

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO
DEPARTAMENTO DE ENGENHARIA DE COMPUTAÇÃO E
AUTOMAÇÃO INDUSTRIAL

“SEQÜENCIAMENTO DE PLANTAS MULTIPROPÓSITO COM
TEMPOS DE PREPARAÇÃO DEPENDENTES DA SEQÜÊNCIA
UTILIZANDO A REPRESENTAÇÃO STN”

Autor: Luiz Carlos de Abreu Rodrigues

Orientador: Prof. Dr. Lluís Gimeno Latre

Dissertação apresentada à Faculdade de Engenharia
Elétrica e de Computação como parte dos requisitos
exigidos para a obtenção do título de Mestre em
Engenharia Elétrica.

Novembro de 1996

Campinas - SP

Este exemplar corresponde à redação final da tese
defendida por LUIZ CARLOS DE ABREU
RODRIGUES e aprovada pela Comissão
Julgadora em 21 / 11 / 1996

Orientador

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA - BAE - UNICAMP

R618s

Rodrigues, Luiz Carlos de Abreu

Seqüenciamento de plantas multipropósito com tempos de preparação dependentes da seqüência utilizando a representação STN / Luiz Carlos de Abreu Rodrigues.-- Campinas, SP: [s.n.], 1996.

Orientador: Lluís Gimeno Latre.

Dissertação (mestrado) - Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Produção - Seqüenciamento. 2. Administração da produção. 3. Processos de fabricação. I. Latre, Lluís Gimeno. II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. III. Título.

AGRADECIMENTOS

À minha esposa Regina e à minha filha Sarah, fontes constantes de alegria na minha vida.

Ao meu pai Delmo, minha mãe Teresinha e aos meus irmãos, por todo o incentivo.

Ao prof. Dr. Lluís Gimeno Latre, pela orientação e ensino dados.

À prof. Dra. Maria Teresa Rodrigues e ao pesquisador Dr. Carlos Alberto Passos, pelos conselhos e críticas construtivas a este trabalho.

A todos os (e as) colegas, professores e funcionários do Desq que me acolheram tão bem durante o meu mestrado.

SUMÁRIO

Resumo

1 - Introdução	1
2 - Modelamento de plantas multipropósito através da Representação Estado-Tarefa	
2.1 - Introdução	5
2.2 - Representação Rede Estado-Tarefa	5
2.3 - Modelamento do problema na Representação STN	8
2.4 - Preparação do processador (<i>Set-up</i>)	17
2.5 - Exemplo de aplicação	20
2.6 - Conclusão	25
3 - Modificações do modelamento Rede Estado-Tarefa para a diminuição da dimensão em problemas de sequenciamento	
3.1 - Introdução	27
3.2 - Janelas de tempo para o processamento dos produtos	28
3.3 - Individualização das bateladas	28
3.4 - Exemplo	29
3.5 - Inclusão de janelas e bateladas no modelamento	35
3.6 - Modelamento da alocação das operações de preparação	41
3.7 - Comparação das duas formulações da alocação das operações de preparação dependentes da sequência	45
4 - Estudos de caso	
4.1 - Descrição do problema de Egli e Rippin (1986)	52

4.2 - Procedimento heurístico utilizado por Egli e Rippin (1986)	59
4.3 - Solução utilizando a Representação Estado-Tarefa (STN)	62
5 - Conclusões	71
Anexo 1 - Rotina do problema do capítulo 4 no GAMS/OSL	72
Anexo 2 - Nomenclatura utilizada no GAMS/OSL	101
Anexo 3 - Modelamento da entrada e saída contínua de produtos	103
Anexo 4 - Modelamento da manutenção preventiva	105
Referências Bibliográficas	107

RESUMO

Com a crescente necessidade de reduzir custos e prazos, as indústrias têm procurado aperfeiçoar os seus métodos de produção, bem como o seu planejamento da produção. Diante da necessidade de gerenciar melhor a produção, aproveitando melhor a capacidade instalada das indústrias, reduzindo os custos com estoque de matérias primas e com capital de giro, vários autores têm apresentado trabalhos relacionados ao planejamento e seqüenciamento da produção. Dentre estes trabalhos há o de Egli e Rippin (1986) que apresenta um problema, relacionado à indústria química, que foi solucionado pelos autores através de um método heurístico. Este problema foi abordado recentemente em uma dissertação de mestrado (Medeiros, 1995) na Faculdade de Engenharia Química da Unicamp, e é novamente abordado nesta dissertação de mestrado.

Kondili *et al.*(1993) apresentaram a Representação Estado-Tarefa (STN), aperfeiçoada por Shah *et al.*(1993), que permite tratar simultaneamente o planejamento e seqüenciamento da produção de plantas multipropósito, e que torna possível tratar muitas das situações complexas encontradas na prática nas indústrias químicas. Esta formulação baseia-se na representação discreta do tempo, no qual o horizonte de planejamento é dividido em um número de intervalos de igual duração, resultando em um problema linear inteiro misto (MILP). A principal deficiência desta formulação está na dimensão do problema MILP resultante, especialmente se houver a necessidade de alocação de operações de preparação com tempos de preparação dependentes da seqüência. Neste caso, o problema normalmente torna-se intratável, já que o número de equações geradas é proporcional ao quadrado do horizonte de planejamento.

O objetivo desta dissertação é o de apresentar uma nova formulação para as operações de preparação dependentes da seqüência, que permita resolver o problema proposto por Egli e Rippin (1986), utilizando a representação Estado-Tarefa (STN). Com isto é rerepresentada a formulação STN para que, posteriormente, seja apresentada uma nova formulação para tratar a preparação dependente da seqüência. Esta nova formulação é, então, utilizada para resolver o problema proposto por Egli e Rippin (1986), solucionado anteriormente apenas por métodos heurísticos.

CAPÍTULO 1- INTRODUÇÃO

As indústrias (plantas) químicas funcionam hoje em operação contínua ou em bateladas. Produtos produzidos em grandes quantidades, com mercado estável ou em expansão, favorecem a adoção de grandes plantas contínuas. Neste caso, todos os componentes do processo são bem conhecidos e a interação destes componentes é fixada na fase de projeto, determinando-se, entre outras coisas, a capacidade de produção para os produtos. Nestas plantas, a produção é indicada, em geral, pela taxa de produção, p. ex. kg/h, havendo entrada e saída contínua de material. Além disso, como as tarefas do processo são bem conhecidas, é possível gerar um projeto bem balanceado da planta e de seus equipamentos. Indústrias petroquímicas, indústrias de química de base (como amônia, ácido sulfúrico etc.), indústrias de fertilizantes e refinarias de petróleo são exemplos de plantas contínuas.

Quando o mercado de um produto tem um comportamento incerto, sofrendo variações de demanda com o tempo, como ocorre com muitos produtos de grande valor agregado, é favorecida a utilização de plantas com operação em bateladas. Na operação em bateladas, a produção é descontínua, tendo tempo de processamento conhecido e o tamanho das bateladas expresso em massa ou volume. Neste caso, a demanda inicial é normalmente pequena, crescendo de acordo com a procura. Além disso, o ciclo de vida dos produtos pode ser incerto, e com pico de demanda difícil de determinar. Tudo isso, força a planta a ser flexível, acomodando-se rapidamente às mudanças de demanda. As indústrias de aço, corantes, herbicidas, bioquímica, alimentos e farmacêutica são exemplos de plantas com operação em bateladas. Segundo Rippin (1991), a maioria dos produtos químicos (em número) são produzidos em bateladas.

Uma boa parte das atenções na produção em bateladas são dedicadas a realocação de produtos e tarefas, e a reconfiguração dos equipamentos em resposta às novas circunstâncias. Por isso, o processamento em bateladas requer o planejamento do tamanho das bateladas dos produtos a serem produzidos e a coordenação da alocação das operações ao longo do tempo.

A maneira como cada planta é operada é fortemente influenciada pela natureza das demandas impostas a esta planta. Se esta planta dispuser de previsões de demanda de longo prazo, ela poderá ser operada em “campanhas”, com todos os recursos da planta

sendo dedicados a um pequeno subconjunto dos produtos por relativamente longos períodos de tempo. O gerenciamento e controle da planta tornam-se mais simples, reduzindo-se ao mínimo as operações de preparação do processador (*set ups*). Um ciclo padrão das operações é estabelecido em cada campanha, com muitas bateladas de cada produto sendo produzidas em seqüência.

Se não houver previsões confiáveis da demanda, a produção é dirigida para atender as vendas disponíveis (e conseqüentemente os seus prazos de entrega). Com isso, o horizonte de planejamento passa a ser relativamente curto, para que os pedidos recebidos sejam planejados para serem produzidos no horizonte de planejamento subsequente. Nestes horizontes de planejamento relativamente curtos não é possível estabelecer nenhum padrão regular de operação, como ocorre nas campanhas, gerando-se o problema de planejamento e seqüenciamento da produção conhecido como “*short-term scheduling*”.

Quando vários produtos devem ser produzidos na mesma planta, o arranjo mais simples de ser feito é a configuração multiproduto (ou *flowshop*), onde os produtos são produzidos seqüencialmente, sendo que todos os produtos percorrem todos os processadores na mesma ordem. No *flowshop* com seqüências de permutação exige-se que a solução seja tal que em cada processador a ordem em que os produtos o ocupam seja sempre a mesma. Isto faz com que a solução do problema possa ser dada em termos de seqüência de produtos, não sendo necessário especificar a seqüência de operações para cada processador porque esta será igual para todos eles.

Nas plantas com configuração multipropósito (ou *jobshop*), os produtos podem seguir por rotas diferentes, sendo que um produto pode ter mais de uma rota de produção dentro da planta e utilizar um subconjunto específico de processadores. Neste caso, é preciso decidir quando uma tarefa de um produto será alocada e em qual processador ela será alocada.

As indústrias dispõem hoje de três “filosofias” de administração da produção (MRP II, *Just in time* e OPT) que visam aumentar a competitividade das empresas. Dentre estas três “filosofias”, a mais utilizada é o MRP II [(Dilworth, 1992) e (Corrêa e Giansi, 1996)].

O sistema MRP II tem como prioridade a redução dos estoques de produtos intermediários e finais. Este sistema assume que há certa capacidade em excesso na planta, e que os *lead-times* (tempo gasto entre o recebimento do pedido e a sua

conclusão) são fixos e conhecidos. Não faz parte da “filosofia” original deste sistema gerar o seqüenciamento da produção (carta de Gantt) e verificar se há recursos disponíveis para que a produção seja atendida no prazo, o que freqüentemente causa a alocação de recursos (como mão de obra ou processadores), que não podem ser compartilhados, para mais de uma tarefa em um dado instante. Isto ocorre porque o planejamento da produção é feito, considerando-se capacidade infinita da planta e os *lead-times* como um dado de entrada do sistema. Para que seja possível gerar uma carta de Gantt factível, mantendo-se os estoques baixos, é preciso complementar o MRP II com algum sistema de planejamento e seqüenciamento da produção.

O sistema *Just in time* tem como prioridade a eliminação (se for possível) dos estoques de produtos intermediários e finais, assumindo as premissas adotadas na configuração multiproduto (*flowshop*), com a produção sendo feita através de “linhas” de produção. Este sistema de administração da produção depende de um bom balanceamento das “linhas” de produção.

O sistema OPT tem como prioridades o aumento do fluxo de produção, redução dos estoques e redução das despesas operacionais. Neste sistema assume-se que todos os gargalos governam tanto o fluxo de produção como os estoques e, portanto, devem ser tratados especialmente. Além disso, assume-se que há certa capacidade em excesso dos recursos não-gargalo. A grande dificuldade deste sistema está na determinação dos gargalos, já que nem sempre eles são conhecidos, ou mesmo facilmente identificáveis.

Apesar de estes três sistemas de administração da produção serem utilizados pelas indústrias com processamento em bateladas, eles nem sempre dão o resultado desejado, já que não são capazes de garantir a factibilidade da solução obtida. Por isso, a interação entre estes sistemas e o seqüenciamento da produção (*scheduling*) tornaria estes sistemas mais confiáveis, garantindo-se que a resposta obtida é factível.

Neste trabalho são apresentadas técnicas de seqüenciamento da produção para plantas descontínuas com *set ups* dependentes da seqüência de produção, utilizando a representação Rede Estado-Tarefa (*State-Task Network*) (Kondili *et al.*, 1993). A formulação desenvolvida aplica-se a plantas químicas multipropósito, com processamento em bateladas, com restrições sobre os recursos compartilhados e com os tempos de preparação dependentes da seqüência.

No capítulo 2 é apresentada a representação Rede Estado-Tarefa (STN), desenvolvida por Kondili *et al.* (1993), bem como o seu modelamento. Ainda neste

capítulo é apresentada a formulação MILP, desenvolvida pelos mesmos autores, para a preparação do processador (*set up*) e é mostrado um exemplo de aplicação desta representação STN.

No capítulo 3 são apresentadas modificações no modelamento da representação STN, para a diminuição da dimensão em problemas de seqüenciamento com preparação do processador (*set up*) dependente da seqüência. Primeiro, são mostradas as janelas de tempo para o processamento dos produtos e a individualização das bateladas, que são introduzidos neste novo modelamento. A seguir, é apresentado o modelamento da alocação das operações de preparação. Por fim, é feita uma comparação entre as formulações dos dois capítulos para a alocação das operações de preparação dependentes da seqüência.

No capítulo 4, é apresentado o problema proposto por Egli e Rippin (1986). A seguir, é feita uma apresentação sucinta da heurística apresentada por estes autores. Finalmente, é dada a solução, através do programa GAMS/OSL, utilizando a representação STN.

No capítulo 5 são apresentadas as conclusões deste trabalho. No anexo 1 é apresentada a rotina utilizada no GAMS/OSL. No anexo 2 é apresentada a nomenclatura utilizada no GAMS/OSL. No anexo 3 são apresentadas as alterações no modelamento da representação STN quando se considera a entrada (alimentação) contínua de matérias primas e a saída contínua de produtos. No anexo 4 são apresentadas as alterações no modelamento da representação STN quando se considera a realização de manutenção ou preparação dependente da freqüência nos processadores.

CAPÍTULO 2

MODELAMENTO DE PLANTAS MULTIPROPÓSITO ATRAVÉS DA REPRESENTAÇÃO REDE ESTADO-TAREFA (STN - *State Task Network*)

2.1. Introdução

Neste capítulo descreve-se a representação Rede Estado-Tarefa (STN - *State Task Network*) apresentada por Kondili *et al.*(1993) que será utilizada neste trabalho. São apresentados os aspectos mais significativos, remetendo o leitor à referência acima para descrição completa.

No capítulo 3 será apresentada uma modificação deste modelamento para tratar de uma forma mais eficiente os tempos de preparação dependentes da sequência.

2.2. Representação Rede Estado-Tarefa

A Representação Rede Estado-Tarefa (STN) foi desenvolvida originalmente para descrever processos químicos complexos, mas a sua forma clara de representar processos produtivos, permite que esta representação seja usada na indústria em geral, tanto em processos contínuos, semi-contínuos ou batelada.

Na rede Estado-Tarefa (STN) existem dois tipos de nós: os nós Estado (*State*), que representam as matérias primas, os produtos intermediários e os produtos finais, e os nós Tarefa (*Task*), que representam as operações de processamento, as quais transformam matérias primas vindas de um ou mais estados de entrada (que alimentam a tarefa) para estados de saída (que são produtos da tarefa realizada). Os nós Estado são representados por círculos e os nós Tarefa por retângulos. Isto permite que não hajam ambigüidades na representação dos processos produtivos, como pode acontecer quando apenas se representam as operações de processamento.

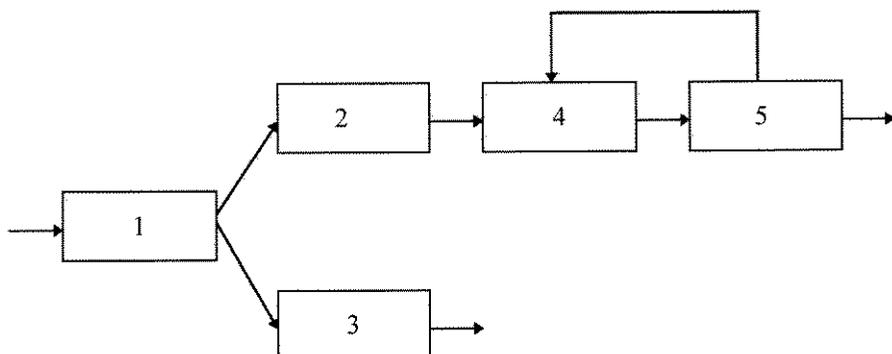


Fig. 2.1

Na figura 2.1, é dado um exemplo de um processo representado apenas por suas receitas operacionais (tarefas). No processo da figura 2.1, observa-se que não está claro se a tarefa 1 gera dois produtos diferentes que são consumidos pelas tarefas 2 e 3, respectivamente, ou se a tarefa 1 gera somente um produto que é consumido pelas tarefas 2 e 3. Também não está claro, nesta figura, se a tarefa 4 consome duas matérias primas diferentes, vindas das tarefas 2 e 5, ou se a tarefa 4 consome somente uma matéria prima, gerada tanto pela tarefa 2 como pela tarefa 5.

A Figura 2.2 mostra duas representações STN diferentes que eliminam as ambigüidades apresentadas pela representação por receitas operacionais (tarefas), dada na figura 2.1. No processo representado pela figura 2.2a, a tarefa 1 gera um único produto que é compartilhado entre as tarefas 2 e 3. A tarefa 4, por sua vez requer uma única matéria prima que é produzida tanto pela tarefa 2 e 5. No entanto, no processo mostrado na figura 2.2b, a tarefa 1 gera dois produtos diferentes que alimentam as tarefas 2 e 3, respectivamente, e a tarefa 4 tem agora duas matérias primas diferentes produzidas pelas tarefas 2 e 5, respectivamente.

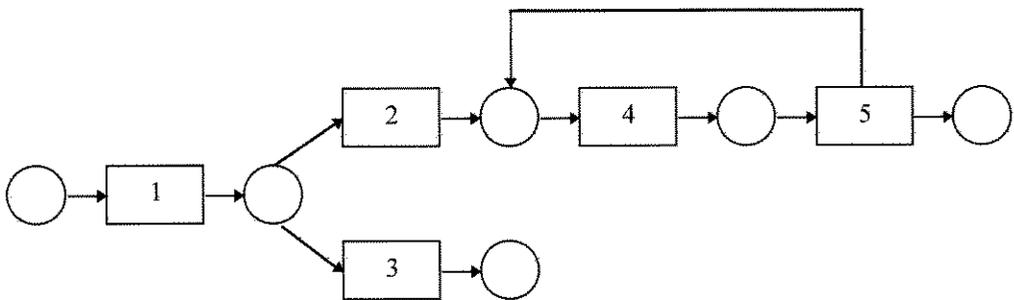


Fig 2.2a

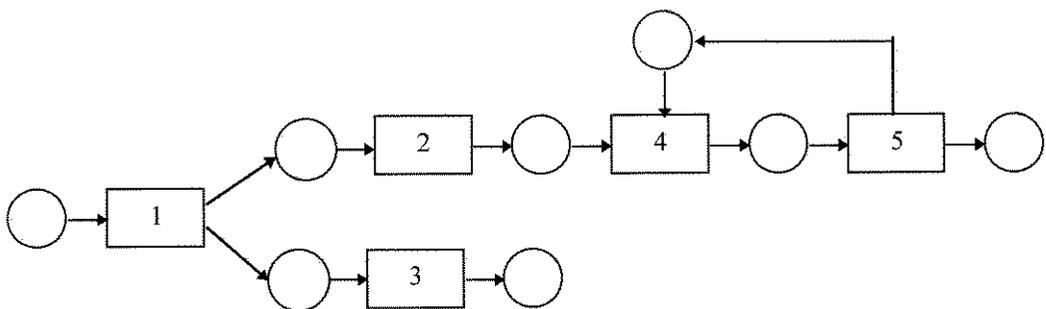


Fig. 2.2b

As regras para a construção de uma representação STN são:

1. Uma tarefa tem a quantidade de estados de entrada (saída) igual ao número de materiais diferentes consumidos (gerados) pela tarefa.

2. Duas ou mais entradas (saídas) em um estado são necessariamente de materiais iguais.

É importante ressaltar que a representação STN não gera necessariamente gráficos interconectados. Muitos casos de planejamento e sequenciamento de produção (*Scheduling*) envolvem a produção de um número de produtos que apesar de unidos pelo consumo dos mesmos recursos da planta, não compartilham as mesmas matérias primas ou os mesmos produtos intermediários. A representação STN para tais processos consiste em uma série de subgrafos disjuntos.

Além disso, a definição da representação STN e a formulação do planejamento e sequenciamento de produção a serem apresentados a seguir, são suficientemente genéricos na definição de uma Tarefa. Na prática, muitas operações de processamento podem ser subdivididas em uma quantidade de operações menores. Por exemplo, uma operação de reação em um processador químico, pode consistir de uma etapa de aquecimento, uma etapa de reação e uma etapa de resfriamento. Como todas estas etapas são realizadas na mesma unidade de processamento, sem interrupções entre elas, então é conveniente representar todas estas etapas como sendo uma única tarefa.

Aos produtos intermediários podem ser associadas variáveis representando a quantidade de material estocada, que podem seguir quatro políticas diferentes de estocagem.

1. **UIS** (*Unlimited Intermediate Storage*) - Indica que este estado pode ser armazenado em um local de estoque com capacidade “infinita” que permite que haja um estoque ilimitado do produto.
2. **FIS** (*Finite Intermediate Storage*) - Indica que este estado pode ser armazenado em um local de estoque com capacidade de estocagem limitada.
3. **NIS** (*No Intermediate Storage*) - Indica que neste estado não há capacidade de estocagem, sendo que neste caso a estocagem só poderá ser feita no próprio processador (que não estará disponível para processar outra tarefa).
4. **ZW** (*Zero Wait*) - Indica que o produto neste estado não pode ser estocado, por exemplo porque este produto é instável ou porque a planta utiliza a “filosofia de produção” *just-in-time*.

É importante enfatizar que a representação STN pode ser usada para representar o processo físico, propriamente dito, independente do tipo de problema que se deseja resolver. As unidades de processamento e a sua conectividade não são mostradas explicitamente neste tipo de representação, assim como nenhum outro recurso disponível é mostrado.

No caso de problemas de planejamento e sequenciamento de produção, assume-se que uma tarefa recebe material de seus estados de alimentação, em proporções fixas e conhecidas, e que esta tarefa produz material para os seus estados de saída, também em proporções fixas e conhecidas. Os tempos de processamento de cada tarefa são considerados independentes da quantidade de material a ser processado e são previamente conhecidos. No entanto, produtos diferentes de uma mesma tarefa podem ter tempos de processamento diferentes (por exemplo, uma tarefa de destilação pode gerar vários produtos e com cada um destes produtos sendo gerado após um dado tempo de processamento).

2.3. Modelamento do problema na representação STN

Uma consideração básica em qualquer programa de sequenciamento e planejamento de produção (*Scheduling*) refere-se à representação do tempo. A formulação a ser utilizada baseia-se numa representação discreta do tempo definida *a priori*. O horizonte de tempo de interesse é dividido em um número de intervalos de igual duração, chamados de “intervalos de tempo” (*slots*) t . Eventos de qualquer tipo, tal como início ou fim de processamento de uma tarefa, mudanças na disponibilidade de um processador e de outros recursos etc, só são permitidos nos extremos desses intervalos de tempo (*slots*).

A principal vantagem deste tipo de representação do tempo é que ela facilita a formulação por criar uma “grade de referência”, de intervalos regulares, na qual todas as operações (tarefas e operações de preparação) que competem por recursos compartilhados são posicionadas. Na prática, a duração de um intervalo de tempo (*slot*) é obtida a partir do máximo divisor comum de todos os tempos de processamento e preparação dos equipamentos envolvidos no problema.

Na tabela 2.1 são apresentados os índices usados nas equações deste capítulo.

Tabela 2.1. Índices usados.

<i>i</i>	tarefas.
<i>j</i>	processadores.
<i>s</i>	estados.
<i>t</i>	intervalos de tempo (<i>slots</i>).
<i>u</i>	recursos utilizados.

A seguir são mostradas as equações apresentadas na formulação de *Kondili et al.* (1993), para o modelamento do problema usando a representação STN. Na apresentação da formulação dada a seguir, sempre que for necessário, será apresentada uma tabela contendo todos os termos (Variáveis, dados das equações e conjuntos utilizados na limitação do domínio das equações) ainda não definidos anteriormente.

a) Equações de Balanço de Massa

O balanço de massa é dado pela equação [2.1] abaixo, e na tabela 2.2 é apresentada a nomenclatura utilizada.

$$S_{st} = S_{s,t-1} + \sum_{i \in OG_s} \rho'_{is} \sum_{j \in J_i} M_{j,i,t-\rho_{is}} - \sum_{i \in OC_s} \rho_{is} \sum_{j \in J_i} M_{jit} + R_{st} - D_{st}, \text{ para } \forall s \text{ e } t. \quad [2.1]$$

Tabela 2.2. Nomenclatura da equação [2.1]

Variáveis das equações:

S_{st}	Estoque do estado <i>s</i> no intervalo de tempo (<i>slot</i>) <i>t</i> .
M_{jit}	Quantidade de material que a tarefa <i>i</i> começou a processar no <i>slot</i> <i>t</i> no processador <i>j</i> .
D_{st}	Quantidade de produto <i>s</i> retirada no intervalo de tempo (<i>slot</i>) <i>t</i> para ser entregue ao cliente.
R_{st}	Quantidade de produto <i>s</i> recebida por fornecimento externo à planta no intervalo de tempo (<i>slot</i>) <i>t</i> .

Dados das equações:

ρ'_{is}	Proporção (fração mássica) do produto <i>s</i> na saída da tarefa <i>i</i> .
ρ_{is}	Proporção (fração mássica) do produto <i>s</i> na entrada da tarefa <i>i</i> .
ρ_{is}	Tempo de processamento da tarefa <i>i</i> para gerar o produto <i>s</i> .

Conjuntos utilizados na limitação do domínio das equações:

OG_s	Conjunto de tarefas produzindo o estado (produto) <i>s</i> .
OC_s	Conjunto de tarefas recebendo o estado (matéria prima) <i>s</i> .
J_i	Conjunto dos processadores que podem realizar a tarefa <i>i</i> .

Esta equação, que é escrita para todos os estados s e para todos os $slots$ t , indica que a variação no estoque de material ($S_{s,t} - S_{s,t-1}$) no estado s , no intervalo de tempo ($slot$) t , é dado pela diferença entre a quantidade produzida e a quantidade consumida de material.

Os somatórios na equação [2.1] servem para discriminar os termos que devem ser considerados no balanço de massa. Assim, o somatório $(\rho_{is} * M_{jit})$ indica a quantidade de material que está sendo retirada do estado s para ser usada na tarefa $i \in OC_s$ no $slot$ t no processador $j \in J_i$. O somatório $(\rho'_{i,s} * M_{j,i,t-p_{is}})$ indica a quantidade de material que está sendo recebida no estado s proveniente da tarefa $i \in OG_s$ no $slot$ t (dado que ela se iniciou no $slot$ $t-p_{is}$), executada no processador $j \in J_i$.

b) Equação de não coexistência

Em qualquer intervalo de tempo ($slot$) t , pode haver no máximo uma única tarefa i sendo realizada no processador j . Esta restrição é atendida através da equação [2.2]:

$$\sum_{i \in I_j} \sum_{t'=t}^{t+pt_i-1} W_{jit'} - 1 \leq U(1 - W_{jit}) \quad , \text{ para } \forall j, i \in I_j, t \in H. \quad [2.2].$$

Ou alternativamente, pela equação [2.3] abaixo:

$$\sum_{i \in I_j} \sum_{t'=t}^{t-pt_i+1} W_{jit'} \leq 1 \quad , \text{ para } \forall j, t \in H. \quad [2.3].$$

Tabela 2.3. Nomenclatura das equações [2.2] e [2.3].

Variáveis da equação:

W_{jit} = 1, Indica que o processamento da tarefa i no processador j iniciou-se no intervalo de tempo ($slot$) t ; = 0, Caso contrário.

Dados da equação:

pt_i Tempo total de processamento da tarefa i . $pt_i \equiv \max_{s \in SG_i} (p_{is})$.

U Constante com um valor positivo suficientemente grande (Shah *et al.*, 1993).

Conjuntos utilizados na limitação do domínio das equações:

H Conjunto de todos os intervalos de tempo ($slots$) exceto o último intervalo de tempo ($slot$). No último $slot$, não há nenhum processamento de tarefa, já que este $slot$ está fora do horizonte de planejamento e só é usado para representar os estoques finais. H representa o horizonte de planejamento, ou seja, o período de tempo em que deseja planejar a produção.

I_j Conjunto das tarefas que podem ser executadas no processador j .

SG_i Conjunto dos estados (produtos) produzidos pela tarefa i .

A equação [2.2] só está ativa como restrição se $W_{jit} = 1$, no lado direito desta equação. Neste caso, o somatório de W_{jit} , tem que ser menor ou igual a 1, e dado que $W_{jit} = 1$, então todos os demais W_{jit} , têm valor igual a zero.

Sendo W_{jit} uma variável binária, a equação [2.2] garante que cada tarefa $i \in I_j$ estará alocada a apenas um intervalo de tempo (*slot*), o qual corresponde ao *slot* de início do processamento desta tarefa i . Caso contrário, $W_{jit} = 0$, a constante grande $U \geq 0$ satisfaz a restrição.

A equação de não coexistência pode ser escrita de diversas maneiras que podem levar a um algoritmo mais eficiente. *Shah et al.* (1993) apresentaram uma formulação alternativa para a equação de não coexistência (equação [2.3]) que resultou numa redução considerável do *gap* de integralidade em relação à equação [2.2].

A equação [2.3] é escrita para cada processador e cada *slot* no horizonte de planejamento. O início de processamento em um *slot* t no processador j de uma tarefa $i \in I_j$ implica que nenhuma outra tarefa poderá ter início neste processador se o seu processamento ocupar o *slot* t . Portanto, assim como a equação [2.2], esta equação especifica que em qualquer *slot* dentro do horizonte de planejamento ($t \in H$) e em qualquer processador j , somente pode ser processada uma tarefa por vez, de tal forma que ou é processada uma tarefa ou o processador fica desocupado.

c) Consumo de recursos compartilhados

As tarefas a serem realizadas podem consumir recursos compartilhados, como vapor, eletricidade, água, mão de obra etc. O consumo de recursos pode variar ao longo do processamento da tarefa, e, além disso, em qualquer instante, a quantidade consumida de um recurso por uma tarefa pode ser constante ou variar proporcionalmente a quantidade de material que está sendo processada. As equações [2.4] e [2.5] representam, respectivamente, a quantidade de recurso u demandada no *slot* t , e a disponibilidade do recurso u no *slot* t .

$$U_{u,t} = \sum_i \sum_{j \in J_i} \sum_{\theta=1}^{\min(p_i, t)} (\alpha_{u,i,\theta} * W_{j,i,t-\theta+1} + \beta_{u,i,\theta} * M_{j,i,t-\theta+1}) \quad , \text{ para } \forall u, t \in H. \quad [2.4].$$

$$U_{u,t}^{\min} \leq U_{u,t} \leq U_{u,t}^{\max} \quad , \text{ para } \forall u, t \in H \quad [2.5].$$

Tabela 2.4. Nomenclatura das eq. [2.4] e [2.5]

Variáveis das equações:

$U_{u,t}$ Consumo total do recurso u (eletricidade, vapor etc.) no *slot* t .

Dados das equações:

θ Indica um intervalo de tempo (*slot*). $1 \leq \theta \leq pt_i$ e $\theta \leq t$.

$\alpha_{u,i,\theta}$ Indica quanto do recurso u é consumido no *slot* θ quando a tarefa i está sendo processada. A dependência em θ permite que o consumo seja variável ao longo do processamento da tarefa.

$\beta_{u,i,\theta}$ Indica quanto do recurso u é consumido no *slot* θ por unidade de material processado pela tarefa i . O consumo deste recurso variará de acordo com a quantidade processada.

$U_{u,t}^{min}$ Consumo mínimo disponível do recurso u no *slot* t .

$U_{u,t}^{max}$ Disponibilidade do recurso u no *slot* t .

