algoritmo} \\ \text{Gomory-Hu} \\ \text{modificado} \end{array}$$

Logo, é necessário que se defina a complexidade do algoritmo gerador da árvore corte.

$$\begin{array}{c} \text{complexidade} \\ \text{do algoritmo} \\ \text{Gomory-Hu} \\ \text{modificado} \end{array} = \begin{array}{c} \text{número de} \\ \text{máquinas} - 1 \\ = m - 1 \end{array} \times \begin{array}{c} \text{complexidade} \\ \text{do algoritmo} \\ \text{MPM} \end{array}$$

Dado que a complexidade do algoritmo MPM é $O[(n+m)^3]$, onde O simboliza complexidade, e $n+m$ é o número de nós da rede (apêndice 1), tem-se que:

$$O(\text{ Algoritmo de Gomory-Hu }) = O[(n+m)^3(m-1)]$$

Logo a complexidade computacional do algoritmo proposto é:

$$O(\text{ Algoritmo Proposto }) = O[NC (n+m)^3(m-1)]$$

3.7. Contra Exemplo

O objetivo da resolução deste exemplo é ilustrar que o método descrito neste capítulo, utiliza uma estratégia gulosa para resolução do problema de particionamento de redes. Esta estratégia é considerada como uma das mais primitivas ferramentas utilizadas por um algoritmo. A idéia básica é a geração de uma solução de forma a fornecer, em cada etapa da resolução, a melhor contribuição possível à função objetivo. Analisando o problema como um todo, esta estratégia pode ser míope e não garante a otimalidade. No problema em estudo constata-se que, a escolha do corte mínimo como ponto de separação de células nem sempre obtém a menor iteração total entre elas. O exemplo a seguir ilustra este fato. Inicialmente, resolve-se o exemplo seguindo os passos normais do algoritmo principal, ou seja, utilizando a estratégia do corte mínimo. Numa segunda fase, o mesmo exemplo é resolvido, porém, sem utilizar esta estratégia.

■ Exemplo - Primeira Resolução com a Estratégia de Corte Mínimo

Seja a matriz 5x7 da figura 3.27, representando 7 tipos de peças ($n=1,\dots,7$), e 5 tipos de máquinas ($m=8,\dots,12$). Esta matriz é gerada no passo 1 do algoritmo principal descrito na seção 3.1. No passo 2, representa-se esta matriz através de um rede capacitada G (figura 3.28). Aplica-se o algoritmo principal com o objetivo de particionar a rede em três subredes (células) compostas no mínimo de uma máquina.

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

Figura 3.27. Matriz 5 x 7 de máquinas x peças antes do agrupamento

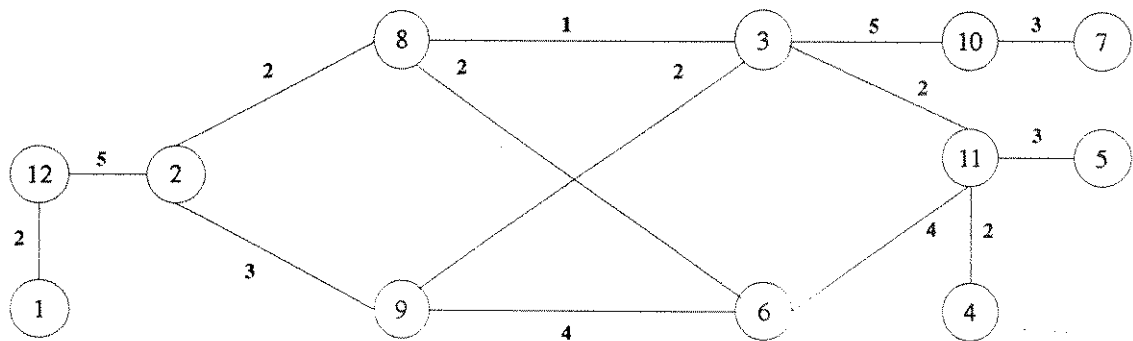
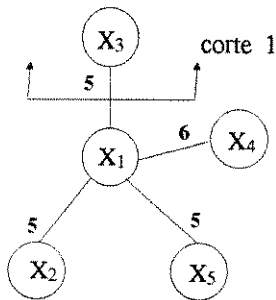


Figura 3.28. Rede original formada da matriz da figura 3.27.

O próximo passo é a construção da árvore corte de Gomory-Hu (figura 3.29) com os valores dos cortes entre os nós. Seleciona-se o ramo da árvore com o menor valor de corte possível. Seleciona-se na figura 3.29 o ramo entre X_1 e X_3 com valor de corte igual a 5. Observe que existe um empate de valores de corte. Dado que ambas as células que seriam formadas atendem a restrição do número de componentes, a escolha é arbitrária. Este corte forma duas subredes compostas dos seguintes nós:

- célula 1: { 1, 12 }
- célula 2: { 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 }



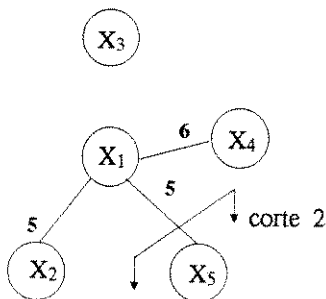
conjunto	componentes
X ₁	2, 6, 9
X ₂	8
X ₃	1, 12
X ₄	4, 5, 11
X ₅	3, 7, 10

Figura 3.29. Árvore corte obtida através do algoritmo de Gomory-Hu.

Neste ponto a função objetivo possui o valor deste corte:

- função objetivo = 5

Estas duas células podem agora ser decompostas em outras células menores. Para isso, é necessário atualizar os valores de corte da árvore. Aplica-se novamente o algoritmo de Gomory-Hu na rede resultante da retirada das arestas do conjunto-corte de G. A nova árvore corte obtida desta operação está ilustrada na figura 3.30.



conjunto	componentes
X ₁	2, 3, 6, 9
X ₂	8
X ₃	1, 12
X ₄	4, 5, 11
X ₅	7, 10

Figura 3.30. Árvore corte após a primeira iteração.

Seleciona-se o ramo entre X₁ e X₅ para ser seccionado. Observa-se neste caso um empate de valores de corte, porém somente o corte entre X₁ e X₅ obtém o número mínimo de elementos numa célula. O valor de corte obtido é 5. Com a formação desta nova célula tem-se:

- célula 1: { 1, 12 }
- célula 2: { 2, 3, 4, 5, 6, 8, 9, 11 }
- célula 3: { 7, 10 }

Atualização da função objetivo:

$$\text{função objetivo} = 5 + 5 = 10$$

Obtidas as três células, a execução do algoritmo principal com a estratégia de corte mínimo está concluída. A matriz da figura 3.31 representa este agrupamento.

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

Figura 3.31. Matriz 5 x 7 de máquinas x peças após o agrupamento realizado através da estratégia de corte mínimo.

■ Exemplo - Segunda Resolução sem a Estratégia de Corte Mínimo

Este procedimento se mantém idêntico ao da primeira resolução até a construção da árvore corte. Nesta resolução o que é modificado são os locais de separação das células. Dado que a estratégia de corte mínimo não está sendo utilizada, escolhe-se como separação entre células, o ramo entre X_1 e X_4 com valor de corte 6 (figura 3.32). Esta escolha é arbitrária.

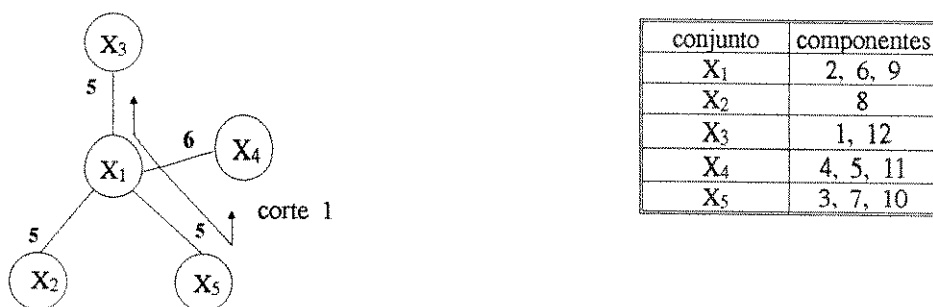


Figura 3.32. Árvore corte com nova escolha de separação inicial de células.

Este corte forma duas subredes compostas dos seguintes nós:

- célula 1: { 4, 5, 11 }
- célula 2: { 1, 2, 3, 6, 7, 8, 9, 10, 12 }

Neste ponto a função objetivo possui o valor deste corte:

- função objetivo = 6

Estas duas células podem agora ser decompostas em outras células menores. Para isso, é necessário atualizar os valores de corte da árvore. Aplica-se novamente o algoritmo de Gomory-Hu na rede resultante da retirada das arestas do conjunto-corte de G. A nova árvore corte obtida desta operação está ilustrada na figura 3.33.

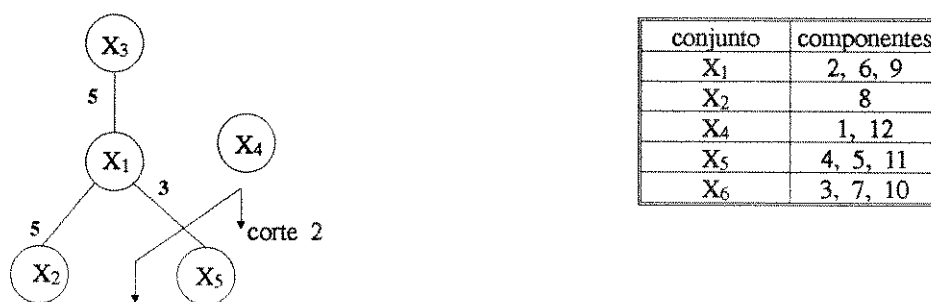


Figura 3.33. Árvore corte após a primeira iteração.

Seleciona-se o ramo entre X_1 e X_5 para ser seccionado. O valor de corte obtido é 3. Com a formação desta nova célula tem-se:

- célula 1: { 4, 5, 11 }
- célula 2: { 1, 2, 6, 8, 9, 12 }
- célula 3: { 3, 7, 10 }

Atualização da função objetivo:

- função objetivo = 6 + 3 = 9

Obtidas as três células, a execução do algoritmo principal sem a estratégia de corte mínimo está concluída. A matriz da figura 3.34 representa este agrupamento.

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

Figura 3.30. Matriz 5 x 7 de máquinas x peças após o agrupamento realizado sem a estratégia de corte mínimo.

Observa-se que o novo valor da função objetivo é melhor que o obtido na primeira resolução. Isto ilustra o fato da estratégia de corte mínimo ser míope.

Capítulo 4

APRESENTAÇÃO E ANÁLISE DOS RESULTADOS

4.1. Introdução

O objetivo deste capítulo é descrever os resultados da implementação do procedimento heurístico descrito no capítulo 3. Este método forma células de máquinas e famílias de peças para aplicação da técnica de produção Tecnologia de Grupo. Na segunda seção são apresentados índices que avaliam a qualidade dos resultados. Na terceira, o algoritmo é aplicado a problemas clássicos da literatura, e seu desempenho é analisado. A seguir expõem-se as características do procedimento constatadas a partir dos testes computacionais.

Os programas computacionais foram escritos na linguagem C e os testes realizados numa estação de trabalho SUN modelo SPARCstation +1. Detalhes sobre a implementação se encontram no apêndice 2.

4.2. Índices Avaliadores

Nesta seção são apresentados índices capazes de fornecer uma base quantitativa para comparar e avaliar diferentes soluções do mesmo problema. Os índices são obtidos da análise da matriz máquinas x peças diagonalizada, ou seja, após o agrupamento. Para tal análise algumas definições são necessárias:

Bloco: submatriz da matriz máquinas x peças formada pela intersecção das colunas representando a família de peças e das linhas representando as células de máquinas.

Forma Bloco Diagonal Perfeita: é a forma de disposição de uma matriz tal seus blocos diagonais contenham somente elementos não nulos e todos os elementos fora destes blocos são nulos (figura 4.1).

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

bloco 1 (aponta para o primeiro bloco diagonal, linhas 10-7, colunas 1-6)

bloco 2 (aponta para o segundo bloco diagonal, linhas 8-9-11, colunas 3-5)

Figura 4.1. Matriz máquinas x peças na forma bloco diagonal perfeita

$\{ x_{ij} \}$: componente da matriz máquinas x peças onde $x_{ij} \neq 0$ se a peça da coluna j visita a máquina da linha i e $x_{ij} = 0$ caso contrário.

$$\{ a_{ij} \}: \begin{cases} a_{ij} = 1 & \text{se } x_{ij} \neq 0 \\ a_{ij} = 0 & \text{se } x_{ij} = 0 \end{cases}$$

m : número de máquinas (linhas).

n : número de peças (colunas).

k: número de blocos diagonais (número de células)

e: número total de operações (número de elementos não nulos da matriz).

$$e = \sum_{i=1}^m \sum_{j=1}^n a_{ij}$$

e_d: número de elementos não nulos nos blocos diagonais.

$$e_d = \sum_{r=1}^k \sum_{i=M_0+\dots+M_{r-1}+1}^{M_0+\dots+M_{r-1}+M_r} \sum_{j=N_0+\dots+N_{r-1}+1}^{N_0+\dots+N_{r-1}+N_r} a_{ij} \quad \text{onde } M_0 = N_0 = 0$$

e_o: número de elementos não nulos fora dos blocos diagonais.

$$e_o = e - e_d$$

e_v: número de elementos nulos nos blocos diagonais.

$$e_v = \sum_{r=1}^k M_r N_r - e_d$$

M_r: número de máquinas (linhas) no bloco r.

