

PROGRAMAÇÃO DA PRODUÇÃO EM INDÚSTRIAS DE MANUFATURA USANDO PROGRAMAÇÃO POR RESTRIÇÕES

Luiz Carlos de Abreu Rodrigues¹

Resumo. Neste trabalho considera-se o problema de programação da produção em indústrias de manufatura, considerando-se a sua solução usando a abordagem de Programação por Restrições. Propõe-se uma fase de planejamento que tem como objetivo a determinação de janelas de tempo para as tarefas, identificando um intervalo de tempo em que o processamento das tarefas é viável. As janelas de tempo permitem uma análise do carregamento dos processadores e da factibilidade do planejamento de produção. Após o planejamento de produção, realiza-se a fase de programação da produção que consiste no ordenamento das tarefas que compartilham um mesmo processador. Foi dado enfoque à solução da produção por bateladas.

Palavras-Chave: Programação da produção, Programação por restrições, Indústria de manufatura.

Abstract: This paper deals with the scheduling of manufacturing industries. The goal of this work is to present the solution of the proposed scheduling problem using Constraint Programming. At first a production planning phase is performed to identify tasks time windows. Time windows indicate the time interval within which the processing of tasks is feasible. Time windows enable the analysis of equipment units loading and the feasibility analysis of production planning. After production planning, scheduling is performed through the ordering of tasks that share the same resources. Batch production is emphasized within this work.

Keywords: scheduling of manufacturing, Constraint Programming, Manufacturing industry

1. Introdução

Pode-se distinguir dois segmentos industriais distintos: indústrias de manufatura e de processos. As indústrias de manufatura têm sua produção

¹ Doutor em Engenharia Elétrica pela UNICAMP. Professor do Departamento Acadêmico de Mecânica da Universidade Tecnológica Federal do Paraná. e-mail: lcan@cefetpr.br

medida em unidades (ou peças) produzidas, podendo-se citar as indústrias metalúrgicas e metal-mecânicas como exemplos deste segmento. Indústrias de processos têm sua produção medida em unidades de volume ou de massa, tendo as indústrias químicas, alimentícias e farmacêuticas como exemplos deste segmento. Como há uma ampla literatura científica sobre problemas de programação da produção nestes dois segmentos, este trabalho enfoca as indústrias de manufatura.

Os processos de produção podem ser separados em pelo menos duas categorias: contínua e por bateladas. Slack (2002) considera a existência de outras categorias de processos de produção, porém os problemas de programação da produção destas categorias adicionais, quando associados a indústrias de manufatura, podem ser enquadrados como casos particulares do mesmo problema encontrado em processamentos por bateladas.

Os processos de produção contínua caracterizam-se pela presença de “linhas de produção”, e até mesmo plantas industriais, dedicadas à produção contínua de um produto. Entre as indústrias que adotam a produção contínua, podem-se citar indústrias petroquímicas, de papel e automobilística, entre outras. Slack (2002) faz uma distinção entre a produção contínua e a produção em massa. Porém, pode-se minimizar tal diferenciação considerando-se que na verdade Slack (2002) distinguiu a produção dedicada e em grandes volumes nas indústrias de processos (produção contínua) e de manufatura (produção em massa). Para este trabalho, uma discussão das diferenças entre produção contínua e de massa não é considerada relevante, pois em ambos os casos não há, efetivamente, um problema de programação da produção, já que a produção é dedicada em ambos os casos. No entanto, o problema de programação da produção pode ser determinante para a competitividade de empresas que operam nos processos de produção por bateladas e por projeto. O caso de produção dita semicontínua, quando uma linha de produção é utilizada para a obtenção de mais de um produto, pode ser solucionado usando a abordagem proposta. Na discussão, apresentada na seqüência deste trabalho, a produção semi-contínua será tratada como produção por batelada com arranjo físico multi-produto (*flowshop*).

A produção por bateladas, também chamada de produção por lotes, é encontrada nas indústrias de manufatura e de processos, sendo que as matérias-primas são processadas em quantidades finitas. Além disso, estes sistemas caracterizam-se pela produção descontínua, já que ocorrem interrupções no processamento entre diferentes bateladas que compartilham um mesmo sistema de produção. Há dois tipos de arranjos físicos associados à produção por bateladas: arranjo físico por processo e por produto. O arranjo físico por produto, também chamado *flowshop*, caracteriza-se pela existência de vários processadores (postos de trabalho) próximos uns dos outros, com uma disposição que reflete a seqüência de processamento dos produtos. Neste caso, os sistemas de produção correspondentes, que podem ser linhas de produção ou sistemas flexíveis

de manufatura, são projetados especialmente para a obtenção de uma família de produtos similares ou, ao menos, produtos que seguem a mesma seqüência de produção. Tais sistemas de produção são projetados visando à otimização do trabalho envolvido e à minimização de estoques em processo. Ao longo deste trabalho o arranjo físico por produto será chamado de produção multi-produto.

O arranjo físico por processo, também chamado *jobshop*, caracteriza-se pela inexistência de uma mesma seqüência de produção para todas as bateladas que são processadas num dado posto de trabalho. Neste tipo de arranjo físico, a fábrica caracteriza-se pela existência de vários departamentos organizados pela semelhança de equipamentos ou habilidades. Ao longo deste trabalho o arranjo físico por processo será chamado de produção multipropósito.

Há diversas técnicas para a programação da produção, pode-se citar a Programação Linear Inteira Mista e métodos chamados heurísticos. A abordagem apresentada neste trabalho para a resolução do problema proposto baseia-se em duas etapas: planejamento e programação da produção. As duas etapas são solucionadas usando a “Programação por Restrições”, que é conhecida em inglês como *Constraint Programming* ou *Constraint Logic Programming*. O objetivo deste trabalho é divulgar esta técnica ainda pouco conhecida no Brasil, mas que já está presente em *softwares* comerciais (Ilog, 1997). Na seção 2 deste trabalho, são apresentadas as duas etapas da abordagem de Programação por Restrições. Na seção 3, é apresentado um exemplo sucinto e na seção 4 são apresentados os resultados finais.