A equação [2.4] expressa o consumo do recurso u por todas as tarefas em processamento no *slot* t . O somatório de θ de um até o mínimo entre pt_i e t ($1 \leq \theta \leq pt_i$ e $\theta \leq t$) garante que em qualquer *slot* t são consideradas todas as tarefas em processamento naquele *slot*. A equação [2.5] expressa que a disponibilidade do recurso u pode variar ao longo do horizonte de planejamento, mas o consumo deste recurso não pode exceder a disponibilidade em nenhum momento. Estas equações são escritas para todos os recursos u e todos os *slots* t dentro do horizonte de planejamento.

d) Tarefa de estocagem

Até o momento, tem-se assumido que toda capacidade de estocagem na planta é dedicada a estados individuais, ou seja, cada estado tem o seu próprio local de estoque. Entretanto, para plantas produzindo uma grande quantidade de produtos, é comum dispor de um número relativamente pequeno de tanques de estocagem, sendo cada tanque compartilhado por diversos produtos.

Esta restrição pode ser absorvida através de uma modificação da representação STN. Considere a representação STN mostrada na figura 2.3a, envolvendo material no estado S_2 , que é produzido pela tarefa I_1 e consumido pela tarefa I_2 . A fim de levar em consideração a disponibilidade de tanques de estocagem compartilhados, já que pode ser possível armazenar o produto do estado S_2 em alguns destes tanques, é produzida uma

“tarefa de estocagem”(i*) como mostra a figura 2.3b. A tarefa de estocagem recebe uma certa quantidade de material do estado S₂ e gera uma quantidade igual após um intervalo de tempo (slot).

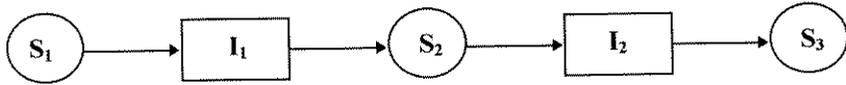


Fig. 2.3a - Representação STN original

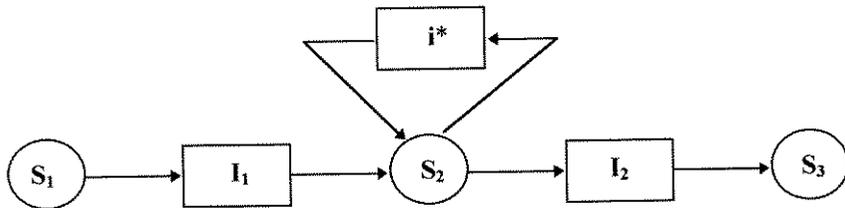


Fig. 2.3b - Representação STN modificada

A mesma modificação pode ser realizada em todos os estados que têm um conjunto de tanques de estocagem compartilhados. Este conjunto de tanques de estocagem compartilhados são, então, tratados como um processador que está disponível para realizar um subconjunto de tarefas de estocagem. Deve-se notar que, neste caso, a capacidade de estocagem C_s para cada estado s deve somente incluir a capacidade de estocagem especificamente dedicada a este estado em particular, porque somente um estado pode ocupar um tanque de estocagem em um intervalo de tempo (slot) qualquer.

Além disso, podem haver processadores habilitados a realizarem a tarefa I₁ na figura 2.3b, e que também podem realizar a tarefa de estocagem i*. A implicação prática disto é que não será mais necessário remover o produto da tarefa I₁, do processador em que foi processada esta tarefa, imediatamente após o fim do seu processamento. Ou seja, este produto pode aguardar no processador por quanto tempo for necessário, implementando assim a política “NIS - no intermediate storage”, na qual o processador pode funcionar como tanque de armazenagem, e que é utilizada em problemas envolvendo processamento em bateladas. Quando uma tarefa (I₁, por exemplo) puder ser realizada em mais de um processador, sendo possível armazenar o produto (S₂) no processador, é preciso garantir que o produto estocado permaneça no processador em que foi realizada a tarefa. Isto é garantido pela equação [2.6] abaixo:

$$M_{ji^*t} \leq M_{j,i^*,t-1} + \sum_{i \in I_j \cap OG_{s_2}} \rho'_{is_2} * M_{j,i,t-p_{is_2}}, \text{ para } \forall s_2, i^* \in OC_{s_2}, j \in J_{i^*}, t \in H. \quad [2.6].$$

A equação [2.6] garante que a quantidade de material estocado no processador j no intervalo de tempo (*slot*) t , não pode ser maior que a quantidade de material estocado no intervalo de tempo (*slot*) anterior mais a quantidade de material que acaba de ser produzida neste processador. Com isso, garante-se que o material S_2 produzido em outro processador não poderá ser estocado no processador j . Esta equação é gerada para todas as tarefas de estocagem i^* , em todos os processadores j em que estas tarefas podem ser produzidas e depois armazenadas, e em todos os *slots* t dentro do horizonte de planejamento.

e) Limitações de capacidade

A quantidade de material a ser processada em cada batelada da tarefa i no processador j pode estar limitada a uma capacidade máxima e mínima de processamento. A equação [2.7] indica como é feita a limitação da capacidade de processamento de uma batelada.

$$W_{jit} * V_{ji}^{min} \leq M_{jit} \leq W_{jit} * V_{ji}^{max} \quad , \text{ para } \forall i, j \in J_i, t \in H. \quad [2.7].$$

Por outro lado, a quantidade de material estocada em um estado s , não pode ser superior à capacidade máxima de estocagem, nem inferior à capacidade mínima de estocagem, dadas na equação [2.8]. Note que se $W_{jit} = 0$, então $M_{jit} = 0$.

$$C_s^{min} \leq S_{st} \leq C_s^{max} \quad , \text{ para } \forall s, t \in H. \quad [2.8].$$

Tabela 2.5. Nomenclatura das equações [2.7] e [2.8]

Dados das equações:

V_{ji}^{max}	Limite máximo de processamento da tarefa i no processador j .
V_{ji}^{min}	Limite mínimo de processamento da tarefa i no processador j .
C_s^{max}	Capacidade máxima de estocagem no estado s .
C_s^{min}	Capacidade mínima de estocagem no estado s .

f) Indisponibilidade temporária de um equipamento

Durante o horizonte de planejamento, certos processadores podem temporariamente tornarem-se indisponíveis devido a sua manutenção, quebra do processador etc.

Esta indisponibilidade de um processador j pode ser levada em consideração no planejamento de produção simplesmente fixando-se o valor das variáveis W_{jt} em zero durante o período em que isto ocorrer.

$$W_{jt} = 0 \quad , \text{ para } \forall i \in I_j, t = t_1 - pt_i + 1, \dots, t_2 - 1. \quad [2.9].$$

Esta equação garante que se o processador j está indisponível entre os slots t_1 e t_2 ($t_2 \geq t_1$), então, nenhuma tarefa $i \in I_j$ pode ser iniciada entre os slots t_1 e $(t_2 - 1)$. Além disso, nenhuma tarefa $i \in I_j$ poderá estar em processamento no slot t_1 , o que implica que nenhuma tarefa poderá ser iniciada após o slot $(t_1 - pt_i + 1)$. Com isso, nenhuma tarefa poderá ser iniciada entre os slots $(t_1 - pt_i + 1)$ e $(t_2 - 1)$.

g) Função Objetivo

Pode-se adotar uma série de critérios como função objetivo. A função objetivo utilizada em Kondili *et al.*(1993) é a maximização dos lucros de produção que são expressos como:

$$\text{Lucro} = \text{Valor dos Produtos [2.10a]} - \text{Despesa com as Matérias Primas [2.10b]} - \text{Despesa de Estocagem [2.10c]} - \text{Despesa com os Recursos [2.10d]} \quad [2.10].$$

Cada um dos termos da equação [2.10] acima é dado a seguir:

$$\text{Valor dos Produtos} = \sum_s (CV_{s,H+1} * S_{s,H+1} + \sum_{t=1}^H CV_{st} * D_{st}) \quad [2.10a].$$

$$\text{Despesa com as Matérias Primas} = \sum_s (CP_{s,0} * S_{s,0} + \sum_{t=1}^H CP_{st} * R_{st}) \quad [2.10b].$$

$$\text{Despesa de Estocagem} = \sum_s \sum_{t=1}^H CE_{st} * S_{st} \quad [2.10c].$$

$$\text{Despesa com os Recursos} = \sum_u \sum_{t=1}^H CU_{ut} * U_{ut} \quad [2.10d].$$

Tabela 2.6. Nomenclatura das equações [2.10a] a [2.10d]

Variáveis das equações:

Lucro Indica o lucro total de produção. Esta é a variável a ser maximizada.

Dados das equações:

CV_{st} Valor de venda do produto no estado *s* no intervalo de tempo (*slot*) *t*. O valor de venda de um produto pode ser variável no tempo de acordo com o mercado.

CP_{st} Custo da matéria prima no estado *s* no *slot t*.

CE_{st} Custo de estoque no estado *s* no *slot t*.

CU_{ut} Custo de consumo do recurso *u* no *slot t*.

A expressão [2.10a] inclui tanto o valor dos produtos entregues aos clientes ao longo do horizonte de planejamento como dos produtos deixados em estoque no final do horizonte de planejamento (*slot H + 1*). Se não for desejado ter estoque de produtos finais no final do horizonte de planejamento, pode-se atribuir um custo negativo (ou seja, uma parcela de perda) ao estoque de produtos no final do horizonte de planejamento ($CV_{s,H+1} \ll 0$), ou então, impor como restrição que o estoque de produtos no final do horizonte de planejamento seja nulo ($S_{s,H+1}=0$).

As despesas com o estoque inicial de matéria prima e com o material recebido ao longo do horizonte de planejamento são levadas em consideração na equação [2.10b]. Observe que as quantidades de R_{st} e V_{st} podem ser tanto fixadas a priori como dados do problema, ou podem ser tratadas como variáveis do problema sujeitas a limitações quanto ao seu valor máximo e mínimo.

A despesa de estocagem, equação [2.10c], apresenta os gastos com estocagem durante o processamento no horizonte de planejamento. Estes gastos podem ser decorrentes, por exemplo, da necessidade de aquecimento ou de refrigeração durante a estocagem. Além disso, pode-se considerar que o giro de capital é prejudicado pelos materiais em estoque resultando em um custo financeiro da operação de estocagem. Se não for desejado ter estoque de produtos intermediários no final do horizonte de planejamento, pode-se adotar o mesmo procedimento usado nos produtos finais, atribuindo-se um custo negativo ao estoque de intermediários no final do horizonte de planejamento ($CE_{s,H+1} \ll 0$), ou então, impondo-se que o estoque de produtos intermediários no final do horizonte de planejamento seja nulo ($S_{s,H+1}=0$).

A despesa com os recursos, equação [2.10d], apresenta os gastos com o consumo dos recursos, como eletricidade, vapor e mão de obra, ao longo do horizonte de planejamento.

2.4 Preparação do Processador (*Set-up*)

Em uma planta multipropósito é comum encontrar processadores em que é necessário realizar uma operação de preparação do processador entre o processamento de duas tarefas neste processador, devido a necessidade de limpar o equipamento ou trocar o ferramental do processador antes de iniciar a tarefa seguinte. Assim como as tarefas, uma operação de preparação tem duração fixa que pode ou não estar relacionada com a tarefa que acabou de ser processada e com a tarefa que vai ser processada, e também pode ou não consumir recursos da planta como mão de obra, eletricidade, vapor etc.

Em Kondili *et al.* (1993) considera-se que o tempo de duração da operação de preparação entre duas tarefas depende da tarefa anterior que acabou de ser processada e da tarefa posterior que ainda vai ser iniciada no mesmo processador. Ou seja, o tempo de duração da preparação é dependente da sequência.

Serão considerados aqui duas formas de tratamento da operação de preparação do processador:

- 1) Alocação.
- 2) Previsão sem alocação.

No primeiro caso, a operação de preparação é considerada como uma outra tarefa que deve ser alocada no tempo. No segundo caso, apenas garante-se que haverá um intervalo de tempo entre as duas tarefas suficiente para que a operação de preparação seja completada, mas não é especificado em que *slot* esta operação será iniciada (no caso de ser deixado entre as duas tarefas um número de *slots* maior que o necessário).

A manutenção preventiva pode também ser classificada como uma operação de preparação, mas que depende da frequência de uso do processador. A manutenção preventiva não é abordada em nenhum dos problemas estudados e por isso é apresentado no anexo 4.

a) Operação de preparação com alocação

Considera-se que as tarefas $i \in I_j$, para cada processador j , podem ser divididas em NF_j famílias de tarefas $I_j^{(k)}$, onde $k = 1, \dots, NF_j$. A equação [2.11], descrita abaixo, garante que se qualquer tarefa i_j da família k_j tem o seu processamento iniciado no

processador j no intervalo de tempo (*slot*) t_1 , e qualquer tarefa i_2 da família k_2 tem o seu processamento iniciado neste mesmo processador no intervalo de tempo (*slot*) t_2 (onde $t_2 \geq t_1$), sendo que nenhuma outra tarefa é processada entre o fim da primeira tarefa i_1 e o início da segunda tarefa i_2 , então haverá uma operação de preparação entre os intervalos de tempo t_1 e $t_2 - 1$.

$$\sum_{t=t_1+1}^{t_2-1} WS_{j,k_1,k_2,t} \geq \sum_{i_1 \in I_j^{(k_1)} \cap NH} W_{j,i_1,t_1} + \sum_{i_2 \in I_j^{(k_2)} \cap NH} W_{j,i_2,t_2} - \sum_{t=t_1+1}^{t_2-1} \sum_{i \in I_j \cap NH} W_{jit} - 1$$

, para $\forall j \in SP, k_1 \in K_j, k_2 \in K_j, t_1, t_2 \geq t_1$. [2.11].

Tabela 2.7. Nomenclatura das equações [2.11] a [2.14]

Variáveis das equações:

$WS_{j,k_1,k_2,t}$ = 1, Indica que a operação de preparação do processador j para realizar uma tarefa (i_2) da família k_2 é iniciada no *slot* t , após a realização de uma tarefa (i_1) da família k_1 ;
= 0, Caso contrário.

Dados das equações:

τ_{j,k_1,k_2} Tempo de duração da operação de preparação de uma tarefa da família k_1 , após o processamento de uma tarefa da família k_2 no processador j .
 CS_{j,k_1,k_2} Custo da operação de preparação de uma tarefa i_2 da família k_2 , após o processamento de uma tarefa i_1 da família k_1 no processador j .

Conjuntos utilizados na limitação do domínio das equações:

NH Conjunto de todas as tarefas de todos os produtos, com exceção das tarefas de armazenamento (*hold*) no próprio processador. Ou seja, as tarefas de *hold* não estão contidas neste conjunto.
SP Conjunto dos processadores onde é preciso efetuar uma operação de preparação do processador para que as tarefas possam ser processadas neles. Existem processadores que não necessitam de uma operação de preparação para realizar qualquer tarefa para a qual esteja habilitado.
 $I_j^{(k)}$ Conjunto das tarefas da família k que podem ser realizadas no processador j .
 K_j Conjunto das famílias de tarefas que podem ser realizadas no processador j .

A equação [2.11] constitui uma restrição ativa no problema quando os dois primeiros termos do lado direito desta equação forem iguais a 1 (uma tarefa $i_1 \in I_j^{(k_1)}$ é iniciada no *slot* t_1 e uma tarefa $i_2 \in I_j^{(k_2)}$ é iniciada no *slot* $t_2 \geq t_1$) e o terceiro termo do lado direito desta equação for igual a zero (nenhuma tarefa é processada entre o término

da tarefa i_1 e o início da tarefa i_2). Neste caso, a soma do lado direito desta equação será igual a 1, o que força a alocação de pelo menos uma operação de preparação. Porém, como a função objetivo deste problema é minimizar o custo de produção ou maximizar o lucro e a realização de uma operação de preparação implica em um custo, então, somente uma operação de preparação será alocada.

Se for usada a operação de preparação com alocação no planejamento de produção, será necessário alterar as equações [2.3], equação de não coexistência, e [2.10], função objetivo. A equação [2.3] será substituída pela equação [2.12], que conterá também o somatório das operações de preparação. Note que ou o processador j está processando uma tarefa $i \in I_j$ ou o processador está realizando uma operação de preparação ou o processador está desocupado no *slot* t .

$$\sum_{i \in I_j} \sum_{t'=t}^{t-p_i+1} W_{j,i,t'} + \sum_{k_1 \in K_j} \sum_{k_2 \in K_j} \sum_{t'=t}^{t-\tau_{i,k_1,k_2}+1} WS_{j,k_1,k_2,t'} \leq 1, \text{ para } \forall j, t \in H. \quad [2.12].$$

A equação [2.10] será substituída pela equação [2.13], que conterá também o custo das operações de preparação, como é mostrado abaixo:

$$\begin{aligned} \text{Lucro} = & \text{Valor dos Produtos [2.10a]} - \text{Despesa com as Matérias Primas [2.10b]} - \\ & \text{Despesa de Estocagem [2.10c]} - \text{Despesa com os Recursos [2.10d]} - \\ & \text{Despesa com as Operações de Preparação [2.13a]} \end{aligned} \quad [2.13].$$

onde:

$$\text{Despesa com Preparação} = \sum_j \sum_{k_1 \in K_j} \sum_{k_2 \in K_j} \sum_{t=1}^H CS_{j,k_1,k_2} * WS_{j,k_1,k_2,t} \quad [2.13a]$$

b) Previsão da operação de preparação sem alocação

O uso da formulação [2.11] pode gerar um grande número de variáveis e equações, já que esta formulação é proporcional a H^2 . Em muitos casos, quando os custos das operações de preparação e o consumo de recursos não são significativos, pode ser suficiente garantir o tempo para o processamento da operação de preparação sem alocação desta operação. Neste caso, basta garantir que haja tempo necessário para realizar a operação de preparação entre as tarefas, não havendo a preocupação de indicar

exatamente quando esta operação será iniciada. Por exemplo, se uma tarefa i_1 de uma família k_1 começar a ser realizada no processador j no slot t , então, nenhuma tarefa i_2 de uma família k_2 poderá ser processada antes do slot $(t + pt_i + \tau_{jkk'})$, ou seja, nenhuma tarefa i_2 de uma família k_2 poderá ser processada antes da conclusão da tarefa i_1 e do tempo necessário para a realização da operação de preparação. É dada a seguir a equação [2.14] (Shah *et al.*, 1993) que garante a realização da operação de preparação sem alocação desta operação.

$$\sum_{i' \in I_j^{(k')}} W_{ji'} + \sum_{i \in I_j^{(k)}} W_{j,i,t-pt_i-\theta} \leq 1 \quad , \text{ para } \forall j \in \text{SP}, t, \theta = 0, \dots, \tau_{jkk'} - 1. \quad [2.14].$$

Esta equação simplesmente assegura que para o processador j começar a processar uma tarefa qualquer i_2 da família k_2 no slot t , nenhuma tarefa i_1 da família k_1 poderia encerrar o seu processamento após o slot $t - \tau_{jkk'}$, ou nenhuma tarefa i_1 da família k_1 poderia ser iniciada após o slot $t - pt_i - \tau_{jkk'}$.

2.5. Exemplo de aplicação

O exemplo apresentado em (Shah *et al.*, 1993) é utilizado para ilustrar a aplicação do modelo STN e a solução de um problema MILP. No exemplo apresenta-se o problema do planejamento e sequenciamento da produção de uma fábrica de fluidos lubrificantes no período de uma semana, sujeita a uma série de restrições técnicas e contratuais.

Quatro produtos, *Blend_A*, *Prod 1*, *Prod 2* e *Prod 3*, são produzidos a partir de seis materias primas, *Feed A*, *Feed B*, *Feed C*, *Feed D*, *Add 1* e *Add 2*, de acordo com a seguinte receita:

a) Receita:

- (1) **Reação:** Reagir *Feed A* para produzir *R_Prod* após 5 h.
- (2) **Dosagem A:** Dosar *R_Prod* (99,9%) com *Add 1* (0,1%) para produzir *Blend A* após 2 h.
- (3) **Dosagem B:** Dosar *R_Prod* (99,9%) com *Add 2* (0,1%) para produzir *Int 1* após 2 h.

- (4) **Mistura 1:** Misturar *Int 1* (38%) com *Feed B* (62%) para produzir *Prod 1* após 2 h.
- (5) **Mistura 2:** Misturar *Int 1* (40%) com *Feed C* (60%) para produzir *Prod 2* após 2 h.
- (6) **Mistura 3:** Misturar *Int 1* (53%) com *Feed D* (47%) para produzir *Prod 3* após 2 h.

A rede Estado-Tarefa (STN) deste problema é dada a seguir:

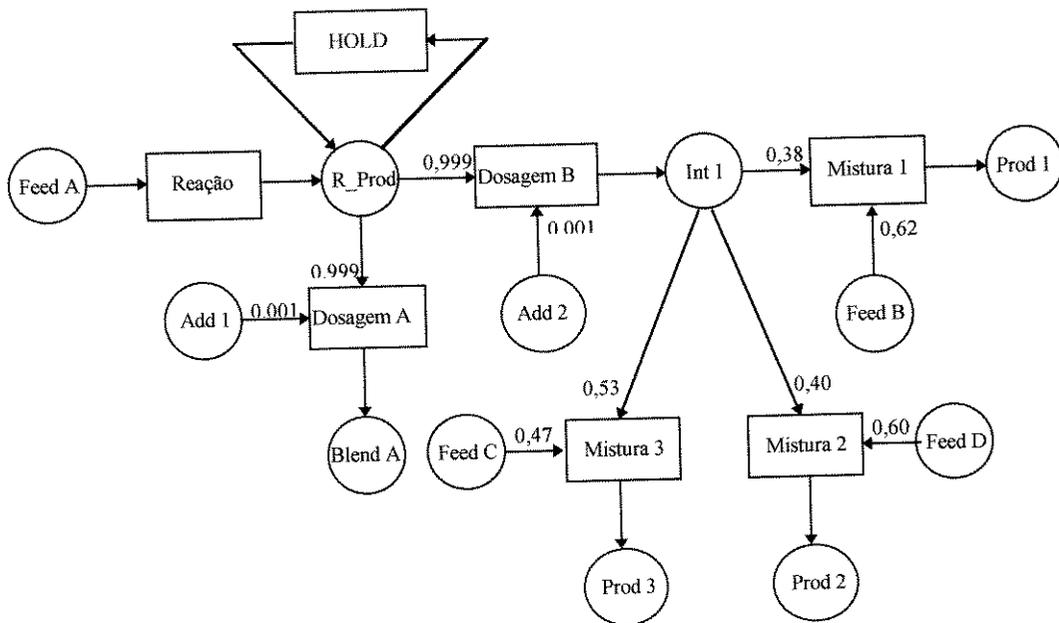


Fig. 2.4

b) Equipamento disponível:

Esta planta possui 3 tipos de processadores que são descritos abaixo:

- (1) **Reatores:** Esta planta dispõe de dois reatores que estão aptos a efetuar a tarefa *Reação*. O reator **R1** tem capacidade de processamento de 30 t e o reator **R2** tem capacidade de processamento de 60 t.
- (2) **Dosadores:** Esta planta dispõe-se de dois dosadores que estão aptos a efetuar as tarefas *Dosagem A* e *Dosagem B*. O dosador **B1** tem capacidade de processamento de 40 t e o dosador **B2** tem capacidade de processamento de 70 t.
- (3) **Misturadores:** Esta planta dispõe-se de dois misturadores que estão aptos a efetuar as tarefas *Mistura 1*, *Mistura 2* e *Mistura 3*. O Misturador

M1 tem capacidade de processamento de 50 t e o misturador **M2** tem capacidade de processamento de 45 t.

c) Capacidade de estocagem:

São indicadas aqui as capacidades de estocagem de todos os estados (locais de estoque):

- (1) As matérias primas (*Feed A*, *Feed B*, *Feed C* e *Feed D*) e os produtos finais (*Prod 1*, *Prod 2*, *Prod 3* e *Blend A*) têm capacidade de estocagem ilimitada.
- (2) O estado intermediário *Int 1* tem capacidade de estocagem de 75 t.
- (3) O produto no estado intermediário *R_Prod* só pode ser armazenado no próprio processador que o produziu (operação de armazenamento ou *hold*).

d) Recursos:

Os recursos mais consumidos (e que estão sendo considerados) são mão de obra e vapor. Note que devido às características da planta, a necessidade de mão de obra para realizar uma tarefa pode ser fracional, como é mostrado abaixo:

- (1) A tarefa **Reação** requer 1 operador durante a primeira hora (carregamento do reator). Além disso, esta tarefa requer vapor a uma taxa de 100 g/t produzida/s (0,36 Kg/Kg produzido por hora) durante a primeira hora de operação.
- (2) A tarefa **Dosagem A** requer 0,5 operadores durante a primeira hora.
- (3) A tarefa **Dosagem B** requer 0,5 operadores durante a primeira hora.
- (4) A tarefa **Mistura 1** requer 0,5 operadores durante a primeira hora.
- (5) A tarefa **Mistura 2** requer 0,5 operadores durante a primeira hora.
- (6) A tarefa **Mistura 3** requer 0,5 operadores durante a primeira hora.

A planta opera 24 h por dia, sete dias por semana, e o horizonte de planejamento é de uma semana, começando segunda-feira às 00:00 h. Durante os dois primeiros turnos (06:00 às 22:00) existem dois operadores disponíveis, enquanto que no turno da noite (22:00 às 06:00) há apenas um operador disponível. O vapor está

disponível a uma taxa de 25 t/h todos os dias, exceto na quinta-feira (devido a manutenção) quando a disponibilidade é de apenas 10 t/h. O reator R1 não está disponível sábado durante as sete primeiras horas (também devido a manutenção).

e) Necessidade de limpeza:

Os dois misturadores são usados para três tarefas diferentes. As tarefas *Mistura 2* e *Mistura 3*, produzem produtos “claros”, enquanto a tarefa *Mistura 1* produz um produto “escuro”. Consequentemente, se um misturador for usado para fazer a *Mistura 1*, ele deverá ser limpo antes de fazer a *Mistura 2* ou *Mistura 3*. Na situação inversa, nenhuma limpeza será necessária.

f) Necessidade de produção:

Há uma imposição quanto a produção mínima e máxima de cada dia, conforme apresentado abaixo:

Obs: Dados da necessidade de produção em toneladas (t).

Tabela 2.8. Produção mínima e máxima, e o valor de venda unitário dos produtos.

Dia	Blend A			Prod 1			Prod 2			Prod 3		
	Min	Max	Valor	Min	Max	Valor	Min	Max	Valor	Min	Max	Valor
Seg	50	200	2,0	50	160	3,0	100	100	2,5	100	100	2,5
Ter	50	150	2,0	100	150	3,0	120	120	2,5	150	200	2,0
										50	50	3,5
Qua	-	-	-	150	150	3,0	120	180	2,0	100	100	2,0
Qui	50	150	2,0	150	180	2,5	100	100	2,0	-	-	-
Sex	80	170	2,0	200	200	2,5	-	-	-	100	350	2,0
Sab	-	-	-	150	200	2,0	-	-	-	-	-	-
Dom	120	350	1,0	200	550	2,0	200	450	1,5	-	-	-

g) Resultados

A solução obtida para o problema acima, usando-se a formulação apresentada neste capítulo, é dada na carta de Gantt da figura 2.5, que indica como será a ocupação dos processadores e o consumo de recursos ao longo do horizonte de planejamento.

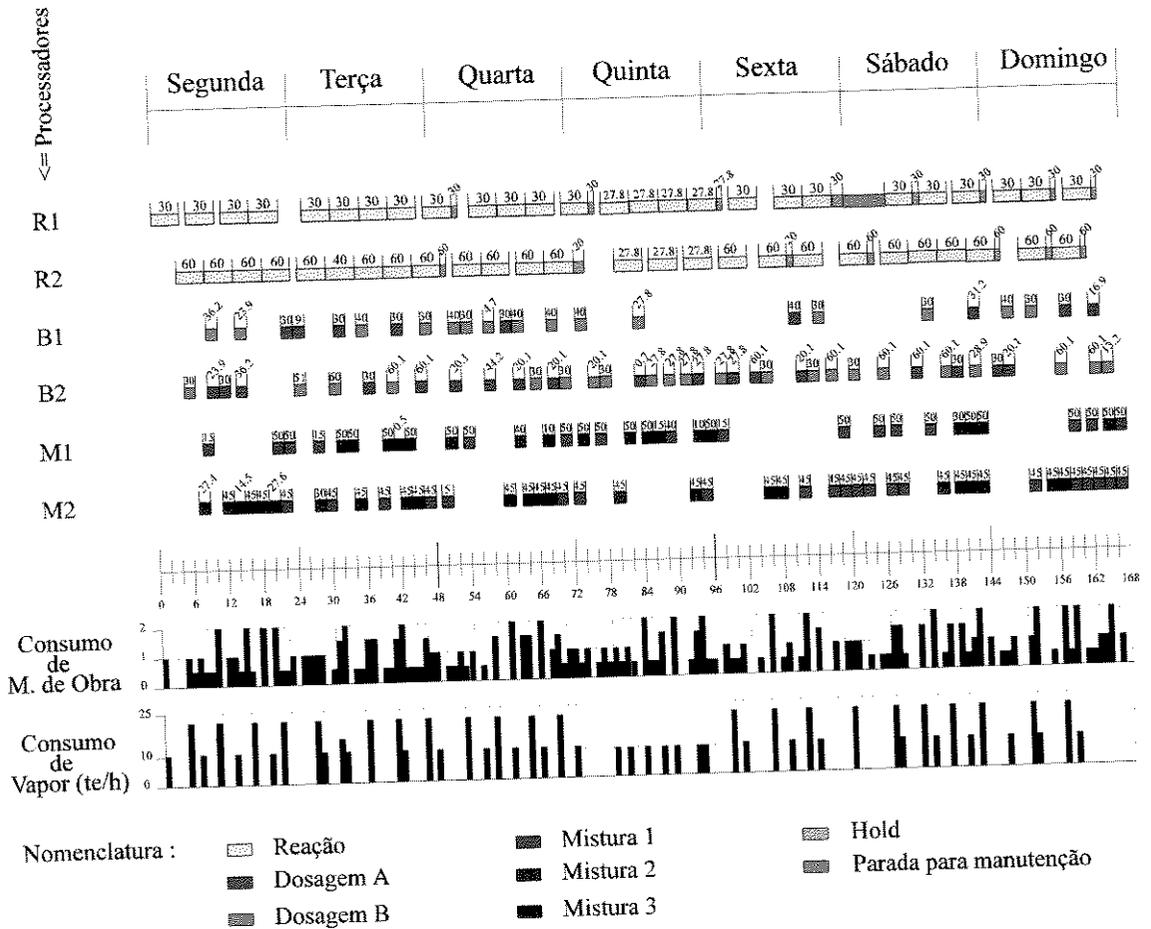


Fig. 2.5

Observe na figura 2.5 que quando as tarefas **Mistura 2** ou **Mistura 3** são processadas logo após a tarefa **Mistura 1**, é deixado um intervalo de pelo menos uma hora entre o fim da tarefa anterior (**Mistura 1**) e o início da tarefa posterior (**Mistura 2** ou **Mistura 3**). No entanto, não há necessidade da alocação da limpeza (*set-up*), já que neste problema a limpeza não consome nenhum recurso compartilhado (mão de obra e vapor) e, também, o seu custo não é considerado no problema. Com isso, é possível usar a equação [2.14] que apenas garante que haja tempo suficiente para a limpeza, sem que esta operação seja alocada.

A tabela 2.9, a seguir, indica um resumo dos resultados obtidos para este problema usando o pacote GAMS/OSL, usando-se a formulação deste capítulo, e com o modelo de Shah *et al.*(1993). Na resolução com o pacote GAMS/OSL de um problema MILP, os ajustes *default* do programa estabelecem como se dará a busca *Branch and Bound*. Shah *et al.*(1993) resolveram este problema usando uma abordagem do tipo *Branch and Bound* sendo que, a cada nó o limitante inferior é calculado usando o pacote

MINOS. Utilizando uma tolerância de 5%, a solução inteira encontrada pelos autores é de 8.754,5.

Tabela 2.9. Resumo da implementação usando GAMS/OSL.