N_r: número de peças (colunas) no bloco r.

q: fator ponderador, $0 \leq q \leq 1$

φ: e_v/e = taxa de elementos nulos em relação ao número total de operações.

ψ: e_o/e = taxa de elementos não nulos fora dos blocos diagonais em relação ao número total de operações.

η: índice de eficiência de agrupamento.

Γ: índice de eficácia de agrupamento.

Obs: As equações matemáticas expostas acima devem ser consideradas como sugestões. O cálculo destas variáveis deve ser realizado da forma que mais se adapte a estrutura dos dados utilizados como referência. O essencial neste cálculo é que não haja modificação no significado da variável.

Os índices de agrupamento estão relacionados basicamente com dois fatores:

- a utilização interna das máquinas da célula, e
- os movimentos intercelulares das peças.

Logo, um melhor agrupamento significa maior utilização interna das máquinas e/ou menor quantidade de movimentos intercelulares das peças. Nas próximas seções, examina-se as expressões matemáticas destes índices.

4.2.1. Eficiência de Agrupamento

Este índice denominado eficiência de agrupamento η , foi proposto por Chandrasekharan e Rajagopalan (1986) e pode ser definido como a média ponderada de dois índices diferentes η_1 e η_2 , ou seja:

$$\text{Eficiência de Agrupamento} = \eta = q \cdot \eta_1 + (1-q) \cdot \eta_2, \quad 0 \leq q \leq 1 \quad (3)$$

onde:

$$\begin{aligned} \eta_1 &= \frac{\text{Número de elementos não nulos nos blocos diagonais}}{\text{Número total de elementos nos blocos diagonais}} = \\ &= \frac{ed}{\sum_{r=1}^k M_r N_r} \end{aligned} \quad (4)$$

$$\eta_2 = \frac{\text{Número de elementos nulos fora dos blocos diagonais}}{\text{Número total de elementos fora dos blocos diagonais}} =$$

$$= 1 - \frac{e_0}{\left(mn - \sum_{r=1}^k M_r N_r \right)} \quad (5)$$

O índice η_1 avalia o nível de utilização interna das máquinas de uma célula e η_2 o nível de movimentação intercelular das peças. O fator ponderador q , permite alterar a ênfase dada a cada um destes índices de acordo com as necessidades específicas de cada problema. Observe que a função η possui a propriedade de não negatividade e só assume valores entre 0 e 1, ou seja, apresenta características básicas para um índice de eficiência. Além disso, $\eta = 1$ caracteriza uma matriz na forma bloco diagonal perfeita.

Mais de 100 casos diferentes foram analisados por Kumar e Chandrasekharan (1990) e foi constatado que os valores deste índice variaram numa faixa entre 0,75 e 1. Mesmo para soluções com um número elevado de elementos fora dos blocos diagonais, o índice apresentou valores em torno de 0,75, o que confirma a baixa sensibilidade deste índice em relação à η_2 (movimentação intercelular). Numa pesquisa posterior sobre este problema, foi proposto um novo índice que pondera com a mesma importância, o nível de utilização interna das máquinas e o nível de movimentação intercelular. Este índice é descrito na próxima seção.

4.2.2. Eficácia de Agrupamento

O índice denominado eficácia de agrupamento Γ foi desenvolvido por Kumar e Chandrasekharan (1990). Estes autores apresentam uma análise deste índice onde suas características básicas são descritas. Este índice pode ser definido por:

$$\text{Eficácia de Agrupamento} = \Gamma = \frac{(1 - \psi)}{(1 + \phi)} \quad (6)$$

onde:

$$\psi = \frac{\text{Número de elementos não nulos fora dos blocos diagonais}}{\text{Número total de operações}} = \frac{e_o}{e}$$

$$\phi = \frac{\text{Número de elementos nulos nos blocos diagonais}}{\text{Número total de operações}} = \frac{e_v}{e}$$

Este índice também pode ser expresso por:

$$\Gamma = \frac{e - e_o}{e + e_v}$$

Esta expressão possui a propriedade da não negatividade e está limitada a valores entre 0 e 1. Além disso, ψ e ϕ são taxas e não são afetadas pela dimensão da matriz. A taxa ψ representa o nível de movimentos intercelulares, e a taxa ϕ o nível de utilização interna das máquinas nas células. Observa-se através da equação (6) que, $\Gamma = 0$ implica em $\psi = 1$. Logo a eficácia nula é um ponto onde todos os elementos não nulos da matriz estão fora dos blocos diagonais. Quando $\Gamma = 1$ implica em $\psi = \phi = 0$. Esta condição corresponde a uma matriz na forma bloco-diagonal perfeita.

A seguir calcula-se os índices de agrupamento para avaliar a solução obtida para um exemplo. O cálculo destes índices é realizado passo a passo.

■ Exemplo 1

Seja a matriz 20x20 de máquinas e peças da figura 4.2 (Harhalakis 1990). Cada elemento x_{ij} da matriz representa o tempo de processamento da peça j na máquina i . O número de células desejado é 5, com no mínimo três componentes (três máquinas).

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
21	2								4			5		3			2			3
22		3		3										4						2
23		2		1							3									1
24					1			5												
25						5	1					3								4
26				3								1			3					
27				4				3				2			2					
28			1							3				1					1	
29	3							4	2			1					1			
30				4										2				4		2
31		1		2		1			3		1									3
32	1								5			4					3			4
33								2							1					
34					2					2					2					
35				2			1					3			1					
36					3	2									3					
37					4	3						4		4						
38	4							1			2									1
39			3							1						4		2		
40	5		2							2								3		

Figura 4.2. Matriz máquinas x peças antes do agrupamento.

Aplicando o algoritmo principal descrito no capítulo 3, obtém-se a matriz diagonalizada mostrada na figura 4.3. Baseado nesta matriz calcula-se os índices de agrupamento.

- Cálculo do Índice de Eficiência

$$\eta = q \cdot \eta_1 + (1-q) \cdot \eta_2$$

onde:

$$\eta_1 = \frac{e_d}{k} \quad e \quad \eta_2 = 1 - \frac{e_o}{\left(mn - \sum_{r=1}^k M_r N_r \right)}$$

$$e = \sum_{i=1}^m \sum_{j=1}^n a_{ij} = \sum_{i=1}^{20} \sum_{j=1}^{20} a_{ij} = 79$$

	13	5	16	8	19	14	4	2	11	15	7	6	17	20	12	9	1	10	18	3	
35	3	2	1	1																	
27	2	4	2	3																	
26	1	3	3																		
24		1		5																	
33			2							1											
31				3		2	1	1			1					3					
22				2	4	3	3														
30					2	4								2						4	
23				1		1	2	3													
25				4						1	5				3						
36										3	2	3									
37	4									4	3	4									
34									2	2		2									
29			4										1		1	2	3				
32													3	4	4	5	1				
38														1	2	1	4				
21					3								2	3	5	4	2				
28					1														3	1	1
40																5			2	3	2
39		4																	1	2	3

Figura 4.3. Matriz máquinas x peças após o agrupamento.

$$e_d = \sum_{r=1}^k \sum_{i=M_0+\dots+M_{r-1}+1}^{M_0+\dots+M_{r-1}+M_r} \sum_{j=N_0+\dots+N_{r-1}+1}^{N_0+\dots+N_{r-1}+N_r} a_{ij} \quad \text{onde } M_0 = N_0 = 0$$

$$e_d = \sum_{i=1}^5 \sum_{j=1}^4 a_{ij} + \sum_{i=6}^{10} \sum_{j=5}^9 a_{ij} + \sum_{i=11}^{13} \sum_{j=10}^{12} a_{ij} +$$

$$\sum_{i=14}^{17} \sum_{j=13}^{17} a_{ij} + \sum_{i=18}^{20} \sum_{j=18}^{20} a_{ij} = 14 + 15 + 8 + 18 + 9 = 64$$

$$e_0 = e - e_d = 79 - 64 = 15$$

$$\eta_1 = \frac{e_d}{k \sum_{r=1} M_r N_r} = \frac{64}{20 + 25 + 9 + 20 + 9} = 0,7711 = 77,11\%$$

$$\eta_2 = 1 - \frac{e_o}{\left(mn - \sum_{r=1}^k M_r N_r \right)} = 1 - \left[\frac{15}{400 - 83} \right] =$$

$$= 1 - 0,047 = 0,9527 = 95,27 \%$$

para $q = 0,5$

$$\eta = \frac{0,7711 + 0,9527}{2} = 0,8619 = 86,19\%$$

- Cálculo do Índice de Eficácia

$$\Gamma = \frac{(1 - \psi)}{(1 + \phi)}$$

onde:

$$e_v = \sum_{r=1}^k M_r N_r - e_d = 83 - 64 = 19$$

$$\psi = \frac{e_o}{e} = \frac{15}{79} = 0,1899$$

$$\phi = \frac{e_v}{e} = \frac{19}{79} = 0,2405$$

$$\Gamma = \frac{(1 - \psi)}{(1 + \phi)} = \frac{1 - 0,1899}{1 + 0,2405} = 0,6531 = 65,31 \%$$

(3)

4.3. Exemplos da Literatura

Nesta seção utiliza-se exemplos conhecidos da literatura para analisar o comportamento do procedimento heurístico proposto neste trabalho. A tabela 4.1. apresenta uma comparação entre alguns algoritmos selecionados com o algoritmo proposto. Esta comparação é realizada em termos de índices de agrupamento.

Referências	Dimensão da matriz	Número de células	Outros Algoritmos		Algoritmo Proposto		Tempo (seg.)
			Eficiência %	Eficácia %	Eficiência %	Eficácia %	
01. King (1980)	5 x 6	2	83,33	70,58	83,33	70,59	0,1
02. King e Nakonrchai (1982)	5 x 7	2	85,33	72,83	85,62	73,68	0,1
03. Kumar <i>et al</i> (1986)	9 x 15	3	84,05	68,08	84,05	68,09	0,4
04. Chandrasekharan <i>et al</i> (1986)	8 x 20	3	95,83	85,25	95,83	85,25	0,1
05. Ventura <i>et al</i> (1990)	11 x 22	3	85,33	72,83	87,53	72,83	0,6
06. Harhalakis <i>et al</i> (1990)	20 x 20	5	85,72	64,64	86,19	65,31	5,7
07. Harhalakis <i>et al</i> (1990)	20 x 20	4	79,6	57,15	74,54	48,92	5,3
08. Kumar <i>et al</i> (1986)	23 x 20	2	59,73	37,27	55,57	25,41	8,2
09. Ventura <i>et al</i> (1990)	16 x 43	2	63,97	29,92	61,36	26,83	2,8
10. Chandrasekharan <i>et al</i> (1986)	20 x 35	4	84	75,14	87,81	75,14	1,6
11. Chandrasekharan <i>et al</i> (1989)	24 x 40	7	100	100	100	100	1,9
12. Chandrasekharan <i>et al</i> (1989)	24 x 40	7	95,2	85,11	95,2	85,11	2,5
13. Chandrasekharan <i>et al</i> (1989)	24 x 40	7	91,14	73,5	91,66	75,17	7,7
14. Chandrasekharan <i>et al</i> (1989)	24 x 40	7	85,04	73,5	91,16	73,51	10,3
15. Chandrasekharan <i>et al</i> (1989)	24 x 40	7	77,31	40,4	56,19	17,79	33,5
16. Kumar <i>et al</i> (1990)	46 x 100	12	98,36	94,17	98,36	94,17	26,3

Tabela 4.1. Índices de agrupamento para exemplos da literatura.

O tempo de processamento do algoritmo proposto para cada exemplo é apresentado na última coluna. As referências da literatura não apresentam os tempos relativos a estes exemplos. Embora sem um parâmetro de comparação, o tempo obtido pelo algoritmo proposto pode ser considerado bom.

Os exemplos da tabela 4.1 são matrizes binárias, com excessão dos exemplos 6 e 7 (Harhalakis *et al* 1990) que utilizam uma matriz com números inteiros positivos. Para estes dois exemplos calcula-se a seguir o nível de utilização interna das máquinas nas células, considerando os valores dos componentes da matriz. Estes valores representam o tempo de processamento.

$$\textit{Utilização Interna} = \frac{\textit{Somatória dos elementos não nulos nos blocos diagonais}}{\textit{Somatória total dos elementos da matriz}}$$

- Cálculo da Utilização Interna do exemplo 6 :

$$UI_{\textit{outros algoritmos}} = \frac{155}{201} \times 100 = 77,11 \%$$

$$UI_{\textit{algoritmo proposto}} = \frac{158}{201} \times 100 = 78,6 \%$$

- Cálculo da Utilização Interna do exemplo 7 :

$$UI_{\textit{outros algoritmos}} = \frac{164}{201} \times 100 = 81,59 \%$$

$$UI_{\textit{algoritmo proposto}} = \frac{169}{201} \times 100 = 84,08 \%$$

Analisando as células finais através destes índices, observa-se que o comportamento do algoritmo proposto para os exemplos 6 e 7 foi melhor que o dos outros algoritmos. A finalidade deste cálculo é apresentar um índice simples que possa avaliar células considerando fatores como tempo de processamento ou volume de cada tipo de peça. A matriz inicial e final do exemplo 6 está ilustrada respectivamente, na figura 4.2 e 4.3.

