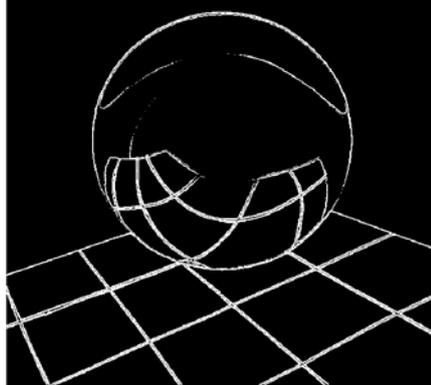




GESTÃO
&
PRODUÇÃO



v.2, n.1 ,p.38-58, abr.1995

UMA ABORDAGEM PRÁTICA PARA A PROGRAMAÇÃO DE SETUP E PARA O DIMENSIONAMENTO DE LOTE EM UMA INDÚSTRIA TÊXTIL

Umit Akinc

School of Business and Accountancy, Wake Forest
University,
Winston-Salem, NC 27109

Tradução de

Paulo dos Santos Ferreira

Reprinted from:

IIE Transactions magazine, March 1993, Copyright 1993,
Institute of Industrial Engineers, 25 Technology Park/Atlanta,
Norcross, Georgia 30092.

Resumo

Estuda-se a programação de preparações de máquinas e atividades produtivas de uma indústria têxtil, localizada na Carolina do Norte, EUA. A firma enfrenta o problema de programar a produção das encomendas dos clientes em diversos teares circulares, que admitem diferentes configurações, mediante instalação de diferentes cilindros, para tecer vários tipos de malha crua. Dado um conjunto de requisitos, para diferentes estilos de malha, o problema consiste em decidir quanto à configuração específica a ser usada em cada máquina e quanto aos específicos pedidos a serem processados nessas configurações. O problema é formulado como um modelo de programação linear inteira. O objetivo é a maximização da contribuição total de todos os pedidos programados sujeitos a restrições impostas à capacidade de produção pelas máquinas e pelas operações de preparação, considerando explicitamente os efeitos dos ajustes programados e das restrições ao atendimento das encomendas dos clientes. São discutidas várias tentativas de solução. Projeta-se um procedimento aproximado que progressivamente adiciona novos ajustes com base em várias heurísticas por meio das quais o "valor" das configurações candidatas às máquinas é avaliado. Essas heurísticas podem ser desenvolvidas tanto para um procedimento interno de programação quanto para uso do programador humano, interativamente, num ambiente de microcomputador.

Palavras-chave: programação da produção, preparação de máquina, carga de máquina, , programação linear.

1. Introdução

Desde os dias do início da era da gestão científica, e do seu pioneiro, Henry Gantt, a programação de operações de manufatura constitui uma área movimentada da atividade de gerência. O estudo desse problema tem atraído, e continua a atrair, o interesse de numerosos acadêmicos, em decorrência, talvez, de sua considerável estrutura quantitativa e da riqueza de suas virtualmente infinitas variações. Como resultado dessa preocupação erudita, acumulou-se ao longo dos anos uma volumosa quantidade de trabalhos de pesquisa, que se estendem desde os clássicos estágio único e dois estágios sequenciais de máquinas (JOHNSON, 1954) até as mais recentes obras sobre a programação de sistemas flexíveis de manufatura (STECKE, 1983). Veja-se a vasta bibliografia apresentada por GRAVES (1981) e por VOLLMANN *et al.* (1988). Por outro lado, aqueles que exercem a gerência adquiriram pleno conhecimento do impacto das práticas de programação da produção sobre critérios de desempenho evidentes tais como os níveis: de estoque, de utilização do equipamento e da mão-de-obra e dos serviços à disposição dos clientes; critérios que, por sua vez, afetam diretamente a decisão final. Talvez seja por isso que uma parcela significativa dos artigos acadêmicos de pesquisa, neste domínio, tenha encontrado, da parte dos dirigentes profissionais, um nível de aceitação que raramente é desfrutado pela pesquisa em outras áreas da ciência da administração.

No seu sentido mais geral, a programação da produção trata da destinação de recursos limitados, para a realização de atividades concorrentes, ao longo de um horizonte de tempo, de modo a satisfazer certos critérios, definidos em termos do dispêndio de tempo ou de dinheiro, ou de ambos. Como tal, ela abrange uma vasta variedade de problemas, que diferem por

ampla gama de aspectos. GRAVES (1981) esta-belece uma estrutura para classificar os problemas de programação segundo algumas de suas propriedades características mais relevantes, citando pesquisas importantes relativas a cada categoria. Considerando a maneira pela qual são geradas as exigências básicas a serem satisfeitas, ele distingue entre fábricas *abertas* e *fechadas*. Numa fábrica aberta aquelas exigências têm sua origem nas específicas encomendas dos clientes, ao passo que numa fábrica fechada as exigências dizem respeito a produtos relativamente padronizados e derivam de decisões quanto à reposição de estoques. Dependendo da complexidade do processo produtivo, e em ordem crescente dessa complexidade, o autor distingue as seguintes categorias de problemas de programação: *um-estágio, uma-instalação; um-estágio, instalações-paralelas; multiestágios, fábrica-fluxo e multiestágios, fábrica-tarefa*. A terceira dimensão da classificação é o tipo de critério empregado para avaliar as programações. Assim, há casos em que as programações são avaliadas a partir de suas conseqüências em termos de *custo monetário*, e outros em que a avaliação se baseia no grau de observância das *datas devidas*, impostas por decisão externa. Embora a maioria dos problemas de programação enfrentados no mundo real não se enquadre precisamente em alguma dessas categorias, a quase totalidade dos trabalhos acadêmicos adota, em linhas mais ou menos gerais, a estrutura delineada por GRAVES (1981). Para um retrospecto em profundidade da literatura sobre programação o leitor poderá consultar GRAVES (1981) e VOLLMANN *et al.* (1988).

Este artigo descreve uma tentativa de modelar e analisar um problema de programação enfrentado por uma pequena indústria de malharia da Carolina do Norte, nos

Estados Unidos, e relata o projeto e a implementação de um sistema de programação baseado em microcomputador. O problema se reveste de algumas características muito peculiares, que impedem, de modo geral, a aplicação direta de alguma das abordagens exaustivamente recapituladas em GRAVES (1981) e VOLLMANN *et al.* (1988), mas deve ter um grau de generalidade suficiente para que a sua solução se torne potencialmente útil, em situações se-

melhantes, quer dentro quer fora da indústria têxtil. A próxima seção apresenta a em-presa e o problema de programação que ela enfrenta. Este é então formalmente exposto como um problema de otimização inteira mista. Seguem-se um esboço da abordagem para uma solução aproximada e uma im-plantação piloto do sistema em um ambiente de microcomputador. A seção final apresenta as conclusões do trabalho.

2. A Empresa e seu Problema de Programação

Adele é uma pequena indústria de malharia, com um contingente de cerca de 100 empregados, formado por homens e mulheres. Produz uma ampla variedade de tecidos de malha crus tubulares, que diferem em largura, estilo de malha, peso e materiais componentes do fio. A empresa mantém um estoque de produtos acabados apenas para alguns poucos destes estilos, aqueles cuja demanda é relativamente estável. Com relação a estes itens Adele tenta, sempre que possível, suprir a demanda a partir do estoque. Pode porém ocorrer, dependendo da posição do estoque, que até mesmo um pedido referente a algum desses itens de maior consumo precise ser total ou parcialmente atendido a partir da produção corrente. O volume da demanda de malha crua origina-se, na sua maior parte, de pedidos específicos dos clientes, e as malhas são tecidas em regime de produção-sob-encomenda. Conseqüentemente, os reflexos da programação sobre o estoque não constituem motivo de grande preocupação, não obstante o fato de que esses estoques são usados como instrumento tático para conseguir um bom nível de utilização do equipamento. A prática usual consiste em programar primeiro as encomendas específicas dos clientes para depois utilizar a eventual capacidade remanescente para a reposição dos estoques daqueles poucos estilos padronizados. Portanto, segundo a classifica-

ção de GRAVES (1981), Adele é uma fábrica que não é nem fechada nem aberta, mas híbrida.

O processo de manufatura de Adele abrange uma série de operações que vão desde o recebimento do fio até a expedição do tecido cru acabado. A tecedura, que é uma das primeiras etapas do processo, constitui o foco do problema de programação, pois é nesse estágio que se adiciona ao produto a maior parte do seu valor. Após a tecedura de um determinado item (estilo de malha) da programação, que envolve possivelmente a operação concomitante de várias máquinas circulares, o produto é reunido em um "lote" único, numa área de espera, para prosseguir com as restantes operações de acabamento.