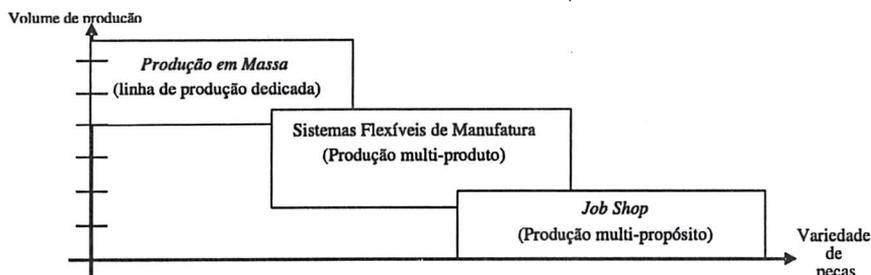


FIGURA 1. Arranjo físico em função do volume de produção.

2. Planejamento e programação da produção

Diversas áreas na indústria utilizam plantas operando em lotes ou batelada. Estas plantas descontínuas caracterizam-se por um conjunto de equipamentos multipropósito, ou seja, úteis para diferentes tarefas em diferentes “receitas de produção”, e com amplas possibilidades de interconexão. A operação das plantas descontínuas pode ser realizada de forma cíclica, pela repetição de campanhas de produção (operação em campanhas), quando a demanda de produtos finais é suficientemente previsível. A fábrica pode ser configurada para produzir

determinados produtos finais que são estocados para garantir a demanda durante um período determinado. O interesse provém da diminuição de tempos de preparação e de uma operação repetitiva no chão-de-fábrica.

Quando existe incerteza na demanda, a operação é determinada pela demanda a curto prazo e, possivelmente, pela reposição de níveis pequenos de estoque, não sendo possível determinar um ciclo a ser repetido (operação de curto prazo). O problema tratado neste trabalho é o planejamento e a programação da produção de curto prazo. A questão central neste tipo de operação é averiguar se a capacidade da planta e o horizonte de tempo disponível são suficientes para atender a demanda, ou inversamente adaptar/negociar a demanda à capacidade de produção da planta. Este problema surge porque a capacidade de uma planta multipropósito não está bem definida, já que ela depende dos produtos que serão fabricados e, especialmente, do *mix* de produção. Nos dois casos extremos pode-se ter uma planta com baixa utilização dos equipamentos, o que provavelmente leva a fabricar produtos adicionais para estoque, ou uma demanda elevada que implica em atrasos de entrega. .

Neste trabalho, a demanda é caracterizada por quantidades específicas de produtos finais com prazos de entrega determinados. Note que tal consideração é válida tanto no caso de política *make-to-stock* como *make-to-order*. A divisão do problema em duas etapas, de planejamento e de programação de produção, constituiu-se em uma abordagem tanto industrial quanto acadêmica. Porém, esta divisão não é trivial: a etapa de planejamento deveria incluir diversos aspectos da programação da produção de forma a gerar planos aceitáveis para esta última. Por exemplo, a hipótese de capacidade infinita ou o uso da análise de capacidade de produção em sistemas MRP pode levar a uma situação infactível devido à definição dos *lead times*. Na literatura acadêmica, vários autores propuseram modelos de planejamento simplificados baseados nos modelos da programação de produção, pela agregação do tempo e da agregação de restrições (Bassett et al., 1996; Subrahmanyam et al., 1996). Neste caso, o objetivo são modelos simplificados que são utilizados, a partir de formulações de programação linear inteira mista, para determinar os requisitos de produção em cada período de tempo agregado. O problema, semelhante ao caso do MRP, é poder garantir que o problema de programação da produção posterior será factível. Quando a situação não é factível, o recurso é a volta ao nível de planejamento e à introdução de limites nos requisitos de produção que podem ser alocados a cada período de tempo.

A abordagem conjunta dos problemas de planejamento e de programação da produção tem sido atacada por formulações de programação linear inteira mista (Kondili et al., 1993; Baker, 1993), mas a necessidade de uma representação suficientemente fina do tempo (para a programação da produção) leva, em geral, a problemas de dimensão demasiadamente grandes.

Neste trabalho propõe-se uma solução em dois níveis, na qual o nível de planejamento fornece, como resultado, um conjunto de janelas de tempo factíveis

em termos de capacidade dos equipamentos sob diversas circunstâncias. Propõe-se uma fase de planeamento que tem como objetivo determinar: *i*) um horizonte de planeamento factível para atender a demanda e *ii*) analisar o carregamento da planta induzido pela demanda. A ferramenta principal para a análise do carregamento da planta é a abordagem de Programação por Restrições (*Constraint Programming*). Ela permite determinar o carregamento dos processadores induzido pelas tarefas habilitadas neles a partir das janelas de tempo em que estas tarefas devem ser processadas. Estas janelas de tempo para processamento são determinadas, na fase de planeamento, por procedimentos semelhantes aos utilizados por sistemas MRP/MRP II, mas sem a utilização de *lead times*.

A fase de planeamento fornece à fase de programação um resultado na forma de uma janela de processamento para cada batelada de cada tarefa. Estas janelas de tempo são factíveis do ponto de vista de capacidade dos processadores sob diversas circunstâncias e fornecem aos usuários uma visão clara do carregamento permitindo; por exemplo, para permitir a análise das folgas necessárias para atender a imprevistos no chão-de-fábrica. Este resultado, em forma de um conjunto de janelas de tempo, é extremamente útil para reduzir a dimensão do problema de programação.