Os exemplos de 11 a 15 compõem um conjunto de problemas com um grau crescente de dificuldade, o que pode ser comprovado pela queda progressiva dos índices. O tempo de processamento para este bloco de exemplos foi crescente, ressaltando o fato do exemplo 15 (24 x 40) ter obtido um tempo pior que o exemplo de maior dimensão da tabela (46 x 100). Desta observação, conclui-se que o tempo de processamento não tem uma relação direta com a dimensão, e que está relacionado com a estrutura da matriz. Chandrasekharan et al (1989) realizaram um interessante estudo sobre matrizes estruturalmente complexas, onde tentam identificar antecipadamente quando a matriz não será facilmente diagonalizada.

Os índices de eficiência e eficácia obtidos pelos outros algoritmos e pelo algoritmo proposto estão respectivamente, na quarta e quinta coluna. Para facilitar a visualização destes valores, foram construídos os gráficos 4.1, 4.2, 4.3 e 4.4 expostos a seguir.

Através destes gráficos pode ser observada a diferença de comportamento entre o algoritmo descrito e os outros algoritmos. No exemplo 15 percebe-se uma grande diferença entre os índices obtidos com cada algoritmo. Este problema foi resolvido detalhadamente e observou-se que a sua árvore corte equivalente possui a forma de uma estrela como exemplificado na figura 4.4.

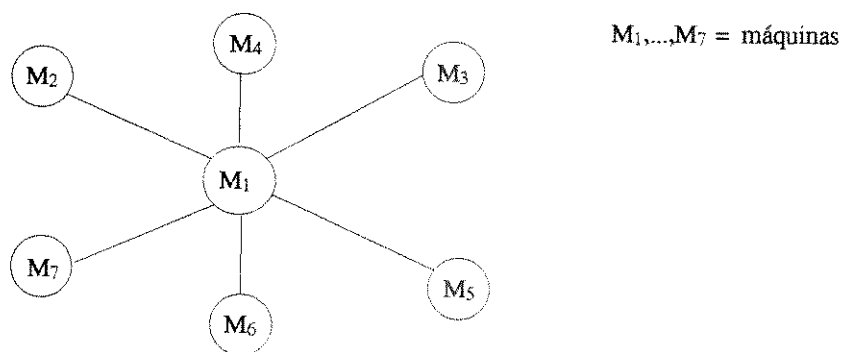


Figura 4.4. Exemplo de árvore corte em forma de estrela.

Devido a este formato todos os cortes possíveis formarão uma célula de somente uma máquina. Logo a qualidade das células será muito ruim e conseqüentemente os índices terão valores baixos.

Índices de Eficiência

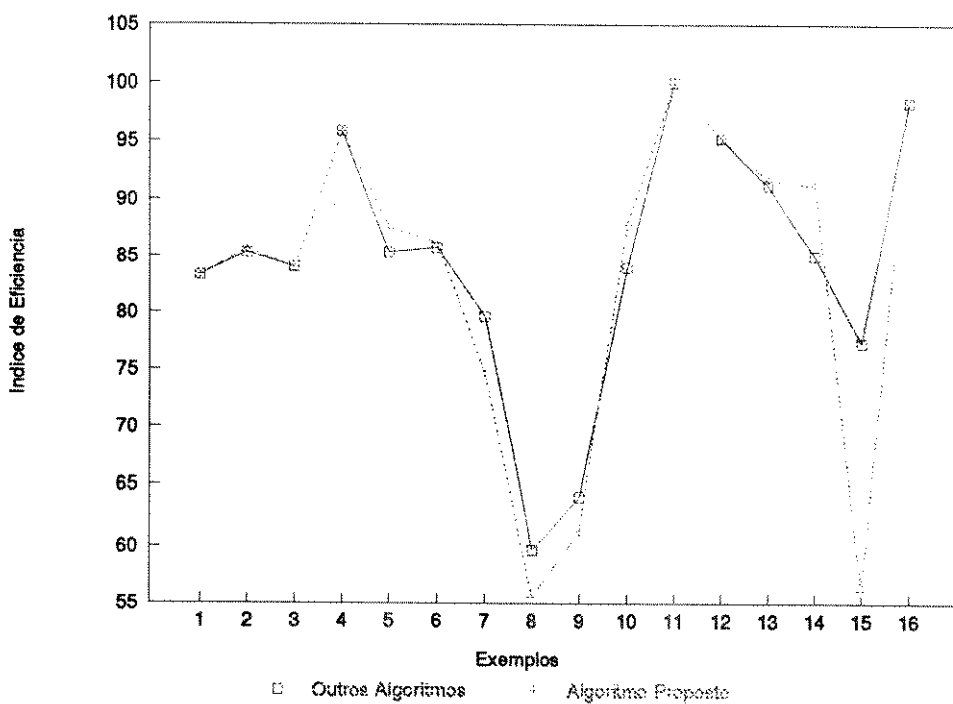


Gráfico 4.1. Comparação entre índices de eficiência.

Diferença entre Índices de Eficiência

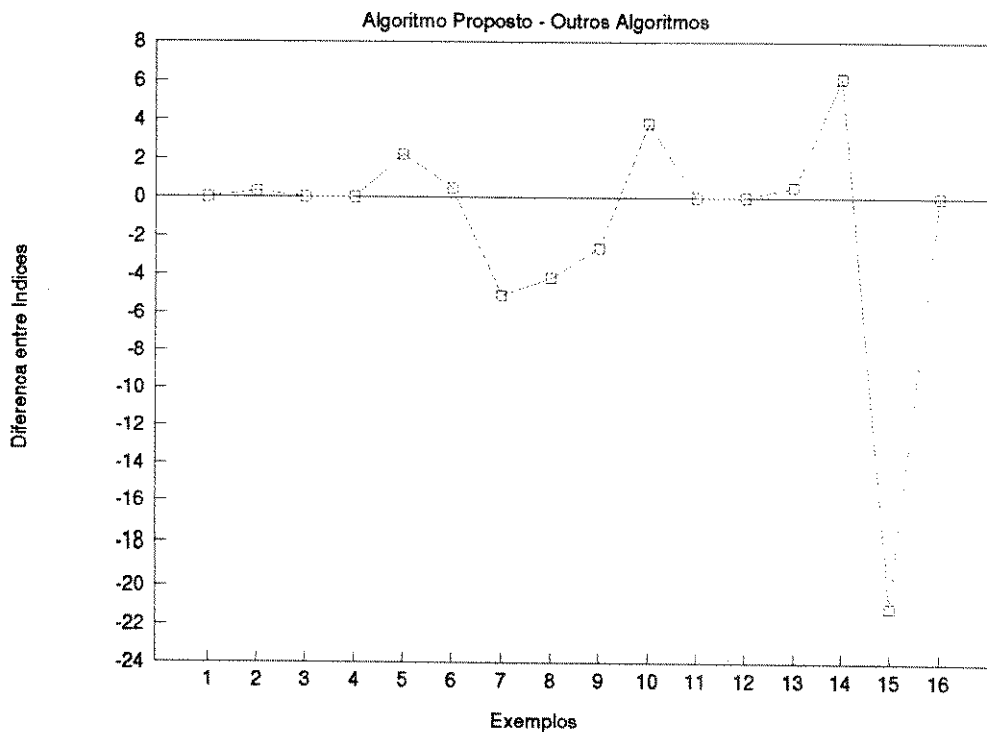


Gráfico 4.2. Variação da diferença entre os algoritmos.

Índices de Eficácia

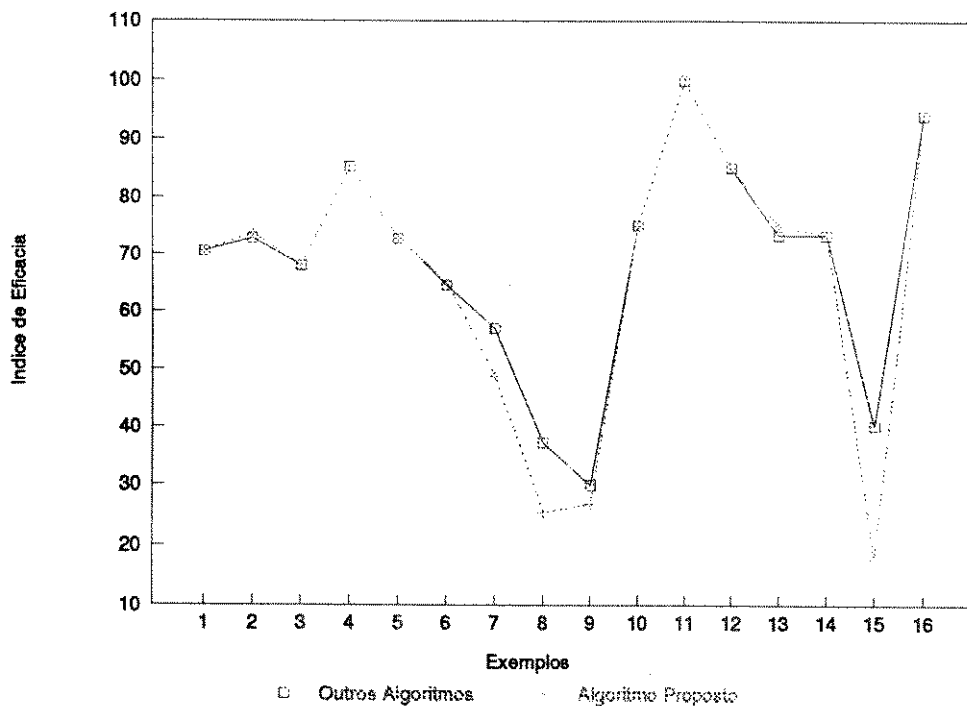


Gráfico 4.3. Comparação entre índices de eficácia.

Diferença entre Índices de Eficácia

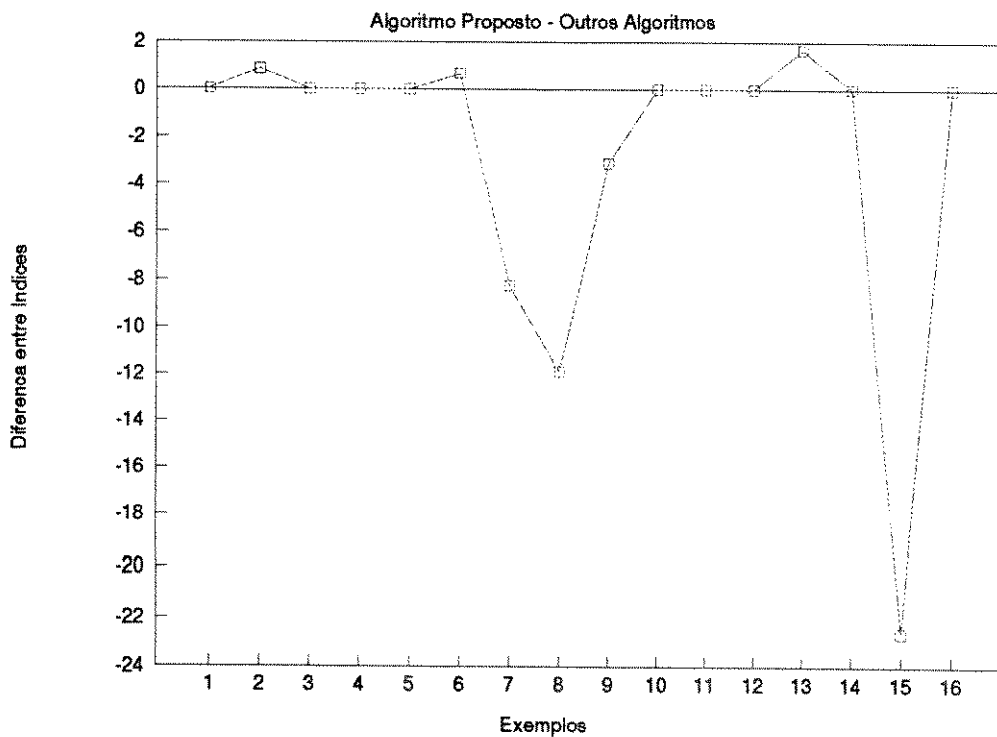


Gráfico 4.4. Variação da diferença entre os algoritmos.

Analisam-se a seguir os valores médios dos índices. Estes valores estão ilustrados através dos gráficos 4.5 e 4.6.