Este estágio está equipado com 48 teares circulares - que diferem bastante quanto a marcas, modelos e idades - cada um dos quais é capaz de tecer um determinado subconjunto do total de tipos (estilos de malha) que compõem a linha de produtos da empresa. As máquinas trabalham com cilindros substituíveis, que admitem o alojamento, com densidade variável, de diferentes tipos de agulhas. Cada particular configuração dessas agulhas (cuja instalação ou mudança constitui uma *preparação menor*) permite à máquina tecer um pequeno subconjunto da variedade total de estilos que ela pode tecer. Para tecer outros estilos, porém,

pode ser necessária a substituição do cilindro (o que constitui uma *prepara-ção maior*). Máquinas da mesma marca e do mesmo modelo podem compartilhar cilindros, mas, em geral, um tipo de cilindro destinado a determinada máquina não pode

ser usado em outra, de diferente marca ou modelo. Cada um dos itens da relação dos estilos de malha a serem produ-zidos pode ser tecido, opcionalmente, em diferentes máquinas, a diferentes taxas de produção, desde que essas máquinas estejam equipadas com o cilindro adequado. Portanto, segundo GRAVES (1981), o problema classifica-se como de estágio-único, com máquinas paralelas porém não-idênticas. Uma vez que as preparações, particularmente as maiores, ocupam precioso tempo da produção e consomem outros recursos, a programação dessas preparações - de modo a resolver a barganha entre rendimento da tecedura e dispêndio de esforços com as preparações - constitui, na Adele, um dos principais objetivos da programação. Os custos diretos e de oportunidade da produção perdida em decorrência das preparações são portanto explicitamente considerados, o que constitui uma outra divergência em relação à quase totalidade dos artigos sobre programação, com poucas exceções, como KAMARKAR (1987) e TRIGEIRO *et al.* (1989). KAMARKAR (1987) examina os efeitos do tamanho dos lotes sobre os tempos de espera de manufatura, levando em conta explicitamente os tempos de ajustes, ao longo de um horizonte infinito, num regime de fila estacionária.

Em artigo mais recente, TRIGEIRO *et al.* (1989) propõem um modelo que, em meio à grande maioria dos trabalhos sobre programação existentes na literatura, é o que mais se aproxima de uma representação da situação na Adele. Trata-se de um modelo multi-período multi-item, para a determinação do tamanho de lotes ao longo de um horizonte finito, que, como o modelo aqui apresentado, leva em conta explicitamente a capacidade consumida pelas

preparações. Entre os principais defeitos desse modelo, para o presente problema, destacam-se: (1) é um modelo de máquina única, ao passo que, na Adele, é necessário programar a operação simultânea de 48 máquinas circulares paralelas; (2) o modelo não permite considerar a capacidade dos instrumentos de preparação (cilindros) que constitui aqui uma característica importante do problema, e (3) como a maioria dos modelos anteriores para a determinação do tamanho de lotes, este atribui primordial importância à barganha entre os custos da manutenção de estoques e os da execução de preparações; sendo portanto mais adequado para fábricas que produzem-para-o-estoque, enquanto a Adele concentra seu maior volume de negócios na produção-sob-encomenda, atribuindo prioridade mais baixa às considerações concernentes ao estoque.

O modelo de programação de que trata este artigo determina, para cada máquina: (1) o intervalo de tempo durante o qual ela deverá trabalhar com um determinado cilindro e (2) as quantidades, dos respectivos estilos específicos, que deverão ser processadas enquanto a máquina estiver operando com esse cilindro. Daqui por diante, neste artigo, essas decisões serão mencionadas como planos de preparação/carga. O objetivo é maximizar alguma soma ponderada da massa total tecida ao longo do horizonte de planejamento, sujeita a restrições da capacidade (tanto de máquinas quanto de cilindros) e os limites superior e inferior da quantidade de produção de diferentes estilos. É óbvio que estas decisões a respeito de cargas de máquina e escolhas de cilindro não podem ser tomadas considerando *esti-lo-por-estilo* nem *máquina-por-máquina*, conforme até agora se tem feito na Adele, mas devem ser tomadas para todo o conjunto simultaneamente, para levar em conta o efeito de uma específica decisão de carga sobre as oportunidades de carregar outros

estilos, outras máquinas e outros cilindros. Quando o problema é considerado no seu próprio contexto, em que existem vários grupos de máquinas idênticas ou semelhantes, muitos tipos de cilindros compartilhados por algumas dessas máquinas, e uma grande quantidade de estilos de malha que podem ser processados, a diferentes taxas de produção, nessas combinações máquina-cilindro, a dificuldade deste problema de

programação desafia o uso de regras extraídas *ad hoc* do simples bom-senso. É necessário um procedimento *sistemático* de avaliação das possibilidades virtualmente ilimitadas de programação, para assegurar que a capacidade disponível será usada de maneira efetiva. Mais ainda, esse sistema deve também ser fácil de usar, requerendo apenas um esforço moderado, para que uma eventual necessidade de reprogramação, decorrente de alterações nos pedidos dos clientes, falhas de máquina, falta de materiais etc., possa ser enfrentada pela gerência de forma rápida, fácil e segura.

A filosofia adotada no projeto do sistema de programação não foi a de montar

uma “caixa-preta” altamente formalizada e rígida, mas, ao contrário, a de prover o programador humano de uma *ferramenta* interativa, que tomará as últimas decisões. Esta ferramenta interativa de programação formaliza as decisões de *carga* e deixa ao programador humano a escolha do *momento oportuno* para ações específicas, tais como trocas de cilindro e fixação dos diversos estilos de malha. Acreditamos que um programador bem treinado, que é quem melhor poderá manter-se ao corrente das mínimas circunstâncias inesperadas, tais como falhas de máquina, falta de materiais e alterações nos pedidos dos clientes, pode usar os planos de preparação/carga fornecidos pelo sistema e elaborar um cronograma de produção, mediante uma técnica tão simples como o gráfico de Gantt. Ao fazer isso, ele pode julgar quando é conveniente proceder a pequenas correções e quando é necessário reformular inteiramente a solução do problema com os dados do momento, caso alguma circunstância inesperada venha exigir um acentuado desvio dos planos de preparação/carga correntes.

3. Formulação do Problema

O problema de programar preparações e cargas de máquinas é formulado como um problema de programação inteira mista, que capta as características essenciais do

problema concernentes ao objetivo da programação, exigências da produção e capacidades de máquinas e cilindros. É usada a seguinte notação:

- i, j, k = índices para máquinas, cilindros e estilos, respectivamente;
- d_{ijk} = margem de contribuição do estilo k tecido na máquina i usando o cilindro j (receita – custos diretos por libra);
- c_{ijk} = taxa de produção do estilo k processado na máquina i usando o cilindro j (libras/hora);
- D_k^u (D_k^l) = limite superior (inferior) na produção do estilo k ;
- H_i = horas-padrão disponíveis da capacidade da máquina i ;
- G_j = horas-padrão disponíveis do cilindro tipo j ;
- S_i = conjunto dos tipos de cilindro que a máquina i pode usar;
- L_{ij} = tempo-padrão para instalar um cilindro do tipo j na máquina i (preparação maior);
- e_{ij} = custo de uma preparação maior (desembolso);
- K_{ijk} = tempo-padrão para iniciar o estilo k na máquina i com o cilindro j (preparação menor);
- f_{ijk} = custo de uma preparação menor (desembolso);

- x_{ij} = duração do tempo em que a máquina i está programada para usar o cilindro j ;
 y_{ijk} = libras do estilo k processadas na máquina i enquanto está usando o cilindro j ;
 z_{ij} = 1 se a máquina i estiver programada para usar o cilindro j durante todo o período da programação, 0 em caso contrário;
 w_{ijk} = 1 se a máquina i estiver programada para processar o estilo k enquanto usar o cilindro j , 0 em caso contrário.