Blocos de equações	10
Blocos de variáveis	7
Equações geradas	5.425
Variáveis simples	6.564
Variáveis discretas	2.328
Elementos não-zeros	23.115
Solução inteira obtida	8.857,90
Solução relaxada obtida	9.030,00
Diferença entre as soluções (%)	1,906 %
OPTCR - Diferença aceitável (%)	5 %
Tempo de CPU (segundos)	1.606,41
Número de iterações feitas	9.879
Número de nós sondados	429
Memória RAM alocada	6,80 Mb

2.6. Conclusão

A formulação do problema através da representação STN permite tratar muitas situações complexas encontradas na prática, tal como o processamento em paralelo de um produto, o compartilhamento de recursos e de processadores por diversos produtos, a mistura e reciclagem de materiais em uma tarefa, além da geração de vários produtos em uma tarefa.

As variáveis binárias influenciam a dimensão e a complexidade do problema, sendo que o número de variáveis binárias do problema tende a ser proporcional ao número de *slots* do horizonte de planejamento. Como a discretização do horizonte de planejamento (definida *a priori*) deve ser suficientemente fina, para que todas as tarefas possam ser iniciadas e encerradas no limite dos *slots*, quanto maior for o número de *slots* no horizonte de planejamento do problema, maior será o número de variáveis binárias do problema.

Diversos autores (Shah *et al.*, 1993; Rapacoulias *et al.*, 1991; Shah e Pantelides, 1991) têm indicado que o uso apenas do número de variáveis binárias não é o melhor indicador da complexidade e da dimensão do problema. Observe que cada equação

contém uma ou mais variáveis, e cada variável na equação gera um elemento não zero. Assim, a complexidade e a dimensão de um problema são influenciados, também, pelo seu número de elementos não zeros.

Shah *et al.* (1993) indicam que a complexidade computacional desta formulação parece depender da estrutura da rede de processamento e da flexibilidade dos equipamentos disponíveis. Problemas envolvendo reciclagem de material ou equipamentos disponíveis para processar diversas tarefas tendem a ser mais complexos que os problemas com estrutura de processamento acíclica e com um uso mais restrito dos equipamentos pelas tarefas. Outros fatores como o número de ordens a serem processadas e o uso de recursos compartilhados têm um efeito menos pronunciado na dimensão e complexidade do problema.

No entanto, a utilização da alocação da operação de preparação num problema, como indicado por Kondili *et al.* (1993), pode resultar num número inaceitavelmente alto de equações escritas e de elementos não zeros gerados. Nos problemas em que apenas se considera a duração (previsão) da operação de preparação, sem que haja alocação, a complexidade e a dimensão do problema são bem menores que nos problemas com alocação da preparação.

Com isso, pode-se observar que a principal limitação desta formulação está na dimensão do problema MILP resultante, especialmente quando o problema envolve alocação da operação de preparação.

CAPÍTULO 3

MODIFICAÇÕES DO MODELAMENTO REDE ESTADO-TAREFA PARA A DIMINUIÇÃO DA DIMENSÃO EM PROBLEMAS DE SEQUENCIAMENTO

3.1. Introdução

A resolução de problemas de planejamento e sequenciamento de plantas multi-propósito através de técnicas MILP utilizando a representação Rede Estado-Tarefa tem sido freqüentemente criticada na literatura pelo problema da dimensão. Para problemas grandes, o número de variáveis e de equações geradas pelo modelamento descrito no capítulo anterior elevam a dimensão destes problemas a tamanhos inviáveis quando se utilizam o *hardware* (microcomputadores e/ou estações de trabalho) e os pacotes de *software* para problemas MILP atualmente disponíveis.

A origem do problema reside na discretização pré-fixada do tempo, que deve ser suficientemente fina para permitir representar todos os eventos de interesse, e na quantidade de variáveis binárias, variáveis contínuas e equações associadas aos intervalos de tempo (*slots*) resultantes da discretização. Note-se que o problema da dimensão resulta do número de elementos não zeros criados pela utilização destas variáveis e equações no modelamento, e não unicamente da discretização propriamente dita.

Neste capítulo propõem-se três modificações do modelamento com o objetivo de reduzir a dimensão em problemas de sequenciamento com alocação do *set-up* dependente da seqüência.

- Individualização das bateladas de um produto.
- Nova formulação de *set-up* dependente da seqüência.
- Utilização de janelas de tempo.

A individualização das bateladas supõe uma definição prévia do número de bateladas de cada produto a serem produzidas para atender a demanda, e portanto, do tamanho das bateladas. Neste trabalho supõe-se que um nível de planejamento anterior definiu o tamanho de batelada, o que permite determinar o número de bateladas a partir da demanda. Apesar de a individualização das bateladas aumentar o número de variáveis binárias do problema, a nova formulação da alocação das operações de preparação (*set-*

up), diminui sensivelmente o número de equações e elementos não-zeros em relação à formulação original (Kondili *et al.*, 1993) descrita na seção 2.4.

A terceira modificação consiste na utilização de janelas de tempo para a execução das tarefas que conjuntamente com a individualização das bateladas permitem trabalhar com janelas de execução para cada uma das tarefas e especialmente com janelas para as tarefas de preparação. O resultado é uma diminuição no número de variáveis binárias e equações geradas pela nova formulação.

3.2. Janelas de tempo para o processamento dos produtos

O tempo durante o qual um produto, e suas tarefas, deve ou pode ser processado está freqüentemente limitado por dados externos. Tipicamente, tem-se um instante inicial a partir do qual as matérias primas estão disponíveis e um instante final definido pela data de entrega. Por outro lado, no horizonte de planejamento, podem existir períodos em que a planta não vai operar (feriados, paradas etc.) ou determinados processadores não vão estar disponíveis (por exemplo, por estarem em manutenção). Estes aspectos podem ser incluídos facilmente na formulação levando a uma limitação do âmbito de definição de variáveis binárias e equações, reduzindo assim a dimensão do problema (Kondili *et al.*, 1993).

Dada a janela de processamento de um produto, as janelas de cada uma das suas tarefas podem ser facilmente determinadas a partir dos tempos de processamento das tarefas. Analogamente, dadas as janelas de duas tarefas num mesmo processador pode ser determinada a janela para a operação de preparação do processador.

3.3. Individualização das bateladas

Freqüentemente a satisfação da demanda de um produto envolve a realização de várias bateladas. A formulação apresentada no capítulo 2 não precisa da individualização destas bateladas, dado que, nada impede que as tarefas que compõem a rota de um produto sejam alocadas várias vezes no horizonte de planejamento, de forma a satisfazer as demandas.

A individualização das bateladas leva a um aumento no número de variáveis binárias. A variável de alocação, W_{jit} , da tarefa i no processador j no slot t , passará a ser W_{bjit} , onde b indica a batelada específica da tarefa i que está sendo alocada. Porém, a utilização conjunta do índice de batelada e das janelas de processamento levam a uma redução sensível da dimensão da nova formulação de preparação (*set-up*) dependente da seqüência. Isto ocorre basicamente devido à redução do número de elementos não zeros gerados.

3.4. Exemplo

Com o objetivo de esclarecer a utilização de janelas de processamento e índices de batelada, apresenta-se nesta seção um exemplo simples extraído do estudo de caso que será discutido no capítulo 4.

Suponha, por exemplo, um produto P , resultante do processamento de três tarefas ($P1$, $P2$ e $P3$), onde a tarefa $P1$ é processada no processador R , a tarefa $P2$ é processada no processador FP e a tarefa $P3$ é processada no processador TRS , como indicado na fig. 3.1.

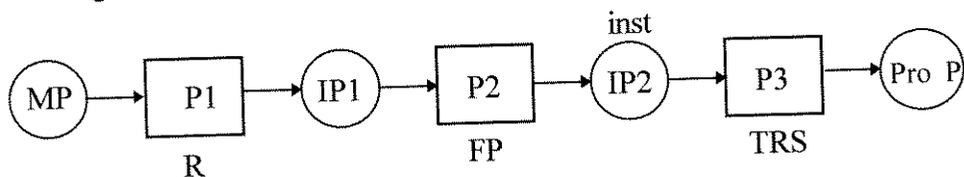


Fig. 3.1 - Representação STN do exemplo

Na representação STN, mostram-se as três tarefas ($P1$, $P2$ e $P3$), o estado das matérias primas (MP), os dois estados intermediários ($IP1$ e $IP2$) e o estado do produto final (Pro_P). Os processadores (R , FP e TRS) podem ser utilizados no processamento de várias tarefas de vários produtos, porém, observe que as três tarefas, dadas acima, são específicas do produto P . Não haverá, portanto, nenhuma outra tarefa $P1$, $P2$ ou $P3$ relacionada a outro produto em outro subgrafo. Por isso, nas equações a serem apresentadas a seguir, HB_i é definido como o conjunto das bateladas da tarefa i que é preciso processar no horizonte de planejamento, não sendo dito qual é o produto n que está relacionado a esta tarefa, já que a tarefa i está relacionada a um único produto n . Com isso, ao indicar uma tarefa i estar-se-á indicando, também, o seu produto n . O mesmo ocorrerá quando nos referirmos a batelada b da tarefa i , sem que seja feita qualquer referência ao produto n relacionado à esta tarefa.

A partir das datas de entregas, das quantidades a serem entregues, do estoque inicial e do estoque final desejado, é possível determinar o número de bateladas do produto n que terão que ser produzidas a fim de atender as necessidades de vendas e do estoque mínimo final, e com isso, determina-se HB_i . Por exemplo, se o estoque inicial do produto P for de 200 kg, o estoque mínimo final for 150 kg, o tamanho da batelada do produto P for igual a 110 kg e as vendas no horizonte de planejamento forem de 490 kg, será necessário produzir quatro bateladas (440 kg) no horizonte de planejamento para satisfazer as necessidades deste produto.

O horizonte de planejamento é de oito dias úteis, das 8:00 h de segunda-feira até as 24:00 h de terça-feira da semana seguinte, sendo que a planta fica fechada das 16:00 h de sábado até as 08:00 h de segunda-feira. Há uma demanda (de vendas) constante do produto P (Pro_P) de 70 kg por dia (não há nenhuma demanda no sábado). O tempo de processamento da tarefa $P1$ é de 16 horas e das tarefas $P2$ e $P3$ é de 8 horas, e o tempo das operações de preparação é de 8 horas.

O primeiro passo é determinar o tamanho de cada intervalo de tempo (*slot*). A partir dos dados dos tempos de processamento e dos tempos das operações de preparação, observa-se que cada dia será composto por três intervalos de tempo (*slots*) de oito horas cada e que o horizonte de planejamento terá vinte e um *slots* úteis, porque, como pode ser visto na figura 3.2, o número total de *slots* no período estudado será de vinte e seis *slots* dos quais cinco *slots* (referentes ao fim de semana) não podem ser utilizados. Note que o último *slot* do sexto dia (sábado / *slot* 16) está ligado ao primeiro *slot* do sétimo dia (segunda-feira / *slot* 17), no entanto nenhuma tarefa pode ser iniciada antes do *slot* 16 e ser concluída após este *slot*.

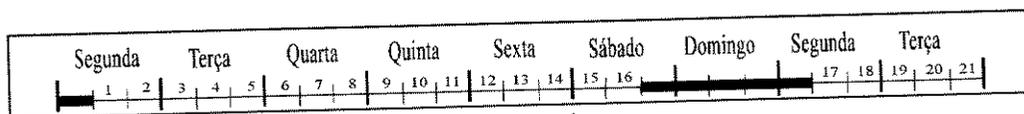


Fig. 3.2 - Slots de tempo

Dada a definição anterior do tamanho do *slot* de tempo, toma-se a demanda diária de 70 kg como uma demanda a ser satisfeita até o último *slot* de cada dia (com exceção do fim de semana quando não há nenhuma demanda), ou seja, o produto necessário para satisfazer a demanda deve estar disponível até o fim do último *slot* de cada dia.

Com isto, pode-se determinar o fim da janela de cada uma das bateladas, como indicado na tabela 3.1 e na figura 3.3.

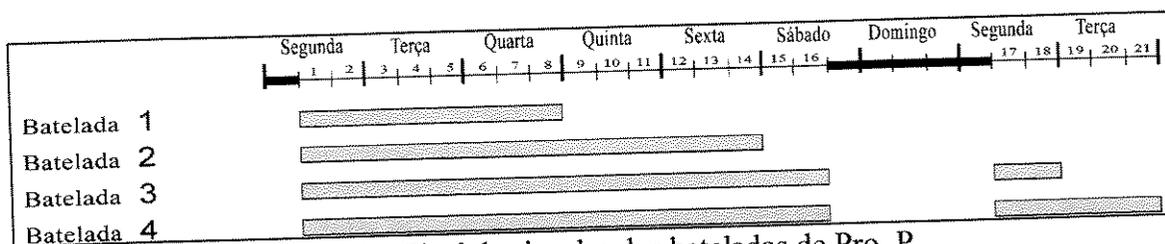


Fig. 3.3 - Final das janelas das bateladas de Pro_P

Tabela 3.1. Cálculo do fim das janelas das bateladas de Pro P.

DIA	NÚMERO DE SLOTS NO DIA	SLOTS DESTE DIA	DEMANDA	ESTOQUE NO INÍCIO DO DIA	ENTRADA EM ESTOQUE DA BATELADA	BAT. POR CONCLUIR (FIM JAN. DA BAT. Pro P)
1º dia - Segunda	2	1 e 2	70	200	-	-
2º dia - Terça	3	3 a 5	70	130	-	-
3º dia - Quarta	3	6 a 8	70	60	110	1
4º dia - Quinta	3	9 a 11	70	100	-	-
5º dia - Sexta	3	12 a 14	70	30	110	2
6º dia - Sábado	2	15 e 16	-	70	-	-
7º dia - Segunda	2	17 e 18	70	70	-	-
8º dia - Terça	3	19 a 21	70	0	220	3 e 4
Fim do horizonte	-	-	-	150	-	-

Dada a não disponibilidade da planta em cinco *slots* do fim de semana, é possível restringir mais as janelas das bateladas 3 e 4. Observe, também, que a terceira batelada deve ficar pronta até o fim do *slot* 18, senão não haverá tempo hábil para a conclusão da quarta batelada na terça-feira (antes do fim do horizonte de planejamento), já que a quarta batelada pode ficar pronta até o fim do *slot* 21, para que o estoque final seja de 150 kg (que é o estoque mínimo final).

Através da disponibilidade (entrega) de matérias primas e do tempo mínimo para o fim de processamento da batelada anterior em cada processador, é possível saber a partir de quando a batelada *b* do produto *n* pode começar a ser processada, como é indicado na figura 3.4.

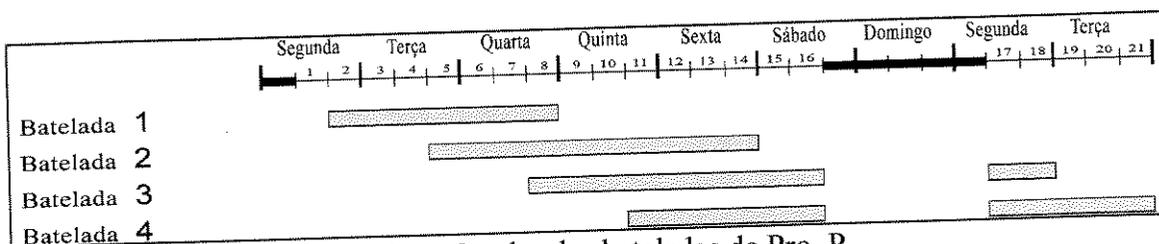


Fig. 3.4 - Janelas das bateladas de Pro_P.

Definindo-se o início e fim da janela da batelada b do produto n , é possível definir o início e fim das janelas da batelada b de cada tarefa i pertencente ao produto n nos processadores j habilitados, gerando a janela JB_{bji} . Note que a janela de qualquer batelada b de qualquer tarefa i , JB_{bji} , indica os *slots* em que o processamento desta batelada pode ser iniciado no processador j , ou seja, a janela de uma batelada b de uma tarefa i , JB_{bji} , é o conjunto de *slots* onde pode haver o início do processamento desta batelada no processador j correspondente.

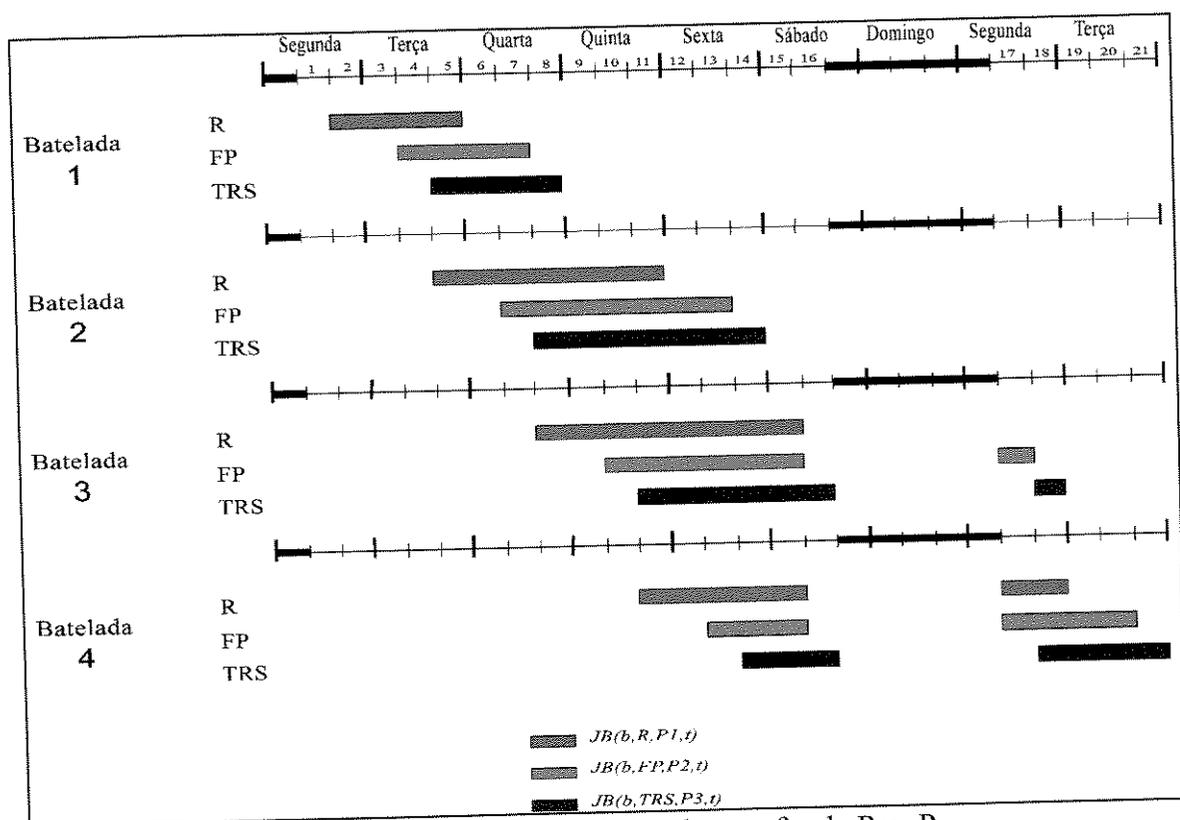


Fig. 3.5 - Janelas das bateladas das tarefas de Pro_P.

Como foi citado na seção 3.3, a individualização das bateladas leva, por um lado, a um aumento no número de variáveis binárias porque existe sobreposição entre as janelas das bateladas da mesma tarefa. No entanto, a utilização das janelas das bateladas reduz consideravelmente este aspecto.

Com os prazos finais de cada batelada do produto P , pode-se calcular o início e fim das janelas das bateladas e das operações de preparação. Na tabela 3.2, e na figura 3.6, mostram-se todos os *slots* mais cedo e mais tarde onde podem ser iniciadas as tarefas de processamento e as operações de preparação. Como as operações só podem ser iniciadas no início dos *slots*, as janelas apresentadas na figura 3.6 são interrompidas

ou encerradas no último *slot* em que as operações ainda podem ser começadas. Por exemplo, se o último *slot* em que uma batelada de uma tarefa pode ser iniciada antes do fim de semana é o *slot* 16, então, a janela desta batelada desta tarefa é interrompida no *slot* 16 para só ser reiniciada após o fim de semana. O início da janela da operação de preparação da primeira batelada das tarefas **P1**, **P2** e **P3** ocorre no início do *slot* 1, pois todos os processadores estão desocupados. Como as operações de preparação duram 1 *slot* (8 horas), o início da janela da primeira batelada da tarefa **P1** ocorre no início do *slot* 2. O tempo de processamento da tarefa **P1** é de 2 *slots* (16 horas), por isso, o início da janela da primeira batelada da tarefa **P2** ocorrerá no início do *slot* 4. A tarefa **P2** tem tempo de processamento de 1 *slot* (8 horas), portanto, o início da janela da primeira batelada da tarefa **P3** ocorrerá no início do *slot* 5.

Tabela 3.2 - Slots de início e fim das janelas das bateladas e das operações de preparação

b_2	i_2	início da janela da bat. b_2 da tar i_2	fim da janela da bat. b_2 da tar i_2	início da jan. da op. de prep. da bat. b_2 da tar i_2	fim da jan. da op. de prep. da bat. b_2 da tar i_2
1	P1	2	5	1	4
	P2	4	7	1	6
	P3	5	8	1	7
2	P1	5	11	4	10
	P2	7	13	5	12
	P3	8	14	6	13
3	P1	8	14	7	13
	P2	10	17	8	16
	P3	11	18	9	17
4	P1	11	18	10	17
	P2	13	20	11	19
	P3	14	21	12	20

Para determinar o fim das janelas das bateladas e das operações de preparação, é feito o processo inverso ao empregado para determinar o início destas janelas. Como a primeira batelada do produto **P** (**Pro_P**) deve ser concluída até o fim do *slot* 8 (fim do terceiro dia), verifica-se que o fim da janela da primeira batelada da tarefa **P3** ocorrerá no (início do) *slot* 8, já que esta tarefa tem tempo de processamento de 1 *slot* (8 horas). O fim da janela da primeira batelada da tarefa **P2** ocorrerá no (início do) *slot* 7, e o fim da janela da primeira batelada da tarefa **P1** ocorrerá no (início do) *slot* 5. Da mesma

forma, verifica-se que a janela da operação de preparação da primeira batelada da tarefa **P3** deverá estar finalizada no (início do) *slot* 7. Para a tarefa **P2**, o fim da janela da operação de preparação da primeira batelada ocorrerá no (início do) *slot* 6. E para a tarefa **P1**, a janela da operação de preparação da primeira batelada será encerrada no (início do) *slot* 4. Continuando este procedimento para as três bateladas seguintes, obter-se-ão as janelas das respectivas bateladas, como pode ser visto na tabela 3.2 e na figura 3.6. Note, por exemplo, que operação de preparação da segunda batelada da tarefa **P1** só pode ser iniciada a partir do início do *slot* 4 porque o processamento da primeira batelada desta tarefa não poderá se encerrar antes do fim do *slot* 3. Observe, também, que nas bateladas 3 e 4, as janelas das bateladas das tarefas são interrompidas antes do fim de semana para que nenhuma tarefa esteja em processamento neste período. Além disso, como o produto da tarefa **P2** é instável, esta tarefa só pode ser processada se a tarefa **P3** puder ser processada logo a seguir, e isso se reflete nas janelas das bateladas das tarefas **P2** e **P3**.

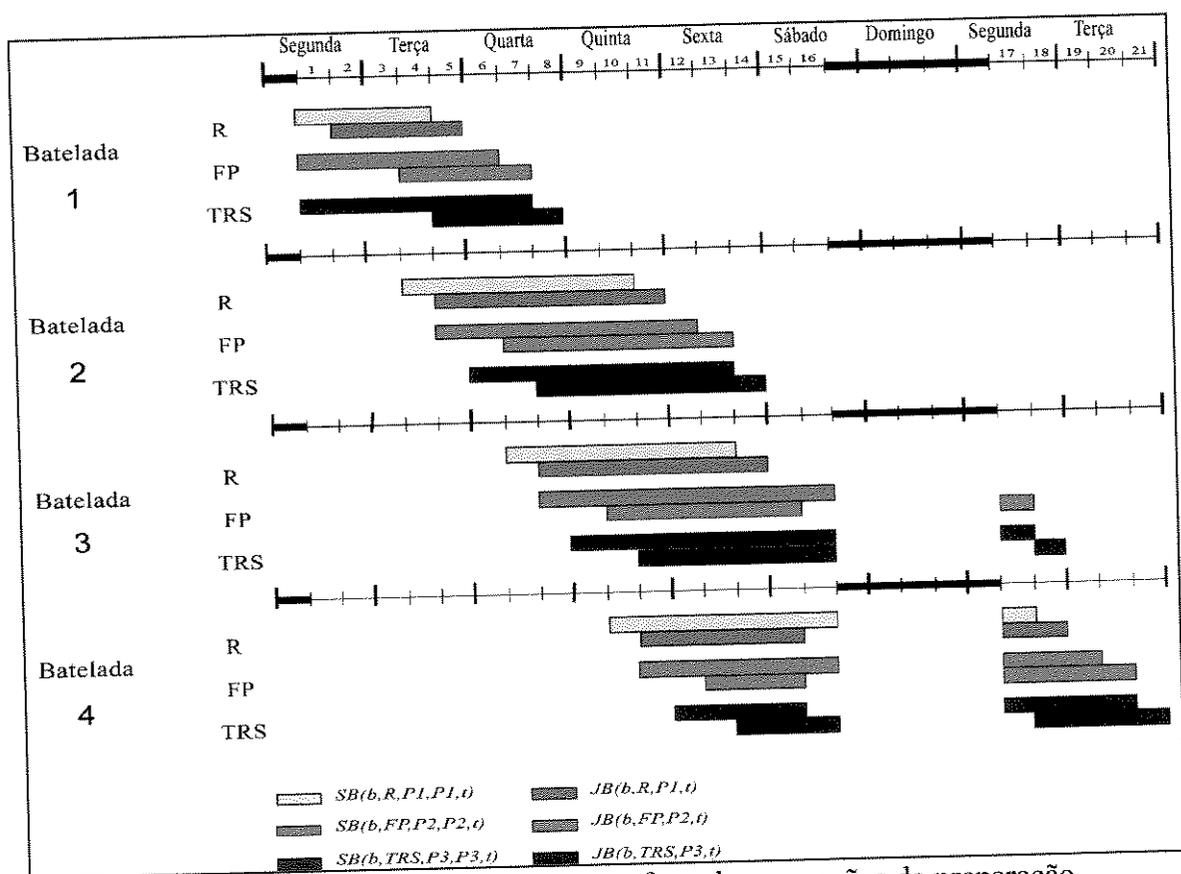


Fig. 3.6 - Janelas das bateladas das tarefas e das operações de preparação

3.5. Inclusão de Janelas e Bateladas no Modelamento

Nesta seção as equações utilizadas no modelamento através da representação STN, apresentadas na seção 2.3, são escritas individualizando as bateladas e considerando as janelas de tempo, para limitar o âmbito das equações. Deve-se observar, como citado na introdução deste capítulo, que os tamanhos das bateladas são definidos por um nível de planejamento anterior. Para facilidade de leitura, nas tabelas estão incluídas todas as variáveis, parâmetros e conjuntos, mesmo que muitos deles já tenham sido descritos na seção 2.3.

A definição prévia do tamanho das bateladas faz com que a variável M_{jit} (na tabela 2.2) passe a ser um parâmetro M_i . A variável de alocação (W_{jit}) passa a ser indexada em batelada (W_{bjit}). As janelas aparecem na formulação como limitantes do âmbito das equações e de alguns dos termos destas.

a) Equações de Balanço de Massa

O balanço de massa é dado através de duas equações distintas. A equação [3.1] é usada para os produtos que tem local de estoque próprio. A equação [3.2] é usada para os estados intermediários, que não possuem local de estoque (pode-se estocar no processador apenas se o estado for estável). A consideração de um tamanho de batelada pré-fixado permite incluir a equação [2.7] na equação de balanço de massa [2.1].

A equação [3.2] também pode ser escrita tal qual a equação [3.1], apenas eliminando-se as variáveis S_{st} , R_{st} e D_{st} . A forma como as equações são dadas refletem a forma como foram utilizadas no caso estudado. A seguir são dadas as equações [3.1], [3.2] e a tabela 3.3.

$$S_{st} = S_{s,t-1} + \sum_{i \in OG_s} \rho'_{is} * M_i * \sum_{j \in J_i} \sum_{b/t-p_i \in JB_{bji}} W_{bjit-p_i} - \sum_{i \in OC_s} \rho_{is} * M_i * \sum_{j \in J_i} \sum_{b/t \in JB_{bji}} W_{bjit} + R_{st} - D_{st}$$

,para $\forall s \in P, t \in HI$. [3.1].

$$\sum_{i \in OG_s} \rho'_{is} * M_i * \sum_{j \in J_i/t-p_i \in JB_{bji}} W_{b,j,i,t-p_i} = \sum_{i \in OC_s} \rho_{is} * M_i * \sum_{j \in J_i/t \in JB_{bji}} W_{b,j,i,t}$$

,para $\forall b, s \in EN, t \in HI$. [3.2].

Tabela 3.3. Nomenclatura das equações [3.1] e [3.2].

Variáveis das equações:

S_{st}	Estoque no estado s no intervalo de tempo (<i>slot</i>) t .
W_{bjt}	= 1, Indica que o processamento da tarefa i da batelada b no processador j iniciou-se no intervalo de tempo (<i>slot</i>) t ; = 0, Caso contrário.
D_{st}	Quantidade de produto retirada do estado s no intervalo de tempo (<i>slot</i>) t para ser entregue ao cliente.
R_{st}	Quantidade do produto s recebida por fornecimento externo à planta no intervalo de tempo (<i>slot</i>) t .

Dados das equações:

ρ'_{is}	Proporção de saída da tarefa i para o estado s .
ρ_{is}	Proporção de entrada da tarefa i vindo do estado s .
M_i	Quantidade de material processado na tarefa i .
p_i	Tempo de processamento da tarefa i .

Conjuntos utilizados na limitação do domínio das equações:

OG_s	Conjunto de tarefas produzindo material para o estado s .
OC_s	Conjunto de tarefas recebendo material do estado s .
J_i	Conjunto dos processadores que realizam a tarefa i .
P	Conjunto dos estados com local próprio para estocagem.
EN	Conjunto dos estados sem local próprio para estocagem (fora do processador).
JB_{bji}	Conjunto dos intervalos de tempo (<i>slots</i>) em que a batelada b da tarefa i pode ser iniciada no processador j .
H	Conjunto de todos os intervalos de tempo (<i>slots</i>) exceto os intervalos de tempo (<i>slots</i>) iniciais usados para definir as condições iniciais do problema. Note que podem ser necessários vários <i>slots</i> para definir as condições iniciais.

b) Equação de não coexistência

Em qualquer intervalo de tempo (*slot*) t , pode haver no máximo uma única tarefa i sendo realizada no processador j ou uma operação de preparação do processador j para realizar a tarefa i_2 . É dada a seguir a equação [3.3] e a tabela 3.4 com a sua nomenclatura:

$$\sum_{i \in I_j} \sum_{t'=t}^{t-p_i+1} \sum_{b/t' \in JB_{bji}} W_{b,j,i,t'} + \sum_{i_1} \sum_{i_2} \sum_{t'=t}^{t-\tau_{i_1,i_2}+1} \sum_{b_2/t' \in SB_{b_2,j,i_1,i_2}} WS_{b_2,j,i_1,i_2,t'} \leq 1$$

, para $\forall j, t \in H$. [3.3].

Tabela 3.4. Nomenclatura da equação [3.3].

Variáveis da equação:

$WS_{b_2, j, i_1, i_2, t}$ = 1, Indica que a operação de preparação do processador j (para realizar a batelada b_2 da tarefa i_2) é iniciada no *slot* t , após a realização da tarefa i_1 . Note que se deseja processar a batelada b_2 da tarefa i_2 , e para isso é necessário determinar a tarefa i_1 , que antecede esta batelada no processador j , dado que o período de ocupação do processador é dependente da seqüência, pois o tempo de processamento da operação de preparação da tarefa i_2 e o seu custo dependem da tarefa i_1 (anterior) e da tarefa i_2 (posterior). No entanto, a operação de preparação independe da batelada específica (b_1) da tarefa i_1 que foi processada antes da batelada b_2 da tarefa i_2 , porque o tempo de processamento da operação de preparação e o seu custo não dependem de qual batelada foi processada (Só dependem das tarefas);
= 0, Caso contrário.

Dados da equação:

τ_{i_1, i_2} Tempo de duração da operação de preparação (limpeza) do processador j para realizar uma tarefa i_2 após processar uma tarefa i_1 .