2.1. Fase de planeamento

Neste trabalho, a modelagem do processo de produção é feita utilizando a representação de Rede Estado-Tarefa (*State Task Network - STN*) introduzida por Kondili et al. (1993). Trata-se de uma representação em forma de grafo no qual os nós representam estados (matérias-primas, produtos intermediários e produtos finais) ou tarefas (estágios na transformação das matérias primas em produtos finais). A Figura 1 mostra um exemplo de representação STN. A tarefa *TA* processa o estado de entrada *S1* sendo gerado o estado *S3*. Já a tarefa *TC* tem dois estados de entrada (matérias primas), *S2* e *S3*, e produz dois estados de saída, *S4* e *S5*. Cabe notar que o grafo STN representa apenas a receita de fabricação, tal como ocorre no *bill of materials* em sistemas MRP; não inclui a alocação de tarefas a processadores, podendo existir várias alocações possíveis como é corrente em plantas multipropósito. Os tempos de processamento das tarefas são considerados fixos, independentes da alocação. Caso contrário, o estágio em questão tem de ser representado por duas (ou mais) tarefas com tempos de processamento diferentes (função dos processadores em que estão habilitadas). A representação STN específica, para cada tarefa, os estados de entrada e saída, assim como suas porcentagens, na entrada e saída, com relação ao tamanho da batelada da tarefa (frações mássicas). As informações na representação STN são análogas às contidas no *bill of materials* utilizado em sistemas MRP, mas é uma ferramenta com maior potencialidade de representação de receitas complexas de produção, tais como recíclós, frequentes na indústria química.

A estrutura para o sistema de planeamento proposto é a indicada na Figura 3. A obtenção das janelas de processamento é feita em três passos:

- determinação da quantidade de bateladas de cada tarefa e dos instantes de término mais tarde (*latest finish times – LFT*), respectivos, a partir da demanda de produtos finais e de uma alocação única de tarefas a processadores;
- determinação dos tempos de início mais cedo (*earliest start times – EST*) para todas as bateladas a partir de um plano de fornecimento de matérias-primas;
- propagação das restrições induzidas pelas janelas de tempo (*EST* e *LFT*), obtendo as janelas de tempo para a fase seguinte de programação da produção.

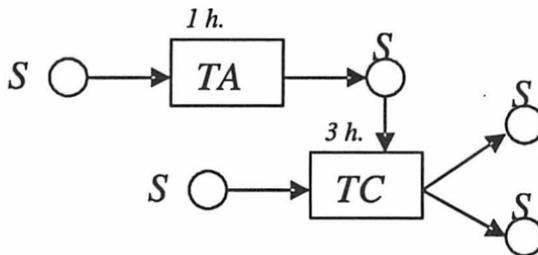


FIGURA 2. Representação Estado-Tarefa.

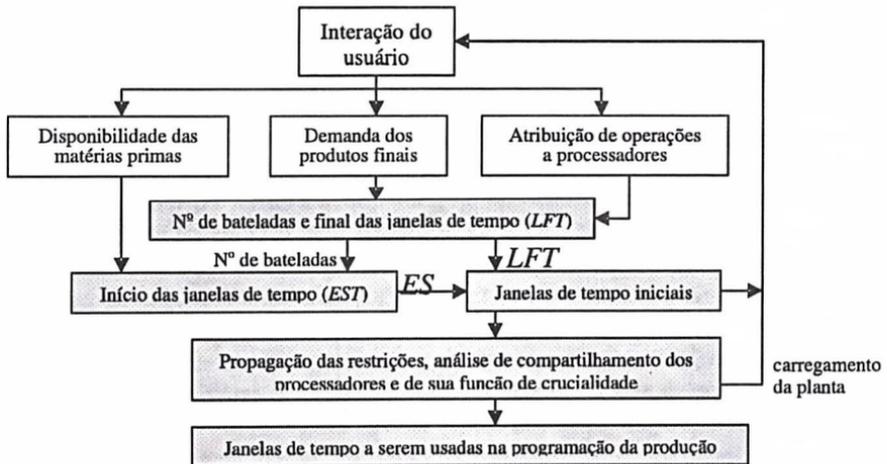


FIGURA 3. Estrutura do sistema de planeamento.

2.1.1 Quantidade de bateladas e instantes de término mais tarde.

O procedimento é semelhante à fase de explosão em sistemas MRP (Orlicky, 1975). Inicia-se com os produtos finais e percorre-se o diagrama STN até as matérias-primas determinando a demanda de cada estado induzida pelos produtos

finais. A representação STN permite determinar facilmente a ordem em que os estados devem ser tratados (*low level coding*). Da mesma forma que em sistemas MRP, nesta fase deve ser determinado o tamanho de batelada para cada estado ou tarefa produzindo cada estado.

Freqüentemente, na indústria química, a liberdade para definir o tamanho de batelada de uma tarefa é muito restrita devido a limitações tecnológicas no equipamento (ex. reatores) ou pouco conhecimento de fatores de escala na receita; no sistema desenvolvido tem-se optado por um tamanho de batelada fixo. A determinação deste tamanho de batelada implica em definir uma alocação única de tarefas a equipamentos (permitindo a alocação de várias tarefas ao mesmo equipamento), o que constitui uma das três entradas do sistema de planejamento. Ou seja, as diferentes opções possíveis de alocação de equipamentos devem ser analisadas como cenários de planejamento diferentes, sendo que a informação sobre o carregamento dos equipamentos em cada caso, fornece ao usuário meios para selecionar as modificações possíveis.