O problema é agora formulado matematicamente como:

$$\text{Max} \sum_i \sum_{j \in S_i} \sum_k (d_{ijk} y_{ijk} - f_{ijk} w_{ijk}) - e_{ij} z_{ij} \quad (1)$$

sujeito a

$$D_k^l \leq \sum_i \sum_{j \in S} y_{ijk} \leq D_k^u \quad \forall k \quad (2)$$

$$\sum_k (y_{ijk} / c_{ijk} + K_{ijk} w_{ijk}) - x_{ij} \leq 0 \quad \forall i, j \in S_i \quad (3)$$

$$\sum_{j \in S_i} (x_{ij} + L_{ij} z_{ij}) \leq H_i \quad \forall i \quad (4)$$

$$\sum_i (x_{ij} + L_{ij} z_{ij}) \leq G_j \quad \forall j \quad (5)$$

$$x_{ij} \leq (H_i - L_{ij}) z_{ij} \quad \forall i, j \in S_i \quad (6)$$

$$w_{ijk} \leq z_{ij} \quad \forall k, i, j \in S_i \quad (7)$$

$$y_{ijk} \leq D_k^u w_{ijk} \quad \forall k, i, j \in S_i \quad (8)$$

$$y_{ijk}, x_{ij} \geq 0 \quad \forall k, i, j \in S_i \quad (9)$$

$$z_{ij}, w_{ijk} \in \{0, 1\} \quad k, i, j \in S_i \quad (10)$$

Na formulação acima, a função objetivo maximiza a contribuição total líquida da programação, ao longo do horizonte de planejamento. Isso é modelado como receita menos custos diretos (material, trabalho e despesas gerais variáveis) menos os custos diretos de todas as preparações, que podem incluir itens tais como: custo direto do trabalho de preparação, apropriações, refugos no início e no final do processo de produção de um estilo específico, etc. As equações (2) definem limitações, derivadas das condições de mercado, sobre a produção de cada estilo. D_k^l , por exemplo, pode representar comprissos da empresa com clientes importantes, enquanto D_k^u pode representar, em relação aos itens produzidos para o estoque, a soma dos pedidos

da firma, previsão de vendas adicionais e máximo ajustamento admissível do estoque. A equação (3) estabelece conexões lógicas entre as variáveis y_{ijk} , x_{ij} e w_{ijk} , estipulando que a soma dos tempos: de duração do processo e de uma preparação menor, programados para a máquina i usando o cilindro j é limitada pelo tempo programado para a operação dessa máquina com esse cilindro. As equações (4) são restrições da capacidade e limitam o tempo total programado para a operação da máquina i ao tempo-padrão disponível (H_i). Este corresponde às horas disponíveis no turno, ajustadas para: rendimento, utilização e para qualquer atividade de não-produção programada, tal como manutenção preventiva, de-

envolvimento de produto, realização de testes etc. As equações (5) modelam as limitações que a disponibilidade de diferentes tipos de cilindro impõe à programação. A disponibilidade de cilindros do tipo j , G_j , também é medida em horas-padrão: o número de cilindros disponíveis vezes o número de horas do turno menos qualquer atividade de não-produção programada. As equações (6), (7) e (8) estipulam conexões lógicas entre as variáveis do problema. As equações (6) forçam $x_{ij} = 0$ se a máquina i não estiver programada para usar o cilindro j (ou seja, se $z_{ij} = 0$); são ociosas, todavia, se $z_{ij} = 1$. As equações (7) só admitem a programação de preparações menores se estiver programada a prepara-

ção maior associada. As equações (8) forçam $y_{ijk} = 0$ quando $w_{ijk} = 0$ e, caso contrário, são ociosas.

Na sua maior parte, os dados que esta formulação requer ou já se encontram, de alguma forma, disponíveis; ou poderão tornar-se disponíveis sem muito esforço. A empresa dispõe de uma base de dados com as taxas de produção de diversos estilos em diversas configurações máquina-cilindro. Embora não se tenha efetuado um estudo de engenharia de produção formal, a gerência "sente" que as diferenças de tempos e custos das preparações maiores envolvendo diferentes pares máquina-cilindro devem ser insignificantes a ponto de poderem ser estimadas, como um parâmetro genérico, com aceitável grau de segurança.

Esta formulação parece representar todas as características proeminentes do atual problema de carga, além de requerer dados que poderiam ser obtidos com moderado esforço. Infelizmente, porém, o tamanho do problema pode, quando relativamente grande, impedir a sua efetiva solução. Na Adele, com cerca de 48 máquinas, de um a cinco tipos de cilindro para cada uma delas e de 25 a 50 estilos a serem processados durante um

dado horizonte de programação, o número de restrições (variáveis zero/um e contínuas) pode facilmente atingir várias dezenas de milhares. É óbvio que a tentativa de resolver semelhante modelo em busca de soluções ótimas para o problema seria altamente impraticável.

Para encontrar um meio razoável de tratar do problema, procuramos simplificar sua formulação de modo a conseguir uma redução significativa do tamanho sem demasiado sacrifício daquilo que é importante. Mais ainda, de acordo com a filosofia de projeto de dar apoio ao programador humano, concentramos nossa atenção num método de solução aproximada capaz de prontamente oferecer sugestões para o esclarecimento dos dilemas que o programador deve resolver.

A primeira simplificação que conduz a uma formulação equivalente porém menor, baseia-se na observação de que existe sempre uma solução ótima para o problema no qual as equações (3) se verificam estritamente como igualdades. Visto que c_{ijk} e K_{ijk} são não-negativas, as equações (3) podem ser resolvidas em relação às variáveis x_{ij} , eliminando-as e removendo do problema um grande número de restrições. (É claro que o problema poderia ter sido formulado desde o início sem estas variáveis x_{ij} . Existe, todavia, alguma utilidade na explícita separação destas duas decisões interligadas: escolhas associadas à designação dos cilindros a serem instalados nas máquinas circulares (as variáveis x_{ij}) e aquelas que estão associadas à designação dos estilos a serem processados durante os intervalos de tempo disponíveis de cada par máquina-cilindro (as variáveis y_{ijk}).)

A segunda simplificação do modelo é obtida mediante a adoção de uma forma implícita e aproximada de tratamento para as preparações menores, o que elimina uma grande quantidade de variáveis zero/um e alivia a necessidade de estimar acuradamente o tempo e os custos associados a preparações menores

específicas. O modo mais comum de fazer isso consiste em fundir esses custos e tempos com outros dados do problema. Pode-se, por exemplo, incorporar o tempo e os custos das preparações menores ao tempo e aos custos do funcionamento da máquina. A exatidão, neste caso, será bastante boa se for possível fazer *a priori* suposições razoáveis quanto ao tamanho dos lotes. Uma outra abordagem consiste em introduzir ajustamentos na função objetivo e na disponibilidade de máquina, de modo a levar em conta o custo e o tempo de produção perdido em decorrência desses ajustes. A precisão desta aproximação dependerá, por outro lado, da variância do número de preparações menores realmente programadas para as máquinas.

Ambas as abordagens seriam aceitáveis no presente caso, uma vez que os recursos consumidos nas preparações menores são realmente insignificantes em comparação com os dispendidos nas maiores. Como já

existe um grau de incerteza no parâmetro H_i , devido à taxa de rendimento, utilização, falhas etc., a gerência julgou que ajustando H_i mediante uma redução de 15% concederia tempo ocioso suficiente para abrigar as preparações menores. Incorporando ao modelo estas duas simplificações, mediante a eliminação das variáveis w_{ijk} e a substituição $x_{ij} = k(1/c_{ijk})y_{ijk}$, o problema assume a forma:

$$\text{Max } \sum_i \sum_{j \in S_i} \left(\sum_k d_{ijk} y_{ijk} - e_{ij} z_{ij} \right) \tag{11}$$

sujeito a

$$\sum_i \sum_{j \in S_i} y_{ijk} \geq D_k^l \quad \forall k \tag{12}$$

$$\sum_i \sum_{j \in S_i} y_{ijk} \leq D_k^u \quad \forall k \tag{13}$$

$$\sum_{j \in S_i} \left[\sum_k (1/c_{ijk}) y_{ijk} + L_{ij} z_{ij} \right] \leq H_i \quad \forall i \tag{14}$$

$$\sum_{j \in S_i} \left[\sum_k (1/c_{ijk}) y_{ijk} + L_{ij} z_{ij} \right] \leq G_j \quad \forall j \tag{15}$$

$$D_k^u z_{ij} \geq y_{ijk} \quad \forall k, i, j \in S_i \tag{16}$$

4. Método de Solução

Para resolver este modelo seria concebível empregar um pacote de programação linear inteira mista, de uso geral. Existem contudo várias razões que compelem a preferir um método aproximado de análise, especificamente projetado para o caso. Em primeiro lugar, sendo Adele uma empresa de pequeno porte, não seria justificável que arcasse com o custo de um pacote comercial, principalmente porque não dispõe do tipo de *hardware* usualmente exigido pela maioria daqueles pacotes. Em segundo,

porque, a despeito da substancial redução conseguida por meio das simplificações, o problema ainda permanece impraticavelmente grande. Com 50 máquinas, de 5 a 10 tipos de cilindro para cada uma delas e de 25 a 50 estilos, seriam aproximadamente de 250 a 500 inteiros, de 6250 a 25000 variáveis contínuas, e um número de restrições entre 6250 e 25000. Obviamente, a menos que se use algum tipo de decomposição, ou que as equações (16) sejam tratadas implicitamente, a

solução direta de (11)-(16) mediante o emprego de um pacote PI de uso geral seria absolutamente impraticável. Em terceiro, prevemos a necessidade de reprocessar várias vezes o problema, de modo específico, na tentativa de testar e refinar a programação de produção, antes de obter a autorização final para implementá-la. Conseqüentemente, a flexibilidade de rápidas revisões da programação, para responder a: mudanças inesperadas das encomendas dos clientes, falhas da maquinaria ou falta de materiais, tem prioridade mais alta que a habilidade para resolver “otimamente” o problema.