Conjuntos utilizados na limitação do domínio das equações:

H Conjunto de todos os intervalos de tempo (*slots*) dentro do horizonte de planejamento.
I_j Conjunto das tarefas que podem ser executadas no processador j .
SB_{b₂, j, i₁, i₂} Conjunto dos intervalos de tempo (*slots*) em que a operação de preparação da batelada b_2 da tarefa i_2 pode ser iniciada no processador j , tendo a tarefa i_1 como tarefa anterior.

A equação [3.3] é escrita para cada processador e cada *slot* no horizonte de planejamento. Nesta equação, o primeiro termo da equação determina se há alguma batelada b de alguma tarefa $i \in I_j$ sendo processada no *slot* t no processador j , através do somatório de todas as bateladas de todas as tarefas que podem estar sendo processadas no *slot* t no processador j . Da mesma forma, o segundo termo da equação determina se há uma operação de preparação de alguma batelada b_2 de alguma tarefa i_2 (após processar a tarefa i_1) sendo processada no *slot* t no processador j , através do somatório de todas as operações de preparação do processador j que podem estar ocorrendo no *slot* t . Note que para que possa haver uma operação de preparação em um *slot* t , é preciso que a tarefa i_2 possa suceder a tarefa i_1 no processador j e que a batelada b_2 esteja dentro da janela da operação de preparação **SB_{b₂, j, i₁, i₂}** no intervalo entre os *slots* $t - \tau_{i_1, i_2} + 1$ e t . Portanto, esta equação especifica que em qualquer *slot* dentro do horizonte de planejamento ($t \in H$) e em qualquer processador j , somente pode acontecer uma

única tarefa ou uma operação de preparação do processador, de tal forma que ou acontece uma tarefa ou uma operação de preparação ou o processador fica desocupado.

c) Consumo de recursos compartilhados:

As equações [3.4], expressando o consumo de recursos, e [3.5], expressando a disponibilidade de recursos, são dadas a seguir. Em relação à seção 2.3, as únicas mudanças são a inclusão do índice batelada e das janela de tempo das bateladas.

$$U_{u,t} = \sum_i \sum_{j \in J_i} \sum_{\theta=1}^{\min(p_i,t)} \sum_{b \in JB_{b,j,i,t-\theta+1}} (\alpha_{u,i,\theta} * W_{b,j,i,t-\theta+1}) \quad , \text{ para } \forall u, t \in H. \quad [3.4].$$

$$U_{u,t}^{\min} \leq U_{u,t} \leq U_{u,t}^{\max} \quad , \text{ para } \forall u, t \in H. \quad [3.5].$$

Tabela 3.5. Nomenclatura das eq. [3.4] e [3.5].

Variáveis das equações:

$U_{u,t}$ Consumo total do recurso u (eletricidade, vapor etc.) no *slot* t .

Dados das equações:

θ Indica um intervalo de tempo (*slot*). $1 \leq \theta \leq p_i$ e $\theta \leq t$.

$\alpha_{u,i,\theta}$ Indica quanto do recurso u é consumido no *slot* θ quando a tarefa i está sendo processada.

$U_{u,t}^{\min}$ Consumo mínimo disponível do recurso u no *slot* t . Seu valor normalmente é zero.

$U_{u,t}^{\max}$ Disponibilidade do recurso u no *slot* t .

d) Equações de vendas

A entrega de um produto num dia deve satisfazer a quantidade de produto previamente estabelecida pelas vendas, sendo que as equações só são escritas para os produtos (estados) finais e para os dias do horizonte de planejamento. A equação [3.6], expressando as entregas de produtos no dia d e a tabela 3.6, são dadas abaixo:

$$E_{s,d} = \sum_{t \in TD_d} D_{s,t} \quad , \text{ para } \forall s \in P, d \in HD. \quad [3.6].$$

Tabela 3.6. Nomenclatura da eq. [3.6].

Dados da equação:

$E_{s,d}$ Quantidade do estado s (produto) a ser entregue no dia d .

Conjuntos utilizados na limitação do domínio das equações:

TD_d Conjunto dos intervalos de tempo (*slots*) que pertencem ao dia d .

HD Conjunto dos dias que pertencem ao horizonte de planejamento.

e) Equação de existência das bateladas

É preciso que se garanta que cada batelada de uma tarefa só ocorre uma vez. Ao mesmo tempo, é preciso que todas as bateladas das tarefas existam. Isto é garantido pela equação [3.7].

$$\sum_{j \in J_i} \sum_{t \in JB_{jt}} W_{bjit} = 1 \quad , \text{ para } \forall i \in \text{NH}, b \in \text{HB}_i. \quad [3.7].$$

Tabela 3.7. Nomenclatura da equação [3.7]

Conjuntos utilizados na limitação do domínio das equações:

HB_i Conjunto das bateladas da tarefa i que devem ser produzidas no horizonte de planejamento. O número de bateladas de um produto a serem produzidas no horizonte de planejamento é obtido a partir dos dados iniciais do problema que indicam estoque de um produto no início e fim do horizonte de planejamento, bem como as suas demandas. Como a tarefa i pertence a um único produto, é possível saber se uma batelada da tarefa i deve ser produzida dentro do horizonte de planejamento.

NH Conjunto de todas as tarefas de todos os produtos, com exceção das tarefas de armazenamento (*hold*) no próprio processador. Ou seja, as tarefas de *hold* não estão contidas neste conjunto.

Quando se faz o somatório de todos os *slots* em que a batelada b da tarefa i pode ocorrer, e este somatório é igual a 1, garante-se que a batelada b da tarefa i só ocorre uma vez. Como este somatório é repetido para todas as bateladas, todas as tarefas e todos os processadores, garante-se que todas as bateladas de todas as tarefas existem.

f) Equação de existência da operação de preparação do processador

Da mesma forma que a equação anterior, é preciso que se garanta que todas as bateladas tenham uma operação de preparação para o processamento destas bateladas. Isto é garantido pela equação [3.8].

$$\sum_{j \in J_1} \sum_{i_1} \sum_{t \in SB_{b_2, j, i_1, i_2}} WS_{b_2, j, i_1, i_2, t} = 1, \text{ para } \forall i_2, b_2 \in HS_{i_2}. \quad [3.8].$$

Tabela 3.8. Nomenclatura da equação [3.8].

Conjuntos utilizados na limitação do domínio das equações:

HS_{<i>i</i>}	Indica se uma batelada da tarefa <i>i</i> tem preparação (limpeza) do processador no horizonte de planejamento. Podem haver bateladas que já tiveram a operação de preparação feita antes do início do horizonte de planejamento.
------------------------------	---

Quando se faz o somatório de todas as tarefas precedentes *i₁* em todos os intervalos de tempo em que a operação de preparação para o processamento da tarefa *i₂* pode ocorrer, e este somatório é igual a 1, garante-se que a operação de preparação para o processamento da batelada *b₂* da tarefa *i₂* só ocorre uma vez. Como este somatório é repetido para todas as bateladas, todas as tarefas que têm *set-up* prévio e todos os processadores, garante-se que todas as bateladas de todas as tarefas (que necessitam de preparação) têm uma operação de preparação.

g) Equação de custo

Pode-se adotar uma série de critérios como função objetivo, como minimizar o *makespan* (tempo total de produção), *lateness* ou defasagem (diferença entre o tempo de término do processamento e o prazo especificado), *tardiness* ou atraso (dado pela defasagem, se positiva, ou zero, se a defasagem for negativa), *earliness* ou antecipação (dado pela antecipação no término do processamento em relação ao prazo especificado), os custos de produção, ou maximizar os lucros, entre outros. No entanto, a função objetivo deste problema será minimizar os custos de produção dados na equação de custos.

$$\begin{aligned} \text{CUSTO} = & \sum_{s \in P} \sum_{t \in US} CE_s * S_{st} + \sum_b \sum_i \sum_{j \in J_1} \sum_{t \in JB_{bj}} CB_i * W_{bjit} + \\ & \sum_{b_2} \sum_j \sum_{i_1} \sum_{i_2} \sum_{t \in SB_{b_2, j, i_1, i_2}} CS_{i_1, i_2} * WS_{b_2, j, i_1, i_2, t} \end{aligned} \quad [3.9].$$

Tabela 3.9. Nomenclatura da equação [3.9].

Variáveis das equações:

CUSTO Indica o custo total de produção do problema. Esta é a variável a ser minimizada.

Dados das equações:

CB_{*i*} Custo de produção da tarefa *i*.

CE_{*s*} Custo de estoque no estado *s*.

CS_{*i₁,i₂*} Custo da operação de preparação da tarefa *i₂*, após o processamento da tarefa *i₁*.

Conjuntos utilizados na limitação do domínio das equações:

US Conjunto dos intervalos de tempo (*slots*) que constituem-se no primeiro *slot* de cada dia dentro do horizonte de planejamento. A quantidade em estoque dos produtos no fim do dia anterior é obtida no início do primeiro *slot* do dia seguinte.

3.6. Modelamento da Alocação das Operações de Preparação

Nesta seção apresenta-se uma nova formulação para a alocação das operações de preparação. Esta formulação substitui a equação [2.11] da formulação original apresentada em Kondili *et al.* (1993).

A equação [2.11] é repetida abaixo, incluindo-se nela o índice batelada, para facilitar a compreensão da modificação proposta.

$$\sum_{t=t_1+1}^{t_2-1} WS_{b_2,j,k_1,k_2,t} \geq \sum_{i_1 \in I_j^{(k_1)} \cap NH} \sum_{b_1 \in HB_{i_1}} W_{b_1,j,i_1,t_1} + \sum_{i_2 \in I_j^{(k_2)} \cap NH} W_{b_2,j,i_2,t_2} - \sum_{t=t_1+1}^{t_2-1} \sum_{i \in I_j \cap NH} \sum_{b \in HB_i} W_{bjt} - 1$$

, para $\forall j \in SP, k_1 \in K_j, k_2 \in K_j, b_2, t_1, t_2 \geq t_1$. [3.10].

Como foi indicado no capítulo 2, esta equação garante que se a batelada *b₁* de uma tarefa *i₁* tem o seu processamento iniciado no processador *j* no intervalo de tempo (*slot*) *t₁*, e uma batelada *b₂* de uma tarefa *i₂* vai ter o seu processamento iniciado neste mesmo processador no intervalo de tempo (*slot*) *t₂* (onde *t₂* ≥ *t₁*), e nenhuma outra tarefa é processada entre o fim da tarefa *i₁* e o início da tarefa *i₂*, então haverá uma operação de preparação entre os intervalos de tempo (*slots*) *t₁* e *t₂* - 1.

Tabela 3.10. Nomenclatura das eq. [3.10], [3.11], [3.12], [3.13].

Dados das equações:

CH Indica o valor da cardinalidade (último *slot*) do horizonte de planeamento.

Variáveis das equações:

Y_{bj} = 1, Indica que a batelada *b* da tarefa *i* é a última batelada processada no processador *j* no horizonte de planeamento. Neste caso, não haverá nenhuma operação de preparação no processador *j* após o processamento da batelada *b* da tarefa *i*. Todos os processadores terão uma batelada *b* de uma tarefa *i* que será a última batelada processada no processador. (Equação 3.13);
=0, Caso contrário.

Conjuntos utilizados na limitação do domínio das equações:

K_j Conjunto das famílias de tarefas que podem ser realizadas no processador *j*.

$I_j^{(k)}$ Conjunto das tarefas da família *k* executáveis no processador *j*.

SP Conjunto dos processadores onde é preciso efetuar uma operação de preparação do processador para que as tarefas possam ser processadas neles. Existem processadores que não necessitam de uma operação de preparação para realizar qualquer tarefa para a qual esteja habilitado.

C_{ji} Conjunto de todas as bateladas (*b*) de todas as tarefas (*i*) (de todos os produtos) que podem ser processadas no processador *j*, e que são candidatas a serem a última batelada processada neste processador no horizonte de planeamento. O que se pretende é indicar se a batelada *b* da tarefa *i* é candidata a ser a última batelada processada no processador *j* no horizonte de planeamento. Uma batelada é candidata a ser a última batelada processada em um processador *j* se esta é a última batelada de um produto e há pelo menos uma tarefa deste produto que deverá ser realizada neste processador.

PS_{b_1, j, i_1, i_2} Conjunto das bateladas (*b*₂) da tarefa *i*₂ que podem ter a batelada *b*₁ da tarefa *i*₁ como a batelada anterior processada no processador *j*. Este conjunto indica a precedência entre bateladas de tarefas diferentes induzida pelas janelas das bateladas das tarefas. Quando é calculada a janela da batelada *b*₂ da tarefa posterior *i*₂, de um produto *n*₂, é possível determinar quais são as bateladas (*b*₁) da tarefa anterior *i*₁, de um produto *n*₁, que podem preceder a batelada *b*₂ da tarefa *i*₂. Este conjunto indica se a batelada *b*₁ da tarefa *i*₁ pode preceder a batelada *b*₂ da tarefa *i*₂ no processador *j*.

O principal problema com esta equação, como já foi discutido na seção 2.4, é o grande número de equações (proporcional a H^2). A modificação aqui proposta (equações [3.11], [3.12] e [3.13]) gera um número de equações proporcional a *H*.

$$\sum_{i_2} \sum_{b_2 \in PS_{b_1, j, i_2}} \sum_{\substack{t' \in SB_{b_2, j, i_1, i_2} \\ t' \geq t + p_{i_1}}} WS_{b_2, j, i_1, i_2, t'} + Y_{b_1, j, i_1} \geq W_{b_1, j, i_1, t}$$

, para $\forall i_1 \in NH, b_1 \in HB_{i_1}, j \in J_{i_1} \cap SP, t \in JB_{b_1, j, i_1}$. [3.11].

$$\sum_{i_1} \sum_{\substack{t' \in SB_{b_2, j, i_1, i_2} \\ t' \leq t - \tau_{i_1, i_2}}} WS_{b_2, j, i_1, i_2, t'} \geq W_{b_2, j, i_2, t}$$

, para $\forall i_2 \in NH, b_2 \in HS_{i_2}, j \in J_{i_2} \cap SP, t \in JB_{b_2, j, i_2}$. [3.12].

$$\sum_{\substack{t' \in JB_{b-1, j, i} \\ t' \leq t - p_i}} W_{b-1, j, i, t'} \geq W_{bjit}$$

, para $\forall i \in NH, j \in J_i, j \notin SP, b \in HB_i, b \geq 1, t \in JB_{bj}$. [3.13].

A equação [3.11] garante que se a tarefa i_1 é alocada no processador j , em um *slot* t , e a tarefa i_1 não é a última tarefa a ser realizada neste processador (isto é, $Y_{b_1, j, i_1} = 0$), existirá uma operação de preparação iniciando em um *slot* t' ($t' \geq t + p_{i_1}$), correspondente à preparação do processador j para processar uma tarefa i_2 após a tarefa i_1 . A variável Y_{b_1, j, i_1} funciona como um critério de parada na alocação de operações de preparação, pois quando o seu valor for 1 não haverá nenhuma operação de preparação após a batelada b_1 da tarefa i_1 dentro do horizonte de planejamento.

O somatório, no lado esquerdo desta equação, seleciona uma operação dentre as operações de preparação que podem suceder a batelada b_1 da tarefa i_1 no processador j . Este conjunto das operações de preparação é dado por PS_{b_1, j, i_1, i_2} . Após o processamento de uma batelada b_1 da tarefa i_1 no processador j , iniciado no *slot* t , o processamento de uma batelada b_2 de uma tarefa i_2 no mesmo processador será iniciado no *slot* t' (onde $t' \geq t + p_{i_1}$) após o tempo de processamento da batelada b_1 da tarefa i_1 (p_{i_1}).

A equação [3.12] garante que a operação de preparação será iniciada num *slot* t' , tal que a preparação esteja finalizada quando se inicia a tarefa i_2 (ou seja, $t' \leq t - \tau_{i_1, i_2}$).

A equação [3.13] é usada quando um processador não tem operação de preparação e se pretende garantir que não haja inversão na ordem das bateladas de uma tarefa i a serem processadas em um processador j . Esta equação garante que uma batelada $b - 1$ de uma tarefa i será iniciada num *slot* t' , tal que esta batelada esteja finalizada quando se inicia a batelada b (seguinte à batelada $b - 1$) da tarefa i (ou seja, $t' \leq t - p_i$).

É preciso que se garanta que todo processador tem uma batelada que é a última batelada processada neste processador no horizonte de planejamento. Isto é garantido pela equação [3.14].

$$\sum_{i \in I_j} \sum_{b \in C_{ji}} Y_{bji} = 1, \text{ para } \forall j \in \text{SP}. \quad [3.14].$$

O somatório de todos os candidatos a serem a última batelada processada no processador j força que haja uma batelada b da tarefa i que será a última batelada processada neste processador. Como este somatório é repetido para todos os processadores, garante-se que todos os processadores têm uma batelada que é a última batelada processada neste processador.

A fim de reduzir o número de equações e de não zeros gerados pelas equações [3.11], [3.12] e [3.13], estas equações podem ser agregadas no tempo, dando lugar às equações [3.11a], [3.12a] e [3.13a], respectivamente. Na seção 3.7 compara-se o desempenho das duas versões.

$$\sum_{i_2} \sum_{b_2 \in PS_{b_1, j_1, i_2}} \sum_{t' \in SB_{b_2, j_1, i_2}} WS_{b_2, j_1, i_2, t'} * t' + Y_{b_1, j_1, i_1} * CH \geq \sum_{t \in JB_{b_1, j_1}} W_{b_1, j_1, i_1, t} * (t + p_{i_1})$$

, para $\forall i_1 \in \text{NH}, b_1 \in \text{HB}_{i_1}, j \in J_{i_1} \cap \text{SP}. \quad [3.11a].$

$$\sum_{i_1} \sum_{t' \in SB_{b_2, j_1, i_2}} WS_{b_2, j_1, i_2, t'} * (t' + \tau_{i_1, i_2}) \leq \sum_{t \in JB_{b_2, j_2}} W_{b_2, j_2, i_2, t} * t$$

, para $\forall i_2 \in \text{NH}, b_2 \in \text{HS}_{i_2}, j \in J_{i_2} \cap \text{SP}. \quad [3.12a].$

Comparando as equações [3.11] e [3.11a], observa-se que elas levam ao mesmo resultado. Se a batelada b_1 da tarefa i_1 é alocada no processador j no slot t , o lado esquerdo da equação [3.11a] estará limitado a $t' \geq t + p_{i_1}$. A multiplicação pelo instante de tempo onde são iniciadas $WS_{b_2, j_1, i_2, t'}$ e $W_{b_1, j_1, i_1, t}$ (que pode ser implementado de forma linear) pode ser introduzido sem alterar o sinal de desigualdade.

O sinal de desigualdade na equação [3.12] é utilizado para “forçar” o *set up*, se a tarefa i_2 é alocada. A limitação em t' faz com que o início do *set up* leve em consideração o tempo de *set up*, especificando que ele só pode ser iniciado suficientemente antes de t .

O sinal de desigualdade da equação [3.12] não pode ser mantido na equação [3.12a], se for empregado o mesmo mecanismo de agregação aplicado na equação [3.11], porque as equações [3.12] e [3.12a] se referem a instantes anteriores da alocação de i_2 , e não posteriores como nas equações [3.11] e [3.11a]. É importante notar que a equação [3.12a] não força o *set up*, mas que se existe *set up*, ele será iniciado num instante tal que o tempo necessário para a realização do *set up* seja garantido. A realização do *set up* será garantida pela equação [3.11a] e a sua duração pela equação [3.12a].

As equações [3.11a] e [3.12a] realizam o ordenamento das bateladas das tarefas e de suas operações de preparação do processador, agindo conjuntamente neste ordenamento, de tal forma que as bateladas das tarefas e suas operações de preparação vão sendo alocadas nos seus respectivos processadores uma a uma até que não haja nenhuma batelada para ser alocada nos processadores. Observe que a equação [3.11a] só aloca uma operação de preparação de uma batelada b_2 de uma tarefa i_2 se na equação [3.12a] esta batelada for alocada no mesmo processador j . Quando não há mais nenhuma batelada para ser alocada no processadores, teremos $Y_{bji} = 1$ e o ordenamento é encerrado neste processador.

Da mesma forma, a equação [3.13a], abaixo, garante o ordenamento das bateladas num processador j que não precisa de preparação antes do processamento das tarefas.

$$\sum_{t' \in JB_{b,j}} W_{b-l,j,i,t'} * (t' + p_i) \leq \sum_{t \in JB_{bj}} W_{bjt} * t$$

, para $\forall j \in J_i, j \notin SP, i \in NH, b \in HB_i, b \geq 1.$ [3.13a].

3.7. Comparação das duas formulações da alocação das operações de preparação dependentes da seqüência.

Nos casos em que as operações de preparação tem um custo significativo ou consomem recursos, sendo necessário considerar a alocação da operação de preparação em um *slot* específico, Kondili *et al.*(1993) apresentaram a equação [2.11]. Neste capítulo foram apresentadas as equações de preparação, [3.11] a [3.14], assim como as

suas versões agregadas, as equações [3.11a] a [3.14a], que foram propostas para substituir a equação [2.11].

Será mostrado que nos casos em que é inevitável usar a operação de preparação com alocação, o uso das equações [3.11a] e [3.12a] resultam num problema de dimensão menor que o gerado pelo uso da equação [2.11]. Isto se deve à redução no número de equações e elementos não zeros gerados, apesar do aumento no número de variáveis binárias.

Na continuação obtém-se as expressões para o número de equações e o número de não zeros para as diferentes formulações, utilizando-se um exemplo para uma comparação numérica.

3.7.1. Comparação entre a equação proposta por Kondili *et al.* (1993) para a alocação do *set up* e a versão não agregada proposta.

a) Número de equações e elementos não zeros gerados pela equação [2.11], sem agrupamento em famílias.

A fim de facilitar as discussões sobre as vantagens no uso de uma ou outra equação, é rerepresentada abaixo a equação [2.11a], que é equivalente a equação [2.11] quando as famílias de tarefas são consideradas unitárias. Esta é a situação no exemplo tratado no capítulo 4.

$$\sum_{t=t_1+1}^{t_2-1} WS_{j,i_1,i_2,t} \geq W_{j,i_1,t_1} + W_{j,i_2,t_2} - \sum_{t=t_1+1}^{t_2-1} \sum_{i \in I_j \cap NH} W_{ji} - 1$$

[2.11a].

,para $\forall j \in SP, i_1 \in I_j \cap NH, i_2 \in I_j \cap NH, t_1, t_2 \geq t_1$.

A tabela 3.11 abaixo indica quantos elementos não-zeros são escritos pela equação [2.11a] para cada par de *slots* t_1 e t_2 (onde $t_2 \geq t_1$) em um processador j . Em cada linha (para cada par de *slots* t_1 e t_2) são indicados o número de elementos não zeros gerados por cada segmento desta equação, considerando-se todas as combinações entre as tarefas que podem ser realizadas neste processador. O número de tarefas que podem

ser processadas em um processador j é indicado por m e o número de *slots* no horizonte de planejamento é indicado por H .

Tabela 3.11. Número de elementos não zeros gerados pela equação [2.11a] para cada par de *slots* t_1 e t_2 em um processador j .

t_1	t_2	$\sum_{t=t_1+1}^{t_2-1} WS_{j,i_1,i_2,t}$	$W_{j,i_1,t_1} + W_{j,i_2,t_2}$	$\sum_{t=t_1+1}^{t_2-1} \sum_{i \in I_j} W_{j,i,t}$	Total
1	2	0	$2m^2$	0	$2m^2$
1	3	m^2	$2m^2$	m^3	$2m^2 + (m^3+m)$
1	4	$2m^2$	$2m^2$	$2m^3$	$2m^2 + 2(m^3+m)$
1	5	$3m^2$	$2m^2$	$3m^3$	$2m^2 + 3(m^3+m)$
1	6	$4m^2$	$2m^2$	$4m^3$	$2m^2 + 4(m^3+m)$
1	$2m^2$
1	$H-4$	$(H-6)m^2$	$2m^2$	$(H-6)m^3$	$2m^2 + (H-6)(m^3+m)$
1	$H-3$	$(H-5)m^2$	$2m^2$	$(H-5)m^3$	$2m^2 + (H-5)(m^3+m)$
1	$H-2$	$(H-4)m^2$	$2m^2$	$(H-4)m^3$	$2m^2 + (H-4)(m^3+m)$
1	$H-1$	$(H-3)m^2$	$2m^2$	$(H-3)m^3$	$2m^2 + (H-3)(m^3+m)$
1	H	$(H-2)m^2$	$2m^2$	$(H-2)m^3$	$2m^2 + (H-2)(m^3+m)$
2	3	0	$2m^2$	0	$2m^2$
2	4	m^2	$2m^2$	m^3	$2m^2 + (m^3+m)$
2	5	$2m^2$	$2m^2$	$2m^3$	$2m^2 + 2(m^3+m)$
2	6	$3m^2$	$2m^2$	$3m^3$	$2m^2 + 3(m^3+m)$
2
2	$H-3$	$(H-6)m^2$	$2m^2$	$(H-6)m^3$	$2m^2 + (H-6)(m^3+m)$
2	$H-2$	$(H-5)m^2$	$2m^2$	$(H-5)m^3$	$2m^2 + (H-5)(m^3+m)$
2	$H-1$	$(H-4)m^2$	$2m^2$	$(H-4)m^3$	$2m^2 + (H-4)(m^3+m)$
2	H	$(H-3)m^2$	$2m^2$	$(H-3)m^3$	$2m^2 + (H-3)(m^3+m)$
...
$H-3$	$H-2$	0	$2m^2$	0	$2m^2$
$H-3$	$H-1$	m^2	$2m^2$	m^3	$2m^2 + (m^3+m)$
$H-3$	H	$2m^2$	$2m^2$	$2m^3$	$2m^2 + 2(m^3+m)$
$H-2$	$H-1$	0	$2m^2$	0	$2m^2$
$H-2$	H	m^2	$2m^2$	m^3	$2m^2 + (m^3+m)$
$H-1$	H	0	$2m^2$	0	$2m^2$

Para o total de m tarefas, tem-se por processador j e para cada *slot* t_1 :

$$\text{Número total de não zeros: } 2(H-t_1)m^2 + [(H-t_1-1)(H-t_1)(m^3+m)]/2$$

$$\text{Número total de equações: } (H-t_1)m^2$$

Considerando-se que também existe operação de preparação quando as tarefas i_1 e i_2 são iguais ($i_1 = i_2$) o número de equações e de elementos não-zeros gerados pela equação [2.11a] para um processador j qualquer é dado por.

$$\text{Número total de equações para um processador } j = m^2 H(H-1)/2.$$

$$\text{Número total de elementos não-zeros em } j = m^2 \sum_{t=1}^{n-1} \{[(m+1)(t+1) + 4] t / 2\}.$$

b) Número de equações e elementos não zeros na nova formulação, versão não agregada (equações [3.11] e [3.12]).

As equações [3.11] e [3.12] são repetidas abaixo.

$$\sum_{i_2} \sum_{b_2 \in PS_{b_1, j, i_1, i_2}} \sum_{\substack{t' \in SB_{b_2, j, i_1, i_2} \\ t' \geq t + p_{i_1}}} WS_{b_2, j, i_1, i_2, t'} + Y_{b_1, j, i_1} \geq W_{b_1, j, i_1, t}$$

, para $\forall i_1 \in NH, b_1 \in HB_{i_1}, j \in J_{i_1} \cap SP, t \in JB_{b_1, j, i_1}$. [3.11].

$$\sum_{i_1} \sum_{\substack{t' \in SB_{b_2, j, i_1, i_2} \\ t' \leq t - \tau_{i_1, i_2}}} WS_{b_2, j, i_1, i_2, t'} \geq W_{b_2, j, i_2, t}$$

, para $\forall i_2 \in NH, b_2 \in HS_{i_2}, j \in J_{i_2} \cap SP, t \in JB_{b_2, j, i_2}$. [3.12].

A nomenclatura adotada para o cálculo do número de equações e de elementos não zeros gerados pela equações de preparação deste capítulo é indicada na tabela 3.20.

Tabela 3.12. Nomenclatura para calcular o número de equações e não zeros	
m	Número de tarefas que podem ser realizadas no processador <i>j</i> .
l	Número total de bateladas que podem ser realizadas no processador <i>j</i> no horizonte de planejamento.
H	Número de slots que compõem o horizonte de planejamento.

As tabelas 3.13 e 3.14 indicam o número de elementos não zeros que são gerados pelas equações [3.11] e [3.12] em um processador *j* para cada *slot* *t*. Para a comparação com os resultados anteriores não se consideram as janelas de tempo, tomando-se estas iguais ao horizonte de planejamento. Além disso, o número de elementos não zeros depende do tempo de processamento das tarefas (p_{i_1}) e do tempo de duração das operações de preparação (τ_{i_1, i_2}), o que não ocorre na formulação de Kondili *et al.* (1993). A situação de pior caso em termos de não zeros é quando estes tempos (p_{i_1} e τ_{i_1, i_2}) são iguais a 1 *slot*.

Tabela 3.13. Número de elementos não zeros gerados pela eq. [3.11] para uma batelada b_1 de uma tarefa i_1 em um *slot* *t* em um processador *j*.

<i>t</i>	$\sum_{i_2} \sum_{b_2 \in PS_{b_1, j, i_1, i_2}} \sum_{\substack{t' \in SB_{b_2, j, i_1, i_2} \\ t' \geq t + p_{i_1}}} WS_{b_2, j, i_1, i_2, t'}$	Y_{b_1, j, i_1}	$W_{b_1, j, i_1, t}$	Total
1	$(l - 1) \cdot (H - 1)$	1	1	$(l - 1) \cdot (H - 1) + 2$
2	$(l - 1) \cdot (H - 2)$	1	1	$(l - 1) \cdot (H - 2) + 2$
3	$(l - 1) \cdot (H - 3)$	1	1	$(l - 1) \cdot (H - 3) + 2$
...
$H - 3$	$(l - 1) \cdot (3)$	1	1	$(l - 1) \cdot (3) + 2$
$H - 2$	$(l - 1) \cdot (2)$	1	1	$(l - 1) \cdot (2) + 2$
$H - 1$	$(l - 1) \cdot (1)$	1	1	$(l - 1) \cdot (1) + 2$
<i>H</i>	0	1	1	2

Considerando-se todas as bateladas que podem ser realizadas em um processador j , em um *slot* t :

Número de equações geradas pela eq.[3.11] para um *slot* t : l

Número de não zeros gerados pela eq.[3.11] para um *slot* t : $(l - 1).(H - t).l + 2.l$

Tabela 3.14. Número de elementos não zeros gerados pela eq. [3.12] para uma batelada b_l de uma tarefa i_l em um *slot* t em um processador j .

t	$\sum_{i_1} \sum_{\substack{t' \in SB_{b_2, j, i_1, i_2} \\ t' \leq t - \tau_{i_1, i_2}}} WS_{b_2, j, i_1, i_2, t'}$	$W_{b_2, j, i_2, t}$	Total
1	$m.(1)$	1	$m.(1) + 1$
2	$m.(2)$	1	$m.(2) + 1$
3	$m.(3)$	1	$m.(3) + 1$
...
$H - 3$	$m.(H - 3)$	1	$m.(H - 3) + 1$
$H - 2$	$m.(H - 2)$	1	$m.(H - 2) + 1$
$H - 1$	$m.(H - 1)$	1	$m.(H - 1) + 1$

Considerando-se todas as bateladas que podem ser realizadas em um processador j , em um *slot* t :

Número de equações geradas pela eq.[3.12] para um *slot* t : l

Número de não zeros gerados pela eq.[3.12] para um *slot* t : $(m.t + 1).l$

A tabela 3.15 abaixo, apresenta o número de equações e de elementos não zeros que são gerados por cada uma das equações, apresentando, também, o total gerado pelas duas equações.

Tabela 3.15. Número de equações e elementos não zeros gerados num processador j pelas equações [3.11] e [3.12].