Diferentemente dos sistemas MRP, a fase de explosão não utiliza *lead times*, mas apenas os tempos de processamento (acrescentados de tempos de *set ups* independentes da seqüência), isto porque nesta fase não se determinam os momentos de liberação das ordens de produção, mas apenas instantes de término mais tarde (LFT - *latest finish time*). A fase de explosão termina com a obtenção de um plano de fornecimento de matérias-primas. Este plano é um plano-limite na medida em que só seria factível se todos os equipamentos fossem dedicados, ou seja, se não existisse compartilhamento de processadores por diferentes tarefas. O plano é também individualizado, na medida em que cada batelada consumindo matéria-prima origina uma demanda com prazo de conclusão específico. Qualquer plano de fornecimento de matéria-prima será certamente mais agregado e deverá preceder no tempo a este plano limite. A introdução de um plano realista é feita pelo usuário e modificada interativamente até conseguir-se um resultado do planejamento factível e que implique num carregamento da planta que seja julgado adequado.

2.1.2. Determinação de tempos de início mais cedo e obtenção das janelas de tempo iniciais

Os instantes de início mais cedo de todas as bateladas (já determinadas no passo anterior) são obtidos a partir de um plano de fornecimento de matérias-primas introduzido pelo usuário. O procedimento consiste apenas em um balanço de massa iniciando com as matérias-primas e percorrendo o STN até os produtos finais. Obtém-se desta forma o tempo de início mais cedo (EST) para cada batelada dos estados intermediários e dos produtos finais. Para cada batelada, o par, (EST e LFT), determina a sua janela de tempo para produção. Dada a alocação única de tarefas a equipamentos, essas janelas de tempo de produção de estados são também as janelas de processamento das bateladas das tarefas produtoras. Cada

janela deve ser obviamente maior que o tempo de processamento, mais qualquer tempo de *set up* necessário, devendo incluir uma folga para permitir o compartilhamento de processadores. Esta folga corresponde à diferença entre o *lead time* e o tempo de processamento em sistemas MRP, mas assim como nestes sistemas a folga é um dado de entrada (através do *lead time*) aqui ela é determinada por uma análise do carregamento dos processadores induzido pelas janelas de tempo das bateladas.

2.1.3. Propagação de restrições

A fase de propagação de restrições tem como objetivo determinar as reduções nas janelas de tempo das bateladas impostas pelas condições do problema. As condições que podem levar a reduções nas janelas de tempo, implementadas no sistema de planejamento, são as seguintes:

- intervalos de tempo com ocupação obrigatória de equipamentos;
- ordenamentos entre bateladas de tarefas diferentes alocadas ao mesmo equipamento;
- dependência entre as janelas de tempo de bateladas produtoras e consumidoras de um estado intermediário (balanço de massa);
- ordenamento entre bateladas sucessivas de uma mesma tarefa;
- restrições sobre as janelas de tempo de bateladas ligadas por estados intermediários com condições de armazenagem restritas.

Intervalos de ocupação obrigatória

Dada a janela de tempo (EST , LFT) de uma batelada, com tempo de processamento TP , se $LFT - TP < EST + TP$, o intervalo ($LFT - TP$, $EST + TP$) será necessariamente utilizado pela batelada. Esta ocupação obrigatória do equipamento, ao qual a tarefa está alocada, implica que o equipamento não estará disponível para outras bateladas neste intervalo. A propagação desta restrição pode levar à redução das janelas de outras bateladas se estas janelas incluem o intervalo.

Ordenamentos entre bateladas de tarefas diferentes alocadas ao mesmo equipamento

A detecção de ordenamentos induzidos pelas janelas de processamento utiliza ferramentas desenvolvidas na área de Programação por Restrições (Le Pape, 1994; Baptiste e Le Pape, 1995), que tem sido incorporadas em *softwares* de programação de produção comerciais (Ilog, 1997). De forma resumida, trata-se de determinar ordenamentos obrigatórios entre pares de bateladas. Sejam A e B duas

bateladas competindo pelo mesmo processador, com janelas de processamento dadas por (EST_A, LFT_A) e (EST_B, LFT_B) e tempos de processamento TP_A e TP_B . Definem-se os instantes de início mais tarde (*latest start time - LST*) e de término mais cedo (*earliest finish time - EFT*) como: $LBT_A = LFT_A - TP_A$ e $EFT_A = EST_A + TP_A$. A possibilidade de ordenamento tem que ser analisada apenas se as duas janelas tem interseção. A situação é infactível, ou seja as bateladas não poderão ser processadas dentro das suas janelas, se a condição 1 é satisfeita. Se a situação é factível, um ordenamento pode ser deduzido se alguma das duas condições 2a ou 2b é satisfeita.

$$\max(LFT_A, LFT_B) - \min(EST_A, EST_B) < TP_A + TP_B \quad (1)$$

$$EFT_B > LBT_A, \text{ então } A \text{ precede } B \quad (2a)$$

$$EFT_A > LBT_B, \text{ então } B \text{ precede } A \quad (2b)$$

Estes ordenamentos podem dar lugar a reduções nas janelas nas seguintes situações:

- Quando *A* precede *B*, EST_B terá um valor mínimo de forma a permitir o processamento de *A* antes, como indicado na equação 3a. Da mesma forma, LFT_A terá um valor máximo para permitir o processamento de *B* depois, como indicado na equação 3b.

$$EST_B \geq EST_A + TP_A \quad (3a)$$

$$LFT_A \leq LFT_B - TP_B \quad (3b)$$

- Quando *B* precede *A*, EST_A terá um valor mínimo de forma a permitir o processamento de *B* antes, como indicado na equação 4a. Da mesma forma, LFT_B terá um valor máximo para permitir o processamento de *A* depois, como indicado na equação 4b.

$$EST_A \geq EST_B + TP_B \quad (4a)$$

$$LFT_B \leq LFT_A - TP_A \quad (4b)$$

Qualquer relação de ordenamento é deduzida da condição $LFT_r - EST_r \geq TP_r$, onde *r* representa qualquer conjunto de bateladas competindo por um processador. Existem $(2^n - 1)$ conjuntos para *n* bateladas. Uma forma de reduzir a dimensão do problema foi apresentada em Caseau e Laburthe (1994) pela utilização do conceito de intervalo de tarefas (*task interval*). Um intervalo de tarefas é definido como o conjunto de tarefas (bateladas) construído a partir de duas tarefas *A* e *B* (que podem ser a mesma) que contém todas as tarefas *i* tais que $EST_i \geq \min(EST_A, EST_B)$ e $LFT_i \leq \max(LFT_A, LFT_B)$. Os autores provam que apenas é necessário considerar estes conjuntos, a vantagem sendo que o número de conjuntos é no máximo n^2 .