Pacotes de uso geral provavelmente não irão prover o usuário de um meio fácil e propício para resolver repetidamente o problema, nesse esforço de refinamento da solução. Será mais vantajoso um procedimento baseado em microcomputador, rápido e de fácil utilização com métodos aproximados, que o programador da produção poderá sentir que domina e usar com segurança.

A abordagem aproximada baseia-se na seguinte estratégia, que é “a da ladeira mais íngreme”: Para um dado vetor z_{ij} o problema é um problema de programação linear. Uma vez que todas as equações (16) e todas as variáveis y_{ijk} associadas a $z_{ij} = 0$ são neste caso eliminadas, o tamanho do problema resultante é muito mais razoável, em comparação com o de (11)-(16). Este programa linear é resolvido para a condição ótima, e, com base nas informações contidas na solução, estima-se a contribuição potencial das designações máquina-cilindro excluídas (variáveis z_{ij} que agora são nulas). Essas estimativas são então utilizadas para decidir quanto às designações adicionais máquina-cilindro que deverão ser programadas (caso existam), e o programa linear é reotimizado. Este procedimento é repetido iterativamente até que nenhuma designação máquina-cilindro adicional revele potencial para melhorar a função

objeto-vo. (Embora o problema de carga possa ser resolvido mediante a aplicação formal deste algoritmo, na implementação real, conforme será discutido mais adiante, cuidou-se de permitir a interferência do usuário, num cenário interativo.) Esta abordagem geral tem sido utilizada com êxito em outras aplicações de programação inteira mista. Por exemplo, TRIGEIRO *et al.* (1989) usam estratégia semelhante para resolver o problema da determinação do tamanho de lotes levando em consideração a capacidade. Eles utilizam um *relaxamento* Lagrangeano com otimização subgradiente para estimar os custos de oportunidade dos tempos de ajuste, na avaliação do benefício potencial de um novo ajuste. A presente abordagem, porém, conta, para esse propósito, com a solução de uma *restrição* de uma programação linear, inspirada no particionamento de BENDERS (1962).

A idéia de tentar resolver um problema de programação linear inteira mista fixando provisoriamente as variáveis inteiras, resolvendo o problema linear resultante e usando as informações assim obtidas para gerar uma “melhor” solução inteira, e assim sucessivamente, é de fato atraente. Na verdade, é precisamente essa a idéia do particionamento de Benders. Resumidamente, esta abordagem começa com um vetor inteiro e um correspondente programa linear (PL) cuja solução constitui um limite superior. A solução dual ótima desse PL é usada para modificar um problema inteiro - o “principal restrito” (PR), cuja solução fornece um limite inferior e uma nova solução inteira. A iteração entre o PL e os programas inteiros prossegue até que os limites superior e inferior coincidam. A dificuldade inerente a esta abordagem reside no fato de que a cada iteração é necessário resolver um programa inteiro. Diversas alterações dessa abordagem têm sido propostas para contornar essa dificuldade. GEOFFRION & GRAVES (1974) e AKINC & ROODMAN (1984), por exemplo, adotam

a estratégia de não resolver o PR para a condição ótima confirmada a cada iteração. Em lugar disso, enquanto enumerando o PR, cada vez que um novo incumbente é encontrado, o trabalho é temporariamente interrompido para resolver o PL correspondente. Modifica-se o PR usando a solução dual para o PL e o trabalho sobre o PR é retomado.

No presente caso, a sucessiva adição de novos ajustes para melhorar o problema de programação global constitui na realidade uma enumeração incompleta do PR nas variáveis do ajuste. A exemplo do que ocorre no particionamento de Benders, usam-se as soluções duais dos PLs associados para gerar uma série de soluções inteiras “melhoradas”.

Para começar a discutir um pouco mais pormenorizadamente a abordagem, considere-se a resolução do problema de carga

$$\text{Max } \sum_i \sum_{j \in \Theta_r} \sum_k d_{ijk} y_{ijk} - \text{Constante} \tag{17}$$

sujeito a

$$\sum_i \sum_{j \in \Theta_r} y_{ijk} \geq D_k^l \quad \forall k \tag{18}$$

$$\sum_i \sum_{j \in \Theta_r} y_{ijk} \leq D_k^u \quad \forall k \tag{19}$$

$$\sum_{j \in \Theta_r} \left[\sum_k (1/c_{ijk}) y_{ijk} + L_{ij} z_{ij} \right] \leq H_i - \sum_{j \in \Theta_r - \{i\}} L_{ij} \quad \forall i \tag{20}$$

$$\sum_i \sum_k [(1/c_{ijk}) y_{ijk}] \leq G_j - \sum_{j \in \Theta_r - \{i\}} L_{ij} \quad \forall j \tag{21}$$

$$y_{ijk} \geq 0 \quad \forall i, j \in \Theta_r, k \tag{22}$$

em que

$$j_r = \{i \mid z_{ij} = 1\} \tag{23}$$

A estrutura deste PL é semelhante à de um problema referente a transportes, exceto pelo fato de existirem, no presente caso, ambos os limites, superior e inferior, no tocante a “remessas” para os “destinos”, e dois conjuntos de restrições de “fornecimento”, correspondendo, respectivamente, à capacidade da máquina e à do cilindro. (Pode ser possível tirar proveito desta

para um determinado período de planejamento t. Faça-se j_r ser definido como $\{j \mid z_{ij} = 1\}$ S_i em alguma iteração r, e j_i denotar o índice do cilindro programado para estar na máquina i no final do período de planejamento t-1. O procedimento iterativo tem início fazendo-se $j_r = \{j_i\}$ uma vez que existe sempre uma solução ótima

tal que $z_{ij_i} = 1$. A apropriação dos custos e tempos desses ajustes já foi feita no período t-1 e eles são portanto “grátis” em t. O problema de programação linear na iteração r é obtido do problema original restringindo-se as variáveis y_{ijk} àquelas que são compatíveis com os ajustes em j_r . O programa linear a ser resolvido na iteração r é:

estrutura especial, projetando um algoritmo para uso específico, mais eficiente que os de uso geral em programação linear.) Suponha-se que este problema está resolvido para uma particular iteração r mediante um conveniente código de PL e que y_{ijk} foram obtidas como uma solução ótima. (Uma vez que as restrições no tocante às capa-

idades de cilindro são na verdade imprecisas, o PL é resolvido sem elas mas a solução é testada quanto à exequibilidade. Caso a solução deixe de satisfazer a alguma dessas restrições, o que é bastante

impro-vável, a exequibilidade será restaurada mediante iterações dual-simplex.) Caso (17)-(22) seja exequível, isto é, se

$$\sum_i \sum_{j \in \Theta_{ir}} y_{ijk} \geq D_k^1 \quad \forall k$$

então y_{ijk} é também uma exequível solução do problema original e seu valor pode ser usado como uma solução candidata. Se, por outro lado, (17)-(22) for inexecuível em decorrência de ajustes em i_r inadequados para satisfazer os requisitos mínimos, nenhuma tal candidata será produzida. Em qualquer caso, novos ajustes são considerados,

numa tentativa de reduzir o montante de inexecuibilidade (quando não exequível) ou aumentar o valor da solução candidata (quando exequível). Isto se faz mediante a progressiva adição de novos ajustes em i_r até que não seja possível encontrar nenhum ajuste para melhorar a solução.

Tornando Exequível a Solução

Enquanto novos ajustes são iterativamente adicionados, na “resolução” do problema de carga, dá-se prioridade àqueles ajustes que propiciam a maior redução da inexecuibilidade. A razão disto é o fato de que, desta forma, quando o PL se torna exequível, a análise de ajustes adicionais para melhorar o valor da solução pode ser feita de modo mais significativo. Além disso, a busca da exequibilidade é a prática corrente em programação - atualmente, ao encontrar um padrão de carga exequível, o

programador pouco faz (talvez pela falta de ferramentas fáceis de usar) para explorar ajustes adicionais que permitam melhorar a solução.