Número de equações geradas num processador j pela eq. [3.11]	$l.H$
Número de elementos não zeros gerados num processador j pela eq. [3.11]	$[(l-1).(H-1)/2 + 2].l.H$
Número de equações geradas num processador j pela eq. [3.12]	$l.(H - 1)$
Número de elementos não zeros gerados num processador j pela eq. [3.12]	$[m.H/2 + 1].l.(H-1)$
Número de equações geradas num processador j pelas eq. [3.11] e [3.12]	$l.(2H - 1)$
Número de não zeros gerados num processador j pelas eq. [3.11] e [3.12]	$[(m+l-1)H/2 + 3]l.(H-1) + 2l$

A formulação não agregada apresentada neste capítulo reduz a dimensão do problema, quando comparada à formulação dada no capítulo 2. O número de equações

geradas pelas equações [3.11] e [3.12] para um processador j em um *slot* t é proporcional ao número de bateladas processadas no horizonte de planejamento (I), enquanto a equação [2.11a] gera um número de equações proporcional a $(m^2.H)$. O número de elementos não zeros gerados pelas equações [3.11] e [3.12] para um processador j em um *slot* t é proporcional a $(I^2.H)$, enquanto a equação [2.11a] gera um número de elementos não zeros proporcional a $(m^3.H^2)$. Como o número de bateladas processadas em j dentro do horizonte de planejamento (I) é muito menor que o horizonte de planejamento (H), o número de equações e elementos não zeros gerados pelas equações [3.11] e [3.12] são bem menores que os gerados pela equação [2.11a].

Para horizontes de planejamento grandes, o número de bateladas que podem ser processadas aumenta, mas tanto a formulação original no capítulo 2 como a modificação aqui proposta não têm como objetivo o planejamento do tipo campanha. Nesta situação a demanda é muito maior do que o tamanho da batelada e certamente uma boa solução supõe a repetição no tempo de uma linha de produção.

A formulação de Kondili *et al.* (1993) não tem bateladas, mas a nova formulação introduzida neste capítulo utiliza o índice batelada. Isto pode causar alguma dúvida quanto as vantagens desta nova formulação, pelo conseqüente aumento do número de variáveis binárias, porém, é importante notar que as janelas de tempo são responsáveis por uma redução sensível na sobreposição das bateladas, diminuindo assim o aumento do número de variáveis binárias.

Para ilustrar a diminuição da dimensão do problema em termos de número de equações e não zeros, utiliza-se a seguir um exemplo simples. Considera-se um único processador, varia-se o número de tarefas habilitadas e o horizonte de planejamento (número de *slots*). A formulação de Kondili origina o número de elementos não zeros e equações dados na tabela 3.16 abaixo.

Tabela 3.16. Número de não zeros e equações gerados pela eq. [2.11a] em j .

Nº de Tarefas	Nº de elementos não zeros		Nº de equações	
	10 slots	20 slots	10 slots	20 slots
2	1.800	15.200	180	760
3	5.130	44.460	405	1.710
4	11.040	97.280	720	3.040
5	20.250	180.500	1.125	4.750
6	33.480	300.960	1.620	6.840

A formulação proposta gera um número de elementos não zeros que depende do número de bateladas. A tabela a seguir representa a situação para dois casos.

Tabela 3.17. Número de não zeros e equações gerados pelas eq. [3.11] e [3.12] em um processador j , com horizonte de 10 e 20 slots, sendo que cada tarefa pode ter 1 ou 4 bateladas no horizonte de planejamento.

Nº de Tarefas	Nº de elementos não zeros				Nº de equações			
	1 Batelada		4 Bateladas		1 Batelada		4 Bateladas	
	10 slots	20 slots	10 slots	20 slots	10 slots	20 slots	10 slots	20 slots
2	328	1.258	3.832	15.672	38	78	152	312
3	762	3.027	7.908	32.628	57	117	228	468
4	1.376	5.556	14.144	58.704	76	156	304	624
5	2.170	8.845	22.180	93.380	95	195	380	780
6	3.144	12.894	32.016	133.656	114	234	456	936

3.7.2. Redução adicional com a formulação agregada (eq. 3.11a) e [3.12a])

As equações [3.11a] e [3.12a], apesar de atuarem como as equações [3.11] e [3.12], permitem uma sensível redução na dimensão dos problemas com alocação das operações de preparação. As equações [3.11a] e [3.12a] são rerepresentadas a seguir.

$$\sum_{i_2} \sum_{b_2 \in PS_{b_1, j, i_1, i_2}} \sum_{t' \in SB_{b_2, j, i_1, i_2}} WS_{b_2, j, i_1, i_2, t'} * t' + \sum_{t/b_1 \in C_{j, i_1}} Y_{b_1, j, i_1} \geq \sum_{t \in JB_{b_1, j, i_1}} W_{b_1, j, i_1, t} * (t + p_{i_1})$$

, para $\forall i_1 \in NH, b_1 \in HB_{i_1}, j \in J_{i_1} \cap SP$. [3.11a].

$$\sum_{i_1} \sum_{t' \in SB_{b_2, j, i_1, i_2}} WS_{b_2, j, i_1, i_2, t'} * (t' + \tau_{i_1, i_2}) \leq \sum_{t \in JB_{b_2, j, i_2}} W_{b_2, j, i_2, t} * t$$

, para $\forall i_2 \in NH, b_2 \in HS_{i_2}, j \in J_{i_2} \cap SP$. [3.12a].

O número de equações geradas pelas equações [3.11a] e [3.12a] para um processador j , é proporcional ao número de bateladas processadas no horizonte de planejamento neste processador, enquanto o número de equações geradas pelas equações [3.11] e [3.12] é proporcional a $(I.H)$. O número de não zeros gerados pela formulação agregada é menor do que na formulação não agregada devido ao fato que nesta última geram-se duas equações para cada valor de slot t (equações [3.11] e [3.12]), enquanto na formulação agregada as equações geradas não dependem deste valor.

CAPÍTULO 4

ESTUDOS DE CASO

Neste capítulo aplica-se a formulação desenvolvida para resolver o problema apresentado por Egli e Rippin (1986). A seção 4.1 descreve o problema e a seção 4.2 resume o procedimento heurístico apresentado por Egli e Rippin (1986).

O problema foi formulado como um problema linear inteiro misto (MILP) e implementado utilizando o *software* GAMS (Brooke *et al.*, 1992) como linguagem de programação e o *software* OSL (IBM - OSL, 1991) para resolver o problema linear inteiro misto. A seção 4.3 apresenta os resultados obtidos e o anexo 1 descreve a implementação realizada.

4.1. Descrição do problema de Egli e Rippin (1986)

O problema tratado nesta referência é um caso real de planejamento e sequenciamento envolvendo a produção de quatro produtos (D, E, F e H) em uma planta química multipropósito num período de tempo definido. A produção é feita em 3 turnos, o que permite o funcionamento contínuo da planta de segunda-feira às 06:00 h até sábado às 18:00 h. Bateladas parcialmente completadas podem permanecer no equipamento, mesmo em fins de semana e feriados, desde que sejam estáveis.

a) Equipamentos disponíveis

Os processadores disponíveis na planta química são listados na tabela 4.1, com as respectivas abreviaturas, pelas quais estes processadores serão identificados no problema. O reator 2 e o reator 6 funcionam simultaneamente (em fase) na planta, e por isso, são identificados como se fossem um único processador ao desempenharem suas tarefas (R26).

Tabela 4.1. Processadores disponíveis na planta

Processador	Abreviatura
Reator com agitação 1	R1
Reatores com agitação 2 e 6	R26
Reator com agitação 3	R3
Reator com agitação 4	R4
Reator com agitação 7	R7
Filtro	FI
Filtro prensa 1	FP1
Filtro prensa 2	FP2
Secador com pás	TRO
Cabine de secagem	TRS

b) Representação STN dos processos de produção

Os processos de produção de cada um dos quatro produtos (D, E, F e H) são descritos na figura 4.1. Note que, na representação STN abaixo, cada tarefa só pode ser atribuída a um único processador, porém um processador pode ser capaz de processar mais de uma tarefa.

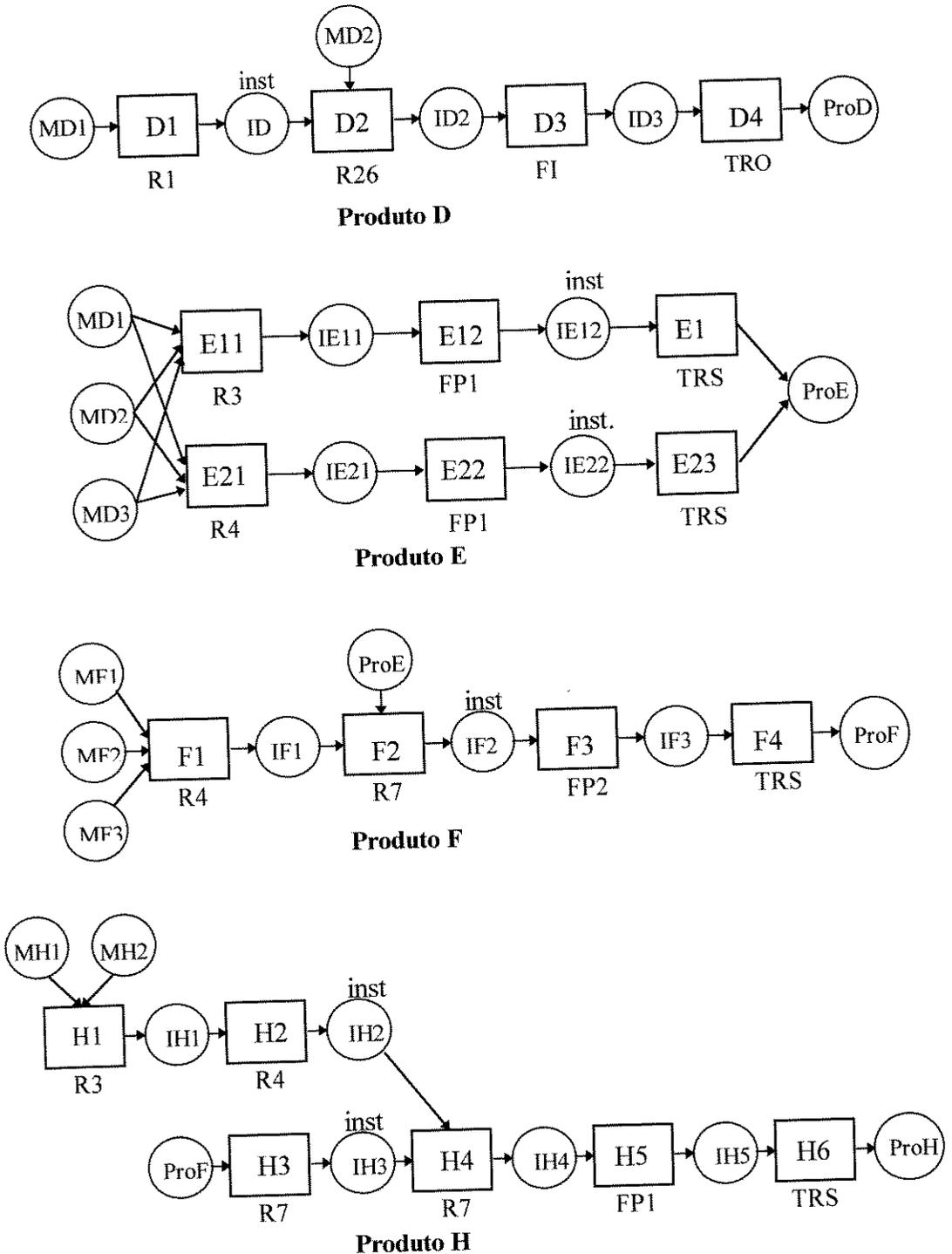


Fig. 4.1. Diagrama STN das linhas de produção

A linha de produção formada pelas tarefas E11, E12 e E13 é denominada *Variante 1 do produto E*, e a linha de produção formada pelas tarefas E21, E22 e E23 é denominada *Variante 2 do produto E*. A diferença entre gerar o produto E pela Variante 1 ou pela Variante 2 está na quantidade gerada deste produto em cada batelada, como pode ser visto na tabela 4.2. A representação STN da figura 4.1 indica sobre cada estado intermediário se o produto é instável.

Tabela 4.2. Tamanho das bateladas dos produtos.

	Pro_D	Pro_E (var 1)	Pro_E (var 2)	Pro_F	Pro_H
Tamanho da batelada	400	300	240	600	300

c) Consumo de Matérias primas

Na figura 4.1, observa-se que os produtos E e F, também são consumidos como matéria prima na produção dos produtos F e H, respectivamente. A seguir, são dados os consumos das matérias primas em cada batelada dos produtos. Na primeira coluna a esquerda da tabela 4.3 são apresentados os produtos finais e nas demais colunas são dadas as quantidades de matérias primas e de produtos intermediários necessários para gerar uma batelada do produto final.

Tabela 4.3. Consumo de Matérias Primas por Batelada de Produto.

Prod. Final	Matérias Primas										Prod. Int.	
	MD1	MD2	ME1	ME2	ME3	MF1	MF2	MF3	MH1	MH2	E	F
D	1000	1500	0	0	0	0	0	0	0	0	0	0
E(v.1)	0	0	740	1020	900	0	0	0	0	0	0	0
E(v.2)	0	0	670	1500	750	0	0	0	0	0	0	0
F	0	0	0	0	0	1000	700	1000	0	0	100	0
H	0	0	0	0	0	0	0	0	700	1400	0	200

d) Dados de Estocagem

Neste problema, são dados o estoque inicial, o estoque mínimo final, o estoque pulmão (Buffer) e o estoque máximo. O estoque pulmão, por razões estratégicas da empresa, não pode ser usado em nenhum momento para atender as necessidades de entrega de produtos. Com isso, o estoque disponível no problema será a diferença entre o estoque total (que não pode ser maior que o estoque máximo) e o estoque pulmão. O estoque inicial, o estoque mínimo final e o estoque máximo têm que ser respeitados.

Tabela 4.4. Dados de estocagem.

	Estoque inicial (Kg)	Estoque mínimo final (Kg)	Estoque pulmão (Buffer)	Estoque máximo (Kg)	Custo manufat (Sfr)
Pro_D	1900	2100	1800	40000	200
Pro_E	2370	2100	2000	30000	250
Pro_F	2170	1800	1500	20000	300
Pro_H	2500	2500	2000	25000	400

O custo de manufatura dado na tabela 4.4 é usado para determinar o capital representado pelo estoque dos produtos, já que o capital retido está sujeito a um custo financeiro de 1% ao mês. Os custos de estocagem de matéria prima não são considerados neste problema, já que a política de estoques de matéria prima é considerada um problema à parte nesta planta.

Os produtos intermediários estão sujeitos à política de armazenagem **NIS** (*No Intermediate Storage*), ou seja, não há estocagem de produtos intermediários fora do processador. Se o produto for estável, ele poderá ser armazenado dentro do processador que o gerou até que o processador seguinte o receba. Desta forma, uma tarefa gerando um produto estável pode ser iniciada independente da disponibilidade do processador seguinte. Para uma tarefa que gera um produto instável, o seu início de processamento está relacionado com a necessidade de se iniciar a tarefa seguinte imediatamente após o término da anterior.

e) Dados de Vendas

O planejamento das necessidades de produção dos produtos deve atender as quantidades que devem ser entregues dentro do horizonte de planejamento. As datas de entregas das vendas com suas respectivas quantidades são especificadas na tabela 4.5. As entregas de vendas devem ser cumpridas sem que hajam atrasos.

Tabela 4.5. Dados das Entregas de Vendas.

Dia		Pro_D Kg	Pro_E Kg	Pro_F Kg	Pro_H Kg
24 jul	Seg	50	20	100	100
25 jul	Ter	100	30	50	150
26 jul	Qua	50	-	20	50
27 jul	Qui	60	-	100	50
28 jul	Sex	80	280	50	70
29 jul	Sab	-	-	-	-
30 jul	Dom	-	-	-	-
31 jul	Seg	40	50	50	130
01 ago	Ter	-	-	-	-
02 ago	Qua	200	100	30	50
03 ago	Qui	-	130	-	30
04 ago	Sex	120	40	20	50
05 ago	Sab	-	-	-	-
06 ago	Dom	-	-	-	-
07 ago	Seg	110	10	10	30
08 ago	Ter	450	20	-	50
09 ago	Qua	10	-	50	50
10 ago	Qui	10	10	60	30
11 ago	Sex	60	-	10	10
12 ago	Sab	-	-	-	-

f) Dados do Consumo de Recursos

Na tabela 4.6, as demandas de eletricidade e vapor são mostradas como sendo uma função de tempo a partir do início de cada tarefa. A quarta coluna (Duração E.E.) e a sexta coluna (Duração Vapor) indicam a duração do consumo de energia elétrica e vapor, respectivamente, a partir do início do processamento. O consumo de outros recursos que não vamos analisar neste problema, como mão de obra, também poderiam ser considerados se isto fosse desejado. Pode-se impor limites na disponibilidade de qualquer recurso. Neste problema, os limites são:

$$\begin{aligned} \text{Oferta máxima de vapor} &= 40.000 \text{ Kcal/h} \\ \text{Oferta máxima de eletricidade} &= 50 \text{ KW} \end{aligned}$$

Tabela 4.6. Dados do Consumo de Recursos.

Produto	Processamento					
	Tarefa (Proc)	Cons. Energ. Elétr. (kW)	Duração E.E (horas)	Cons. Vapor (kcal/h)	Duração Vapor (horas)	Duração da Tarefa (horas)
D	D1 (R1)	5	4	39500	4	4
	D2 (R26)	5	15	18400	5	15
	D3 (F1)	15	10	0	10	10
	D4 (TRO)	30	13	0	13	13
E (var. 1)	E11 (R3)	10	8	10000	8	8
	E12 (FP1)	10	6	0	6	6
	E13 (TRS)	20	15	0	15	15
E (var. 2)	E21 (R4)	5	8	10000	6	8
	E22 (FP1)	5	4	0	4	4
	E23 (TRS)	20	13	0	13	13
F	F1 (R4)	5	11	10000	8	11
	F2 (R7)	10	6	15000	4	6
	F3 (FP2)	12	3	0	3	3
	F4 (TRS)	42	22	0	22	22
H	H1 (R3)	5	7	20000	6	7
	H2 (R4)	0	2	0	2	2
	H3/H4 (R7)	10	8	25000	8	8
	H5 (FP1)	10	10	0	6	10
	H6 (TRS)	20	35	0	35	35

g) Tempo de Duração e Custo da Operação de Preparação (Set-up)

A seguir, são indicados os processadores e as tarefas que necessitam de preparação, e também são dados o custo e a duração de cada operação de preparação. Observa-se na tabela 4.7 que as operações de preparação tem duração e custo dependentes da seqüência.

Tabela 4.7. Dados da operação de preparação

PROC	Tarefa anterior	Tarefa atual	Duração (h)	Custo (Sfr)
R1	D1	D1	24	500
R26	D2	D2	24	600
F1	D3	D3	36	500
TRO	D4	D4	-	-
R3	E11	E11	10	200
	E11	H1	48	500
	H1	H1	10	100
	H1	E11	60	550
FP1	E12/E22	E12/E22	24	300
	E12/E22	H5	36	200
	H5	H5	-	-
	H5	E12/E22	30	400
TRS	E13/E23	E13/E23	8	100
	E13/E23	F4	16	200
	E13/E23	H6	16	200
	F4	E13/E23	20	200
	F4	F4	6	40
	F4	H6	20	200
	H6	E13/E23	24	200
	H6	F4	24	200
	H6	H6	10	100
FP2	F3	F3	12	250
R4	E21	E21	10	50
	E21	F1	20	120
	E21	H2	48	300
	F1	E21	16	100
	F1	F1	10	200
	F1	H2	20	300
	H2	E21	12	120
	H2	F1	24	400
R7	H2	H2	-	-
	F2	F2	8	100
	F2	H3	12	150
	H4	F2	24	200
	H4	H3	10	100
	H3	H4	-	-

h) Recebimento de Matérias Primas

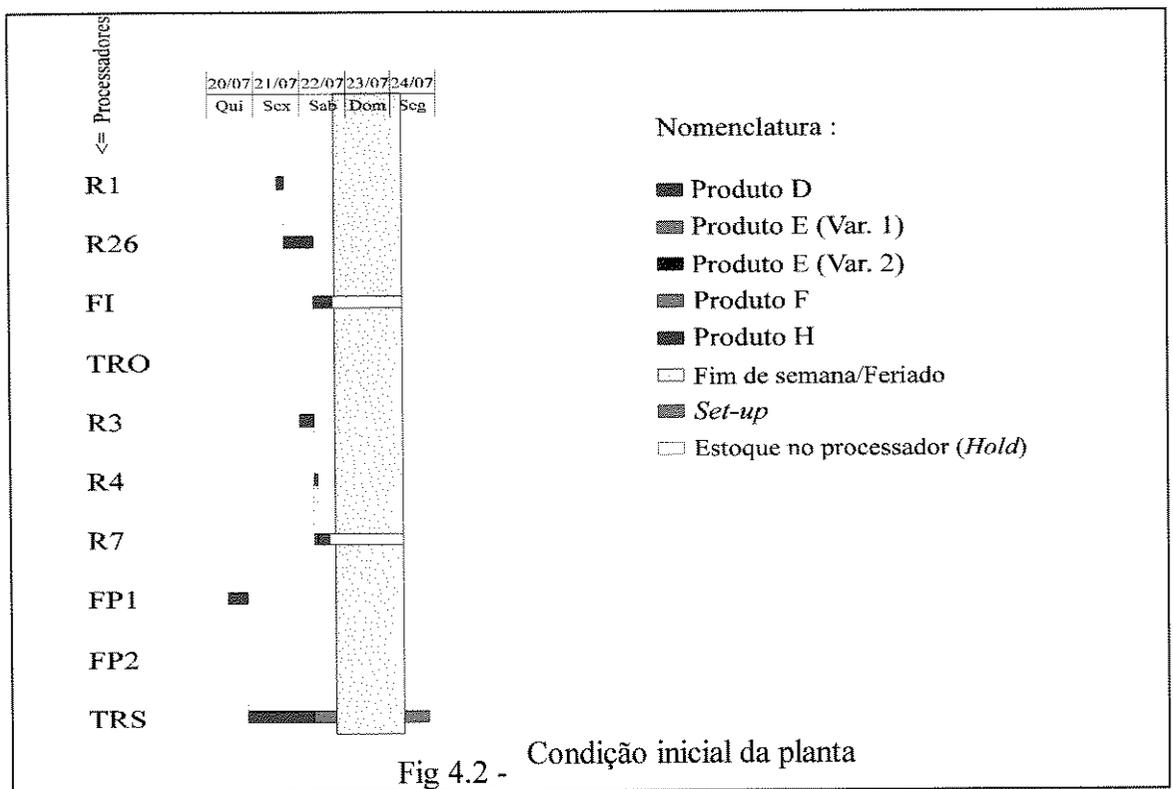
As matérias primas são consideradas disponíveis em estoque em quantidades suficientes para suprir quaisquer demandas, a exceção de ME1 que tem restrições no instante de disponibilidade e quantidade dadas pela tabela 4.8.

Tabela 4.8. Prazo máximo e quantidades a serem recebidas da matéria prima ME1.

	Dia	Horário	Qtde
Recebimento de matéria prima	24 jul	Seg_1	1000
	5 ago	Sab_2	1500
	13 ago	-	12:00

i) Condição Inicial do Problema

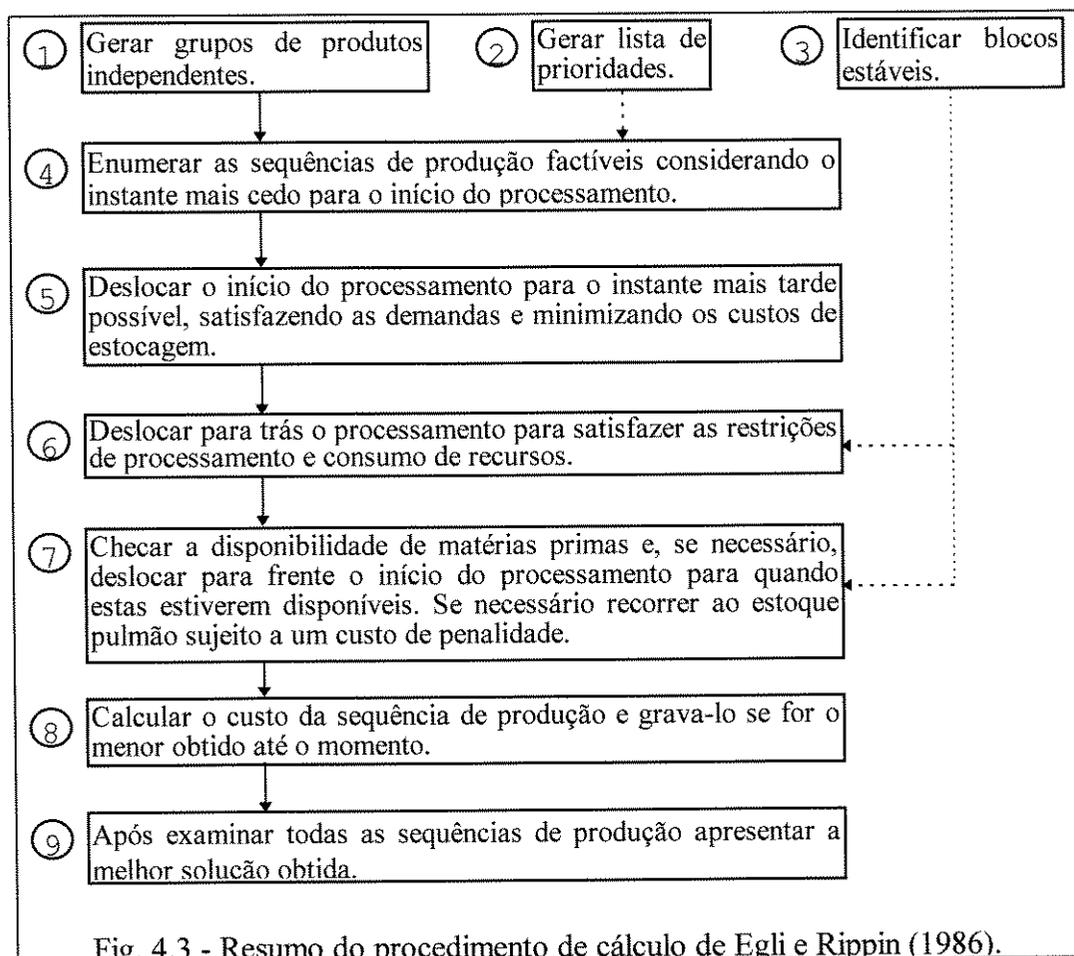
O planejamento deve ser feito a partir do dia 24 de julho (Seg), e para isto é preciso levar em consideração todos os acontecimentos ocorridos pouco antes deste dia e no início deste dia. Na figura 4.2 é apresentada a carta de Gantt da condição inicial da planta, com todas as tarefas e operações de preparação que ocorreram pouco antes do planejamento ou que vão estar sendo processadas no início do período em questão.



4.2. Procedimento heurístico utilizado por Egli e Rippin.

Egli e Rippin (1986) desenvolveram um método heurístico complexo para determinar uma solução para o problema de sequenciamento de produção, que

resumidamente está representado na figura 4.3. Basicamente, o método consiste na geração de todas as sequências possíveis das variantes de produção através de um procedimento enumerativo. Sequências não factíveis ou não ótimas são progressivamente eliminadas de tal forma que somente os candidatos favoráveis estão sujeitos a uma avaliação completa. Para cada sequência, os custos de estocagem são minimizados alocando-se as bateladas das tarefas o mais tarde possível, mas atendendo os prazos de entrega.



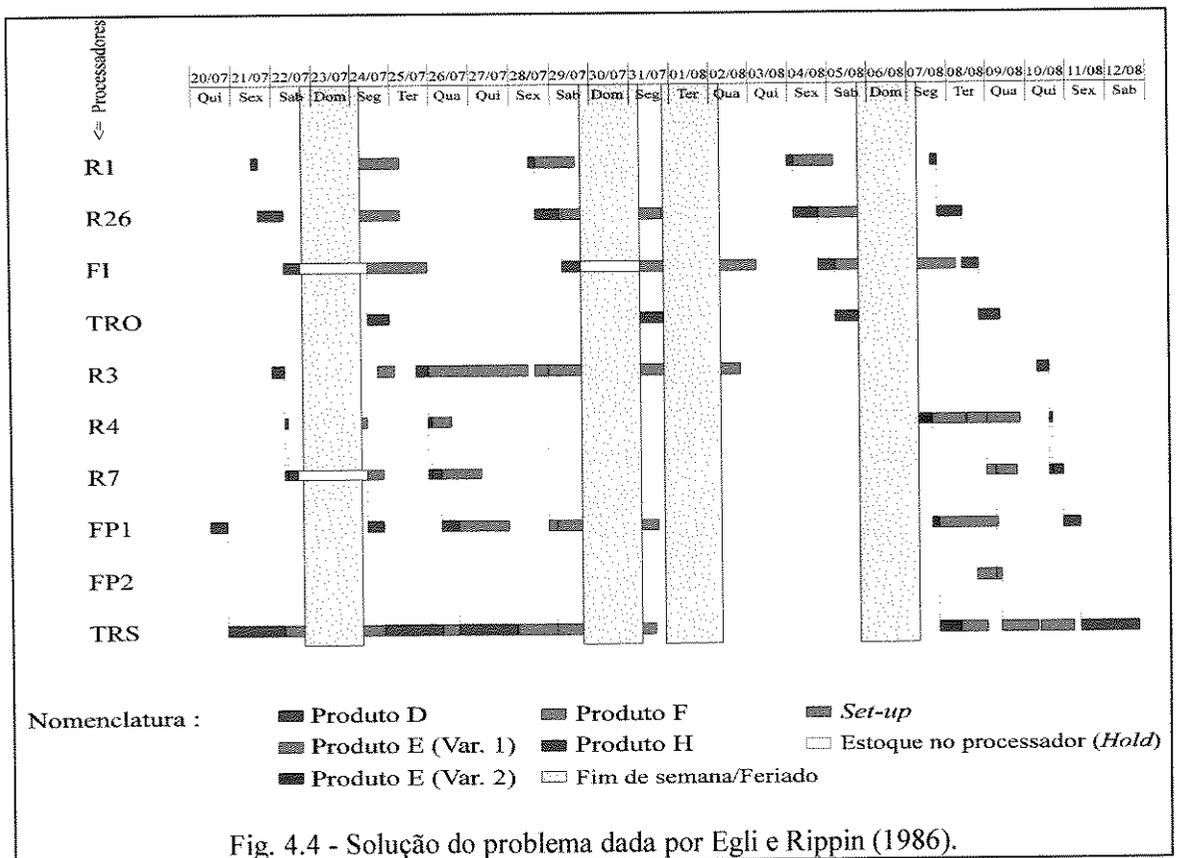
Um consumo de recursos superior ao disponível pode ser causado pela demanda simultânea de diversas tarefas do mesmo produto ou de produtos diferentes. Partindo-se da sua última ocorrência, o consumo excessivo de recursos é corrigido deslocando-se o início do processamento das tarefas ou blocos de tarefas estáveis para algum instante anterior ao previsto. O início do processamento de uma ou mais tarefas é deslocado até que o consumo dos recursos seja inferior ao seu limite. Entre as tarefas que contribuem para o consumo excessivo de recursos, as candidatas a serem deslocadas são avaliadas partindo-se da tarefa com início de processamento mais cedo até a última tarefa a iniciar

o seu processamento. Outras seqüências de produção destas tarefas serão avaliadas até que a restrição no consumo de recursos seja satisfeita.

Todas as maneiras possíveis de corrigir um excesso de demanda são consideradas a cada violação das restrições. Quando um excesso na demanda de recursos ocorre, são testadas várias seqüências de produção para as tarefas que causam este excesso de demanda. Em cada uma destas seqüências, consideram-se as tarefas a serem deslocadas na ordem em que elas aparecem na seqüência de produção, ou seja, partindo-se da tarefa com início de processamento mais cedo até a última tarefa a iniciar o seu processamento.

Quando é gerada a seqüência de produção, alocando-se as tarefas o mais tarde possível, considerando-se apenas os prazos de entrega, são gerados conflitos no consumo de recursos. O método proposto avalia estes conflitos, partindo daquele que está mais próximo do fim do horizonte de planejamento até chegar ao conflito mais próximo do início deste horizonte.

O custo de cada seqüência de produção, obtida através desta heurística, é calculado e a seqüência de menor custo é gravada, ou se desejado, um número de seqüências com custo próximo ao menor obtido são gravadas. A seguir é apresentada a solução encontrada pelos autores usando este método heurístico.