O ordenamento entre uma tarefa A e um intervalo de tarefas Ω ($A \notin \Omega$), chamado de *edge-finding*, é apresentado a seguir. Para um conjunto Ω , o LFT_{Ω} é definido como o máximo LFT das tarefas pertencentes a Ω ou $\max(LFT_A, LFT_B)$. EST_{Ω} é definido como o mínimo EST das tarefas pertencentes a Ω ou $\min(EST_A, EST_B)$. O tempo de processamento (TP) é definido como a soma dos tempos de processamento das tarefas. Definindo $\phi = \Omega \cup A$, a situação será inactível se $LFT_{\phi} - EST_{\phi} < TP_{\Omega} + TP_A$. Em uma situação factível, o ordenamento pode ser deduzido em quatro casos, sendo os dois primeiros considerados “fracos” e os dois últimos “fortes”:

- (a) Se $LFT_{\Omega} - EST_A < TP_{\Omega} + TP_A$, então A não precede o conjunto Ω , indicando que pelo menos uma das tarefas contidas no conjunto Ω deverá ser processada antes de A . Portanto, a seguinte restrição deve ser satisfeita:

$$EST_A \geq \min_{i \in \Omega} (EST_i + TP_i) \quad (5)$$

- (b) Se $LFT_A - EST_{\Omega} < TP_{\Omega} + TP_A$, então o conjunto Ω não precede A , indicando que pelo menos uma das tarefas contidas no conjunto Ω deverá ser processada após o processamento de A . Portanto, a seguinte restrição deve ser satisfeita:

$$LFT_A \leq \max_{i \in \Omega} (LFT_i - TP_i) \quad (6)$$

- (c) Se $LFT_{\Omega} - EST_{\Omega} < TP_{\Omega} + TP_A$, significa que A não pode ser processada entre as tarefas i pertencentes a Ω . Se, adicionalmente, $LFT_{\Omega} - EST_A < TP_{\Omega} + TP_A$, condição indicada no caso (a), então, como A não precede o conjunto Ω , necessariamente o conjunto Ω precede A . Portanto, as restrições (7) e (8) deverão ser satisfeitas. A restrição (7) indica que, como Ω precede A , EST_A deve ser maior ou igual ao instante de término mais cedo do processamento de Ω . De forma análoga, a restrição (8) limita os finais de processamento das tarefas i pertencentes a Ω à posterior necessidade de processamento de A .

$$EST_A \geq EST_{\Omega} + TP_{\Omega} \quad (7)$$

$$LFT_i \leq LFT_A - TP_A \quad \forall i \in \Omega \quad (8)$$

- (d) Se $LFT_{\Omega} - EST_{\Omega} < TP_{\Omega} + TP_A$, apresentada no caso (c), significa que A não pode ser processada entre as tarefas i pertencentes a Ω . Se, adicionalmente, $LFT_A - EST_{\Omega} < TP_{\Omega} + TP_A$, condição indicada no caso (b), então, como o conjunto Ω não precede A , necessariamente A precede o conjunto Ω . Portanto, as restrições (9) e (10) deverão ser satisfeitas. A restrição (9) indica que, como A precede Ω , LFT_A deve ser menor ou igual ao instante de início mais tarde do processamento de Ω . De forma análoga, a restrição (10) limita o início de processamento das tarefas i pertencentes a Ω à necessidade de processamento de A antes de Ω .

$$LFT_A \leq LFT_{\Omega} - TP_{\Omega} \quad (9)$$

$$EST_i \geq EST_A + TP_A \quad \forall i \in \Omega \quad (10)$$

A determinação de ordenamentos induzidos pelas janelas de tempo de bateladas competindo pelo mesmo processador pode originar desta forma reduções nas janelas de tempo através de aumentos de EST e/ou diminuições de LFT.

Propagação de restrições

A fase de planejamento, pelo processo de *edge-finding*, permite determinar as relações de precedência entre bateladas produtoras e consumidoras do mesmo estado. A precedência entre duas bateladas *A* e *B* implica que $EST_B > EST_A + TP_A$ e $LFT_A < LFT_B - TP_B$. Desta forma, por exemplo, qualquer aumento no EST da batelada produtora *A* se traduz no aumento do EST da batelada consumidora *B*. Torna-se assim necessária uma fase de propagação de restrições através da receita (STN), a chamada propagação intra-ordem, complementando a propagação inter-ordem que focaliza os conflitos nos processadores. Além disso, bateladas sucessivas de uma mesma tarefa estão alocadas no mesmo processador e, portanto, entre duas bateladas *A1* e *A2* pode-se impor que $EST_{A2} > EST_{A1} + TP_{A1}$ e $LFT_{A1} < LFT_{A2} - TP_{A2}$.