A procura por solução exequível é feita mediante uma heurística muito simples, que se baseia na maximização da utilização da capacidade ociosa de máquina ou de cilindro no PL corrente, para dirigi-lo rumo à exequibilidade. Para tanto, definimos:

$$H'_i = \text{Max} \left\{ 0, \left(H_i - \sum_{j \in \Theta_{ir}} \left(\sum_k (1/c_{ijk}) y_{ijk} - L_{ij} z_{ij} \right) \right) \right\}; \quad (24)$$

$$G'_j = \text{Max} \left\{ 0, \left(G_j - \sum_i \left(\sum_k (1/c_{ijk}) y_{ijk} - L_{ij} z_{ij} \right) \right) \right\}; \quad (25)$$

$$D'_k = \text{Max} \left\{ 0, \left(D_k^1 - \sum_i \sum_{j \in \Theta_{ir}} y_{ijk} \right) \right\}; \quad (26)$$

H'_i , G'_j e D'_k são, respectivamente, a capacidade ociosa da máquina i , e do cilindro tipo j e o déficit do estilo k abaixo do seu requisito mínimo de produção (o valor de uma variável artificial). Também, para um

fixado \tilde{i} tal que $H'_{\tilde{i}} > 0$ e um fixado $\tilde{j} \in \Theta_{i_r}$ tal que: $G'_{\tilde{j}} > 0$, seja $T_{\tilde{i}\tilde{j}} = \{k \mid D'_k > 0, c_{\tilde{i}\tilde{j}k} > 0\}$, isto é, o conjunto dos estilos com déficit tal que pode ser

reduzido pela adição de \tilde{j} ao conjunto Θ_{ir} . Um limite inferior para o montante da redução do déficit total devida à anexação

$$R_{\tilde{ij}} = \text{Max } \sum_k \alpha_k \quad (27)$$

$$0 \leq \alpha_k \leq D'_k \quad \forall k \in T_{\tilde{ij}} \quad (28)$$

$$\sum_k (1/c_{\tilde{ij}k}) \alpha_k \leq \text{Min } \{H'_i, G'_j\} - L_{\tilde{ij}} \quad (29)$$

Supondo os k dispostos segundo a ordem decrescente de $c_{\tilde{ij}k}$, resolve-se facilmente o problema somando D'_k e $D'_k/c_{\tilde{ij}k}$ até que o resultado da segunda soma iguale o segundo membro de (29).

O fato de $R_{\tilde{ij}}$ ser um limite inferior da redução da inexequibilidade assegura que a adição de novos ajustes na ordem decrescente de $R_{ij} > 0$ resultará numa redução monotônica da inexequibilidade. Para mostrar que $R_{\tilde{ij}}$ é realmente um limite inferior, suponha-se que o corrente PL inexequível seja aumentado com as variáveis adicionais: $y_{\tilde{ij}k} = \alpha_k > 0$. Esta solução é uma nova solução básica do PL. Para constatar isso, considere-se o caso em que, na condição ótima de (27)-(29), (29) não seja estrita. Então, cada nova variável

$y_{\tilde{ij}k}$ simplesmente substitui a variável artificial correspondente na base. Por outro lado, se (29) for estrita, então deve existir um k^* tal que, para $k < k^*$, $k = k^*$ e $k > k^*$ se tenha, respectivamente, $\alpha_k = D'_k$, $\alpha_k \leq D'_k$ e $\alpha_k = 0$. Neste caso, para $k < k^*$, cada $y_{\tilde{ij}k}$ substitui uma variável artificial, enquanto $y_{\tilde{ij}k^*}$ substitui uma variável ociosa (associada a \tilde{i} ou a \tilde{j}) na base. Em qual-quer dos casos, o acréscimo das novas variáveis manterá básica a solução. Em decorrência de sua construção, esta solução básica tem algumas propriedades adicionais: (1) ela não violará nenhuma das restrições satisfeitas pela solução original; (2) ela pode satisfazer algumas

de \tilde{j} a Θ_{ir} pode ser obtido por meio da solução do seguinte problema simples:

das restrições violadas pela anterior, e (3) ela pode violar as condições para a solução ótima de PL (isto é, ela pode ser dual inexequível). As duas primeiras propriedades implicam que a solução aumentada reduzirá de $R_{\tilde{ij}}$ o montante da inexequibilidade. Caso se verifique, porém, a violação mencionada em (3), ganhos superiores a $R_{\tilde{ij}}$ poderão ser conseguidos por meio das etapas adicionais de reprocessamento do PL para restabelecer a solução ótima.

Intuitivamente: esta última possibilidade pode ocorrer se determinados estilos cuja produção não é deficitária puderem ser produzidos mais eficientemente na máquina \tilde{i} (pelo fato de que agora ela conta com o uso do cilindro tipo \tilde{j}), liberando alhures alguma capacidade que poderá ser aproveitada para conseguir maiores reduções do déficit. Uma vez que a matriz das restrições do PL é relativamente rarefeita (significando que somente um pequeno número de combinações máquina-cilindro pode processar um determinado estilo), são remotas as chances de conseguir tais aumentos da redução do déficit total mediante reprocessamentos do PL, que exigem dispendioso trabalho de computação. Conseqüentemente, $R_{\tilde{ij}}$ pode ser considerado como altamente confiável para prever a redução total da inexequibilidade, para efeito de instruir a escolha da próxima combinação máquina-cilindro a ser programada.

A heurística da procura de exequibilidade consiste, em resumo, em:

1. Resolver (27)-(29) para todos os i, j tais que $H'_i > 0$, $G'_j > 0$ e $T_{ij} \neq \emptyset$, parar se não existir nenhuma solução;
2. Fazer $R_{\tilde{ij}} = \text{Max } R_{ij}$, parar se $R_{ij} = 0$;
3. Fazer $\Theta_{\tilde{ir}} = \Theta_{ir} \cup \{\tilde{j}\}$;
4. Substituir adequadamente as variáveis artificiais ou ociosas pelas novas $y_{\tilde{ij}} = \alpha_k$, recalculando H'_i , G'_j e D'_k ;
5. Repetir os passos de 1 a 4, acima.

Quando a condição de parada for satisfeita, com todos os $D'_k = 0$, ter-se-á encontrado uma solução exequível para o problema. Neste ponto o PL pode *opcionalmente* ser reotimizado, caso a inserção das novas variáveis tenha ocasionado a violação de alguma das condições da solução ótima. Se, por outro lado, todos os $H'_i = 0$ ou todos os $G'_j = 0$ e ainda houver

alguns $D'_k > 0$, o procedimento terá falhado enquanto tentativa de encontrar uma solução exequível. Mesmo assim, contudo, o mon-tante de inexecuibilidade em (17)-(22), $\Sigma D'_k$, terá sofrido uma redução, e a solução será utilizada para dar início ao estágio seguinte do procedimento.

Eliminação de Emparelhamentos Máquina-Cilindro Não-promissores

Quer se tenha encontrado quer não, no estágio precedente, uma solução exequível, há que considerar a formação de pares máquina-cilindro adicionais, capazes de favorecer a produção. Essa exploração se faz mediante avaliação do incremento do valor objetivo de (17)-(22) em decorrência da adição de um novo ajuste candidato.

Antes, porém, de obter tais estimativas para todas as possíveis designações máquina-cilindro não programadas, é feita uma tentativa de eliminar as associações máquina-cilindro que oferecem potencial muito baixo, ou nenhum, de acordo com um jul-

gamento baseado nas informações já disponíveis quanto ao custo reduzido. Na iteração r , sejam B e $\mu_k^l, \mu_k^u, \pi_i, \sigma_j$, respectivamente, a base ótima e os multiplicadores duais associados às restrições (18)-(21) do PL corrente. O custo reduzido associado a alguma variável, $y_{\tilde{ijk}}$, não incluída no PL corrente pode ser calculado a partir do produto do vetor dos multiplicadores duais pelo vetor dos coeficientes de $y_{\tilde{ijk}}$, conforme segue:

$$\delta_{\tilde{ijk}} = d_{\tilde{ijk}} - [\mu_k^u - \mu_k^l + (1/c_{\tilde{ijk}})(\pi_i + \sigma_j)] \quad (30)$$

O valor $\delta_{\tilde{ijk}}$ mede a taxa local de mudança no valor do PL quando $y_{\tilde{ijk}}$ se torna básica. Quaisquer variáveis $y_{\tilde{ijk}}$ para as quais $\delta_{\tilde{ijk}} \leq 0$ são postas fora de consideração, pois não é provável que o seu potencial para melhorar a solução se torne positivo quando, em posteriores iterações, ajustes adicionais se fizerem disponíveis para o PL. Obviamente, também será posta

fora de consideração qualquer z_{ij} que venha a ter eliminadas todas as variáveis y a ela relacionadas. Mais ainda, durante uma sessão interativa com o usuário, e a critério deste, também poderão ser abandonadas aquelas $y_{\tilde{ijk}}$ com $\delta_{\tilde{ijk}}$ positivo porém “pe-queno” e “pequeno” D_k^u , pois é improvável que venham a oferecer melhoras capazes de justificar seus elevados custos de preparação.