4.3. Solução utilizando a Representação Estado-Tarefa (STN)

a) Planejamento da Produção

A primeira etapa na solução deste problema é o planejamento da produção, quando se define quantas bateladas de cada produto devem ser produzidas dentro do horizonte de planejamento. O primeiro passo no planejamento é definir o tamanho dos *slots*. Neste problema decidiu-se que cada *slot* equivale a 4 horas. Para que fosse adotado este tamanho de *slot* foram feitos arredondamentos nos tempos de processamento e *set up*, já que o máximo divisor comum destes tempos (tabelas 4.6 e 4.7) é uma hora. Isto foi feito para que fosse possível resolver o problema proposto em um microcomputador PC 486 de 100 Mhz com 32 Mbytes de memória RAM.

A tabela 4.9 indica os tempos de processamento em *slots* e o consumo de recursos das tarefas. A tabela 4.10 indica o tempos de *set up* em *slots* e o seu custo.

Tabela 4.9. Dados do Consumo de Recursos.

Produto	Tarefa (Proc)	Processamento				
		Cons. Energ. Elétr. (kW)	Duração E.E (slots)	Cons. Vapor (kcal/h)	Duração Vapor (slots)	Duração da Tarefa (slots)
D	D1 (R1)	5	1	39500	1	1
	D2 (R26)	5	4	18400	1	4
	D3 (F1)	15	2	0	2	2
	D4 (TRO)	30	3	0	3	3
E (var. 1)	E11 (R3)	10	2	10000	2	2
	E12 (FP1)	10	1	0	1	1
	E13 (TRS)	20	3	0	3	3
E (var. 2)	E21 (R4)	5	2	10000	2	2
	E22 (FP1)	5	1	0	1	1
	E23 (TRS)	20	3	0	3	3
F	F1 (R4)	5	3	10000	2	3
	F2 (R7)	10	1	15000	1	1
	F3 (FP2)	12	1	0	1	1
	F4 (TRS)	42	5	0	5	5
H	H1 (R3)	5	2	20000	1	2
	H2 (R4)	0	1	0	1	1
	H3/H4 (R7)	10	2	25000	2	2
	H5 (FP1)	10	2	0	2	2
	H6 (TRS)	20	7	0	7	7

Tabela 4.10. Dados da operação de preparação

PROC	Tarefa anterior	Tarefa atual	Duração (slots)	Custo (Sfr)
R1	D1	D1	6	500
R26	D2	D2	6	600
F1	D3	D3	9	500
TRO	D4	D4	-	-
R3	E11	E11	3	200
	E11	H1	12	500
	H1	H1	3	100
	H1	E11	15	550
FP1	E12/E22	E12/E22	6	300
	E12/E22	H5	9	200
	H5	H5	-	-
	H5	E12/E22	8	400
TRS	E13/E23	E13/E23	2	100
	E13/E23	F4	4	200
	E13/E23	H6	4	200
	F4	E13/E23	0	200
	F4	F4	2	40
	F4	H6	5	200
	H6	E13/E23	6	200
	H6	F4	6	200
	H6	H6	3	100
FP2	F3	F3	3	250
R4	E21	E21	3	50
	E21	F1	5	120
	E21	H2	12	300
	F1	E21	4	100
	F1	F1	3	200
	F1	H2	5	300
	H2	E21	3	120
	H2	F1	6	400
	H2	H2	-	-
R7	F2	F2	2	100
	F2	H3	3	150
	H4	F2	6	200
	H4	H3	3	100
	H3	H4	-	-

A partir dos dados de vendas (tabela 4.5) e das demandas internas é possível calcular quantas bateladas de cada produto terão que ser produzidas dentro do horizonte

de planejamento. As demandas internas correspondem ao consumo de um produto como matéria prima para a produção de outro produto. A tabela 4.3 indica estas demandas (Cada batelada do produto F consome 100 kg do produto E e cada batelada do produto H consome 200 kg do produto F). O total vendido dentro do horizonte de planejamento de cada produto é apresentado na tabela 4.11.

Tabela 4.11. Total vendido dos produtos.

PRODUTO	TOTAL
D	1.340 kg
E	690 kg
F	550 kg
H	850 kg

Antes de calcular o número de bateladas a serem produzidas para cada produto, é preciso determinar o total de estoque disponível para ser entregue aos clientes, já que o estoque pulmão não deve ser utilizado. A tabela 4.12 indica os valores do estoque inicial, do estoque mínimo final de cada produto, descontando-se o estoque pulmão, e a diferença entre estes estoques. Se a diferença entre o estoque inicial e o estoque mínimo final for positiva, ela indicará o total disponível em estoque para satisfazer às vendas. Se esta diferença for negativa, ela indicará o total de produto que deverá ser produzido para satisfazer o estoque mínimo final.

Tabela 4.12. Estoque inicial e estoque mínimo final disponíveis, e a sua diferença.

Produto	Estoque Inicial (kg)	Estoque mínimo final (kg)	Diferença entre estoque mínimo final e inicial (kg)
D	100	300	200
E	370	100	- 270
F	670	300	- 370
H	500	500	0

Para se determinar o número de bateladas a serem produzidas para um produto é preciso levar em consideração as vendas do produto (tabela 4.11), as demandas internas (tabela 4.3) e a diferença entre o estoque inicial e o estoque mínimo final. A tabela 4.13 indica a produção requerida de cada produto, atendendo-se as demandas e o estoque mínimo final. Dividindo-se a produção requerida pelo tamanho da batelada do produto,

obtem-se o número de bateladas do produto a serem produzidas. Observe que a ordem em que são apresentados os produtos E, F e H visa facilitar a compreensão do leitor das demandas internas dos produtos, já que o número de bateladas a serem produzidas dos produtos H e F indicam quais serão as demandas internas dos produtos F e E, respectivamente.

Tabela 4.13. Produção requerida e o número de bateladas a serem produzidas

Produto	Vendas	Dem int	Dif. est.	Prod req	Tam bat	P.r./T.b.	Total bat
D	1.340	0	200	1.540	400	3,85	4
H	850	0	0	850	300	2,833	3
F	550	400 *	- 370	580	600	0,967	1
E (var 1)	690	100	- 270	520	300	1,733	2
E (var 2)	690	100	- 270	520	240	2,16	3

onde:

Vendas Indica o total de produto vendido (tabela 4.11).

Dem. int. Indica a demanda interna do produto. No produto F, a demanda interna é de 400 kg (e não 600 kg) porque a primeira batelada do produto H já está em processamento no início do horizonte de planejamento.

Dif. est. Indica a diferença entre o estoque mínimo final e o estoque inicial.

Prod. req. Indica a produção requerida.

Tam. bat. Indica o tamanho da batelada.

P.r./T.b. Indica o valor da divisão da produção requerida pelo tamanho da batelada.

Total bat. Indica o número de bateladas do produto a ser produzida.

As tabelas 4.14 e 4.15 indicam até quando as bateladas dos produtos devem ficar prontas (cálculo do fim das janelas dos produtos). A finalidade destas duas tabelas é calcular o fim das janelas das bateladas dos produtos e checar se não há falta de matéria prima (tabela 4.8) para atender as demandas dos produtos. Apesar de o único produto com recebimento de matéria prima ser o produto E, é preciso determinar quando o produto F irá consumir o produto E, e quando o produto H consumirá o produto F.

Nas tabelas 4.14 e 4.15, a primeira coluna de cada produto (**Deman.**), indica as quantidades de produto a serem entregues aos clientes naquele dia (apresentadas nestas tabelas em caracteres normais) e as demandas internas deste produto (apresentadas em caracteres itálicos). A segunda coluna de cada produto (**Estoque**) indica a quantidade de

produto disponível em estoque no início daquele dia. A terceira coluna de cada produto (**Batel.**) indica até quando uma batelada de um produto deve ficar pronta, apresentando a quantidade de produto que ficará pronta. A coluna de **Estoque** é rachurada toda vez que o estoque de produto é insuficiente para atender às demandas, indicando que até este dia deve ficar pronta uma batelada deste produto.

Tabela 4.14. Cálculo do fim das janelas dos produtos D, H e F.

DIA	Produto D			Produto H			Produto F		
	Deman.	Estoque	Batel.	Deman.	Estoque	Batel.	Deman.	Estoque	Batel.
24 jul - Seg	50	100	-	100	500		100	670	-
25 jul - Ter	100	50	400	150	400	-	50	570	-
26 jul - Qua	50	350	-	50	250	-	20	520	-
27 jul - Qui	60	300	-	50	200	-	100	500	-
28 jul - Sex	80	240	-	70	150	-	50	400	-
29 jul - Sab	-	160	-	-	80	-	-	350	-
30 jul - Dom	-	160	-	-	80	-	-	350	-
31 jul - Seg	40	160	-	130	80	300	50	350	-
01 ago - Ter	-	120	-	-	250	-	-	300	-
02 ago - Qua	200	120	400	50	250	-	30	300	-
03 ago - Qui	-	320	-	30	200	-	-	270	-
04 ago - Sex	120	320	-	50	170	-	20	270	-
05 ago - Sab	-	200	-	-	120	-	-	250	-
06 ago - Dom	-	200	-	-	120	-	-	250	-
07 ago - Seg	110	200	-	30	120	-	10	250	-
08 ago - Ter	450	90	400	50	90	-	200	240	-
09 ago - Qua	10	40	-	50	40	300	50	40	600
10 ago - Qui	10	30	-	30	290	-	60 + 200	590	-
11 ago - Sex	60	20	400	10	260	-	10	330	-
12 ago - Sab	-	360	-	-	250	300	-	320	-
13 ago - Dom	-	360	-	-	550	-	-	320	-

As demandas internas do produto F que aparecem na tabela 4.14 nos dias 08 e 10 de agosto, são consequência da necessidade de concluir as bateladas do produto H até os dias 09 e 12 de agosto. Observe que após o consumo (demanda interna) do produto F pelo produto H, são necessários pelo menos 11 *slots* para que o produto H fique pronto.

Na tabela 4.15 são apresentadas 2 combinações diferentes das variantes do produto E. O primeiro caso (Produto E - var 1) corresponde à produção de duas bateladas da variante 1 do produto E. O segundo caso (Produto E - var1+var2) corresponde à produção de uma batelada da variante 1 e depois uma batelada da variante

2. O terceiro caso (Produto E - var 2) corresponde à produção de 3 bateladas da variante 2. Teoricamente seria possível apresentar um quarto caso produzindo uma batelada da variante 2, e depois, produzindo uma batelada da variante 1. Porém, como será visto adiante, no terceiro caso (Produto E - var 2), se a primeira batelada produzida do produto E for da variante 2, esta sequência de produção será infactível.

Ao calcular o número de bateladas de cada produto a serem produzidas no horizonte de planejamento, na tabela 4.13, determinou-se que é preciso produzir 2 bateladas da variante 1 do produto E ou 3 bateladas da variante 2 do produto E. Como pode ser visto na tabela 4.15, seria preciso concluir a segunda batelada da variante 2 até o dia 04 de agosto (sexta-feira). Entretanto, não é possível concluir esta batelada neste prazo, já que não há matéria prima ME1 para iniciar esta segunda batelada, visto que esta matéria prima (tabela 4.8) só vai ser recebida no dia 05 de agosto (sábado). Portanto, é infactível produzir 3 bateladas da variante 2 ou produzir qualquer outra combinação em que a primeira batelada do produto E seja da variante 2.

Tabela 4.15. Cálculo do fim das janelas do produto E.

DIA	Produto E - var 1			Produto E - var1+var2			Produto E - var 2		
	Deman.	Estoque	Batel	Deman.	Estoque	Batel	Deman.	Estoque	Batel
24 jul - Seg	20	370	-	20	370	-	20	370	-
25 jul - Ter	30	350	-	30	350	-	30	350	-
26 jul - Qua	-	320	-	-	320	-	-	320	-
27 jul - Qui	-	320	-	-	320	-	-	320	-
28 jul - Sex	280	320	-	280	320	-	280	320	-
29 jul - Sab	-	40	-	-	40	-	-	40	-
30 jul - Dom	-	40	-	-	40	-	-	40	-
31 jul - Seg	50	40	300	50	40	300	50	40	240
01 ago - Ter	-	290	-	-	290	-	-	230	-
02 ago - Qua	100	290	-	100	290	-	100	230	-
03 ago - Qui	130	190	-	130	190	-	130	130	-
04 ago - Sex	40	60	-	40	60	-	40	0	infactível
05 ago - Sab	-	20	-	-	20	-	-		
06 ago - Dom	-	20	-	-	20	-	-		
07 ago - Seg	10	20	-	10	20	-	10		
08 ago - Ter	20 + 100	10	300	20 + 100	10	240	20 + 100		
09 ago - Qua	-	190	-	-	130	-	-		
10 ago - Qui	10	190	-	10	130	-	10		
11 ago - Sex	-	180	-	-	120	-	-		
12 ago - Sab	-	180	-	-	120	-	-		
13 ago - Dom	-	180	-	-	120	-	-		

A partir das tabelas 4.14 e 4.15 é possível gerar as janelas de tempo das tarefas e das operações de preparação, seguindo-se os mesmos passos feitos na seção 3.4. Ao gerar as janelas de tempo, conclui-se a fase de planejamento da produção, partindo-se para a fase de sequenciamento da produção.

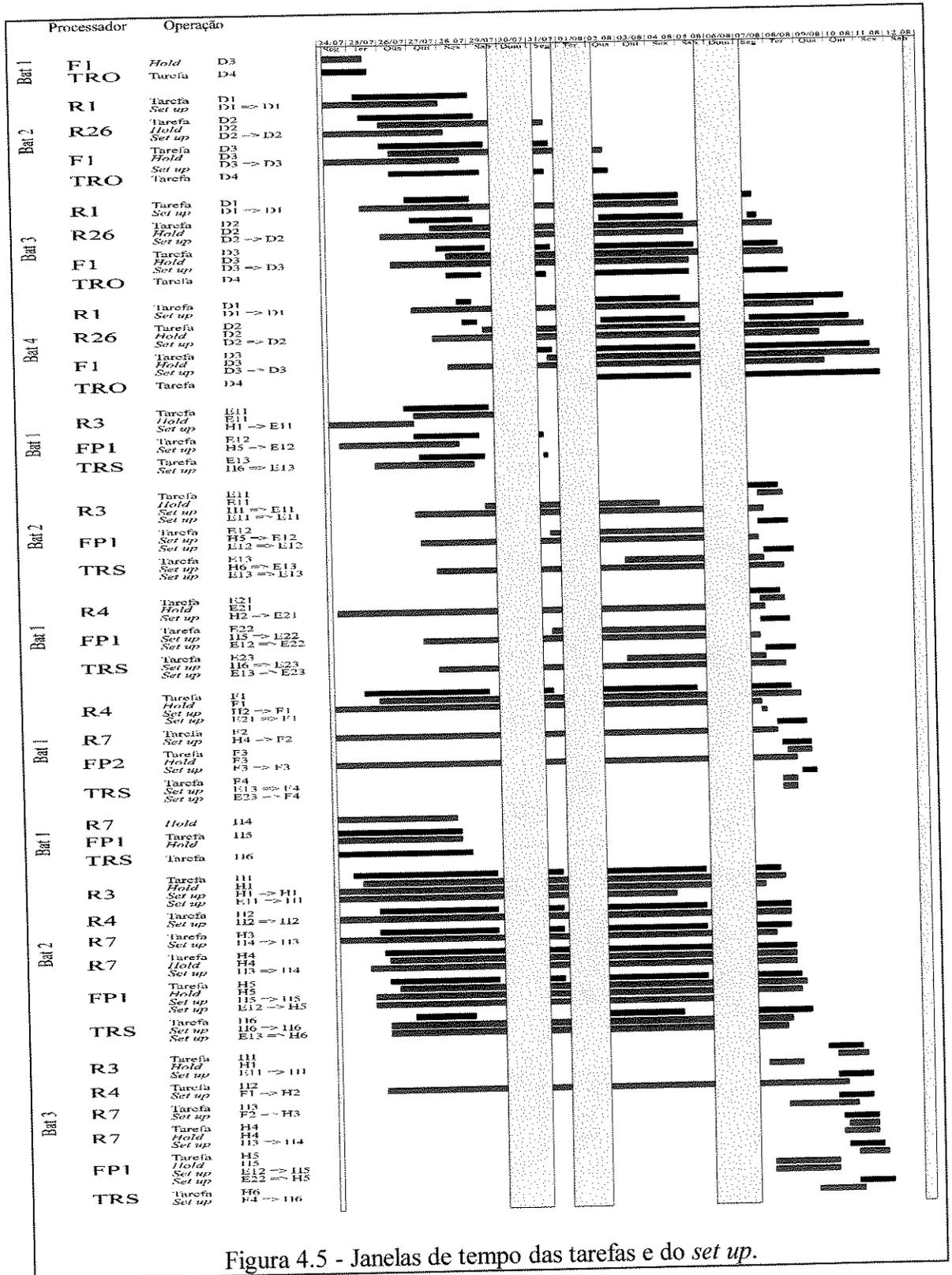


Figura 4.5 - Janelas de tempo das tarefas e do set up.

A figura 4.5 apresenta as janelas de tempo das tarefas e das operações de preparação (*set up*) para todas as bateladas produzidas dentro do horizonte de planejamento em todos os processadores. Nesta figura só são apresentadas as tarefas e operações de preparação que serão executadas dentro do horizonte de planejamento. Por isso, as janelas de tempo da batelada 1 das tarefas **D1**, **D2**, **H1**, **H2** e **H3**, e todas as operações de preparação que foram iniciadas antes do horizonte de planejamento, não são apresentadas nesta figura. Observe que na tabela 4.14, a primeira batelada do produto **H** poderia ser encerrada até a segunda feira (dia 31/07). Como não é possível iniciar e encerrar o processamento da tarefa **H6** no dia 31/07, considera-se que esta batelada deve ficar pronta até o sábado (dia 29/07).

b) Sequenciamento da Produção

A tabela 4.16 apresenta os resultados obtidos para o sequenciamento deste problema, usando-se a formulação apresentada no capítulo 3 e utilizando as equações [3.11a] e [3.12a] para a alocação do *set-up*. Como usando-se as equações [3.11] e [3.12] para a alocação do *set-up*, não foi encontrada nenhuma solução factível para o problema em 8.000 segundos (de tempo de CPU) de busca, apenas a dimensão do problema será apresentada na tabela 4.17. A solução usando-se a formulação do capítulo 2 não foi obtida neste microcomputador por exceder a memória RAM disponível, não sendo possível sequer apresentar a sua dimensão. A rotina correspondente ao caso (b) é apresentada no anexo 1.

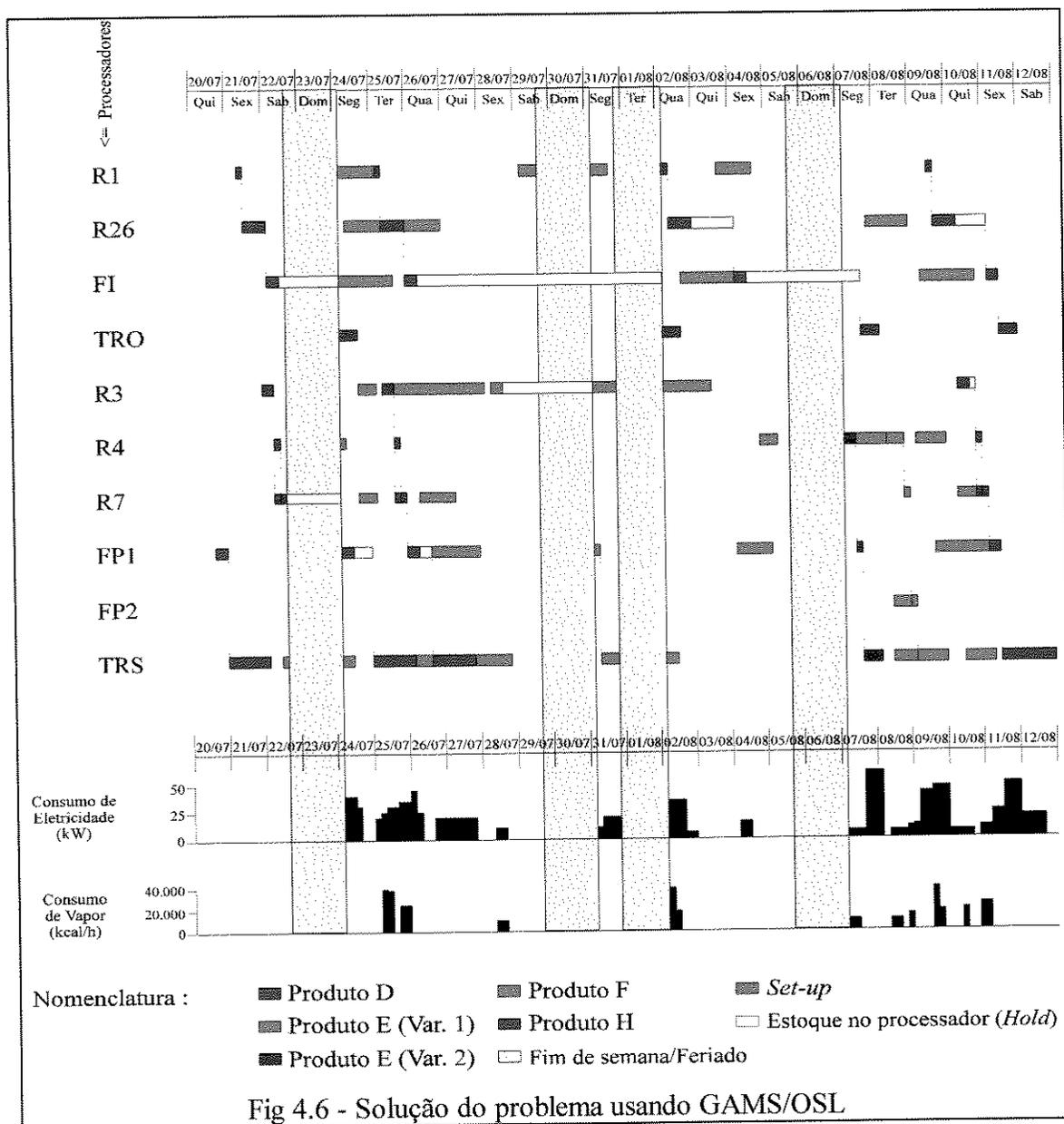
Tabela 4.16. Resumo da solução do problema usando-se as equações [3.11a] e [3.12a] para a alocação do *set up*.

Blocos de equações	22
Blocos de variáveis	8
Equações geradas	2.520
Variáveis simples	4.374
Variáveis discretas	3.441
Elementos não zeros	31.431
Solução inteira obtida	11.693.378
Solução relaxada obtida	10.330,351
Diferença entre as soluções (%)	13.194 %
OPTCR - Diferença aceitável (%)	15 %
Tempo de CPU (segundos)	1.856
Número de iterações feitas	19.401
Número de nós sondados	337
Memória RAM alocada	7.56

Tabela 4.17. Resumo da dimensão do problema usando-se as equações [3.11] e [3.12] para a alocação do *set up*.

Blocos de equações	22
Blocos de variáveis	8
Equações geradas	4.389
Variáveis simples	4.374
Variáveis discretas	3.441
Elementos não zeros	88.572
Memória RAM alocada	18.99

A solução deste problema, utilizando-se as equações [3.11a] e [3.12a] para a alocação do *set up*, é apresentada na figura 4.6.



CAPÍTULO 5 - CONCLUSÃO

Neste trabalho foi apresentada uma nova formulação para o seqüenciamento da produção em plantas, com alocação de *set ups* dependentes da seqüência de produção, utilizando a representação Rede Estado-Tarefa (*State-Task Network*). Esta formulação aplica-se a plantas multipropósito, com processamento em bateladas, e com restrições sobre os recursos compartilhados.

Com a formulação proposta para a alocação de *set ups* dependentes da seqüência de produção, tornou-se possível solucionar o problema apresentado por Egli e Rippin (1986), utilizando a representação Rede Estado-Tarefa (*State-Task Network*). A utilização da formulação apresentada por Kondili *et al* (1993) para a alocação de *set ups* dependentes da seqüência de produção gera um problema com dimensão muito mais elevada (impossibilitando a solução no mesmo computador).

A formulação proposta foi desenvolvida para problemas de seqüenciamento com três características: i) a alocação das operações de preparação deve ser modelada porque elas tem custos associados e (ou) utilizam recursos compartilhados capacitados, ii) as operações de preparação são dependentes da seqüência e iii) a porcentagem de operações que exigem preparação dos processadores é grande. Na situação em que existem poucas operações de preparação a formulação de Kondili *et al.* (1993) segue sendo outra alternativa dado que utiliza um menor número de variáveis binárias, definindo estas apenas para pares de operações entre as quais existe uma operação de preparação.

Como prosseguimento deste trabalho está em andamento a introdução desta formulação numa estratégia de horizonte rolante.

Anexo 1 - Rotina do problema do capítulo 4 no GAMS/OSL

Dados do Problema (REGBATH2.INC)

```
$TITLE Artigo Egli (1986)
$OFFUPPER
$OFFSYMXREF
$OFFSYMLIST

SETS
  DIA /SEM_0
      SEG_1,TER_1,QUA_1,QUI_1,SEX_1,SAB_1
      SEG_2,      QUA_2,QUI_2,SEX_2,SAB_2
      SEG_3,TER_3,QUA_3,QUI_3,SEX_3,SAB_3/
  SLOT Slots:Horizonte incluindo o slot H+1(Shah)/SL1*SL97/
  HORIZ(SLOT) Horizonte de operacao(sem H+1);
  HORIZ(SLOT)=YES;
  LOOP(SLOT,
    IF(ORD(SLOT) EQ CARD(SLOT),
      HORIZ(SLOT)=NO;
    );
  );

SETS
  HOR_INI(SLOT) Horizonte factivel inicial;
  HOR_INI(SLOT)=NO;
  LOOP(SLOT,
    IF(ORD(SLOT) GE 6 AND ORD(SLOT) LE CARD(SLOT),
      HOR_INI(SLOT)=YES;
    );
  );

SETS
  HOR_DIA(DIA) Dias pertencentes ao horizonte factivel;
  HOR_DIA(DIA) = YES;
  HOR_DIA('SEM_0') = NO;

  ALIAS(SLOT,SLOTL);
  ALIAS(SLOT,SLOT1);
  ALIAS(SLOT,SLOT2);

SETS
  SLOT_DIA(SLOT,DIA) Slots que pertencem ao DIA;
  SLOT_DIA(SLOT,DIA)=NO;
  LOOP((SLOT,DIA),
    IF(ORD(SLOT) LE 5,
      SLOT_DIA(SLOT,'SEM_0')=YES;
    );
    IF(ORD(SLOT) GT 5 AND ORD(SLOT) LE 10,
      SLOT_DIA(SLOT,'SEG_1')=YES;
    );
    IF(ORD(DIA) GT 2 AND ORD(DIA) LE 6 AND ORD(SLOT) GT
      ((ORD(DIA)-1)*6-2) AND ORD(SLOT) LE (ORD(DIA)*6-2),
      SLOT_DIA(SLOT,DIA)=YES;
    );
    IF(ORD(SLOT) LE 38 AND ORD(SLOT) GT 34,
      SLOT_DIA(SLOT,'SAB_1')=YES;
    );
    IF(ORD(SLOT) LE 42 AND ORD(SLOT) GT 38,
      SLOT_DIA(SLOT,'SEG_2')=YES;
    );
    IF(ORD(SLOT) LE 47 AND ORD(SLOT) GT 42,
```

```

        SLOT_DIA(SLOT, 'QUA_2')=YES;
    );
    IF(ORD(DIA) GT 9 AND ORD(DIA) LE 11 AND ORD(SLOT) GT
        ((ORD(DIA)-1)*6-7) AND ORD(SLOT) LE ORD(DIA)*6-7,
        SLOT_DIA(SLOT, DIA)=YES;
    );
    IF(ORD(SLOT) LE 63 AND ORD(SLOT) GT 59,
        SLOT_DIA(SLOT, 'SAB_2')=YES;
    );
    IF(ORD(SLOT) LE 68 AND ORD(SLOT) GT 63,
        SLOT_DIA(SLOT, 'SEG_3')=YES;
    );
    IF(ORD(DIA) GT 13 AND ORD(DIA) LE 17 AND ORD(SLOT) GT
        ((ORD(DIA)-1)*6-10) AND ORD(SLOT) LE ORD(DIA)*6-10,
        SLOT_DIA(SLOT, DIA)=YES;
    );
    IF(ORD(SLOT) LE 97 AND ORD(SLOT) GT 92,
        SLOT_DIA(SLOT, 'SAB_3')=YES;
    );
);

```

SETS

```

BAT Indice das bateladas/BAT1*BAT4/
TETA Indice para consumo variavel de recursos(dimensao tp)/T0*T6/
PROC Processadores/R1, R26, F1, TRO, R3, R4, R7, FP1, FP2, TRS/
OPR Operacoes/D1, D2, D3, D4, HOLD2, HOLD3
    E11, E12, E13, HOLE11
    E21, E22, E23, HOLE21
    F1, F2, F3, F4, HOLF1, HOLF3
    H1, H2, H3, H4, H5, H6, HOLH1, HOLH4, HOLH5/

```

```

ESTADOS Produtos e intermediarios/IH1, IH2, IH3, IH4, IH5, PRO_H
    ID3, ID2, ID1, PRO_D
    IE11, IE12, PRO_E
    IE21, IE22
    IF1, IF2, IF3, PRO_F/

```

```

PRD Linhas de produtos/D, E1, E2, F, H/

```

```

PRODUTOS (ESTADOS);
PRODUTOS (ESTADOS)=NO;
PRODUTOS ('PRO_D')=YES;
PRODUTOS ('PRO_E')=YES;
PRODUTOS ('PRO_F')=YES;
PRODUTOS ('PRO_H')=YES;

```

SETS

```

HOR_BAT(BAT, OPR) Horizonte das bateladas;
HOR_BAT(BAT, OPR)=NO;
HOR_BAT('BAT1', 'D1')=YES;
HOR_BAT('BAT1', 'D2')=YES;
HOR_BAT('BAT1', 'HOLD2')=YES;
HOR_BAT('BAT1', 'D3')=YES;
HOR_BAT('BAT1', 'HOLD3')=YES;
HOR_BAT('BAT1', 'D4')=YES;
HOR_BAT('BAT2', 'D1')=YES;
HOR_BAT('BAT2', 'D2')=YES;
HOR_BAT('BAT2', 'HOLD2')=YES;
HOR_BAT('BAT2', 'D3')=YES;
HOR_BAT('BAT2', 'HOLD3')=YES;
HOR_BAT('BAT2', 'D4')=YES;
HOR_BAT('BAT3', 'D1')=YES;
HOR_BAT('BAT3', 'D2')=YES;
HOR_BAT('BAT3', 'HOLD2')=YES;
HOR_BAT('BAT3', 'D3')=YES;

```

```

HOR_BAT('BAT3','HOLD3')=YES;
HOR_BAT('BAT3','D4')=YES;
HOR_BAT('BAT4','D1')=YES;
HOR_BAT('BAT4','D2')=YES;
HOR_BAT('BAT4','HOLD2')=YES;
HOR_BAT('BAT4','D3')=YES;
HOR_BAT('BAT4','HOLD3')=YES;
HOR_BAT('BAT4','D4')=YES;
HOR_BAT('BAT1','E11')=YES;
HOR_BAT('BAT1','HOLE11')=YES;
HOR_BAT('BAT1','E12')=YES;
HOR_BAT('BAT1','E13')=YES;
HOR_BAT('BAT2','E11')=YES;
HOR_BAT('BAT2','HOLE11')=YES;
HOR_BAT('BAT2','E12')=YES;
HOR_BAT('BAT2','E13')=YES;
HOR_BAT('BAT1','E21')=YES;
HOR_BAT('BAT1','HOLE21')=YES;
HOR_BAT('BAT1','E22')=YES;
HOR_BAT('BAT1','E23')=YES;
HOR_BAT('BAT1','F1')=YES;
HOR_BAT('BAT1','HOLF1')=YES;
HOR_BAT('BAT1','F2')=YES;
HOR_BAT('BAT1','F3')=YES;
HOR_BAT('BAT1','HOLF3')=YES;
HOR_BAT('BAT1','F4')=YES;
HOR_BAT('BAT1','H1')=YES;
HOR_BAT('BAT1','HOLH1')=YES;
HOR_BAT('BAT1','H2')=YES;
HOR_BAT('BAT1','H3')=YES;
HOR_BAT('BAT1','H4')=YES;
HOR_BAT('BAT1','HOLH4')=YES;
HOR_BAT('BAT1','H5')=YES;
HOR_BAT('BAT1','HOLH5')=YES;
HOR_BAT('BAT1','H6')=YES;
HOR_BAT('BAT2','H1')=YES;
HOR_BAT('BAT2','HOLH1')=YES;
HOR_BAT('BAT2','H2')=YES;
HOR_BAT('BAT2','H3')=YES;
HOR_BAT('BAT2','H4')=YES;
HOR_BAT('BAT2','HOLH4')=YES;
HOR_BAT('BAT2','H5')=YES;
HOR_BAT('BAT2','HOLH5')=YES;
HOR_BAT('BAT2','H6')=YES;
HOR_BAT('BAT3','H1')=YES;
HOR_BAT('BAT3','HOLH1')=YES;
HOR_BAT('BAT3','H2')=YES;
HOR_BAT('BAT3','H3')=YES;
HOR_BAT('BAT3','H4')=YES;
HOR_BAT('BAT3','HOLH4')=YES;
HOR_BAT('BAT3','H5')=YES;
HOR_BAT('BAT3','HOLH5')=YES;
HOR_BAT('BAT3','H6')=YES;