Restrições de armazenagem

As limitações nas condições de armazenagem são muito frequentes em indústrias de processos dada a natureza dos produtos intermediários. No entanto, tais restrições também podem ser adotadas ao tratar o problema de programação da produção em indústrias de manufatura, impondo limitações sobre o tempo de fila e de espera na produção multipropósito (*jobshop*). De fato, tal consideração deveria ser adotada como uma alternativa para as empresas que não têm volumes de demanda que justifiquem a implantação de linhas de produção ou células de manufatura, mas que desejariam adotar o *Just in Time*, ou quase isso, para produtos de alto valor agregado. No mínimo, pode-se afirmar que a simulação de restrições de armazenagem permite identificar o impacto destas restrições sobre os custos de produção, ainda que sejam um desafio para o controle da produção no chão de fábrica. Na literatura, as restrições de armazenagem têm sido classificadas como: i) FIS (*Finite Intermediate Storage*) quando há capacidade limitada de armazenamento; ii) NIS (*No Intermediate Storage*) quando não existe local para a armazenagem mas o produto pode ser armazenado no processador; e iii) ZW (*Zero Wait*) quando o produto terá tempo de fila nulo. Na produção multiproduto (*flowshop*) pode-se considerar que a restrição FIS ocorre em sistemas de produção com esteira de alimentação, já que a extensão da esteira entre dois postos de trabalho permite uma capacidade limitada de armazenamento. Tal consideração não será aceitável em sistemas de produção em que a alimentação entre dois

postos de trabalho é feita por um mesmo robô. Neste ocorrerá a restrição NIS, se for possível que um produto “espere” pela próxima tarefa de processamento dentro da máquina que acaba de processá-lo. Se esta hipótese não for aceitável (por exemplo, por restrições técnicas), deverá ser adotada a restrição ZW entre postos de trabalho.

Essas restrições não podem ser representadas por relações de precedência entre bateladas, a exceção sendo o caso ZW. Considere, por exemplo, a situação representada na Figura 4. Dada a condição ZW sobre o estado intermediário, uma mudança (redução) no *LFT* de uma batelada consumidora (*TB*) exige uma redução no *LFT* da batelada produtora (*TA*) correspondente. Esta propagação pode ser feita utilizando apenas as relações de precedência impondo $LFT_{TA} = LFT_{TB} - TP_{TB}$. Da mesma forma, na situação em que ocorre um aumento no *EST* de uma batelada consumidora (*TB*), tem-se $EST_{TB} = EST_{TA} + TP_{TA}$.

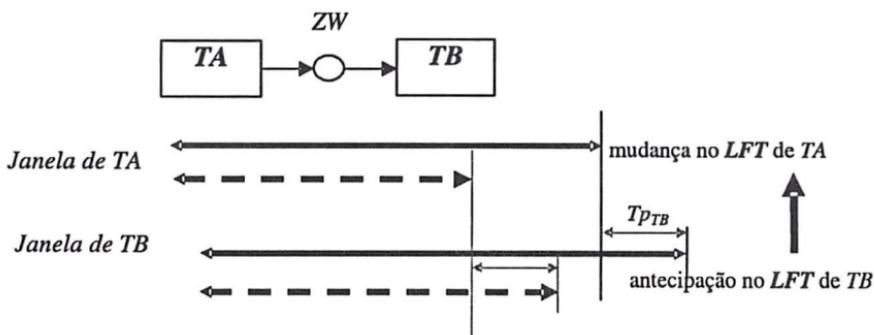


FIGURA 4. Propagação da antecipação do *LFT* de *TB*.

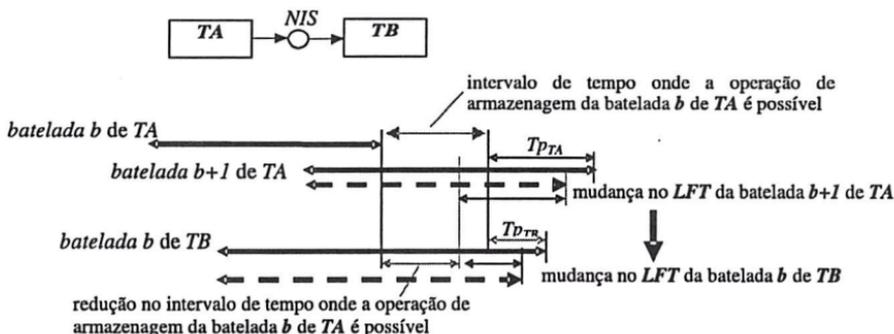


FIGURA 5. Propagação da restrição NIS.

A situação NIS não pode ser resolvida apenas pelas relações de precedência, como é ilustrado na Figura 5. Supõe-se uma mudança (redução) no *LFT* da batelada *b+1* da tarefa produtora *TA*. Inicialmente, isto origina uma redução do intervalo de tempo em que o produto intermediário resultado da batelada *b* pode

ser armazenado no processador da tarefa *TA*. Como consequência, o *LFT* da batelada *b* da tarefa consumidora *TB* deve ser reduzido.

A situação FIS exige a análise dos perfis de produção e consumo mais tarde ou mais cedo como exemplificado na Figura 6. Supõem-se duas bateladas da tarefa produtora *TA*, cujo tamanho do lote é de 40 unidades, e quatro bateladas da tarefa consumidora *TB*, cujo tamanho do lote é de 15 unidades, e uma mudança (redução) no *LFT* da segunda batelada produtora. A propagação desta redução para a primeira batelada origina um perfil de produção mais tarde que, junto com o perfil de consumo mais tarde, implica ultrapassar a capacidade de armazenagem. Alguns *LFT* das bateladas consumidoras devem ser reduzidos, ou, em outras palavras, alguns *LFT* das bateladas consumidoras não são possíveis.

Os dois últimos exemplos mostram que as condições de armazenagem podem introduzir restrições, que não são detectadas quando a propagação é feita utilizando apenas restrições de precedência. Torna-se necessária uma propagação baseada em balanços de massa (receita de produção) e mais abrangente.