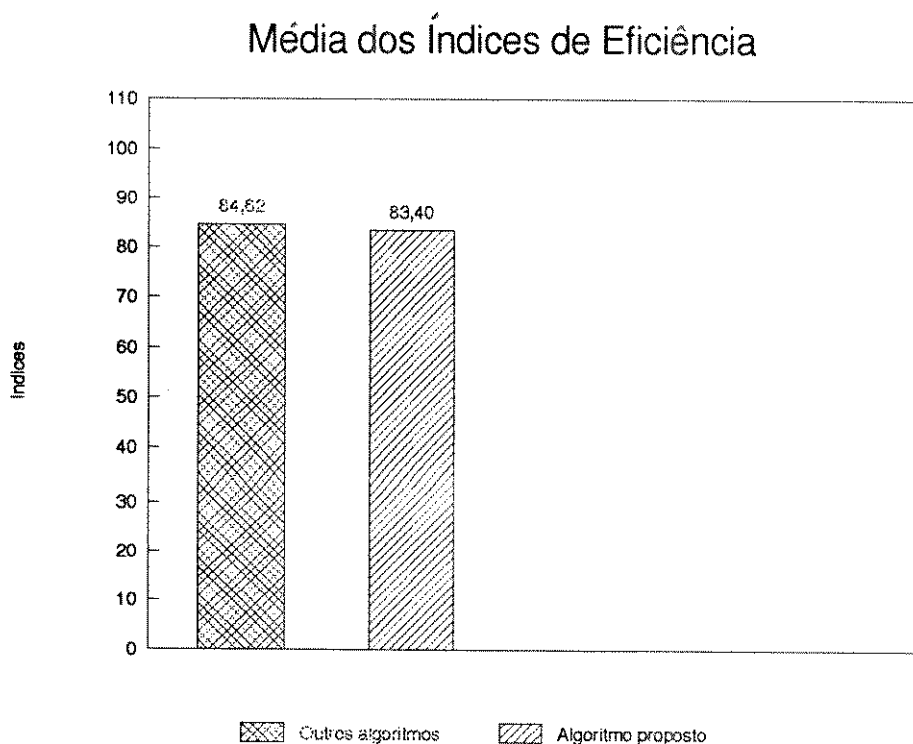


Gráfico 4.5. Valores médios dos índices de eficiência.

A diferença percentual entre os valores médios do índice de eficiência é:

$$D_1 = \frac{84,62 - 83,4}{84,62} \times 100 = 1,44 \%$$

Dada a proximidade dos valores médios, aplica-se o teste estatístico t-pareado (Hines & Montgomery, 1990) para obter um novo parâmetro de comparação. Por este teste obteve-se os seguintes resultados:

nível	tcalculado	tcritico
5%	0,822	2,131

Como $t_{calculado} < t_{critico}$, a hipótese de que a média das diferenças é igual a zero não é rejeitada a um nível de significância de 5 %, ou seja, o desempenho médio, em termos de eficiência, do método VCCC é equivalente ao dos outros algoritmos.

Média dos Índices de Eficácia

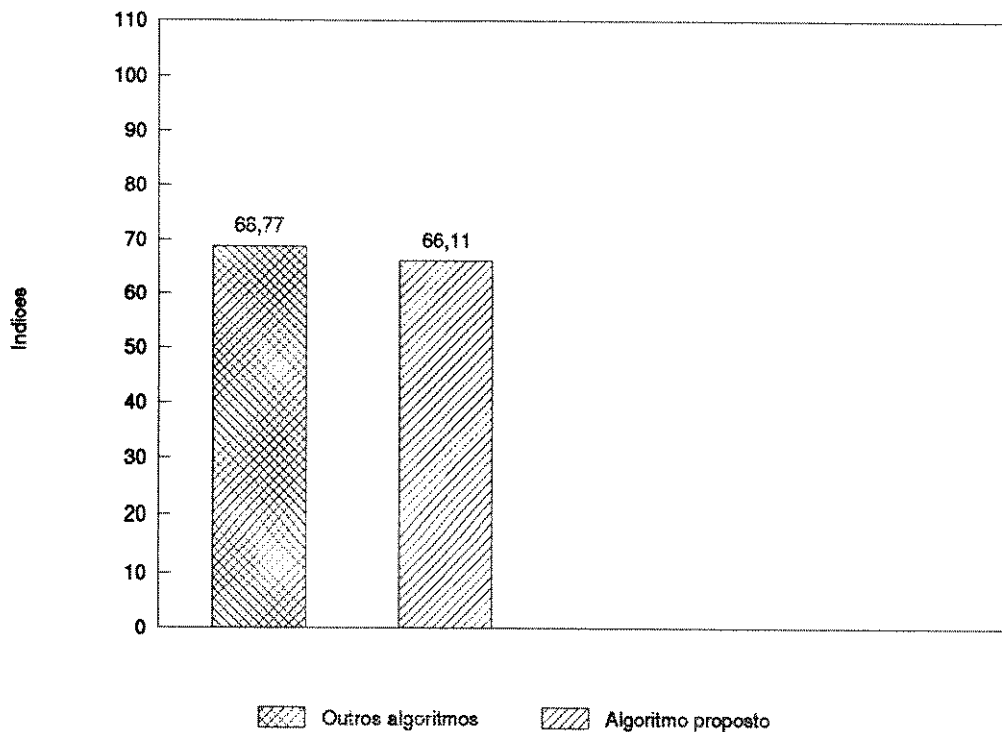


Gráfico 4.6. Valores médios dos índices de eficácia.

A diferença percentual entre os valores médios do índice de eficácia é:

$$D_2 = \frac{68,77 - 66,11}{68,77} \times 100 = 3,87 \%$$

Aplicando o teste estatístico t-pareado obteve-se os seguintes resultados:

nível	tcalculado	tcritico
5%	1,655	2,131

Neste caso, também observa-se que o $t_{calculado} < t_{critico}$ para o nível de significância especificado. Baseado nisto conclui-se que o método VCCC, em termos de eficácia, possui um desempenho médio equivalente ao dos outros algoritmos.

Capítulo 5

COMENTÁRIOS FINAIS

O objetivo desta tese foi propor um método heurístico capaz de formar células de máquinas associadas a famílias de peças. Este método foi implementado e vários exemplos da literatura foram testados. Deste trabalho algumas observações devem ser destacadas.

O primeiro ponto a ser ressaltado é o tipo de informação tratado pelo método. A maioria das técnicas da literatura lidam com matrizes binárias, enquanto que o algoritmo proposto não impõem nenhuma restrição quanto a lidar com números inteiros positivos. Estes números podem representar tanto o tempo de processamento, como tratado neste trabalho, quanto outros parâmetros úteis como volume de produção de cada peça.

Vale a pena citar que este mesmo algoritmo pode ser aplicado a Sistemas Flexíveis de Manufatura (FMS). Um FMS consiste de um conjunto de máquinas com controle computadorizado capazes de realizar um número de diferentes operações, interligadas por um sistema automatizado de manuseio de material. As peças são carregadas em plataformas numa estação de carregamento, e então seguem numa sequência de operações nas máquinas até serem finalmente descarregadas. A versatilidade das máquinas e a possibilidade de ter várias alternativas de roteamento fazem do planejamento da produção para FMS uma tarefa mais difícil do que para as linhas de produção tradicionais. O processamento de peças requer diversas ferramentas, e uma ferramenta normalmente pode ser usada em diversas máquinas. Porém uma ferramenta só pode ser designada para uma máquina de cada vez. Neste sistema, a quantidade de ferramentas que pode ser acoplada em cada máquina é um grande fator restritivo. Porém, se as ferramentas forem trocadas enquanto as máquinas estão operando esta restrição é sensivelmente reduzida.

Para simplificar este problema realiza-se um agrupamento de peças e ferramentas, onde cada peça pertencerá a uma família e cada família terá um conjunto de ferramentas designado para seu processamento. Gerenciar uma família de peças e seu conjunto de ferramentas é certamente uma tarefa mais fácil do que lidar com todo o sistema. Dado as características do problema, o algoritmo proposto pode ser diretamente utilizado para este agrupamento.

Pesquisas futuras no sentido de melhorar a eficiência deste método e do estudo de técnicas para implantação de Tecnologia de Grupo podem ser realizadas. São elas:

- modificações na localização do corte de separação das células. Poder-se-ia desenvolver novos estudos sobre locais de corte como ilustrado no contra exemplo do capítulo 3.
- desenvolvimento de uma nova heurística baseada neste método que tenha habilidade de lidar com o número máximo de elementos numa célula.
- estudos sobre outros possíveis índices de agrupamento que considerem outros parâmetros como o volume de cada tipo de peça, o tempo de processamento e o roteiro de fabricação. Existem algumas sugestões de índices como Eficiência Global, Eficiência de Grupo e Eficiência da Tecnologia de Grupo (Harhalakis *et al* 1990).

Apêndice I

Algoritmo MPM

Neste apêndice são apresentados os requisitos básicos para resolução do problema de fluxo máximo através do algoritmo MPM. Este problema se resume em encontrar o fluxo máximo que pode ser transmitido de um nó fonte (s) para um nó destino (t) numa rede capacitada. Na primeira seção este problema é formalmente descrito. A filosofia básica do algoritmo é substituir a rede original por outras mais fáceis de lidar, e através destas encontrar o fluxo máximo. O procedimento para geração destas redes substitutas, conhecidas como redes em camadas, está detalhado na segunda seção. Na terceira, descreve-se os passos do MPM, ilustrados por um exemplo. Na quarta seção, analisa-se algumas características do algoritmo. Este apêndice se baseia nas informações apresentadas por Chen (1990); algumas modificações foram realizadas para melhor entendimento.

1. O Problema de Fluxo Máximo

Considere uma rede direcionada representada pela quádrupla $G(V,E,c,f)$. Um conjunto de fluxos $\{ f(i,j) \}$ associados com E é factível em $G(V,E,c,f)$, e de valor f_{st} , do nó origem s ao nó destino t , se satisfaz as seguintes restrições. Para cada $i \in V$:

$$\sum_j f(i,j) - \sum_j f(j,i) = f_{st} \quad i = s$$

$$\sum_j f(i,j) - \sum_j f(j,i) = 0 \quad i \neq s, t$$

$$\sum_j f(i,j) - \sum_j f(j,i) = -f_{st} \quad i = t$$

$$c(i,j) \geq f(i,j) \geq 0 \quad (i,j) \in E$$

A seguir será apresentado o algoritmo para construção das redes em camadas. Este algoritmo será utilizado em conjunto com o algoritmo MPM, para encontrar o valor máximo de f_{st} respeitando as restrições acima.

2. Rede em Camadas

A geração da rede em camadas é um processo capaz de detectar os caminhos úteis de s (nó fonte) até t (nó destino) utilizando um fluxo f como guia, onde caminho útil é aquele que é composto de arcos que possuam capacidade de transmitir fluxo. Estes arcos são denominados arcos úteis. Se (i,j) é um arco útil tem-se que (i,j) é um arco não saturado, $f(i,j) < c(i,j)$, ou (j,i) é um arco com fluxo não nulo, $f(j,i) > 0$.

Apresenta-se a seguir um algoritmo para formação de uma rede em camadas $N(V',E',c',f')$ a partir de uma rede $G(V,E,c,f)$ de n nós.

■ Algoritmo para Formação de Redes em Camadas

Seja $U_i \subset V$ e $V_i \subset U_i$ para $i = 0,1,2,\dots,k \leq n-1$.

Passo 1: $U_0 = \{s\}$. Faça $i = 0$.

Passo 2: $U_{i+1} = \left\{ \begin{array}{l} y \mid (x,y) \text{ ou } (y,x) \in E, x \in U_i, y \notin U_0 \cup U_1 \cup \dots \cup U_i \\ \text{e } c(x,y) - f(x,y) + f(y,x) > 0 \text{ (arco útil)} \end{array} \right\}$

Passo 3: Se $t \in U_{i+1}$, faça $k = i+1$ e pare. Vá para o passo 5.

Passo 4: Faça $i = i+1$. Retorne ao passo 2.

Passo 5: $V_k = \{t\}$. Faça $j = k$.

Passo 6: $V_{j-1} = \left\{ \begin{array}{l} x \mid (x,y) \text{ ou } (y,x) \in E, y \in V_j, x \in U_{j-1} \\ \text{e } c(x,y) - f(x,y) + f(y,x) > 0 \end{array} \right\}$

Passo 7: Se $j = 1$, pare.

Passo 8: Faça $j = j-1$. Retorne ao passo 6.

A rede em camadas $N(V',E',c',f')$ é a rede definida pelos conjuntos:

$$V' = V_0 \cup V_1 \cup \dots \cup V_k;$$

$$E' = \left\{ \begin{array}{l} (x,y) \mid x \in V_j, y \in V_{j+1}, \text{ para } j=0,1,\dots,k-1, (x,y) \text{ ou } (y,x) \in E \\ \text{e } c(x,y) - f(x,y) + f(y,x) > 0 \end{array} \right\};$$

$$c'(x,y) = c(x,y) - f(x,y) + f(y,x), (x,y) \in E';$$

e f' é zero.

Observe que na rede em camadas a capacidade de seus arcos, $c'(x,y)$, representa a capacidade residual dos arcos em ambas as direções entre x e y . Outro aspecto interessante desta rede é que, em contraste com uma rede geral, cada caminho do nó origem s a um nó fixo t na camada k contém exatamente k arcos.

■ Exemplo de Construção de uma Rede em Camadas

Seja a rede original $G(V,E,c,f)$ da figura 1, onde observa-se a representação da capacidade e fluxo de cada arco. A partir do fluxo $f = 0$, a rede em camadas $N1(V',E',c',f')$ será construída (figura 2).

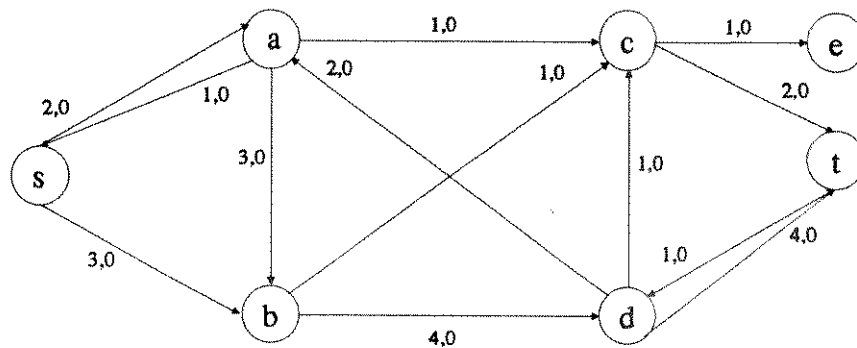


Figura 1. Rede original $G (V, E, c, f)$.