```

SETS

```

HOR_SET(BAT,OPR) Indica se BAT de OPR tem setup precedente;
HOR_SET(BAT,OPR)=NO;
HOR_SET('BAT2','D1')=YES;
HOR_SET('BAT2','D2')=YES;
HOR_SET('BAT2','D3')=YES;
HOR_SET('BAT3','D1')=YES;
HOR_SET('BAT3','D2')=YES;
HOR_SET('BAT3','D3')=YES;
HOR_SET('BAT4','D1')=YES;
HOR_SET('BAT4','D2')=YES;

```

```

HOR_SET('BAT4','D3')=YES;
HOR_SET('BAT1','E11')=YES;
HOR_SET('BAT1','E12')=YES;
HOR_SET('BAT1','E13')=YES;
HOR_SET('BAT2','E11')=YES;
HOR_SET('BAT2','E12')=YES;
HOR_SET('BAT2','E13')=YES;
HOR_SET('BAT1','E21')=YES;
HOR_SET('BAT1','E22')=YES;
HOR_SET('BAT1','E23')=YES;
HOR_SET('BAT1','F1')=YES;
HOR_SET('BAT1','F2')=YES;
HOR_SET('BAT1','F3')=YES;
HOR_SET('BAT1','F4')=YES;
HOR_SET('BAT1','H4')=YES;
HOR_SET('BAT2','H1')=YES;
HOR_SET('BAT2','H2')=YES;
HOR_SET('BAT2','H3')=YES;
HOR_SET('BAT2','H4')=YES;
HOR_SET('BAT2','H5')=YES;
HOR_SET('BAT2','H6')=YES;
HOR_SET('BAT3','H1')=YES;
HOR_SET('BAT3','H2')=YES;
HOR_SET('BAT3','H3')=YES;
HOR_SET('BAT3','H4')=YES;
HOR_SET('BAT3','H5')=YES;
HOR_SET('BAT3','H6')=YES;

```

SETS

```

SET_PROC(PROC) Indica se PROC tem setup;
SET_PROC(PROC)=YES;
SET_PROC('TRO')=NO;

```

SETS

PROC; CAND_ULF(BAT,PROC,OPR) Candidatos a serem a ultima OPR realizada em

```

CAND_ULF('BAT4','R1','D1')=YES;
CAND_ULF('BAT4','R26','D2')=YES;
CAND_ULF('BAT4','FI','D3')=YES;
CAND_ULF('BAT4','TRO','D4')=YES;
CAND_ULF('BAT1','FP2','F3')=YES;
CAND_ULF('BAT3','R3','H1')=YES;
CAND_ULF('BAT3','R4','H2')=YES;
CAND_ULF('BAT3','R7','H4')=YES;
CAND_ULF('BAT3','FP1','H5')=YES;
CAND_ULF('BAT3','TRS','H6')=YES;

```

SETS

```

ORD_BAT(BAT) Quando a ordem da batelada tem que ser maior que 1;
ORD_BAT(BAT)=YES;
ORD_BAT('BAT1')= NO;

```

```

ALIAS(BAT,BATL1);
ALIAS(BAT,BATL2);
ALIAS(OPR,OPRL);
ALIAS(OPR,OPR1);
ALIAS(OPR,OPR2);

```

SETS

```

NOT_HOLD(OPR) Quando OPR nao pode ser HOLD;
NOT_HOLD(OPR)=YES;
NOT_HOLD('HOLD2')=NO;
NOT_HOLD('HOLD3')=NO;
NOT_HOLD('HOLE11')=NO;
NOT_HOLD('HOLE21')=NO;

```

```

NOT_HOLD('HOLF1')=NO;
NOT_HOLD('HOLF3')=NO;
NOT_HOLD('HOLH1')=NO;
NOT_HOLD('HOLH4')=NO;
NOT_HOLD('HOLH5')=NO;

```

SETS

```

PROC OPR(PROC,OPR)      Processadores que podem efetuar as operacoes
/R1.    D1
R26.   (D2,HOLD2)
FI.    (D3,HOLD3)
TRO.   D4
R3.    (E11,HOLE11,H1,HOLH1)
R4.    (E21,HOLE21,F1,HOLF1,H2)
R7.    (F2,H3,H4,HOLH4)
FP1.   (E12,E22,H5,HOLH5)
FP2.   (F3,HOLF3)
TRS.   (E13,E23,F4,H6)/

```

```

OPR SET(PROC,OPR,OPRL) Processadores com setup de OPRL apos OPR
/R1.   D1.          D1
R26.  D2.          D2
FI.   D3.          D3
R3.   (E11,H1).    E11
FP1.  (E12,H5).    E12
TRS.  (E13,H6).    E13
R4.   H2.          E21
FP1.  (E12,H5).    E22
TRS.  (E13,H6).    E23
R4.   (E21,H2).    F1
R7.   H4.          F2
FP2.  F3.          F3
TRS.  (E13,E23).   F4
R3.   (E11,H1).    H1
R4.   (F1,H2).     H2
R7.   (F2,H4).     H3
R7.   H3.          H4
FP1.  (E12,E22,H5). H5
TRS.  (E13,F4,H6). H6/

```

```

PREC SET(BATL1,BATL2,PROC,OPR1,OPR2) Precedencia de setup
/BAT1.  BAT2.  R1.    D1.    D1
BAT2.  BAT3.  R1.    D1.    D1
BAT3.  BAT4.  R1.    D1.    D1
BAT1.  BAT2.  R26.  D2.    D2
BAT2.  BAT3.  R26.  D2.    D2
BAT3.  BAT4.  R26.  D2.    D2
BAT1.  BAT2.  FI.   D3.    D3
BAT2.  BAT3.  FI.   D3.    D3
BAT3.  BAT4.  FI.   D3.    D3
BAT1.  BAT1.  R3.   H1.    E11
BAT2.  BAT1.  R3.   H1.    E11
BAT1.  BAT2.  R3.   H1.    E11
BAT2.  BAT2.  R3.   H1.    E11
BAT1.  BAT2.  R3.   E11.   E11
BAT1.  BAT1.  FP1.  H5.    E12
BAT2.  BAT1.  FP1.  H5.    E12
BAT1.  BAT2.  FP1.  H5.    E12
BAT2.  BAT2.  FP1.  H5.    E12
BAT1.  BAT2.  FP1.  E12.   E12
BAT1.  BAT1.  TRS.  H6.    E13
BAT2.  BAT1.  TRS.  H6.    E13
BAT1.  BAT2.  TRS.  H6.    E13
BAT2.  BAT2.  TRS.  H6.    E13
BAT1.  BAT2.  TRS.  E13.   E13

```

BAT2.	BAT1.	R4.	H2.	E21
BAT1.	BAT1.	FP1.	E12.	E22
BAT2.	BAT1.	FP1.	H5.	E22
BAT1.	BAT1.	TRS.	E13.	E23
BAT2.	BAT1.	TRS.	H6.	E23
BAT1.	BAT1.	R4.	E21.	F1
BAT2.	BAT1.	R4.	H2.	F1
BAT2.	BAT1.	R7.	H4.	F2
BAT2.	BAT1.	TRS.	E13.	F4
BAT1.	BAT1.	TRS.	E23.	F4
BAT1.	BAT2.	R3.	H1.	H1
BAT1.	BAT2.	R3.	E11.	H1
BAT2.	BAT2.	R3.	E11.	H1
BAT1.	BAT3.	R3.	E11.	H1
BAT2.	BAT3.	R3.	E11.	H1
BAT2.	BAT3.	R3.	H1.	H1
BAT1.	BAT2.	R4.	H2.	H2
BAT1.	BAT3.	R4.	F1.	H2
BAT1.	BAT2.	R7.	H4.	H3
BAT1.	BAT3.	R7.	F2.	H3
BAT1.	BAT1.	R7.	H3.	H4
BAT2.	BAT2.	R7.	H3.	H4
BAT3.	BAT3.	R7.	H3.	H4
BAT1.	BAT2.	FP1.	H5.	H5
BAT1.	BAT2.	FP1.	E12.	H5
BAT2.	BAT2.	FP1.	E12.	H5
BAT1.	BAT3.	FP1.	E22.	H5
BAT2.	BAT3.	FP1.	E12.	H5
BAT1.	BAT2.	TRS.	H6.	H6
BAT1.	BAT2.	TRS.	E13.	H6
BAT2.	BAT2.	TRS.	E13.	H6
BAT1.	BAT3.	TRS.	F4.	H6/

PROC_HOLD(PROC,OPR) Processadores que tem armazenagem

/R26.	HOLD2
FI.	HOLD3
R3.	(HOLE11,HOLH1)
R4.	(HOLE21,HOLF1)
R7.	HOLH4
FP1.	HOLH5
FP2.	HOLF3/

OPR_G_PROD(OPR,ESTADOS) Operacoes que geram produtos

/D1.	ID1
(D2,HOLD2).	ID2
(D3,HOLD3).	ID3
D4.	PRO_D
(E11,HOLE11).	IE11
E12.	IE12
E13.	PRO_E
(E21,HOLE21).	IE21
E22.	IE22
E23.	PRO_E
(F1,HOLF1).	IF1
F2.	IF2
(F3,HOLF3).	IF3
F4.	PRO_F
(H1,HOLH1).	IH1
H2.	IH2
H3.	IH3
(H4,HOLH4).	IH4
(H5,HOLH5).	IH5
H6.	PRO_H/

OPR_C_PROD(OPR,ESTADOS) Operacoes que consomem produtos

```

/D2.          ID1
(D3,HOLD2).   ID2
(D4,HOLD3).   ID3
(E12,HOLE11). IE11
E13.         IE12
(E22,HOLE21). IE21
E23.         IE22
(F2,HOLF1).   IF1
F3.          (IF2,PRO__E)
(F4,HOLF3).   IF3
(H2,HOLH1).   IH1
H3.          IH1
H3.          PRO_F
H4.          (IH2,IH3)
(H5,HOLH4).   IH4
(H6,HOLH5).   IH5/

```

----- Estocagem limitada -----

```
EST_LIM(ESTADOS);
```

```
EST_LIM(ESTADOS)=NO;
EST_LIM('PRO_D')=YES;
EST_LIM('PRO_E')=YES;
EST_LIM('PRO_F')=YES;
EST_LIM('PRO_H')=YES;
```

SETS

```
EST_NULO(ESTADOS);
```

```
EST_NULO(ESTADOS)=YES;
EST_NULO('PRO_D')=NO;
EST_NULO('PRO_E')=NO;
EST_NULO('PRO_F')=NO;
EST_NULO('PRO_H')=NO;
```

SETS

HOLD_OPR(OPR,OPRL) Estados estaveis que permitem HOLD

```

/D2.          HOLD2
D3.          HOLD3
E11.         HOLE11
E21.         HOLE21
F1.          HOLF1
F3.          HOLF3
H1.          HOLH1
H4.          HOLH4
H5.          HOLH5/;

```

SETS

PROD_OPR(PRD,OPR) Operacoes que pertencem a um produto

```

/D.          (D1,D2,HOLD2,D3,HOLD3,D4)
E1.         (E11,HOLE11,E12,E13)
E2.         (E21,HOLE21,E22,E23)
F.          (F1,HOLF1,F2,F3,HOLF3,F4)
H.          (H1,HOLH1,H2,H3,H4,HOLH4,H5,HOLH5,H6)/;

```

SETS

PROD_BAT(PRD,BAT) Indica se PRD ocorre em BAT

```

/D.          (BAT1,BAT2,BAT3,BAT4)
E1.         (BAT1,BAT2)
E2.         BAT1
F.          BAT1
H.          (BAT1,BAT2,BAT3)/;

```

PARAMETERS

```
PCOPR_PROD(OPR,ESTADOS) Porcentagem de ESTADOS gerado por OPR
/(D1.          ID1) 1
```

```

(D2.      ID2)  1
(HOLD2.   ID2)  1
(D3.      ID3)  1
(HOLD3.   ID3)  1
(D4.      PRO_D) 1
(E11.     IE11) 1
(HOLE11.  IE11) 1
(E12.     IE12) 1
(E13.     PRO_E) 1
(E21.     IE21) 1
(HOLE21.  IE21) 1
(E22.     IE22) 1
(E23.     PRO_E) 1
(F1.      IF1)  1
(HOLF1.   IF1)  1
(F2.      IF2)  1
(F3.      IF3)  1
(HOLF3.   IF3)  1
(F4.      PRO_F) 1
(H1.      IH1)  1
(HOLH1.   IH1)  1
(H2.      IH2)  1
(H3.      IH3)  1
(H4.      IH4)  1
(HOLH4.   IH4)  1
(H5.      IH5)  1
(HOLH5.   IH5)  1
(H6.      PRO_H) 1/;

```

PARAMETERS

PCPROD_OPR(OPR,ESTADOS) Porcentagem de ESTADOS na alimentacao de OPR

```

/(D2.     ID1)  1
(HOLD2.   ID2)  1
(D3.      ID2)  1
(HOLD3.   ID3)  1
(D4.      ID3)  1
(HOLE11.  IE11) 1
(E12.     IE11) 1
(E13.     IE12) 1
(HOLE21.  IE21) 1
(E22.     IE21) 1
(E23.     IE22) 1
(HOLF1.   IF1)  1
(F2.      IF1)  1
(F3.      IF2)  0.8333333333333333
(F3.      PRO_E) 0.1666666666666667
(HOLF3.   IF3)  1
(F4.      IF3)  1
(HOLH1.   IH1)  1
(H2.      IH1)  1
(H3.      IH1)  0.2
(H3.      PRO_F) 0.8
(H4.      IH3)  0.8333333333333333
(H4.      IH2)  0.1666666666666667
(HOLH4.   IH4)  1
(H5.      IH4)  1
(HOLH5.   IH5)  1
(H6.      IH5)  1/;

```

PARAMETERS

TP(OPR) Tempos de processamento de operacoes gerando produtos

```

/D1      1
D2       4
D3       2
D4       3

```

```

HOLD2      1
HOLD3      1
(E11,E21)  2
(E12,E22)  1
(E13,E23)  3
HOLE11     1
HOLE21     1
F1         3
F2         1
F3         1
F4         5
HOLF1      1
HOLF3      1
H1         2
H2         1
H3         1
H4         1
H5         2
H6         7
HOLH1      1
HOLH4      1
HOLH5      1/;

```

PARAMETERS

```

T SET(OPR1,OPR2) Tempo de setup de OPRL apos executar OPR
/ (D1. D1) 6
 (D2. D2) 6
 (D3. D3) 9
 (E11. E11) 3
 (E11. H1) 12
 (H1. H1) 3
 (H1. E11) 15
 (E21. F1) 5
 (F1. H2) 5
 (H2. E21) 3
 (H2. F1) 6
 (H2. H2) 0
 (F2. H3) 3
 (H4. F2) 6
 (H4. H3) 3
 (H3. H4) 0
 (E12. E12) 6
 (E12. E22) 6
 (E12. H5) 9
 (E22. H5) 9
 (H5. H5) 0
 (H5. E12) 8
 (H5. E22) 8
 (F3. F3) 3
 (E13. E13) 2
 (E13. E23) 2
 (E13. F4) 4
 (E13. H6) 4
 (E23. F4) 4
 (F4. H6) 5
 (H6. H6) 3
 (H6. E13) 6
 (H6. E23) 6
 (H6. F4) 6/;

```

PARAMETERS COST_EST(ESTADOS) Custo de estocagem por Kg
/ (PRO_D,PRO_E,PRO_F,PRO_H) 0.000166/;

PARAMETERS COST_PUL(ESTADOS) Custo do estoque pulmao por Kg
/ (PRO_D,PRO_E,PRO_F,PRO_H) 1.5/;

```

PARAMETERS COST_BAT(OPR) Custo por batelada
/(D1, D2, D3, D4) 50
(E11, E12, E13) 83.33
(E21, E22, E23) 83.33
(F1, F2, F3, F4) 75
(H1, H3, H4, H5, H6) 80/;

```

```

PARAMETERS COST_SET(OPR1, OPR2) Custo de setup de OPR1 apos executar OPR
/(D1. D1) 500
(D2. D2) 600
(D3. D3) 500
(E11. E11) 200
(E11. H1) 500
(H1. H1) 100
(H1. E11) 550
(E12. E12) 300
(E12. E22) 300
(E12. H5) 200
(E22. H5) 200
(H5. H5) 0
(H5. E12) 400
(H5. E22) 400
(E13. E13) 100
(E13. E23) 100
(E13. F4) 200
(E23. F4) 200
(E13. H6) 200
(E23. H6) 200
(F4. E13) 200
(F4. E23) 200
(F4. F4) 40
(F4. H6) 200
(H6. E13) 200
(H6. E23) 200
(H6. F4) 200
(H6. H6) 100
(F3. F3) 250
(E21. E21) 50
(E21. F1) 120
(E21. H2) 300
(F1. E21) 100
(F1. F1) 200
(F1. H2) 300
(H2. E21) 120
(H2. F1) 400
(H2. H2) 0
(F2. F2) 100
(F2. H3) 150
(H4. F2) 200
(H4. H3) 100
(H3. H4) 0 /;

```

SETS

```

OPR_C_EL(OPR) Operacoes que utilizam eletricidade;
OPR_C_EL(OPR)=YES;
OPR_C_EL('HOLD2')=NO;
OPR_C_EL('HOLD3')=NO;
OPR_C_EL('HOLE11')=NO;
OPR_C_EL('HOLE21')=NO;
OPR_C_EL('HOLF1')=NO;
OPR_C_EL('HOLF3')=NO;
OPR_C_EL('HOLH1')=NO;
OPR_C_EL('H2')=NO;
OPR_C_EL('HOLH4')=NO;

```

OPR_C_EL('HOLH5')=NO;

TABLE CREL(OPR,TETA) Consumo de eletricidade

	T0	T1	T2	T3	T4	T5	T6
D1	5	0	0	0	0	0	0
D2	5	5	5	5	0	0	0
D3	15	15	0	0	0	0	0
D4	30	30	30	0	0	0	0
E11	10	10	0	0	0	0	0
E12	10	0	0	0	0	0	0
E13	20	20	20	0	0	0	0
E21	5	5	0	0	0	0	0
E22	5	0	0	0	0	0	0
E23	20	20	20	0	0	0	0
F1	5	5	5	0	0	0	0
F2	10	0	0	0	0	0	0
F3	12	0	0	0	0	0	0
F4	42	42	42	42	42	0	0
H1	5	5	0	0	0	0	0
H2	0	0	0	0	0	0	0
H3	10	0	0	0	0	0	0
H4	10	0	0	0	0	0	0
H5	10	10	0	0	0	0	0
H6	20	20	20	20	20	20	20;

SETS

OPR_C_VAP(OPR) Operacoes que utilizam vapor;
 OPR_C_VAP(OPR)=YES;
 OPR_C_VAP('HOLD2')=NO;
 OPR_C_VAP('D3')=NO;
 OPR_C_VAP('HOLD3')=NO;
 OPR_C_VAP('D4')=NO;
 OPR_C_VAP('HOLE11')=NO;
 OPR_C_VAP('E12')=NO;
 OPR_C_VAP('E13')=NO;
 OPR_C_VAP('HOLE21')=NO;
 OPR_C_VAP('E22')=NO;
 OPR_C_VAP('E23')=NO;
 OPR_C_VAP('HOLF1')=NO;
 OPR_C_VAP('F3')=NO;
 OPR_C_VAP('HOLF3')=NO;
 OPR_C_VAP('F4')=NO;
 OPR_C_VAP('HOLH1')=NO;
 OPR_C_VAP('H2')=NO;
 OPR_C_VAP('HOLH4')=NO;
 OPR_C_VAP('H5')=NO;
 OPR_C_VAP('HOLH5')=NO;
 OPR_C_VAP('H6')=NO;

TABLE CRVAP(OPR,TETA)

	T0	T1	T2	T3	T4	T5	T6
D1	39.5	0	0	0	0	0	0
D2	18.4	0	0	0	0	0	0
E11	10	10	0	0	0	0	0
E21	10	10	0	0	0	0	0
F1	10	10	0	0	0	0	0
F2	15	0	0	0	0	0	0
H1	20	0	0	0	0	0	0
H3	25	0	0	0	0	0	0
H4	25	0	0	0	0	0	0;

TABLE ENTREGA(ESTADOS,DIA) Entrega requerida de ESTADOS no DIA

	SEM_0	SEG_1	TER_1	QUA_1	QUI_1	SEX_1	SAB_1
PRO_D	0	50	100	50	60	80	0

PRO_E	0	20	30	0	0	280	0
PRO_F	0	100	50	20	100	50	0
PRO_H	0	100	150	50	50	70	0

+	SEG_2	QUA_2	QUI_2	SEX_2	SAB_2		
PRO_D	40	200	0	120	0		
PRO_E	50	100	130	40	0		
PRO_F	50	30	0	20	0		
PRO_H	130	50	30	50	0		

+	SEG_3	TER_3	QUA_3	QUI_3	SEX_3	SAB_3	
PRO_D	110	450	10	10	60	0	
PRO_E	10	20	0	10	0	0	
PRO_F	10	0	50	60	10	0	
PRO_H	30	50	50	30	10	0 ;	

PARAMETERS MAT(OPR) Tamanho das bateladas
 / (D1,D2,D3,D4,HOLD2,HOLD3) 400
 (E11,E12,E13,HOLE11) 300
 (E21,E22,E23,HOLE21) 240
 (F1,F2,HOLF1) 500
 (F3,F4,HOLF3) 600
 (H1,HOLH1) 100
 H2 50
 H3 250
 (H4,H5,H6,HOLH4,HOLH5) 300/;

Janelas das Tarefas e das Operações de Preparação (JAN_TUDO.INC)

SETS

JANELAS(BAT,PROC,OPR,SLOT) Indica se pode haver OPR de BAT em SLOT
 JAN_SET(BAT,PROC,OPR,OPRL,SLOT) Indica se pode haver setup de BAT;
 JANELAS(BAT,PROC,OPR,SLOT) = NO;
 JANELAS(BAT,'R1','D1',SLOT) = YES;
 JANELAS(BAT,'R26','D2',SLOT) = YES;
 JANELAS(BAT,'F1','D3',SLOT) = YES;
 JANELAS(BAT,'TRO','D4',SLOT) = YES;
 JANELAS(BAT,'R26','HOLD2',SLOT) = YES;
 JANELAS(BAT,'F1','HOLD3',SLOT) = YES;
 JANELAS('BAT1','R3','E11',SLOT) = YES;
 JANELAS('BAT1','FP1','E12',SLOT) = YES;
 JANELAS('BAT1','TRS','E13',SLOT) = YES;
 JANELAS('BAT1','R3','HOLE11',SLOT) = YES;
 JANELAS('BAT2','R3','E11',SLOT) = YES;
 JANELAS('BAT2','FP1','E12',SLOT) = YES;
 JANELAS('BAT2','TRS','E13',SLOT) = YES;
 JANELAS('BAT2','R3','HOLE11',SLOT) = YES;
 JANELAS('BAT1','R4','E21',SLOT) = YES;
 JANELAS('BAT1','FP1','E22',SLOT) = YES;
 JANELAS('BAT1','TRS','E23',SLOT) = YES;
 JANELAS('BAT1','R4','HOLE21',SLOT) = YES;
 JANELAS('BAT1','R4','F1',SLOT) = YES;
 JANELAS('BAT1','R7','F2',SLOT) = YES;
 JANELAS('BAT1','FP2','F3',SLOT) = YES;
 JANELAS('BAT1','TRS','F4',SLOT) = YES;
 JANELAS('BAT1','R4','HOLF1',SLOT) = YES;
 JANELAS('BAT1','FP2','HOLF3',SLOT) = YES;
 JANELAS('BAT1','R3','H1',SLOT) = YES;
 JANELAS('BAT1','R4','H2',SLOT) = YES;
 JANELAS('BAT1','R7','H3',SLOT) = YES;
 JANELAS('BAT1','R7','H4',SLOT) = YES;

JANELAS ('BAT1', 'FP1', 'H5', SLOT) = YES;
 JANELAS ('BAT1', 'TRS', 'H6', SLOT) = YES;
 JANELAS ('BAT1', 'R3', 'HOLH1', SLOT) = YES;
 JANELAS ('BAT1', 'R7', 'HOLH4', SLOT) = YES;
 JANELAS ('BAT1', 'FP1', 'HOLH5', SLOT) = YES;
 JANELAS ('BAT2', 'R3', 'H1', SLOT) = YES;
 JANELAS ('BAT2', 'R4', 'H2', SLOT) = YES;
 JANELAS ('BAT2', 'R7', 'H3', SLOT) = YES;
 JANELAS ('BAT2', 'R7', 'H4', SLOT) = YES;
 JANELAS ('BAT2', 'FP1', 'H5', SLOT) = YES;
 JANELAS ('BAT2', 'TRS', 'H6', SLOT) = YES;
 JANELAS ('BAT2', 'R3', 'HOLH1', SLOT) = YES;
 JANELAS ('BAT2', 'R7', 'HOLH4', SLOT) = YES;
 JANELAS ('BAT2', 'FP1', 'HOLH5', SLOT) = YES;
 JANELAS ('BAT3', 'R3', 'H1', SLOT) = YES;
 JANELAS ('BAT3', 'R4', 'H2', SLOT) = YES;
 JANELAS ('BAT3', 'R7', 'H3', SLOT) = YES;
 JANELAS ('BAT3', 'R7', 'H4', SLOT) = YES;
 JANELAS ('BAT3', 'FP1', 'H5', SLOT) = YES;
 JANELAS ('BAT3', 'TRS', 'H6', SLOT) = YES;
 JANELAS ('BAT3', 'R3', 'HOLH1', SLOT) = YES;
 JANELAS ('BAT3', 'R7', 'HOLH4', SLOT) = YES;
 JANELAS ('BAT3', 'FP1', 'HOLH5', SLOT) = YES;
 JAN_SET (BAT, PROC, OPR1, OPR2, SLOT) = NO;
 JAN_SET (BAT, 'R1', 'D1', 'D1', SLOT) = YES;
 JAN_SET (BAT, 'R26', 'D2', 'D2', SLOT) = YES;
 JAN_SET (BAT, 'FI', 'D3', 'D3', SLOT) = YES;
 JAN_SET ('BAT1', 'R3', 'H1', 'E11', SLOT) = YES;
 JAN_SET ('BAT2', 'R3', 'E11', 'E11', SLOT) = YES;
 JAN_SET ('BAT2', 'R3', 'H1', 'E11', SLOT) = YES;
 JAN_SET ('BAT1', 'FP1', 'H5', 'E12', SLOT) = YES;
 JAN_SET ('BAT2', 'FP1', 'E12', 'E12', SLOT) = YES;
 JAN_SET ('BAT2', 'FP1', 'H5', 'E12', SLOT) = YES;
 JAN_SET ('BAT1', 'TRS', 'H6', 'E13', SLOT) = YES;
 JAN_SET ('BAT2', 'TRS', 'E13', 'E13', SLOT) = YES;
 JAN_SET ('BAT2', 'TRS', 'H6', 'E13', SLOT) = YES;
 JAN_SET ('BAT1', 'R4', 'H2', 'E21', SLOT) = YES;
 JAN_SET ('BAT1', 'FP1', 'E12', 'E22', SLOT) = YES;
 JAN_SET ('BAT1', 'FP1', 'H5', 'E22', SLOT) = YES;
 JAN_SET ('BAT1', 'TRS', 'E13', 'E23', SLOT) = YES;
 JAN_SET ('BAT1', 'TRS', 'H6', 'E23', SLOT) = YES;
 JAN_SET ('BAT1', 'R4', 'E21', 'F1', SLOT) = YES;
 JAN_SET ('BAT1', 'R4', 'H2', 'F1', SLOT) = YES;
 JAN_SET ('BAT1', 'R7', 'H4', 'F2', SLOT) = YES;
 JAN_SET ('BAT1', 'FP2', 'F3', 'F3', SLOT) = YES;
 JAN_SET ('BAT1', 'TRS', 'E13', 'F4', SLOT) = YES;
 JAN_SET ('BAT1', 'TRS', 'E23', 'F4', SLOT) = YES;
 JAN_SET ('BAT1', 'R7', 'H3', 'H4', SLOT) = YES;
 JAN_SET ('BAT2', 'R3', 'H1', 'H1', SLOT) = YES;
 JAN_SET ('BAT2', 'R3', 'E11', 'H1', SLOT) = YES;
 JAN_SET ('BAT2', 'R4', 'H2', 'H2', SLOT) = YES;
 JAN_SET ('BAT2', 'R7', 'H4', 'H3', SLOT) = YES;
 JAN_SET ('BAT2', 'R7', 'H3', 'H4', SLOT) = YES;
 JAN_SET ('BAT2', 'FP1', 'H5', 'H5', SLOT) = YES;
 JAN_SET ('BAT2', 'FP1', 'E12', 'H5', SLOT) = YES;
 JAN_SET ('BAT2', 'TRS', 'E13', 'H6', SLOT) = YES;
 JAN_SET ('BAT2', 'TRS', 'H6', 'H6', SLOT) = YES;
 JAN_SET ('BAT3', 'R3', 'H1', 'H1', SLOT) = YES;
 JAN_SET ('BAT3', 'R3', 'E11', 'H1', SLOT) = YES;
 JAN_SET ('BAT3', 'R4', 'F1', 'H2', SLOT) = YES;
 JAN_SET ('BAT3', 'R7', 'F2', 'H3', SLOT) = YES;
 JAN_SET ('BAT3', 'R7', 'H3', 'H4', SLOT) = YES;
 JAN_SET ('BAT3', 'FP1', 'E12', 'H5', SLOT) = YES;
 JAN_SET ('BAT3', 'FP1', 'E22', 'H5', SLOT) = YES;
 JAN_SET ('BAT3', 'TRS', 'F4', 'H6', SLOT) = YES;

* DADOS INICIAIS

```
LOOP(SLOT,  
  W_SETUP.FX('BAT1','R1','D1','D1',SLOT)=0;  
  JAN_SET('BAT1','R1','D1','D1',SLOT) = NO;  
  W_SETUP.FX('BAT1','R26','D2','D2',SLOT)=0;  
  JAN_SET('BAT1','R26','D2','D2',SLOT) = NO;  
  W_SETUP.FX('BAT1','F1','D3','D3',SLOT)=0;  
  JAN_SET('BAT1','F1','D3','D3',SLOT) = NO;  
  W_SETUP.FX('BAT1','R3','H1','H1',SLOT)=0;  
  JAN_SET('BAT1','R3','H1','H1',SLOT) = NO;  
  W_SETUP.FX('BAT1','R3','E11','H1',SLOT)=0;  
  JAN_SET('BAT1','R3','E11','H1',SLOT) = NO;  
  W_SETUP.FX('BAT1','R4','F1','H2',SLOT)=0;  
  JAN_SET('BAT1','R4','F1','H2',SLOT) = NO;  
  W_SETUP.FX('BAT1','R7','F2','H3',SLOT)=0;  
  JAN_SET('BAT1','R7','F2','H3',SLOT) = NO;  
  W_SETUP.FX('BAT1','R7','H4','H3',SLOT)=0;  
  JAN_SET('BAT1','R7','H4','H3',SLOT) = NO;  
);
```

```
W.FX('BAT1','R1','D1','SL1') = 1;  
W.FX('BAT1','R26','D2','SL1') = 1;  
W.FX('BAT1','F1','D3','SL1') = 1;  
W.FX('BAT1','R26','HOLD2',SLOT) = 0;  
JANELAS('BAT1','R26','HOLD2',SLOT) = NO;  
W.FX('BAT1','F1','HOLD3','SL5') = 1;  
W.FX('BAT1','R3','H1','SL1') = 1;  
W.FX('BAT1','R4','H2','SL1') = 1;  
W.FX('BAT1','R7','H3','SL1') = 1;  
W.FX('BAT1','R7','H4','SL2') = 1;  
W.FX('BAT1','R3','HOLH1',SLOT) = 0;  
JANELAS('BAT1','R3','HOLH1',SLOT) = NO;  
W.FX('BAT1','R7','HOLH4','SL5') = 1;  
W_SETUP.FX('BAT1','TRS','H6','H6','SL5') = 1;  
W_SETUP.FX('BAT1','R7','H3','H4','SL2') = 1;
```