Nas técnicas de Programação por Restrições, a propagação de restrições através da receita é feita utilizando relações temporais entre *EST* e *LFT* de bateladas produtoras e consumidoras. Trata-se basicamente de relações de precedência que podem ser obtidas a partir do procedimento de *edge-finding*. Curiosamente, essas relações temporais são dados de entrada nos sistemas atualmente disponíveis (Ilog, 1997), devendo ser completados com uma fase de pré-processamento (Das et al., 1998; Das et al., 1999) que forneça estes dados. Os dois primeiros passos na fase de planejamento aqui proposta têm essa finalidade. Estes mecanismos de propagação de restrições têm sido implementados em um sistema interativo de planejamento e programação da produção.

2.2. Fase de programação da produção

Em Rodrigues et al. (2000), é discutido o uso de duas abordagens para o problema de programação da produção que usam os resultados da fase de planejamento: uma formulação mista (Mixed Integer Linear Problem - MILP) com discretização uniforme do tempo e uma aplicação de Simulated Annealing. No presente trabalho utilizou-se a Programação por Restrições também na fase de programação da produção, devido à sua simplicidade. Note que durante a fase de planejamento executou-se uma análise de ordenamentos entre bateladas de tarefas diferentes alocadas ao mesmo equipamento. A partir deste procedimento, é possível identificar disjunções, ou seja, situações onde não foi definido um ordenamento entre bateladas. Esta fase consiste em, sucessivamente, definir um ordenamento entre um par de bateladas disjuntas e repetir o procedimento de propagação das restrições apresentados na fase de planejamento. Para evitar a escolha aleatória das disjunções, utiliza-se a informação do carregamento dos processadores para identificar qual par de bateladas disjuntas será ordenada (Rodrigues et al., 2000).

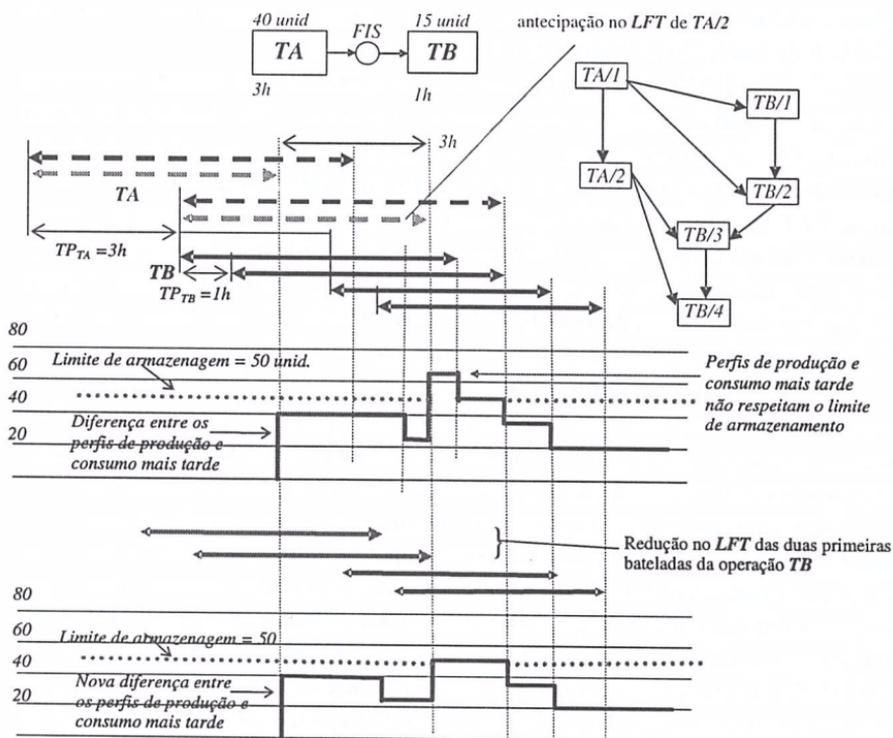


FIGURA 6. Propagação da restrição FIS.

3. Solução de um exemplo

O problema possui sete processadores que são usados na produção de três produtos finais. A figura 7 apresenta a receita de produção de cada um dos produtos finais usando a representação STN. A tabela 1 apresenta a demanda de produtos finais adotada no exemplo. A tabela 2 apresenta a atribuição de operações a processadores e a política de armazenagem em cada um dos estados.

TABELA 1. Demandas para o exemplo.

Produto Final	Demanda	Prazo de conclusão
ProA	100	32
	90	56
ProB	270	32
ProC	75	56

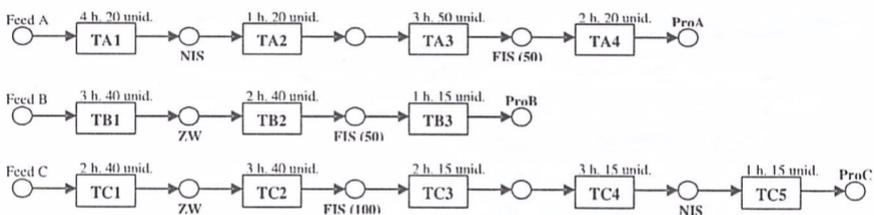


FIGURA 7. Representação STN da receita do exemplo tratado.

TABELA 2. Dados de entrada

processador	operação	tamanho da batelada	estados gerados pela operação	política de armazenagem
P1	TA1	20	A1	NIS
P1	TC1	40	C1	ZW
P2	TA2	20	A2	
P2	TA4	20	ProA	
P3 e P4	TB1	40	B1	ZW
P4	TC4	15	C4	NIS
P5	TB2	40	B2	FIS (50)
P5	TC2	40	C2	FIS (100)
P6	TB3	15	ProB	
P6	TC3	15	C3	
P7	TA3	50	A3	FIS (50)
P7	TC5	15	ProC	

Para o exemplo dado, a figura 8 apresenta as janelas de tempo e o carregamento dos processadores resultantes da propagação de restrições. Na figura 9, é apresentada a solução ótima para a programação da produção.