Identificação de Emparelhamentos Máquina-Cilindro Promissores

Para simplificar a discussão, seja $A_r = \{(i, j) \mid z_{ij} = 1\}$, isto é, o conjunto de todas as combinações máquina-cilindro que estão disponíveis na interação r , e seja $Z[A_r]$ o valor objetivo ótimo de PL quando estiverem em uso as preparações pertencentes a A_r .

Defina-se $\Delta_{\tilde{i}\tilde{j}} = Z[A_r \cup \{\tilde{i}, \tilde{j}\}] - Z[A_r]$ como a *melhora marginal exata* na função objetivo, decorrente da programação de \tilde{j} em \tilde{i} . Esta melhora, contudo, é exata somente em relação ao PL corrente, e o seu real efeito na solução final poderá ser bem diferente, dependendo das outras prepara-

$$\Delta_{\tilde{i}\tilde{j}} = \text{Max}_{\beta_k} \sum_{k \in N_{\tilde{i}\tilde{j}}} \delta_{\tilde{i}\tilde{j}k} \beta_k - d_B B_r^{-1} t_{\tilde{i}\tilde{j}} L_{\tilde{i}\tilde{j}} \quad (31)$$

sujeito a

$$\sum_{k \in N_{\tilde{i}\tilde{j}}} B_r^{-1} P_{\tilde{i}\tilde{j}k} \beta_k \leq B_r^{-1} (P_o - t_{\tilde{i}\tilde{j}} L_{\tilde{i}\tilde{j}}) \quad (32)$$

$$\beta_k \geq 0 \quad k \in N_{\tilde{i}\tilde{j}} \quad (33)$$

Nas expressões acima $P_{\tilde{i}\tilde{j}k}$ e P_o são vetores do problema original (11)-(16) e denotam a coluna de coeficientes de $y_{\tilde{i}\tilde{j}k}$ e o segundo membro das restrições, respectivamente. O vetor $t_{\tilde{i}\tilde{j}}$ tem valor 1 (um) nas posições correspondentes às restrições de capacidade da máquina \tilde{i} com o cilindro \tilde{j} e 0 (zero) em todas as demais. Como antes, B_r denota a base do PL corrente e d_B o subvetor dos coeficientes das variáveis básicas correntes da função objetivo.

Falando por alto, (31)-(33) estima o efeito negativo de um novo ajuste, devido ao seu consumo de capacidade (o termo constante, na função objetivo), e o efeito positivo decorrente da contribuição das novas variáveis, habilitadas pelo ajuste $\tilde{i}\tilde{j}$. Este último efeito é estimado mediante a reotimização do PL num sentido restrito, ou seja: considerando-se apenas as novas variáveis $y_{\tilde{i}\tilde{j}k}$, $k \in N_{\tilde{i}\tilde{j}}$. Para justificar este arrazoado, considere-se a reotimização total do PL corrente após a inclusão das no-

ções que vierem a estar presentes naquele momento. Assim sendo, não vale a pena insistir na obtenção, comparativamente dispendiosa, de tais valores exatos. Estimativas mais simples de $\Delta_{\tilde{i}\tilde{j}}$ serão perfeitamente adequadas para a busca de associações proveitosas máquina-cilindro.

Seja N_{ij} o conjunto de todos os estilos que podem ser tecidos na máquina i com o cilindro j , ou seja: $N_{ij} = \{k \mid c_{ijk} > 0\}$. Um limite inferior para $\Delta_{\tilde{i}\tilde{j}}$ é obtido como o valor ótimo de:

vas variáveis $y_{\tilde{i}\tilde{j}k}$ e aproximando para baixo, por $L_{\tilde{i}\tilde{j}}$, os segundos membros das restrições de capacidade de \tilde{i} e \tilde{j} . No caso particular em que $N_{\tilde{i}\tilde{j}}$ contenha somente uma variável, a maximização precedente equivale à determinação do valor exato da melhora de $Z(A_r)$ depois de executado *uma única vez* o processamento do PL. Isso pode ser feito do seguinte modo: (1) restaurar a exequibilidade, usando o método dual-simplex caso a redução nas capacidades da máquina \tilde{i} e do cilindro \tilde{j} tenha tornado inexecutável o PL; (2) introduzir na base todas as *novas* variáveis, até que todos os seus custos reduzidos sejam não-positivos e, finalmente, (3) introduzir quaisquer outras variáveis não-básicas *antigas* cujos custos reduzidos se tenham tornado positivos ao longo dos passos precedentes. Conforme se mostra, com maior rigor, no Apêndice A, resolver (31)-(33) é equivalente a implementar somente o passo (2) anterior. A qualidade deste salto depende obviamente da frequência e da extensão com que seria necessário

executar os passos (1) e (3) para a total reotimização, caso esta fosse levada a efeito.

O passo (1) será desnecessário sempre que $H_{\bar{i}} - L_{\bar{i}\bar{j}}$ e $G_{\bar{j}} - L_{\bar{i}\bar{j}}$ não forem menores do que os limites inferiores dos intervalos de $H_{\bar{i}}$ e $G_{\bar{j}}$, respectivamente, dentro dos quais B_r se mantenha exequível. Se este não for o caso, então poderá não haver nenhuma solução exequível para (31)-(33), o que tornaria inútil a tentativa de efetuar o salto proposto. Os PLs, toda-via, particularmente nas primeiras iterações, são caracterizados pela abundante disponibilidade de capacidades de máquina e cilindro, pelo fato de terem sido programados bem poucos cilindros para cada máquina. Como resultado, a condição de exequibilidade da antiga base é satisfeita na maioria dos casos. Resta ainda a possibilidade de que, em algum ponto do algoritmo, todos os pares remanescentes máquina-cilindro, tornando (31)-(33) inexecuível, invalidem o salto em questão. Caso isto ocorra, o próximo par máquina-cilindro deverá ser selecionado a partir de alguma outra heurística ou intuição do programador para superar o impasse. A escolha poderá basear-se, por exemplo, no mínimo $d_B B_r^{-1} t_{ij} L_{ij}$, na mínima inexecuibilidade (soma dos componentes negativos do segundo membro de (32)) ou na solução de (31)-(33) tendo como segundos membros: $\text{Max} \{0, B_r^{-1} (P_o - t_{\bar{i}\bar{j}} L_{\bar{i}\bar{j}})\}$.

A solução de (31)-(33) dará o valor exato de $\Delta_{\bar{i}\bar{j}}$ sempre que o passo (3) for desnecessário. Experiência limitada à duração do trabalho de desenvolvimento do sistema produziu alguma evidência de que o

passo (3) é na verdade raramente requerido. A seleção do novo par $\bar{i}\bar{j}$ dentre os candidatos em determinada iteração baseia-se no máximo $\bar{\Delta}_{\bar{i}\bar{j}}$ e uma reotimização plena é efetuada após adicionar-se o par *selecionado*. Esta

reotimização plena raramente exigiu que se introduzisse na base uma variável antiga não-básica. Uma razão para esse comportamento poderá ser a rarefação da matriz das restrições do problema original: cada estilo só pode ser tecido num restrito número de combinações máquina-cilindro. Portanto, o ingresso de uma nova variável y_{ijk} na base raramente abre oportunidades substanciais de incremento da eficiência para variáveis já existentes, associadas a outros estilos, diferentes de k , a serem tecidos em pares alternativos máquina-cilindro.