Passo 1: $U_0 = \{ s \}$, $i = 0$

Passo 2: $U_1 = ?$ Os nós a e b estão em U_1 , pois (s,a) e (s,b) são arcos úteis.

$$c'(s,a) = c(s,a) - f(s,a) + f(a,s) = 2 - 0 + 0 > 0$$

$$c'(s,b) = c(s,b) - f(s,b) + f(b,s) = 3 - 0 + 0 > 0$$

$$U_1 = \{ a, b \}$$

Passo 3: $t \in U_1$? Não

Passo 4: $i = 1$

Passo 2: $U_2 = ?$

$$c'(a,c) = c(a,c) - f(a,c) + f(c,a) = 1 > 0$$

$$c'(b,c) = c(b,c) - f(b,c) + f(c,b) = 1 > 0$$

$$c'(b,d) = c(b,d) - f(b,d) + f(d,b) = 4 > 0$$

$$U_2 = \{ c, d \}$$

Passo 3: $t \in U_2$? Não

Passo 4: $i = 2$

Passo 2: $U_3 = ?$

$$c'(c,t) = c(c,t) - f(c,t) + f(t,c) = 2 > 0$$

$$c'(c,e) = c(c,e) - f(c,e) + f(e,c) = 1 > 0$$

$$c'(d,t) = c(d,t) - f(d,t) + f(t,d) = 4 > 0$$

$$U_3 = \{ t, e \}$$

Passo 3: $t \in U_3$? Sim. $k = 3$

Passo 5: $V_3 = \{ t \}$, $j = 3$

Passo 6: $V_2 = ?$

$$V_2 = \{ c, d \}$$

Passo 7: $j = 1$? Não

Passo 8: $j = 1$

Passo 6: $V_1 = ?$

$$V_1 = \{ a, b \}$$

Passo 7: $j = 1$? Sim, pare.

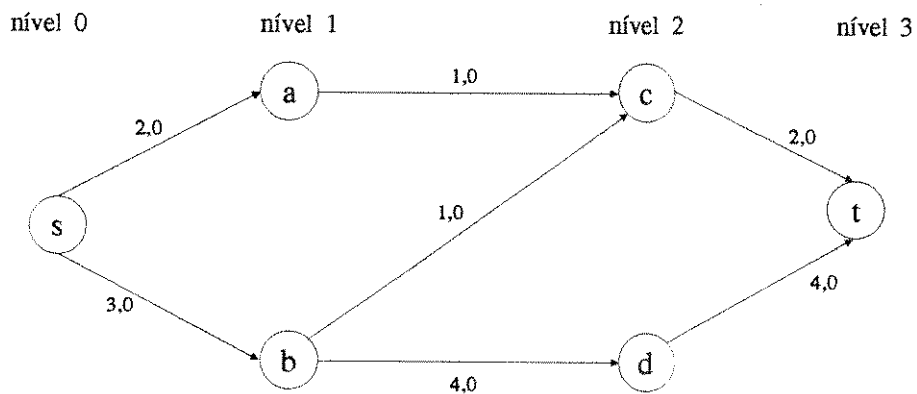


Figura 2. Rede em Camadas N1 (V', E', c', f') com fluxo zero.

3. Fases do Algoritmo

Nesta seção será apresentado como utilizar o algoritmo MPM para encontrar o fluxo máximo numa dada rede $G(V,E,c,f)$. Inicia-se o procedimento com a construção da rede em camadas $N1$ utilizando o fluxo zero como guia. Dado que a rede em camadas foi construída, busca-se encontrar o fluxo bloqueador desta rede. O fluxo f' é considerado bloqueador numa rede em camadas, se todo caminho entre s (fonte) e t (destino) contiver pelo menos um arco saturado ($f'(i,j)=c'(i,j)$). Encontrado o fluxo bloqueador f' , aumenta-se o fluxo da rede original G do valor de f' . Com o novo fluxo f^* em G como guia, constroem-se uma nova rede em camadas $N2$, encontra-se o fluxo bloqueador e aumenta-se o fluxo em G . O conjunto de etapas, constituído de construção da rede em camadas, busca do fluxo bloqueador e aumento do fluxo em G , é conhecido como uma fase do algoritmo para encontrar o fluxo máximo. O procedimento continua executando as fases até que não mais se possa obter uma rede em camadas. Neste estágio o fluxo em G é máximo. Este procedimento, foi elaborado por Malhortra, Pramodh-Kumar e Maheshwari (1978).

Descreveu-se na seção anterior como construir uma rede em camadas $N(V',E',c',f')$ de uma dada rede $G(V,E,c,f)$ usando um fluxo f como guia. Agora apresenta-se como encontrar o fluxo bloqueador em N , através do algoritmo MPM.

O cálculo do fluxo bloqueador se baseia nos potenciais de fluxo dos nós da rede N . A seguir será explicado como obtê-los.

Para cada nó y da rede em camadas $N(V',E',c',f')$, o potencial de entrada de fluxo P_e de y é a soma das capacidades de todos os arcos (x,y) que chegam em y :

$$P_e(y) = \sum_{(x,y) \in E'} c'(x,y)$$

O potencial de saída de fluxo P_s de y é a soma das capacidades de todos os arcos (y,x) que saem de y :

$$P_s(y) = \sum_{(y,x) \in E'} c'(y,x)$$

O potencial de fluxo de y é o mínimo entre o potencial de entrada e o de saída:

$$P(y) = \min [P_e(y), P_s(y)]$$

Em outras palavras, o potencial de fluxo de um nó é o mínimo entre a capacidade total dos arcos que entram e a capacidade total dos arcos que saem do nó. A quantidade máxima de fluxo que pode passar por todos os nós não pode exceder o menor potencial de fluxo dentre os nós com potencial de fluxo não nulo.

■ Algoritmo MPM para Encontrar o Fluxo Bloqueador em N

Passo 1: Encontre um nó $x \neq s, t$ de menor potencial de fluxo em $N(V', E', c', f')$.

$$P_{\min} = \min_j P(j)$$

Passo 2: Envie P_{\min} unidades de fluxo para os arcos de saída $(x, y), (x, y) \in E'$, de N saturando tantos quanto forem possíveis.

Passo 3: Siga o fluxo $f'(x, y) \neq 0, (x, y) \in E'$ para cada y no nível imediatamente superior, e repita o passo 2, enviando $f'(x, y)$ unidades de fluxo através dos arcos de saída de y . Repita este procedimento para o próximo maior nível até que o destino t seja encontrado.

Passo 4: Aloque P_{\min} unidades de fluxo para os arcos de chegada $(y, x), (y, x) \in E'$, de N saturando tantos quanto forem possíveis.

Passo 5: Siga o fluxo $f'(y, x) \neq 0, (y, x) \in E'$, para cada y no nível imediatamente inferior, e repita o passo 4 designando $f'(y, x)$ unidades de fluxo através dos arcos de chegada de y . Repita este procedimento para o próximo menor nível até que a fonte s tenha sido encontrada.

Passo 6: Aumente o fluxo $f(x, y), (x, y) \in E$, de G trocando cada $f(x, y)$ por $f(x, y) + f'(x, y)$. Atualize as capacidades dos arcos da rede N de $c'(x, y)$ para $c'(x, y) - f'(x, y)$.

Passo 7: Remova todos os arcos de capacidade residual zero e todos os nós distintos de s e t , de potencial de fluxo zero, junto com todos os arcos ligados a estes, da rede resultante do passo 6. Se a fonte s e o destino t ainda estão conectados, retorne ao passo 1 e repita todos os passos, senão pare.

O algoritmo completo está ilustrado no exemplo a seguir.

■ Exemplo da Aplicação do MPM

O objetivo deste exemplo é obter o fluxo máximo entre os nós s e t da rede $G(V,E,c,f)$ da figura 1. Assuma inicialmente que o fluxo em G é zero ($f = 0$).

Primeira Fase

A rede em camadas $N1$ de G , com relação a $f = 0$, é mostrada na figura 2. Sua construção está detalhada no exemplo da segunda seção.

Passo 1: Potencial de fluxo dos nós de $N1$ (figura 3):

$$P(a) = \min [P_e(a), P_s(a)] = \min [2,1] = 1$$

$$P(b) = \min [3,5] = 3$$

$$P(c) = \min [2,2] = 2$$

$$P(d) = \min [4,4] = 4$$

$$P_{\min} = \min \{ P(a), P(b), P(c), P(d) \} = P(a) = 1$$

Passo 2: Envie 1 unidade de fluxo através de (a,c) ou $f'(a,c) = 1$

Passo 3: Siga $f'(a,c)$ para o arco (c,t) obtendo $f'(c,t) = 1$

Passo 4: Designe 1 unidade de fluxo para o arco de entrada (s,a) , $f'(s,a) = 1$.

Passo 5: Não é necessário. A figura 3 mostra os novos fluxos obtidos nos passos 2, 3 e 4.

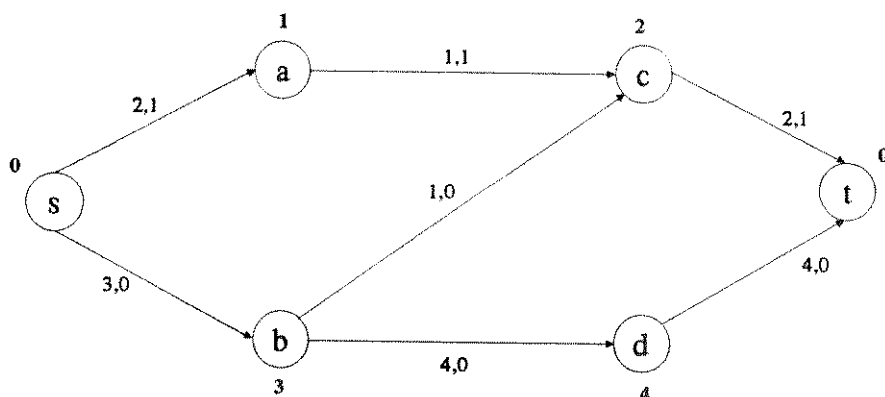


Figura 3. Rede em camadas N1 (V', E', c', f') com os novos fluxos. Os potenciais de fluxo estão indicados pelos números adjacentes aos nós.

Passo 6: Troque o fluxo f de G para:

$$f(s,a) = 1, \quad f(a,c) = 1, \quad f(c,t) = 1$$

Todos os outros fluxos devem ser mantidos em zero. Atualize as capacidades dos arcos (s,a) , (a,c) , e (c,t) :

$$c'(s,a) = c(s,a) - f(s,a) = 1$$

$$c'(a,c) = 0$$

$$c'(c,t) = 1$$

Passo 7: Remova o nó a e os arcos (s,a) e (a,c) ligados ao nó a . A rede resultante $N1'$ é ilustrada na figura 4.

Passo 1: $P_{\min} = P(c) = 1$

Passo 2: Mande 1 unidade de fluxo através de (c,t) , $f'(c,t) = 1$

Passo 3: Não é necessário

Passo 4: Designe 1 unidade de fluxo para o arco (b,c) , $f'(b,c) = 1$

Passo 5: Designe 1 unidade de fluxo para o arco (s,b) , $f'(s,b) = 1$. A figura 4 mostra os novos fluxos obtidos nos passos 2, 3, 4 e 5.

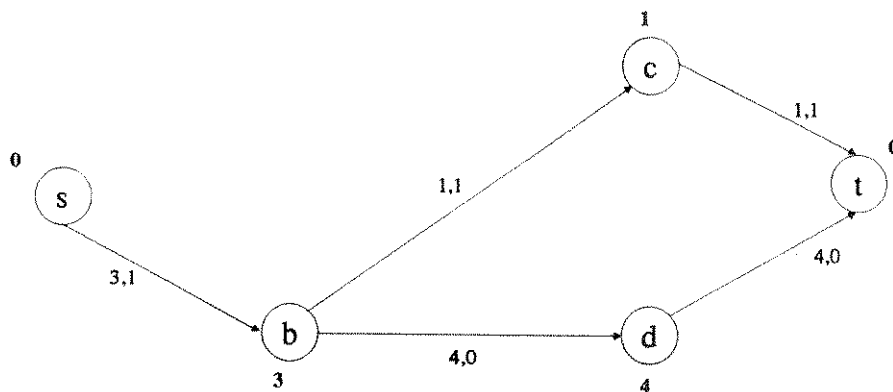


Figura 4. Rede em camadas $N1'$ (V',E',c',f') com novos fluxos.

Passo 6: Aumente o fluxo f de G :

$$f(s,b) = f(s,b) + f'(s,b) = 1$$

$$f(b,c) = 1$$

$$f(c,t) = 2$$

Atualize as capacidades dos arcos:

$$c'(s,b) = c'(s,b) - f'(s,b) = 3 - 1 = 2$$

$$c'(b,c) = 0$$

$$c'(c,t) = 0$$

Passo 7: Remova o nó c e os arcos (b,c) e (c,t) ligados ao nó c . A rede resultante $N1''$ é mostrada na figura 5.