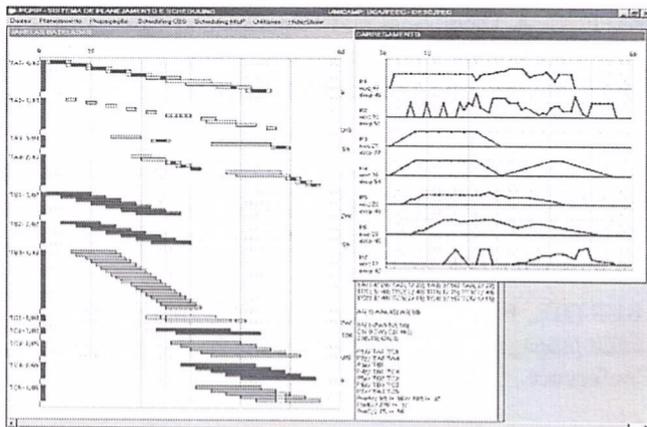


FIGURA 8
Janelas de tempo e carregamentos de processadores para o exemplo dado.

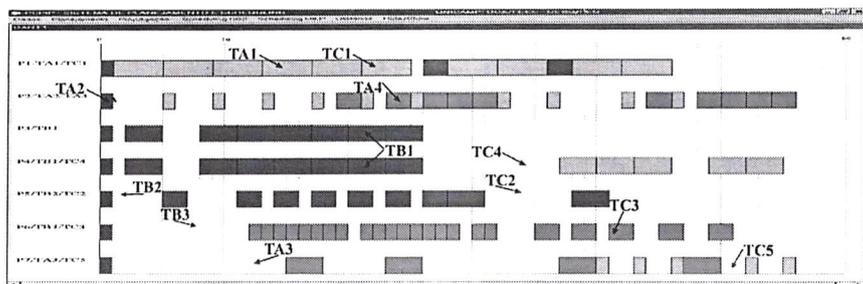


FIGURA 9. Solução global.

4. Conclusões

O trabalho desenvolvido tem como objetivo principal apresentar uma nova abordagem para a solução de problemas de planejamento e programação da produção, a Programação por Restrições. O sistema de planejamento implementado permite que o usuário tenha informação sobre a factibilidade do cenário de produção gerado e sobre o carregamento dos processadores decorrente deste cenário. A possibilidade de interação do usuário com o sistema permite que novos cenários de produção sejam gerados sempre que for detectada alguma infactibilidade ou quando o usuário achar conveniente. Também foi apresentada uma nova visão da programação da produção em indústrias de manufatura, incorporando restrições de armazenagem, permitindo a simulação tanto de problemas *jobshop* como *flowshop*.

5. Referências

- BAKER, K.R. Requirements Planning. **Handbooks in OR & MS**. Nova York: Ed. Elsevier, 1993.
- BAPTISTE, P.; LE PAPE, C. A Theoretical and Experimental Comparison of Constraint Propagation Techniques for Disjunctive Scheduling. **Proceedings of 14th International Joint Conference on Artificial Intelligence**. Montreal: Canadá, 1995
- BASSET, M.H.; DAVE, P.; DOYLE III, F.J.; KUDVA, G.K.; PEKNY, J.F.; REKLAITIS, G.V.; SUBRAHMANYAM, S.; MILLER, D.L.; ZENTNER, M.G. Perspectives on model based integration of process operations. **Computers and Chemical Engineering**, vol. 20, 1996, pp. 821-844.
- CASEAU, Y.; e LABURTHER, F. Improving branch and bound for job shop scheduling with constraint propagation. **Proceedings of the 8th Franco-Japanese 4th Franco-Chinese Conference**, 1995.

DAS, B.P.; SHAH, N.; CHUNG, P.W.H. Off-line scheduling a simple chemical batch process production plant using the ILOG scheduler. **Computers and Chemical Engineering**, vol. 22, 1998, pp. S947-S950.

DAS B.P.; SHAH, N.; CHUNG, P.W.H.; HUANG, W. A comparative study of time-based and activity based approaches to multipurpose multistage batch process production scheduling. **Proceedings of PRES'99**, 1999, pp. 229-234.

Ilog, **Scheduler 4.0. User's Manual**. Mountain View-EUA: ILOG, 1997.

KONDILI, E.; PANTELIDES, C.C.; SARGENT, R.W.H. A general algorithm for short term scheduling of batch operations – I. MILP formulation. **Computers and Chemical Engineering**, vol. 17, 1993, pp. 211-227.

LE PAPE, C. Implementation of resource constraints in ILOG Scheduler: A library for the development of constrained based scheduling systems. **Intelligent Systems Engineering**, vol. 3, 1994, pp. 55-66.

ORLICKY, J. **Material Requirements Planning**. Nova York-EUA: McGraw-Hill, 1975.

RODRIGUES, L.C.A.; GRAELLS, M.; CANTÓN, J.; GIMENO, L.; RODRIGUES, M.T.M.; ESPUÑA, A.; PUIGJANER, L. Utilization of processing time windows to enhance planning and scheduling in short term multipurpose batch plants. **Computers and Chemical Engineering**, vol. 24, 2000, pp. 353-359.

SLACK, N.; CHAMBERS, S.; JOHNSTON, R. **Administração da produção**. 2. ed. São Paulo: Atlas, 2002.

SUBRAHMANYAM, S.; PEKNY, J.F.; REKLAITIS, G.V. Decomposition approaches to batch plant design and scheduling. **Industrial Engineering and Chemical Research**, vol. 35, 1996, pp. 1866-1876.

6. Nota de Responsabilidade

O autor é o único responsável pelo material impresso incluído neste artigo.