Comparado à plena reotimização do PL para calcular $\Delta_{\bar{i}\bar{j}}$, o cálculo de $\bar{\Delta}_{\bar{i}\bar{j}}$ para todos os pares potenciais máquina-cilindro requer um esforço de computação bastante pequeno. A condição de manter exequível a base corrente pode ser facilmente testada, na medida em que informações suscetíveis às mudanças no segundo membro das restrições são prontamente disponíveis mediante a solução do PL. $P_{\bar{i}\bar{j}k}$ tem somente quatro componentes não nulos: o valor 1 (um), nas posições correspondentes às restrições de demanda (18) e (19), e $1/c_{\bar{i}\bar{j}k}$ nas posições correspondentes às restrições de demanda (20) e (21). O vetor $B_r^{-1} P_{\bar{i}\bar{j}k}$, associado à nova variável $y_{\bar{i}\bar{j}k}$, pode portanto ser calculado como uma soma ponderada das quatro colunas correspondentes de B_r^{-1} . Os segundos membros são computados a partir de $B_r^{-1} P_o - B_r^{-1} t_{\bar{i}\bar{j}} L_{\bar{i}\bar{j}}$ cujo primeiro termo é o valor da solução ótima corrente e $B_r^{-1} t_{\bar{i}\bar{j}}$ é simplesmente a soma de duas colunas de B_r^{-1} . Embora o problema (31)-(33) tenha o mesmo número de restrições que o próprio PL, devido à extrema rarefação de B_r^{-1} , os vetores $B_r^{-1} t_{\bar{i}\bar{j}}$ contêm principalmente zeros, fazendo com que se abandone a maioria das restrições em (31)-(33). Além disso, este problema, ao contrário do PL original, tem poucas variáveis: o

número destas é a cardinalidade de $N_{\tilde{i}\tilde{j}}$. Conseqüentemente, estes saltos podem ser quase sempre obtidos mediante inspeção. Nos raros casos em que isto não possa ser

feito, o dual do problema, tendo muito poucas restrições, poderá ser resolvido por meio do algoritmo simplex.

Sumário do Algoritmo para a Solução Aproximada do Problema de Carga:

1. $r \leftarrow 1$; $\Theta_{ir} \leftarrow \{j_i\} \forall i$; CAND = 0.
2. Resolva o programa linear resultante (17)-(22).
3. Se $\sum_i \sum_j y_{ijk} \geq D'_k \forall k$, go to 7.
4. Usando (27)-(29), determine (\tilde{i}, \tilde{j}) que minimize a inexequibilidade. Se não existir nenhum (i, j) que reduza a inexequibilidade, go to 9.
5. $\Theta_{ir} \leftarrow \Theta_{ir} \cup \{\tilde{j}\}$,
6. Go to 3.
7. Reotimize (17)-(22) (opcional).
8. CAND = Z(A_r).
9. Compute $\delta_{ijk}, \forall i, j \in S_i - \Theta_{ir}$, e k.
10. $S_i \leftarrow S_i - \{j\}, \forall (i, j)$ tal que $\delta_{ijk} \leq 0, \forall k$, ou qualquer outra regra de eliminação.
11. Se $i_r = S_i \quad i$, pare.
12. Compute $\bar{\Delta}_{ij} \forall i, j \in S_i - \Theta_{ir}$, tal que B_r permaneça exequível com respeito a $H_{\tilde{i}} - L_{\tilde{i}\tilde{j}}$ e $G_{\tilde{j}} - L_{\tilde{i}\tilde{j}}$.
13. Faça $\bar{\Delta}_{\tilde{i}\tilde{j}} = \text{Max } \bar{\Delta}_{ij}$
14. Se $\bar{\Delta}_{\tilde{i}\tilde{j}} < 0$, go to 18.
15. Atualize o PL, inserindo $y_{\tilde{i}\tilde{j}k} \forall k$.
16. CAND = Z[A_r]; $r \leftarrow r + 1$; $\Theta_{ir} \leftarrow \Theta_{ir-1} \cup \{\tilde{j}\}$
17. Go to 11.
18. Escolha \tilde{i}, \tilde{j} utilizando uma heurística alternativa.
19. Atualize o PL, inserindo $y_{\tilde{i}\tilde{j}k} \forall k$.
20. Se Z[A_r] < CAND, então
 - a. Restaure o antigo PL;
 - b. $S_{\tilde{i}} \leftarrow S_{\tilde{i}} - \{\tilde{j}\}$;
 - c. Go to 18.
21. Go to 16.

5. Implementação do Sistema

O sistema de programação é parcialmente implementado, como piloto para a avaliação da exequibilidade da sua

implantação total. Nesta etapa os arquivos e programas do piloto constituem módulos independentes cuja interface é feita

manualmente. Por exemplo: o programa de uma base de dados de microcomputador é usado para criar e manter os arquivos do sistema; utiliza-se uma planilha para introduzir os dados referentes aos requisitos; um programa em BASIC prepara os dados para o PL, cuja solução se faz por meio do algoritmo simplex, escrito em FORTRAN. A implementação total envolverá simplesmente a integração dos módulos numa ferramenta interativa de fácil uso orientado por menu.

A principal fonte dos dados que entram no sistema é o arquivo de PADRÕES que já vem sendo mantido no computador da Adele. Mediante a interligação de um microcomputador IBM instalado, o programador da produção tem acesso imediato a esse arquivo, que contém cerca de 300 registros, com os seguintes campos:

- Número da máquina
- Nome da máquina
- Descrição da máquina
- Cilindro
- Estilo
- Produção padrão (em 24 horas).

Este é um arquivo relativamente estável, cujo conteúdo só é alterado quando uma nova máquina é adquirida ou uma das existentes é retirada. Foi esclarecido porém que ele pode não conter obrigatoriamente todas as possíveis

combinações máquina-cilindro-estilo, mas simplesmente arrola todas aquelas que foram usadas em alguma ocasião. Isto não acarreta nenhum problema grave para a implementação plena, pois à medida que novas combinações são descobertas e submetidas à experiência prática, o arquivo pode ser convenientemente atualizado.

Um novo arquivo, denominado PREPARAÇÃO, foi criado e mantido no microcomputador. Este arquivo contém as informações necessárias a respeito das relações máquina-preparações. Para cada uma das 48 máquinas ele mantém uma lista dos tipos

de cilindro que a máquina aceita, além de manter um registro dinâmico do tipo de cilindro correntemente instalado.

Na implementação piloto, o programador inicia o processo introduzindo a PROGRAMAÇÃO DE REQUISITOS semanal, mediante um programa de planilha eletrônica. Trata-se de uma lista de estilos, com as respectivas margens de contribuição e limites superior e inferior de produção.

O PL inicial, correspondente a $\Theta_{ir} = \{j_i\}$, é resolvido e o plano experimental de preparação/carga é resumidamente apresentado ao programador, sob a forma de vários relatórios:

1. PROGRAMAÇÃO EXPERIMENTAL (Máquinas) – Relatório impresso ou na tela, especificando, para cada máquina:
 - os cilindros que estão sendo usados (programação de preparação da máquina),
 - os estilos a serem processados mediante cada um dos cilindros programados para serem usados,
 - utilização da máquina (capacidade ociosa),
 - o valor da capacidade adicional (preço projetado da capacidade).
2. PROGRAMAÇÃO EXPERIMENTAL (Estilos) – Relatório impresso ou na tela, sobre cada estilo que figura na PROGRAMAÇÃO DE REQUISITOS:
 - limite inferior de produção,
 - limite superior de produção,
 - produção programada corrente,
 - custo marginal da violação dos limites superior e inferior (preço projetado dos requisitos)

Com base nesta última peça de informação, o programador pode querer revisar a programação de requisitos (limites superior e inferior da produção de cada estilo). Afinal, esses limites superiores e inferiores são apenas previsões, e, como tal, podem

3. AÇÕES POTENCIAIS – Uma lista das designações potencialmente benéficas envolvendo máquina, cilindro e estilo (aquelas com y_{ijk} positivo), com as respectivas taxas padrão de produção.
4. NOVOS CILINDROS POTENCIAIS – Uma lista das designações potenciais máquina-cilindro, e dos valores R_{ij} ou $\bar{\Delta}_{ij}$ a elas associados.

Com base nos relatórios 3 e 4 acima, e na exequibilidade do PL corrente, o programador determina uma nova preparação (i, j) a ser implementada. O PL é reotimizado com a inserção das novas variáveis y_{ijk} admitidas pela nova preparação. Este processo iterativo é executado sob o controle do programador até que o plano de preparação/carga seja aceito, numa determinada iteração, ou que não restem mais ajustes benéficos (conclusão normal). Neste ponto o programador recebe o RELATÓRIO FINAL DE CARGA: - uma lista das

preparações maiores em cada máquina,

preparações menores em cada máquina, quantidades de cada estilo a serem produzida em cada máquina.

Com a implementação piloto buscamos as respostas para várias questões: (1) É possível conseguir boas programações de produção mediante a abordagem de programação linear aplicada iterativamente? (2) Pode semelhante sistema ser implementado sem exigir grandes investimentos em *hardware* e em *software*, a despeito de sua dependência no tocante à utilização de

estar sujeitos a grandes folgas. Esta é uma das únicas características desta abordagem interativa em que não somente a solução, mas o próprio problema pode mudar de uma iteração para outra.

um instrumental matemático altamente sofisticado? (3) Ele poderá ser entendido e eficientemente utilizado por um programador inteligente? e (4) Os procedimentos de computação necessários poderão ser executados com suficiente rapidez, para permitir que o programador possa usá-los iterativamente?