Passo 1: $P_{\min} = P(b) = 2$

Passo 2: Envie 2 unidades de fluxo através de (b,d) , $f'(b,d) = 2$

Passo 3: Envie 2 unidades de fluxo através de (d,t) , $f'(d,t) = 2$

Passo 4: Envie 2 unidades de fluxo através de (s,b) , $f'(s,b) = 2$

Passo 5: Não é necessário. A figura 5 mostra os novos fluxos obtidos nos passos 2, 3 e 4.

Passo 6: Aumente o fluxo f de G :

$$f(s,b) = f(s,b) + f'(s,b) = 3$$

$$f(b,d) = 2$$

$$f(d,t) = 2$$

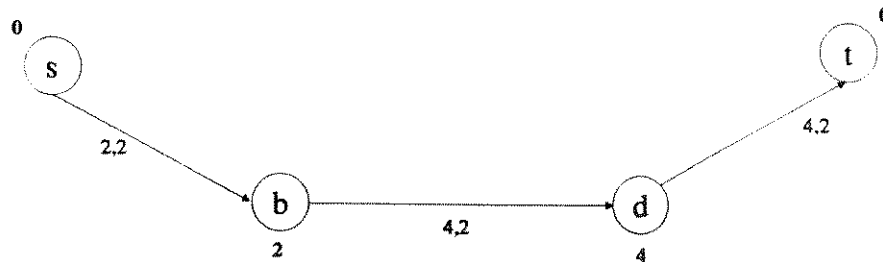


Figura 5: Rede em camadas \$N1''\$ (\$V',E',c',f'\$) com novos fluxos.

Atualize as capacidades dos arcos: $c'(s,b) = c'(s,b) - f'(s,b) = 0$

$$c'(b,d) = 2$$

$$c'(d,t) = 2$$

Passo 7: Remova o nó \$b\$ e os arcos \$(s,b)\$ e \$(b,d)\$. Não é possível construir uma nova rede em camadas. Observe que todo caminho direto de \$s\$ a \$t\$, da rede em camadas \$N1\$ (figura 6), contém pelo menos um arco saturado. Logo o fluxo de bloqueio em \$N1\$ foi encontrado. Pare.

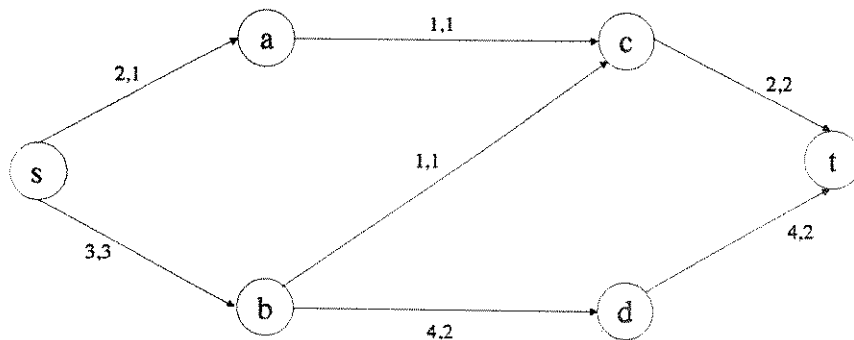


Figura 6. Rede em camadas \$N1\$ com o fluxo de bloqueio obtido com superposição dos fluxos de \$N1'\$ e \$N1''\$.

Isto completa a primeira fase das operações para encontrar o fluxo máximo em \$G\$ utilizando redes em camadas. Para iniciar a fase 2, a rede em camadas \$N2\$ é construída.

Esta nova rede em camadas tem como guia o fluxo da rede $G(V,E,c,f^*)$ da figura 7, obtido na fase 1.

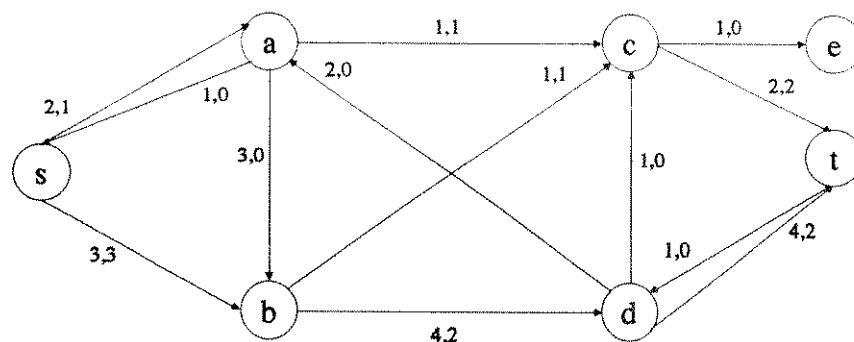


Figura 7. Rede G com os novos fluxos obtidos da fase 1.

Segunda Fase

A rede em camadas $N2$ está ilustrada na figura 8. O algoritmo para sua construção está detalhado na segunda seção.

Passo 1: Potencial de fluxo dos nós de $N2$:

$$P(a) = \min [P_e(a), P_s(a)] = \min [1, 3] = 1$$

$$P(b) = \min [3, 2] = 2$$

$$P(d) = \min [2, 2] = 2$$

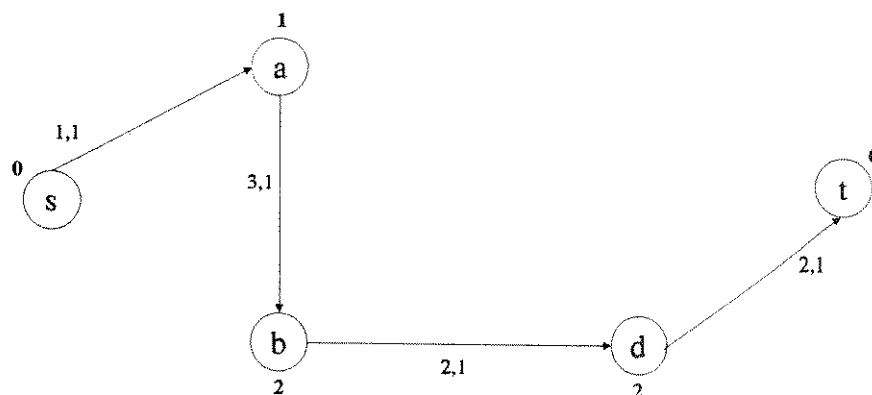


Figura 8. Rede em camadas $N2$ com os novos fluxos.

$$P_{\min} = \min \{ P(a), P(b), P(d) \} = P(a) = 1$$

Passo 2: Envie 1 unidades de fluxo através de (a,b), $f'(a,b) = 1$

Passo 3: Envie 1 unidades de fluxo através de (b,d) e (d,t), $f'(b,d) = 1$ e

$$f'(d,t) = 1$$

Passo 4: Envie 1 unidades de fluxo através de (s,a), $f'(s,a) = 1$

Passo 5: Não é necessário

Passo 6: Aumente o fluxo f de G :

$$f(a,b) = f(a,b) + f'(a,b) = 1$$

$$f(b,d) = 3$$

$$f(d,t) = 3$$

$$f(s,a) = 2$$

Atualize as capacidades dos arcos:

$$c'(a,b) = c(a,b) - f(a,b) = 3 - 1 = 2$$

$$c'(b,d) = 1$$

$$c'(d,t) = 1$$

$$c'(s,a) = 0$$

Passo 7: Remova o nó a e os arcos (s,a) e (a,b) ligados ao nó a da rede $N2$. Com a remoção deste nó, completa-se a segunda fase e o fluxo bloqueador em $N2$ é o ilustrado na figura 8. Iniciar-se-ia então a terceira fase, porém, não é possível contruir uma nova rede em camadas, pois não existe nenhum caminho de s a t composto de arcos úteis. Logo o fluxo atual de G é máximo e tem valor 5 (figura 9).

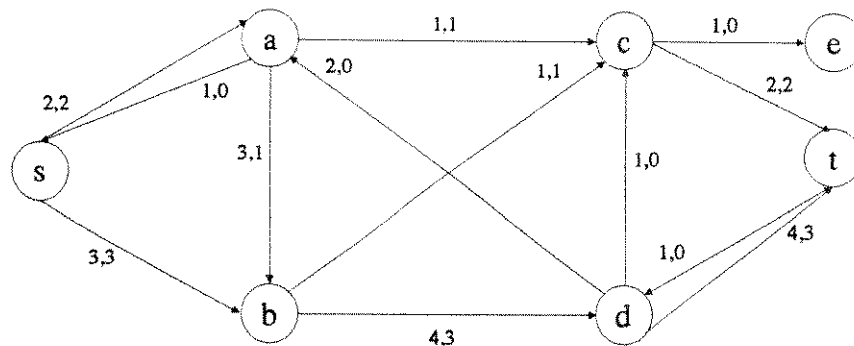


Figura 9. Rede original G com fluxo máximo entre s e t .

4. Características do Algoritmo

Esta seção tem por finalidade expor algumas características básicas do algoritmo.

■ Complexidade do Algoritmo

Inicialmente analisa-se o número máximo de redes em camadas que serão construídas antes de se obter o fluxo máximo. Isto será derivado a partir do teorema sobre as alturas das redes em camadas, exposto a seguir. Altura de uma rede em camadas significa o número de arcos em qualquer caminho de s a t em N , ou seja, o nível máximo atingido pela rede. Por exemplo, a rede da figura 2 possui altura 3.

Teorema: *As alturas das redes em camadas que ocorrem nas consecutivas fases de resolução do problema de fluxo máximo formam uma seqüência estritamente crescente de inteiros positivos.* (Chen 1990)

Considerando o teorema apresentado acima, e o fato de que cada nível possui pelo menos um nó, conclui-se que o número máximo de redes em camadas e conseqüentemente de fases possíveis na resolução do problema de fluxo máximo é $n-1$.

Agora será estimado o número de operações necessárias para o estabelecimento do fluxo de bloqueio através do algoritmo MPM.

Primeiro, para encontrar um nó, diferente de s e t ($P(s) = P(t) = 0$), com potencial de fluxo mínimo (passo 1) numa rede em camadas com n nós, na primeira vez, necessita-se de $n-3$ comparações, na segunda, no máximo $n-4$ comparações. Logo haverá no máximo $(n-3) + (n-4) + (n-5) + \dots + 1 = (n-2).(n-3)/2$ comparações. Nos passos 2 a 6, desde que cada arco só pode ser saturado uma vez, o tempo para saturar todos os arcos de G é no máximo $O(m)$, onde m é o número de arcos da rede. Para cada nó de potencial mínimo, visita-se no máximo $n-1$ outros nós e existem no máximo n nós de potencial mínimo. Logo o trabalho total requerido para atualizar a capacidade dos arcos é no máximo $O(n^2)$. Como resultado, o trabalho necessário para o algoritmo MPM encontrar o fluxo de bloqueio numa rede em camadas é:

$$O(n^2) + O(m) + O(n^2) = O(n^2), \text{ onde } m < n^2$$

Dado que existem no máximo $n-1$ redes em camadas derivadas da rede original G , para encontrar o fluxo bloqueador, o algoritmo MPM será aplicado no máximo $n-1$ vezes, até que o fluxo máximo seja encontrado. Então, este algoritmo para encontrar o fluxo máximo numa rede tem complexidade $O(n^3)$ e depende somente do número de nós da rede.

■ Aplicação do Teorema do Fluxo Máximo - Corte Mínimo

Nas seções anteriores mostrou-se como resolver o problema de fluxo máximo. Agora será apresentado como, com este fluxo, obtém-se o local de corte mínimo. Inicialmente é necessário apresentar o seguinte teorema e seu corolário (Chen 1990).

Teorema do Fluxo Máximo - Mínimo Corte: O valor do fluxo máximo em uma rede G é igual à capacidade do corte mínimo de G .

Corolário: Um corte $s-t$ (W, \overline{W}) é mínimo se e somente se para o fluxo máximo f :

(i) *todo arco $e \in (W, \overline{W})$ estiver saturado, e*

(ii) *todo arco $e \in (\overline{W}, W)$ satisfizer $f(e) = 0$*

No algoritmo descrito neste apêndice, o fluxo máximo f^* de uma rede direcionada G é encontrado quando não é possível construir uma nova rede em camadas utilizando f^* como guia. Esta tentativa de se obter esta rede falha devido a existência de um conjunto bloqueio de nós A . Estes nós estão em todo caminho que sai do nó s , e possuem as seguintes características,

$$c(x,y) - f(x,y) = 0 \quad \text{para todo } (x,y) \in E \text{ e } x \in A,$$

$$f(x,y) = 0 \quad \text{para todo } (y,x) \in E \text{ e } x \in A,$$

ou seja, não são arcos úteis.

O conjunto bloqueio A , devido a suas características básicas, separa o nó s do nó t . Obviamente $t \notin A$, senão esta seria uma rede em camadas útil. Então, determina-se dois conjuntos de nós, W e \overline{W} , onde W é composto dos nós de s até o conjunto bloqueio A , e $\overline{W} = V - W$ com $t \in \overline{W}$. Observa-se através das características dos arcos que compõem o corte (W, \overline{W}) , que este satisfaz as restrições impostas pelo corolário para ser o corte mínimo da rede G . Logo na última tentativa de construção da rede em camadas também se localiza o corte mínimo.

Exemplificando esta aplicação do teorema, apresenta-se na figura 10 o corte mínimo obtido na construção da última rede em camadas do exemplo da terceira seção, onde o conjunto bloqueio $A = \{ s \}$.

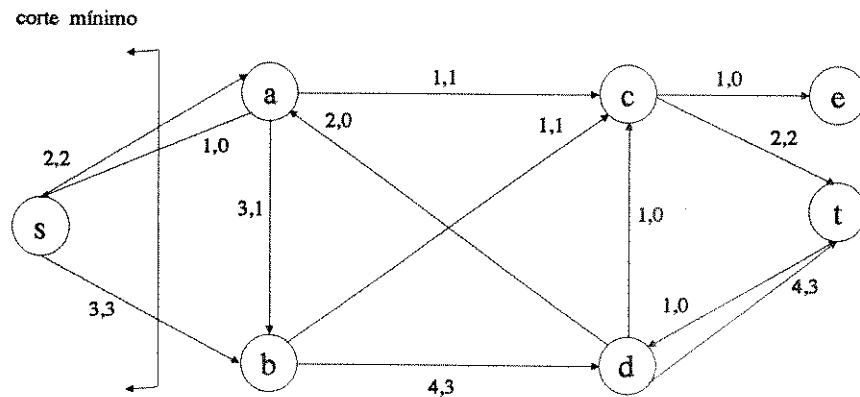


Figura 10. Rede original G com corte mínimo entre s e t .

Apêndice II

Estruturas de Dados da Implementação

Uma estrutura é uma coleção de uma ou mais variáveis, possivelmente de tipos diferentes, colocadas juntas sob um único nome para manipulação conveniente. As estruturas ajudam a organizar dados complicados, particularmente em grandes programas, pois permitem que um grupo de variáveis relacionadas sejam tratadas como uma unidade ao invés de entidades separadas.

A escolha da estrutura é uma das mais importantes decisões na implementação de um algoritmo. Para os mesmos dados, podem ser definidos diferentes estruturas, que requerem menos ou mais espaço e levam a algoritmos menos ou mais eficientes.

Para a implementação do algoritmo proposto, a estrutura que melhor se adequou foi a estrutura de lista ligada. Maiores detalhes sobre esta estrutura serão dados a seguir.

■ Lista Ligada

Uma das vantagens primárias da lista ligada é que estas não necessitam de uma definição prévia de tamanho máximo. A segunda vantagem é que estas acrescentam flexibilidade nos algoritmos. Esta flexibilidade é garantida pelo rápido acesso aos itens da lista em determinadas situações. Isto ficará mais aparente ao se exemplificar algumas propriedades e operações básicas da lista ligada.

Uma lista ligada é um conjunto de itens organizados seqüencialmente, tal que cada elemento (nó) contenha uma ligação para o próximo elemento (nó), conforme figura 1.

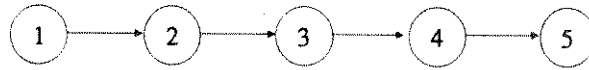


Figura 1. Exemplo de lista ligada.

Nesta representação (figura 1) observa-se que cada elemento possui uma ligação para o próximo nó, logo o último nó deve especificar algum próximo nó. É definido então, um nó nulo denominado z. O último nó é ligado ao nó z e este é ligado a ele mesmo. Deve existir também um nó que indique o início da lista. Este nó é o cabeçalho da lista (figura 2). O objetivo destes nós é facilitar a manipulação dos nós na lista.

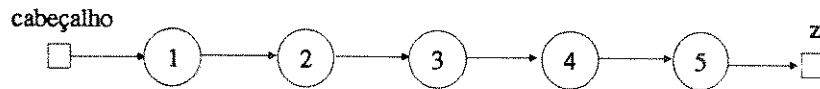


Figura 2. Lista ligada com nós auxiliares.

Suponha, por exemplo, que se queira deslocar o nó 5 para o início da lista. Conforme figuras 3.a e 3.b, somente três ligações devem ser trocadas, independente do tamanho da lista.

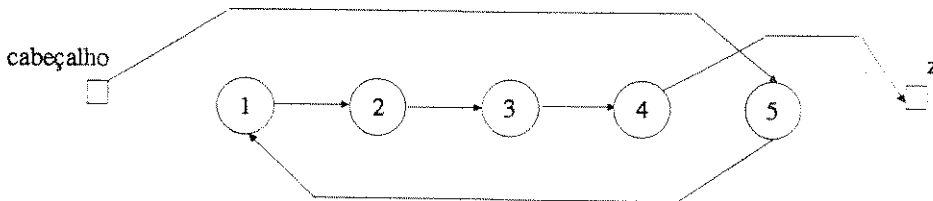


Figura 3.a. Operação de troca na lista ligada.

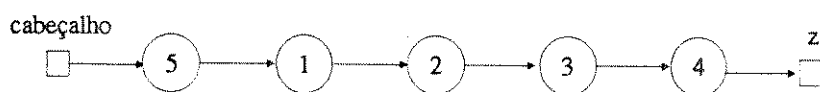


Figura 3.b. Novo arranjo da lista depois da troca.

Outras duas operações são realizadas com simplicidade nesta estrutura: inserção e retirada de elementos. Para a operação de inserção do nó 10 na lista da figura 2, somente duas ligações são trocadas (figura 4). Observa-se que depois da operação de retirada deste nó, este ainda existe, porém não pode mais ser acessado seguindo as ligações que se iniciam no cabeçalho da lista.

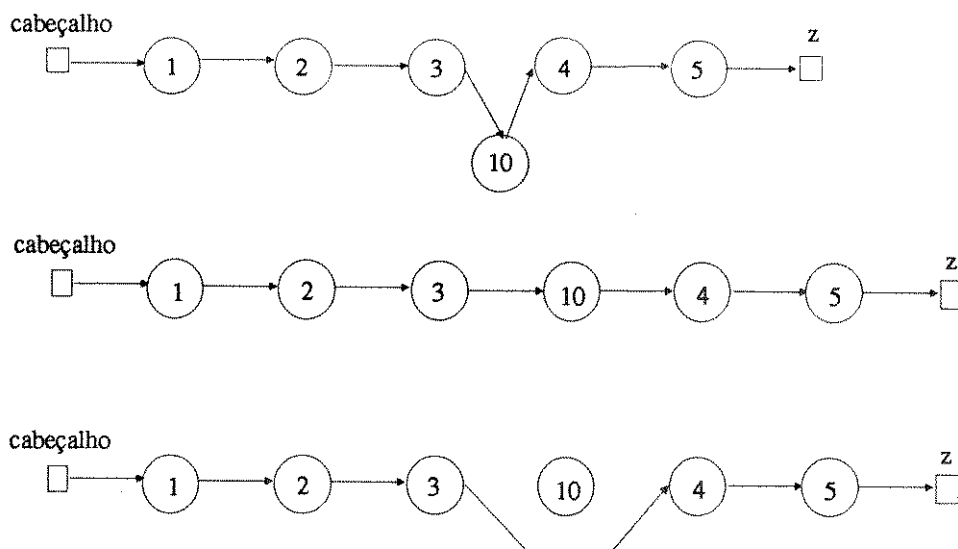


Figura 4. Operação de inserção e retirada de um elemento.

Por outro lado, existem operações em que as listas ligadas não são tão bem sucedidas. A mais óbvia delas, é a de encontrar o k-ésimo elemento de uma lista. Somente realizando um caminho através de k ligações da lista obtém-se este elemento.

Outra operação que não é realizada facilmente é a de encontrar o elemento antes de um outro dado elemento. Por exemplo, se o nó antes do nó 5 tiver que ser eliminado, será preciso transitar pela lista até encontrar o nó 5, e então eliminar o nó 4.

■ Estrutura de Lista Ligada para Grafos

Descreve-se nesta seção como utiliza-se a estrutura de lista ligada para grafos. Na figura 5 ilustra-se um grafo que será representado através desta estrutura. Associado a cada aresta tem-se a sua capacidade e seu fluxo.

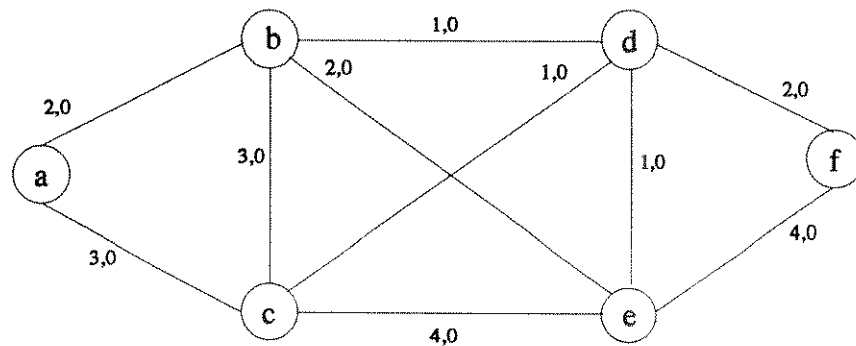


Figura 5. Grafo original.

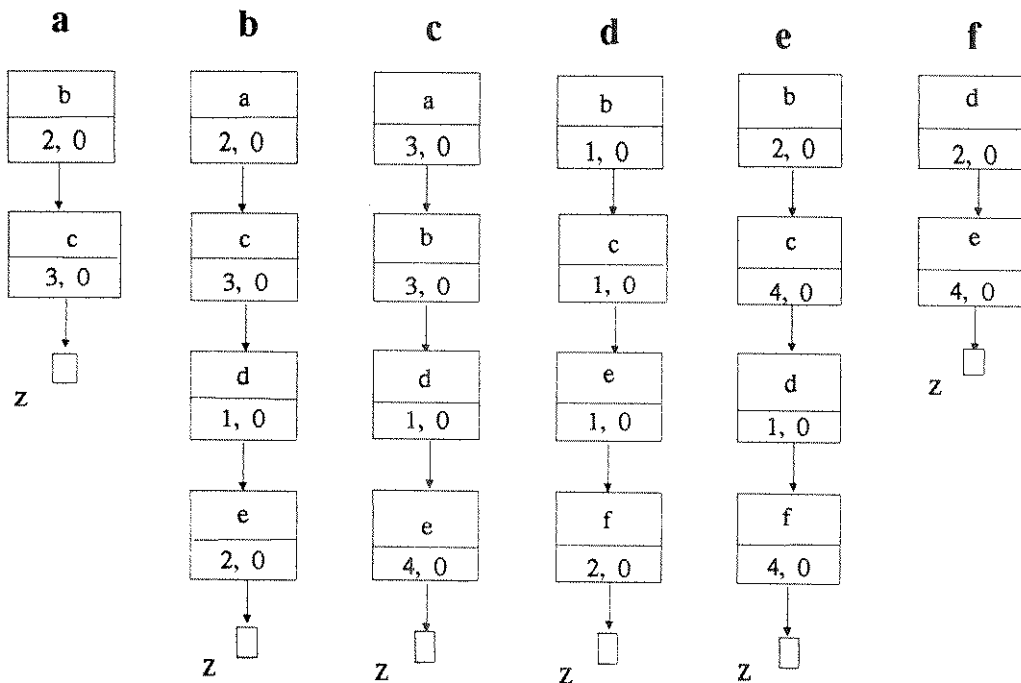


Figura 6. Representação do grafo por lista ligada.

Note na figura 6 que cada aresta é duplamente representada, ou seja, uma aresta que conecta x e y é representada na lista de x e y. É importante que a aresta tenha

esta dupla representação, senão uma simples pergunta como “Quais os nós que estão diretamente conectados ao nó x?”, não poderia ser facilmente respondida.

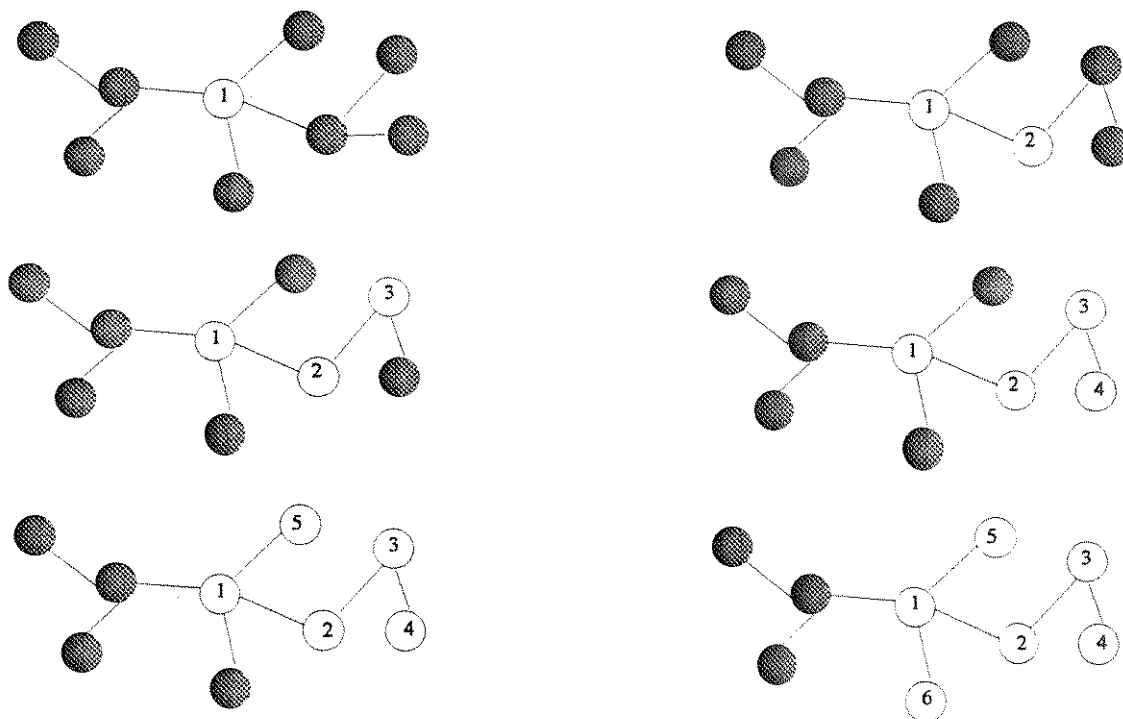
Observe que a cada nó está associado não só o seu nome, como também a capacidade e o fluxo da aresta que interliga este nó ao nó que nomeia a lista. Estas informações são básicas para a representação do grafo.