A experiência limitada com a implantação piloto, sob a forma de programação retrospectiva, foi realmente animadora. O “sistema” gerou, de forma consistente, programações que se mostraram superiores quando equiparadas àquelas produzidas manualmente. Uma vez que o programador, que tem íntima familiaridade com o ambiente de programação da produção, esteve envolvido no projeto do sistema, sua dificuldade para entender a finalidade e o funcionamento deste, e o próprio papel a ser desempenhado pelo usuário no processo, foi realmente muito pequena. Finalmente, em quase todas as tentativas, foi necessário resolver de dois a cinco programas lineares relativamente pequenos, cada um dos quais demandou cerca de um minuto de processamento, num PC AT, dotado de coprocessador aritmético.

6. Conclusões

A programação da produção constitui uma área intrinsecamente difícil, no que concerne às decisões da gerência. Há inúmeros objetivos conflitantes que devem ser considerados simultaneamente, incluindo, além de outros, o atendimento de pedidos

dos clientes, o uso efetivo da capacidade instalada e os reflexos da programação nos estoques. Esta é provavelmente a razão pela qual os módulos de programação mestre da maioria dos sistemas de programação industrial tipo MRP II sejam os

mais fracos. Essencialmente, para a maior parte das implementações da programação mes-tre, o acionamento de todo o sistema de programação foi deixado a cargo da intuição de um “programador mestre”. Este trabalho tem como principal preocupação a de prover esse programador de uma ferramenta de apoio para suas complexas decisões. Embora a proposta ferramenta de apoio à decisão tenha sido projetada tendo-se em mente uma empresa específica, acreditamos que a abordagem geral aqui adotada pode ser aplicada a outros ambientes de programação. Na maioria dos casos, os custos das preparações, apesar de importantes, raramente dominarão a estrutura geral dos custos de programação. Assim, acreditamos que um algoritmo de programação baseado na judiciosa enumeração incompleta das variáveis referentes às preparações deverá

gerar provavelmente programações de produção satisfatórias.

Sendo este sistema uma ferramenta interativa, seu uso impõe alguns requisitos específicos ao usuário. Este deverá ter, como mínimo absolutamente indispensável, a capacidade de entender: (1) as entradas do sistema, onde e como são mantidos e quando e como atualizar os dados básicos que irão instruir a solução do problema; (2) as saídas do sistema sob a forma de telas ou de relatórios, compreendendo o significado das informações fornecidas, especialmente aquelas menos intuitivas, como “custos reduzidos”. É portanto essencial que o usuário do sistema seja submetido a um treinamento prévio, que deverá ser rápido e fácil, pois é presumível que o responsável pela programação tenha boa sensibilidade quanto aos objetivos deste sistema.

Apêndice A

Limites inferiores no valor de uma nova preparação

Usando a notação normal da programação linear, considere-se o problema:

$$(O) \quad \begin{array}{ll} \text{Max } c'x & \\ \text{sujeito a } & Ax = b \\ & x \geq 0 \end{array}$$

Seja B uma base ótima e $c = (c_B, c_N)$ e $x = (x_B, x_N)$ os correspondentes particionamentos de c e x . Dado um novo conjunto de variáveis x_V com vetor função objetivo

c_V e matriz de restrições V e um vetor arbitrário Δ , definam-se (E) e (R) relacionadas a (O) como:

$$(E) \quad \begin{array}{ll} \text{Max } c'x + c_V x_V & \\ \text{sujeito a } & Ax + Vx_V = b - \Delta \\ & x, x_V \geq 0 \end{array}$$

$$\begin{aligned}
 & \text{Max } c_B x_B + c_V x_V \\
 \text{(R)} \quad & \text{sujeito a} \quad Bx_B + Vx_V = b - \Delta \\
 & \quad \quad \quad x, x_V \geq 0
 \end{aligned}$$

Denotando-se por Z_E e Z_R , respectivamente, os valores ótimos de (E) e de (R), temos então que $Z_R \geq Z_E$ pois (E), com $x_N = 0$, é equivalente a (R). Além disso, se B for exequível em (E) e (R), então $Z_O \leq Z_R \leq Z_E$.

$$\begin{aligned}
 & \text{Max } c_B B^{-1}(b - \Delta) + 0 + (c_V - c_B B^{-1}V)x_V \\
 & \text{sujeito a} \quad I x_B + B^{-1}Vx_V = B^{-1}(b - \Delta) \\
 & \quad \quad \quad x_V, x_B \geq 0
 \end{aligned}$$

em que I é a matriz identidade. As variáveis x_B têm a forma de “folgas” e, por

$$\begin{aligned}
 & \text{Max} \\
 & Z_R = Z_O + \text{Max } (c_V - c_B B^{-1}V)x_V - c_B B^{-1}\Delta \\
 \text{(R')} \quad & \text{sujeito a} \quad B^{-1}Vx_V \leq B^{-1}(b - \Delta) \\
 & \quad \quad \quad x_V \geq 0
 \end{aligned}$$

Obviamente, o valor da precedente maximização é igual a $Z_R - Z_O$ e como $Z_E \geq Z_R$ ele é um limite inferior de $Z_E - Z_O$. O problema (31)-(33) é precisamente a maximização acima. Portanto, sua solução fornece um limite inferior para a mudança que a

Mostraremos agora a relação de (R) com (31)-(33). Na avaliação do custo a partir de (R) com base B, (R) pode ser escrita como:

definição, $Z_O = c_B B^{-1}b$. O problema pode então ser escrito como:

função objetivo sofre como resultado da adição de uma nova preparação e das novas variáveis a ela associadas.

Referências Bibliográficas

- AKINC, U. & ROODMAN, G. M.:** “A Model for Capacity/Location Analysis in an Integrated Manufacturing Strategy”. *Regional Review*, Vol. 14, 24-40, Maio/1984.
- BENDERS, J. F.:** “Partitioning Procedures for Solving Mixed Variable Programming Problems”. *Numerische Mathematik*, Vol. 4, 238-252, 1962.
- GEOFFRION, A. M. & GRAVES, G. W.:** “Multicommodity Distribution System Design by Benders’ Decomposition”. *Management Science*, Vol. 20, No. 5, 822-844, Janeiro/1974.
- GRAVES, S. C.:** “A Review of Production Scheduling”. *Operations Research*, Vol. 29, No. 4, 646-675, Julho-Agosto/1981.
- JOHNSON, S. M.:** “Optimal Two and Three-Stage Production Schedules with Setup Time Considered”. *Naval Research Logistics Quarterly*, Vol. 1, No. 1, 61-68, 1954.
- KAMARKAR, Uday S.:** “Lot Sizes, Lead Times and In Process Inventories”. *Management Science*, Vol. 33, No. 3, 409-418, Março/1987.
- STECHE, K. E.:** “Formulation and Solution of Non-Linear Integer Produc-

tion Planning Problems for Flexible Manufacturing". *Management Science*, Vol. 29, No. 3, 273-288, Março/1983.

TRIGEIRO, W. W., TBOMAS, L. J. & MCCLAIN, J. O.: "Capacitated Lot Sizing with Setup Times". *Manage-*

ment Science, Vol. 35, No. 1, 353-366, 1989.

VOLLMANN, T. E., BERRY, W. L. & WHYBARK, D. C.: *Manufacturing Planning and Control Systems*, 2nd ed.. Irwin, Homewood (Ill.), 1988.

A PRACTICAL APPROACH TO LOT AND SETUP SCHEDULING AT A TEXTILE FIRM

Abstract

This is a study of scheduling of setups and production activities of a textile firm, located in North Carolina, USA. The firm faces the problem of scheduling customer orders on a number of knitting machines which can be configured differently by installing different knitting cylinders, to knit various types of greige cloth. Given a set of requirements for different styles of cloth, the problem is to decide on the specific configurations to be used on each machine and on the specific orders to be run on these configurations. The problem is formulated as an integer linear programming model. The objective is the maximization of total contribution of

all the scheduled orders subject to capacity constraints of machines and that of tooling, which explicitly consider the effect of scheduled setups and constraints on customer orders. Various solution approaches are discussed. An approximate procedure is devised which incrementally adds new setups based on several heuristics by which the "value" of candidate configurations for the machines are evaluated. These heuristics can either be developed into a self contained scheduling procedure or can interactively be utilized by a human scheduler in a microcomputer environment.

Key-words: production scheduling, machine setup, machine load, linear programming.