■ Buscas no Grafo

Na implementação do algoritmo proposto existem várias questões que só podem ser respondidas através de visitas aos vários nós do grafo. Estas questões podem ser respondidas de forma eficiente utilizando técnicas conhecidas como busca em profundidade e busca em largura.

■ Busca em Profundidade

Na figura 7 pode-se acompanhar o processo de busca em profundidade num grafo. Inicia-se o processo do nó 1.



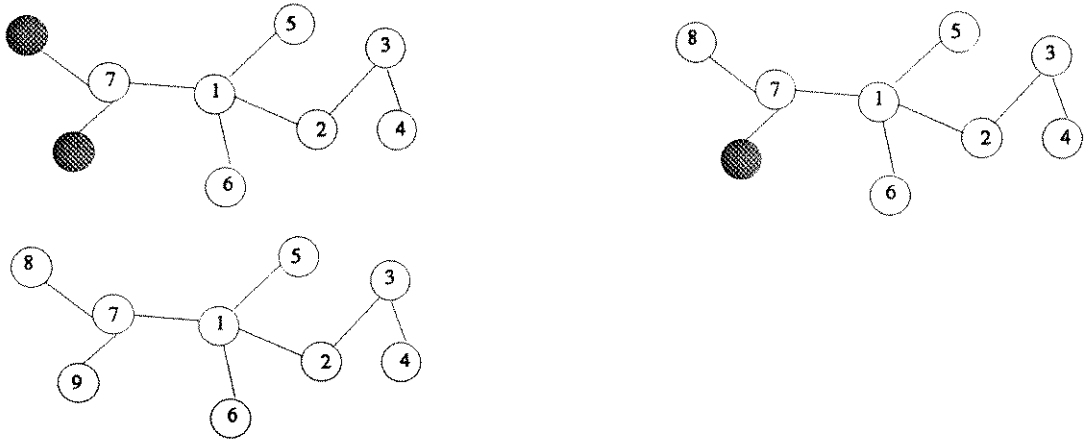
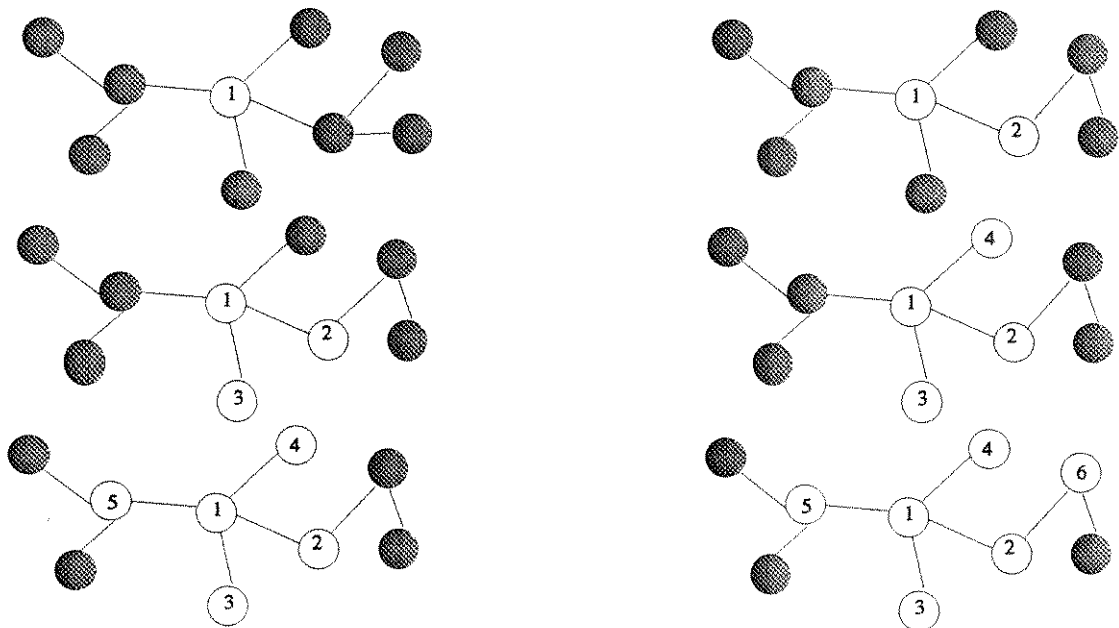


Figura 7. Seqüência de visitas numa busca em profundidade.

A realização desta busca num grafo representado por listas ligadas requer um tempo proporcional a $V+E$, onde V são os nós do grafo e E os arcos. Este tipo de busca foi utilizada em várias rotinas da implementação. Dentre elas, a formação das árvores condensadas e a contagem dos elementos de cada grupo. Caso a estrutura escolhida fosse a matricial, esta mesma busca acarretaria um tempo proporcional a V^2 .

■ Busca em Largura

Na figura 8 acompanha-se uma busca em largura num grafo. Inicia-se o processo do nó 1.



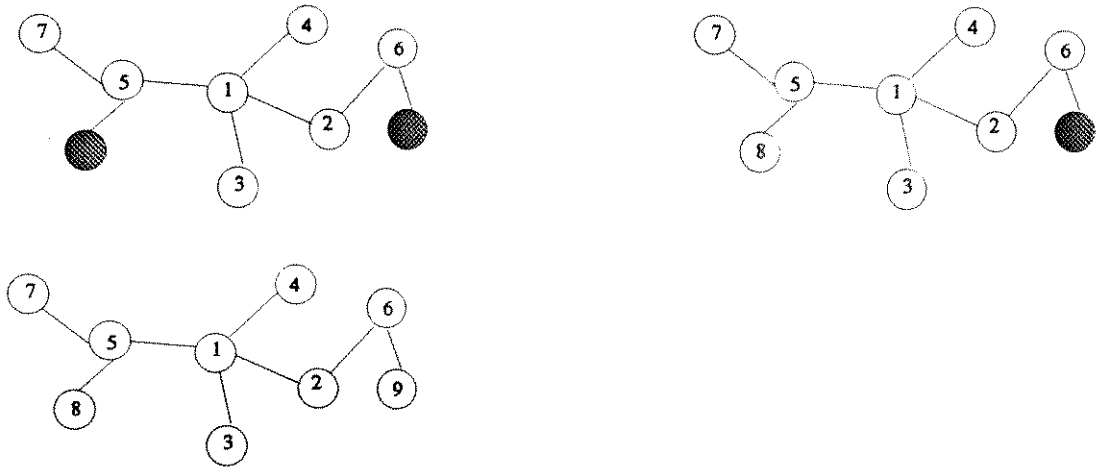


Figura 8. Sequência de visitas numa busca em largura.

Este tipo de busca foi utilizado no algoritmo MPM para encontrar o fluxo máximo entre dois nó de um grafo direcionado (apêndice I).

As informações contidas neste apêndice têm como referência Sedgewick (1990).

Bibliografia

- Agostinho, O.L., 1991, *Sistemas de Manufatura*, UNICAMP, Campinas, SP.
- Baase, S., 1989, *Computer Algorithms: Introduction to Design and Analysis*, Addison Wesley, Reading Massachusetts.
- Burbidge, J.L., 1975, *The Introduction of Group Technology*, John Wiley & Sons, New York.
- Chandrasekaran, M.P. e Rajagopalan, R., 1986, *An Ideal Seed Non-Hierarchical Clustering Algorithm for Cellular Manufacturing*, International Journal of Production Research, vol. 24, no. 2, pp. 451-464.
- Chandrasekaran, M.P. e Rajagopalan, R., 1987, *ZODIAC - An Algorithm for Concurrent Formation of Part-Families and Machine-Cells*, International Journal of Production Research, vol. 25, no. 6, pp. 835-850.
- Chandrasekaran, M.P. e Rajagopalan, R., 1989, *GROUPABILITY : An Analysis of Properties of Binary Data Matrices for Group Tecnology*, International Journal Production of Research, vol. 25, no. 6, pp. 835-850.
- Chen, W., 1990, *Theory of Nets: Flows in Networks*, John Wiley & Sons, New York.
- Chu, C.H. e Tsai, M., 1990, *A Comparison of Three Array-Based Clustering Techniques for Manufacturing Cell Formation*, International Journal of Production Research, vol. 28, no. 8, pp. 1417-1433.
- Franco Jr., H., Chacon, P.P., 1980, *História Econômica Geral e do Brasil*, Atlas, São Paulo.

Garey, M.R., Johnson, D.S., 1979, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W.H. Freeman and Company, New York.

Gomory, R.E. e Hu, T.C., 1961, *Multi-Terminal Network Flows*, J. Soc. Indust. Appl. Math., vol.9, no. 4, pp. 551-570.

Gupta, T. e Seifoddini, H., 1990, *Production Data Based Similarity Coefficient for Machine-Component Grouping Decisions in the Design of a Cellular Manufacturing System*, International Journal of Production Research, vol. 28, no. 7, pp. 1247-1269.

Harhalakis, G., Nagi, R. e Proth, J.M., 1990, *An Efficient Heuristic in Manufacturing Cell Formation for Group Technology Applications*, International Journal of Production Research, vol. 28, no. 1, pp. 185-198.

Kernighan, B.W. e Lin, S., 1970, *An Efficient Heuristic Procedure for Partitioning Graphs*, The Bell System Technical Journal, vol. 49, pp. 291-307.

King, J.R., 1980, *Machine-Component Grouping in Production Flow Analysis: An Approach Using a Rank Order Clustering Algorithm*, International Journal of Production Research, vol. 18, no. 2, pp. 213-232.

King, J.R. e Nakornchai, V., 1982, *Machine-Component Group Formation in Group Technology: Review and Extension*, International Journal of Production Research, vol. 20, no. 2, pp. 117-133.

Kumar, K.R. e Chandrasekharan, M.P., 1990, *Grouping Efficacy: A Quantitative Criterion for Goodness of Block Diagonal Forms of Binary Matrices in Group Technology*, International Journal of Production Research, vol. 28, no. 2, pp. 233-243.

Kumar, K.R., Kusiak, A., Vannelli, A., 1986, *Grouping of Parts and Components in Flexible Manufacturing Systems*, European Journal of Operational Research, vol. 24, pp. 387-397.

Kusiak, A., 1987, *The Generalized Group Technology Concept*, International Journal of Production Research, vol. 25, no. 4, pp. 561-569.

Malhotra, V.M., Pramodh Kumar, M. e Maheshwari, S.N., 1978, *An $O(|V|^3)$ Algorithm for Finding Maximum Flows in Networks*, Information Processing Letters, vol. 7, no. 6, pp. 277-278.

McAuley, J., 1972, *Machine Grouping for Efficient Production*, Production Engineer, Fevereiro 1972, vol. 51, pp. 53-57.

McCormick, W.T., Schweitzer, P.J. e White, T.W., 1972, *Problem Decomposition and Data Recognition by a Clustering Technique*, Operations Research, vol. 20, no. 5, pp. 993-1009.

Moon, J.W. e Moser, L., 1965, *On Cliques in Graphs*, Israel J. Math., vol.3, no. 23.

Moura, R.A., 1987, *Flexibilidade Total: Homem x Máquina*, IMAM, São Paulo.

Rajagopalan, R. e Batra, J.L., 1975, *Design of Cellular Production Systems: A Graph Theoretic Approach*, International Journal of Production Research, vol. 13, no. 6, pp. 567-579.

Sedgewick, R., 1990, *Algorithms in C*, Addison-Wesley, Reading Massachusetts.

Seifoddini, H., Wolfe, P.M., 1986, *Application of the Similarity Coefficient Method in Group Technology*, Institute of Industrial Engineers Transactions, vol. 18, pp. 271-277.

Seifoddini, H., 1990, *Machine-Component Group Analysis Versus the Similarity Coefficient Method in Cellular Manufacturing Applications*, Computers & Industrial Engineering, vol. 18, no. 3, pp. 333-339.

Sério, L.C., 1990, *Tecnologia de Grupo no Planejamento de um Sistema Produtivo*, Ícone Editora, São Paulo.

Vannelli, A. e Kumar, K.R., 1986, *A Method for Finding Minimal Bottle-Neck Cells for Grouping Part-Machine Families*, International Journal of Production Research, vol. 24, no. 2, pp. 387-400.

Ventura, J.A., Chen, F.F., Wu, C.H., 1990, *Grouping Parts and Tools in Flexible Manufacturing Systems Production Planning*, International Journal of Production Research, vol. 28, no. 6, pp. 1039-1056.

Vohra, T., Chen, D., Chang, J.C. e Chen, H., 1990, *A Network Approach to Cell Formation in Cellular Manufacturing*, International Journal of Production Research, vol. 28, no. 11, pp. 2075-2084.

Waghodekar, P.H. e Sahu, S., 1984, *Machine-Component Cell Formation in Group Technology: MACE*, International Journal of Production Research, vol. 22, no. 6, pp. 937-948.