* RESTRICOES DO ESTOQUE INTERMEDIARIO E FINAL

```
STOCK.UP('PRO_D',SLOT) = 38200;  
STOCK.UP('PRO_E',SLOT) = 28000;  
STOCK.UP('PRO_F',SLOT) = 18500;  
STOCK.UP('PRO_H',SLOT) = 23000;  
STOCK.FX('PRO_D','SL5') = 100;  
STOCK.FX('PRO_E','SL5') = 370;  
STOCK.FX('PRO_F','SL5') = 670;  
STOCK.FX('PRO_H','SL5') = 500;  
STOCK.LO('PRO_D','SL97') = 300;  
STOCK.LO('PRO_E','SL97') = 100;  
STOCK.LO('PRO_F','SL97') = 300;  
STOCK.LO('PRO_H','SL97') = 500;  
V.FX(ESTADOS,'SL6') = 0;
```

* RESTRICOES NO CONSUMO DE RECURSOS

```
QEL.UP(SLOT) = 50;  
QVAP.UP(SLOT) = 40;  
LOOP(SLOT,  
  IF(ORD(SLOT) LT 6,  
    QEL.FX(SLOT) = 0;  
    QVAP.FX(SLOT) = 0;  
  );
```

);

* Produto D

* A BATELADA 1 (PARCIALMENTE COMPLETA) DEVE FICAR PRONTA ATE 25 JUL (SL17)
 LOOP(SLOT,

```

    IF(ORD(SLOT) GT 1,
      W.FX('BAT1','R1','D1',SLOT)=0;
      JANELAS('BAT1','R1','D1',SLOT) = NO;
      W.FX('BAT1','R26','D2',SLOT)=0;
      JANELAS('BAT1','R26','D2',SLOT) = NO;
      W.FX('BAT1','FI','D3',SLOT)=0;
      JANELAS('BAT1','FI','D3',SLOT) = NO;
    );
    IF(ORD(SLOT) LT 5 OR ORD(SLOT) GE 17 - TP('D4'),
      W.FX('BAT1','FI','HOLD3',SLOT) = 0;
      JANELAS('BAT1','FI','HOLD3',SLOT) = NO;
    );
    IF(ORD(SLOT) LT 6 OR ORD(SLOT) GT 17 - TP('D4'),
      W.FX('BAT1','TRO','D4',SLOT) = 0;
      JANELAS('BAT1','TRO','D4',SLOT) = NO;
    );
  );

```

* A BATELADA 2 DEVE FICAR PRONTA ATE 02 AGO (SL49)
 * NAO PODEM HAVER BATELADAS PARCIALMENTE COMPLETAS EM 01 AGO (SL43)
 * NAO PODE HAVER NENHUMA BATELADA EM PROCESSAMENTO NO FIM DE SEMANA (SL39)

```

LOOP(SLOT,
  IF(ORD(SLOT) LT 6+T_SET('D1','D1') OR ORD(SLOT) GT 39-TP('D1')-TP('D2'),
    W.FX('BAT2','R1','D1',SLOT)=0;
    JANELAS('BAT2','R1','D1',SLOT) = NO;
  );
  IF(ORD(SLOT) LT 6 OR ORD(SLOT) GT 39-TP('D1')-TP('D2')-T_SET('D1','D1'),
    W_SETUP.FX('BAT2','R1','D1','D1',SLOT)=0;
    JAN_SET('BAT2','R1','D1','D1',SLOT) = NO;
  );
  IF(ORD(SLOT) LT 6+TP('D1')+T_SET('D1','D1') OR ORD(SLOT) GT 39-TP('D2'),
    W.FX('BAT2','R26','D2',SLOT)=0;
    JANELAS('BAT2','R26','D2',SLOT) = NO;
  );
  IF(ORD(SLOT) LT 6+TP('D1')+TP('D2')+T_SET('D1','D1') OR
    ORD(SLOT) GE 43-TP('D3'),
    W.FX('BAT2','R26','HOLD2',SLOT)=0;
    JANELAS('BAT2','R26','HOLD2',SLOT) = NO;
  );
  IF(ORD(SLOT) LT 6 OR ORD(SLOT) GT 39-TP('D2')-T_SET('D2','D2'),
    W_SETUP.FX('BAT2','R26','D2','D2',SLOT)=0;
    JAN_SET('BAT2','R26','D2','D2',SLOT) = NO;
  );
  IF(ORD(SLOT) LT 6+T_SET('D1','D1')+TP('D1')+TP('D2') OR
    ORD(SLOT) GT 43-TP('D3') OR
    (ORD(SLOT) GT 39-TP('D3') AND ORD(SLOT) LE 38),
    W.FX('BAT2','FI','D3',SLOT)=0;
    JANELAS('BAT2','FI','D3',SLOT) = NO;
  );
  IF(ORD(SLOT) LT 6+TP('D1')+TP('D2')+TP('D3')+T_SET('D1','D1') OR
    ORD(SLOT) GE 48-TP('D4'),
    W.FX('BAT2','FI','HOLD3',SLOT)=0;
    JANELAS('BAT2','FI','HOLD3',SLOT) = NO;
  );
  IF(ORD(SLOT) LT 6 OR ORD(SLOT) GT 43-TP('D3')-T_SET('D3','D3'),
    W_SETUP.FX('BAT2','FI','D3','D3',SLOT)=0;
    JAN_SET('BAT2','FI','D3','D3',SLOT) = NO;
  );
  IF(ORD(SLOT) LT 6+T_SET('D1','D1')+TP('D1')+TP('D2')+TP('D3') OR
    ORD(SLOT) GT 48-TP('D4') OR
    (ORD(SLOT) GT 39-TP('D4') AND ORD(SLOT) LE 38) OR

```

```

(ORD(SLOT) GT 43-TP('D4') AND ORD(SLOT) LE 42),
  W.FX('BAT2','TRO','D4',SLOT)=0;
  JANELAS('BAT2','TRO','D4',SLOT) = NO;
);
);

* A BATELADA 3 DEVE FICAR PRONTA ATE 08 AGO (SL75)
* NAO PODEM HAVER BATELADAS PARCIALMENTE COMPLETAS EM 01 AGO (SL43)
* NAO PODE HAVER NENHUMA BATELADA EM PROCESSAMENTO NOS FINS DE SEMANA

LOOP(SLOT,
  IF(ORD(SLOT) LT 6+2*T_SET('D1','D1')+TP('D1') OR
    ORD(SLOT) GT 75-TP('D1')-TP('D2')-TP('D3')-TP('D4') OR
    ORD(SLOT) LT 6+T_SET('D1','D1')+T_SET('D2','D2')+TP('D2') OR
    (ORD(SLOT) GT 39-TP('D1')-TP('D2') AND ORD(SLOT) LE 38) OR
    (ORD(SLOT) GT 43-TP('D1')-TP('D2') AND ORD(SLOT) LE 42) OR
    (ORD(SLOT) GT 64-TP('D1')-TP('D2') AND ORD(SLOT) LE 63),
    W.FX('BAT3','R1','D1',SLOT)=0;
    JANELAS('BAT3','R1','D1',SLOT) = NO;
  );
  IF(ORD(SLOT) LT 6+T_SET('D1','D1')+TP('D1') OR ORD(SLOT) GT
    75-TP('D1')-TP('D2')-TP('D3')-TP('D4')-T_SET('D1','D1'),
    W_SETUP.FX('BAT3','R1','D1','D1',SLOT)=0;
    JAN_SET('BAT3','R1','D1','D1',SLOT) = NO;
  );
  IF(ORD(SLOT) LT 6+T_SET('D1','D1')+TP('D1')+T_SET('D2','D2')+TP('D2') OR
    ORD(SLOT) GT 75-TP('D2')-TP('D3')-TP('D4') OR
    (ORD(SLOT) GT 39-TP('D2') AND ORD(SLOT) LE 38+TP('D1')) OR
    (ORD(SLOT) GT 43-TP('D2') AND ORD(SLOT) LE 42+TP('D1')) OR
    (ORD(SLOT) GT 64-TP('D2') AND ORD(SLOT) LE 63+TP('D1')),
    W.FX('BAT3','R26','D2',SLOT)=0;
    JANELAS('BAT3','R26','D2',SLOT) = NO;
  );
  IF(ORD(SLOT) LT 6+T_SET('D1','D1')+TP('D1')+T_SET('D2','D2')+2*TP('D2') OR
    ORD(SLOT) GE 75-TP('D3')-TP('D4'),
    W.FX('BAT3','R26','HOLD2',SLOT)=0;
    JANELAS('BAT3','R26','HOLD2',SLOT) = NO;
  );
  IF(ORD(SLOT) LT 6+T_SET('D1','D1')+TP('D1')+TP('D2') OR
    ORD(SLOT) GT 75-TP('D2')-TP('D3')-TP('D4')-T_SET('D2','D2'),
    W_SETUP.FX('BAT3','R26','D2','D2',SLOT)=0;
    JAN_SET('BAT3','R26','D2','D2',SLOT) = NO;
  );
  IF(ORD(SLOT) LT 6+T_SET('D1','D1')+TP('D1')+TP('D2')+
    TP('D3')+T_SET('D3','D3') OR
    ORD(SLOT) GT 75-TP('D3')-TP('D4') OR
    (ORD(SLOT) GT 39-TP('D3') AND ORD(SLOT) LE 38) OR
    (ORD(SLOT) GT 43-TP('D3') AND ORD(SLOT) LE 42) OR
    (ORD(SLOT) GT 64-TP('D3') AND ORD(SLOT) LE 63),
    W.FX('BAT3','FI','D3',SLOT)=0;
    JANELAS('BAT3','FI','D3',SLOT) = NO;
  );
  IF(ORD(SLOT) LT 6+TP('D1')+TP('D2')+2*TP('D3')+T_SET('D1','D1')+
    T_SET('D3','D3') OR ORD(SLOT) GE 75-TP('D4'),
    W.FX('BAT3','FI','HOLD3',SLOT)=0;
    JANELAS('BAT3','FI','HOLD3',SLOT) = NO;
  );
  IF(ORD(SLOT) LT 6+T_SET('D1','D1')+TP('D1')+TP('D2')+TP('D3') OR
    ORD(SLOT) GT 75-TP('D3')-TP('D4')-T_SET('D3','D3'),
    W_SETUP.FX('BAT3','FI','D3','D3',SLOT)=0;
    JAN_SET('BAT3','FI','D3','D3',SLOT) = NO;
  );
  IF(ORD(SLOT) LT 6+T_SET('D1','D1')+TP('D1')+TP('D2')+2*TP('D3')+
    T_SET('D3','D3') OR ORD(SLOT) GT 75-TP('D4') OR
    (ORD(SLOT) GT 39-TP('D4') AND ORD(SLOT) LE 38) OR

```

```

(ORD(SLOT) GT 43-TP('D4') AND ORD(SLOT) LE 42) OR
(ORD(SLOT) GT 64-TP('D4') AND ORD(SLOT) LE 63),
  W.FX('BAT3','TRO','D4',SLOT)=0;
  JANELAS('BAT3','TRO','D4',SLOT) = NO;
);
);

* A BATELADA 4 DEVE FICAR PRONTA ATE 08 AGO (SL93)
* NAO PODEM HAVER BATELADAS PARCIALMENTE COMPLETAS EM 01 AGO (SL43)
* NAO PODE HAVER NENHUMA BATELADA EM PROCESSAMENTO NOS FINS DE SEMANA

LOOP(SLOT,
  IF(ORD(SLOT) LT 6+3*T_SET('D1','D1')+2*TP('D1') OR
    ORD(SLOT) LT 6+T_SET('D1','D1')+2*T_SET('D2','D2')+2*TP('D2') OR
    ORD(SLOT) GT 93-TP('D1')-TP('D2')-TP('D3')-TP('D4') OR
    (ORD(SLOT) GT 39-TP('D1')-TP('D2') AND ORD(SLOT) LE 38) OR
    (ORD(SLOT) GT 43-TP('D1')-TP('D2') AND ORD(SLOT) LE 42) OR
    (ORD(SLOT) GT 64-TP('D1')-TP('D2') AND ORD(SLOT) LE 63),
    W.FX('BAT4','R1','D1',SLOT)=0;
    JANELAS('BAT4','R1','D1',SLOT) = NO;
  );
  IF(ORD(SLOT) LT 6+2*T_SET('D1','D1')+2*TP('D1') OR
    ORD(SLOT) LT 6+T_SET('D1','D1')+TP('D1')+T_SET('D2','D2')+TP('D2') OR
    ORD(SLOT) GT 93-TP('D1')-TP('D2')-TP('D3')-TP('D4')-T_SET('D1','D1'),
    W_SETUP.FX('BAT4','R1','D1','D1',SLOT)=0;
    JAN_SET('BAT4','R1','D1','D1',SLOT) = NO;
  );
  IF(ORD(SLOT) LT 6+T_SET('D1','D1')+TP('D1')+2*T_SET('D2','D2')+
    2*TP('D2') OR
    ORD(SLOT) GT 93-TP('D2')-TP('D3')-TP('D4') OR
    (ORD(SLOT) GT 39-TP('D2') AND ORD(SLOT) LE 38+TP('D1')) OR
    (ORD(SLOT) GT 43-TP('D2') AND ORD(SLOT) LE 42+TP('D1')) OR
    (ORD(SLOT) GT 64-TP('D2') AND ORD(SLOT) LE 63+TP('D1')),
    W.FX('BAT4','R26','D2',SLOT)=0;
    JANELAS('BAT4','R26','D2',SLOT) = NO;
  );
  IF(ORD(SLOT) LT 6+T_SET('D1','D1')+TP('D1')+2*T_SET('D2','D2')+
    3*TP('D2') OR
    ORD(SLOT) GE 93-TP('D3')-TP('D4'),
    W.FX('BAT4','R26','HOLD2',SLOT)=0;
    JANELAS('BAT4','R26','HOLD2',SLOT) = NO;
  );
  IF(ORD(SLOT) LT 6+T_SET('D1','D1')+TP('D1')+2*TP('D2')+T_SET('D2','D2') OR
    ORD(SLOT) GT 93-TP('D2')-TP('D3')-TP('D4')-T_SET('D2','D2'),
    W_SETUP.FX('BAT4','R26','D2','D2',SLOT)=0;
    JAN_SET('BAT4','R26','D2','D2',SLOT) = NO;
  );
  IF(ORD(SLOT) LT 6+T_SET('D1','D1')+TP('D1')+TP('D2')+
    2*TP('D3')+2*T_SET('D3','D3') OR ORD(SLOT) GT 93-TP('D3')-TP('D4') OR
    (ORD(SLOT) GT 39-TP('D3') AND ORD(SLOT) LE 38) OR
    (ORD(SLOT) GT 43-TP('D3') AND ORD(SLOT) LE 42) OR
    (ORD(SLOT) GT 64-TP('D3') AND ORD(SLOT) LE 63),
    W.FX('BAT4','FI','D3',SLOT)=0;
    JANELAS('BAT4','FI','D3',SLOT) = NO;
  );
  IF(ORD(SLOT) LT 6+TP('D1')+TP('D2')+3*TP('D3')+T_SET('D1','D1')+
    2*T_SET('D3','D3') OR ORD(SLOT) GE 93-TP('D4'),
    W.FX('BAT4','FI','HOLD3',SLOT)=0;
    JANELAS('BAT4','FI','HOLD3',SLOT) = NO;
  );
  IF(ORD(SLOT) LT 6+T_SET('D1','D1')+TP('D1')+TP('D2')+2*TP('D3')+
    T_SET('D3','D3') OR ORD(SLOT) GT 93-TP('D3')-TP('D4')-T_SET('D3','D3'),
    W_SETUP.FX('BAT4','FI','D3','D3',SLOT)=0;
    JAN_SET('BAT4','FI','D3','D3',SLOT) = NO;
  );
);

```

```

IF(ORD(SLOT) LT 6+T_SET('D1','D1')+TP('D1')+TP('D2')+3*TP('D3')+
  2*T_SET('D3','D3') OR ORD(SLOT) GT 93-TP('D4') OR
  (ORD(SLOT) GT 39-TP('D4') AND ORD(SLOT) LE 38) OR
  (ORD(SLOT) GT 43-TP('D4') AND ORD(SLOT) LE 42) OR
  (ORD(SLOT) GT 64-TP('D4') AND ORD(SLOT) LE 63),
  W.FX('BAT4','TRO','D4',SLOT)=0;
  JANELAS('BAT4','TRO','D4',SLOT) = NO;
);
);
*****
*                               Produto E - variante 1
LOOP(SLOT,
  IF(ORD(SLOT) LT 6+T_SET('H1','E11') OR
    (ORD(SLOT) GT 39-TP('E11') AND ORD(SLOT) LE 38) OR
    (ORD(SLOT) GT 43-TP('E11') AND ORD(SLOT) LE 42) OR
    ORD(SLOT) GT 43-TP('E11')-TP('E12')-TP('E13'),
    W.FX('BAT1','R3','E11',SLOT) = 0;
    JANELAS('BAT1','R3','E11',SLOT) = NO;
  );
  IF(ORD(SLOT) LT 6+T_SET('H1','E11')+TP('E11') OR
    ORD(SLOT) GE 43-TP('E12')-TP('E13'),
    W.FX('BAT1','R3','HOLE11',SLOT) = 0;
    JANELAS('BAT1','R3','HOLE11',SLOT) = NO;
  );
  IF(ORD(SLOT) LT 6+T_SET('H1','E11')+TP('E11') OR
    (ORD(SLOT) GT 39-TP('E12')-TP('E13') AND ORD(SLOT) LE 38) OR
    ORD(SLOT) GT 43-TP('E12')-TP('E13'),
    W.FX('BAT1','FP1','E12',SLOT) = 0;
    JANELAS('BAT1','FP1','E12',SLOT) = NO;
  );
  IF(ORD(SLOT) LT 6+T_SET('H1','E11')+TP('E11')+TP('E12') OR
    (ORD(SLOT) GT 39-TP('E13') AND ORD(SLOT) LE 38+TP('E12')) OR
    ORD(SLOT) GT 43-TP('E13'),
    W.FX('BAT1','TRS','E13',SLOT) = 0;
    JANELAS('BAT1','TRS','E13',SLOT) = NO;
  );
  IF(ORD(SLOT) LT 64 OR ORD(SLOT) GT 75-TP('E11')-TP('E12')-TP('E13'),
    W.FX('BAT2','R3','E11',SLOT) = 0;
    JANELAS('BAT2','R3','E11',SLOT) = NO;
  );
  IF(ORD(SLOT) LT 64+TP('E11') OR ORD(SLOT) GE 75-TP('E12')-TP('E13'),
    W.FX('BAT2','R3','HOLE11',SLOT) = 0;
    JANELAS('BAT2','R3','HOLE11',SLOT) = NO;
  );
  IF(ORD(SLOT) LT 64+TP('E11') OR ORD(SLOT) GT 75-TP('E12')-TP('E13'),
    W.FX('BAT2','FP1','E12',SLOT) = 0;
    JANELAS('BAT2','FP1','E12',SLOT) = NO;
  );
  IF(ORD(SLOT) LT 64+TP('E11')+TP('E12') OR ORD(SLOT) GT 75-TP('E13'),
    W.FX('BAT2','TRS','E13',SLOT) = 0;
    JANELAS('BAT2','TRS','E13',SLOT) = NO;
  );
  IF(ORD(SLOT) LT 6 OR
    ORD(SLOT) GT 43-TP('E11')-TP('E12')-TP('E13')-T_SET('H1','E11'),
    W_SETUP.FX('BAT1','R3','H1','E11',SLOT) = 0;
    JAN_SET('BAT1','R3','H1','E11',SLOT) = NO;
  );
  IF(ORD(SLOT) LT 6+TP('H5') OR
    ORD(SLOT) GT 43-TP('E12')-TP('E13')-T_SET('H5','E12'),
    W_SETUP.FX('BAT1','FP1','H5','E12',SLOT) = 0;
    JAN_SET('BAT1','FP1','H5','E12',SLOT) = NO;
  );
  IF(ORD(SLOT) LT 5+T_SET('H6','H6')+TP('H6') OR

```

```

ORD(SLOT) GT 43-TP('E13')-T_SET('H6','E13'),
W_SETUP.FX('BAT1','TRS','H6','E13',SLOT) = 0;
JAN_SET('BAT1','TRS','H6','E13',SLOT) = NO;
);
IF(ORD(SLOT) LT 6+T_SET('H1','E11')+TP('E11') OR
ORD(SLOT) GT 75-TP('E11')-TP('E12')-TP('E13')-T_SET('E11','E11'),
W_SETUP.FX('BAT2','R3','E11','E11',SLOT) = 0;
JAN_SET('BAT2','R3','E11','E11',SLOT) = NO;
);
IF(ORD(SLOT) LT 6+T_SET('H1','E11')+TP('E11')+T_SET('E11','H1')+
TP('H1') OR
ORD(SLOT) GT 75-TP('E11')-TP('E12')-TP('E13')-T_SET('H1','E11'),
W_SETUP.FX('BAT2','R3','H1','E11',SLOT) = 0;
JAN_SET('BAT2','R3','H1','E11',SLOT) = NO;
);
IF(ORD(SLOT) LT 6+TP('E11')+T_SET('H1','E11')+TP('E12') OR
ORD(SLOT) GT 75-TP('E12')-TP('E13')-T_SET('E12','E12'),
W_SETUP.FX('BAT2','FP1','E12','E12',SLOT) = 0;
JAN_SET('BAT2','FP1','E12','E12',SLOT) = NO;
);
IF(ORD(SLOT) LT 6+T_SET('H1','E11')+TP('E11')+T_SET('E11','H1')+
TP('H1')+TP('H2')+TP('H4')+TP('H5') OR
ORD(SLOT) GT 75-TP('E12')-TP('E13')-T_SET('H5','E12'),
W_SETUP.FX('BAT2','FP1','H5','E12',SLOT) = 0;
JAN_SET('BAT2','FP1','H5','E12',SLOT) = NO;
);
IF(ORD(SLOT) LT 6+T_SET('H1','E11')+TP('E11')+T_SET('E11','H1')+
TP('H1')+TP('H3')+TP('H4')+TP('H5')+TP('H6') OR
ORD(SLOT) GT 75-TP('E13')-T_SET('H6','E13'),
W_SETUP.FX('BAT2','TRS','H6','E13',SLOT) = 0;
JAN_SET('BAT2','TRS','H6','E13',SLOT) = NO;
);
IF(ORD(SLOT) LT 6+T_SET('H1','E11')+TP('E11')+TP('E12')+TP('E13') OR
ORD(SLOT) GT 75-TP('E13')-T_SET('E13','E13'),
W_SETUP.FX('BAT2','TRS','E13','E13',SLOT) = 0;
JAN_SET('BAT2','TRS','E13','E13',SLOT) = NO;
);

```

* Produto E - variante 2

```

IF(ORD(SLOT) LT 64 OR ORD(SLOT) GT 75-TP('E21')-TP('E22')-TP('E23'),
W.FX('BAT1','R4','E21',SLOT) = 0;
JANELAS('BAT1','R4','E21',SLOT) = NO;
);
IF(ORD(SLOT) LT 64+TP('E21') OR ORD(SLOT) GE 75-TP('E22')-TP('E23'),
W.FX('BAT1','R4','HOLE21',SLOT) = 0;
JANELAS('BAT1','R4','HOLE21',SLOT) = NO;
);
IF(ORD(SLOT) LT 64+TP('E21') OR ORD(SLOT) GT 75-TP('E22')-TP('E23'),
W.FX('BAT1','FP1','E22',SLOT) = 0;
JANELAS('BAT1','FP1','E22',SLOT) = NO;
);
IF(ORD(SLOT) LT 64+TP('E21')+TP('E22') OR ORD(SLOT) GT 75-TP('E23'),
W.FX('BAT1','TRS','E23',SLOT) = 0;
JANELAS('BAT1','TRS','E23',SLOT) = NO;
);
IF(ORD(SLOT) LT 6+T_SET('H2','H2')+TP('H2') OR
ORD(SLOT) GT 75-TP('E21')-TP('E22')-TP('E23')-T_SET('H2','E21'),
W_SETUP.FX('BAT1','R4','H2','E21',SLOT) = 0;
JAN_SET('BAT1','R4','H2','E21',SLOT) = NO;
);
IF(ORD(SLOT) LT 6+TP('E11')+T_SET('H1','E11')+TP('E12') OR
ORD(SLOT) GT 75-TP('E22')-TP('E23')-T_SET('E12','E22'),
W_SETUP.FX('BAT1','FP1','E12','E22',SLOT) = 0;
JAN_SET('BAT1','FP1','E12','E22',SLOT) = NO;

```

```

);
IF(ORD(SLOT) LT 6+T_SET('H1','E11')+TP('E11')+T_SET('E11','H1')+TP('H1')+
  TP('H2')+TP('H4')+TP('H5') OR
  ORD(SLOT) GT 75-TP('E22')-TP('E23')-T_SET('H5','E22'),
  W_SETUP.FX('BAT1','FP1','H5','E22',SLOT) = 0;
  JAN_SET('BAT1','FP1','H5','E22',SLOT) = NO;
);
IF(ORD(SLOT) LT 6+T_SET('H1','E11')+TP('E11')+T_SET('E11','H1')+TP('H1')+
  TP('H2')+TP('H4')+TP('H5')+TP('H6') OR
  ORD(SLOT) GT 75-TP('E23')-T_SET('H6','E23'),
  W_SETUP.FX('BAT1','TRS','H6','E23',SLOT) = 0;
  JAN_SET('BAT1','TRS','H6','E23',SLOT) = NO;
);
IF(ORD(SLOT) LT 6+T_SET('H1','E11')+TP('E11')+TP('E12')+TP('E13') OR
  ORD(SLOT) GT 75-TP('E23')-T_SET('E13','E23'),
  W_SETUP.FX('BAT1','TRS','E13','E23',SLOT) = 0;
  JAN_SET('BAT1','TRS','E13','E23',SLOT) = NO;
);

```

```

*****
*                               Produto F

```

```

IF(ORD(SLOT) LT 6+T_SET('H2','F1') OR
  (ORD(SLOT) GT 39-TP('F1') AND ORD(SLOT) LE 38) OR
  (ORD(SLOT) GT 43-TP('F1') AND ORD(SLOT) LE 42) OR
  (ORD(SLOT) GT 64-TP('F1') AND ORD(SLOT) LE 63) OR
  ORD(SLOT) GT 81-TP('F1')-TP('F2')-TP('F3')-TP('F4'),
  W.FX('BAT1','R4','F1',SLOT) = 0;
  JANELAS('BAT1','R4','F1',SLOT) = NO;
);
IF(ORD(SLOT) LT 6+T_SET('H2','F1')+TP('F1') OR
  ORD(SLOT) GE 81-TP('F2')-TP('F3')-TP('F4'),
  W.FX('BAT1','R4','HOLF1',SLOT) = 0;
  JANELAS('BAT1','R4','HOLF1',SLOT) = NO;
);
IF(ORD(SLOT) LT 64+TP('E11')+TP('E12')+TP('E13')-TP('F2') OR
  ORD(SLOT) GT 81-TP('F2')-TP('F3')-TP('F4'),
  W.FX('BAT1','R7','F2',SLOT) = 0;
  JANELAS('BAT1','R7','F2',SLOT) = NO;
);
IF(ORD(SLOT) LT 64+TP('E11')+TP('E12')+TP('E13') OR
  ORD(SLOT) GT 81-TP('F3')-TP('F4'),
  W.FX('BAT1','FP2','F3',SLOT) = 0;
  JANELAS('BAT1','FP2','F3',SLOT) = NO;
);
IF(ORD(SLOT) LT 64+TP('E11')+TP('E12')+TP('E13')+TP('F3') OR
  ORD(SLOT) GE 81-TP('F4'),
  W.FX('BAT1','FP2','HOLF3',SLOT) = 0;
  JANELAS('BAT1','FP2','HOLF3',SLOT) = NO;
);
IF(ORD(SLOT) LT 64+TP('E11')+TP('E12')+TP('E13')+T_SET('E13','F4') OR
  ORD(SLOT) GT 81-TP('F4'),
  W.FX('BAT1','TRS','F4',SLOT) = 0;
  JANELAS('BAT1','TRS','F4',SLOT) = NO;
);
IF(ORD(SLOT) LT 64+TP('E21') OR
  ORD(SLOT) GT 81-TP('F1')-TP('F2')-TP('F3')-TP('F4')-T_SET('E21','F1'),
  W_SETUP.FX('BAT1','R4','E21','F1',SLOT) = 0;
  JAN_SET('BAT1','R4','E21','F1',SLOT) = NO;
);
IF(ORD(SLOT) LT 6 OR
  ORD(SLOT) GT 81-TP('F1')-TP('F2')-TP('F3')-TP('F4')-T_SET('H2','F1'),
  W_SETUP.FX('BAT1','R4','H2','F1',SLOT) = 0;
  JAN_SET('BAT1','R4','H2','F1',SLOT) = NO;
);
IF(ORD(SLOT) LT 6 OR

```

```

ORD(SLOT) GT 81-TP('F2')-TP('F3')-TP('F4')-T_SET('H4','F2'),
W_SETUP.FX('BAT1','R7','H4','F2',SLOT) = 0;
JAN_SET('BAT1','R7','H4','F2',SLOT) = NO;
);
IF(ORD(SLOT) LT 6 OR
ORD(SLOT) GT 81-TP('F3')-TP('F4')-T_SET('F3','F3'),
W_SETUP.FX('BAT1','FP2','F3','F3',SLOT) = 0;
JAN_SET('BAT1','FP2','F3','F3',SLOT) = NO;
);
IF(ORD(SLOT) LT 64+TP('E11')+TP('E12')+TP('E13') OR
ORD(SLOT) GT 81-TP('F4')-T_SET('E13','F4'),
W_SETUP.FX('BAT1','TRS','E13','F4',SLOT) = 0;
JAN_SET('BAT1','TRS','E13','F4',SLOT) = NO;
);
IF(ORD(SLOT) LT 64+TP('E21')+TP('E22')+TP('E23') OR
ORD(SLOT) GT 81-TP('F4')-T_SET('E23','F4'),
W_SETUP.FX('BAT1','TRS','E23','F4',SLOT) = 0;
JAN_SET('BAT1','TRS','E23','F4',SLOT) = NO;
);
);
);
*****
*
*          Produto H
*
* A BATELADA 1 (PARCIALMENTE COMPLETA) DEVE FICAR PRONTA ATE 31 JUL (SL43)
LOOP(SLOT,
IF(ORD(SLOT) GT 1,
W.FX('BAT1','R3','H1',SLOT)=0;
JANELAS('BAT1','R3','H1',SLOT) = NO;
W.FX('BAT1','R4','H2',SLOT)=0;
JANELAS('BAT1','R4','H2',SLOT) = NO;
W.FX('BAT1','R7','H3',SLOT)=0;
JANELAS('BAT1','R7','H3',SLOT) = NO;
);
IF(ORD(SLOT) EQ 1 OR ORD(SLOT) GT 2,
W.FX('BAT1','R7','H4',SLOT)=0;
JANELAS('BAT1','R7','H4',SLOT) = NO;
W_SETUP.FX('BAT1','R7','H3','H4',SLOT) = 0;
JAN_SET('BAT1','R7','H3','H4',SLOT) = NO;
);
IF(ORD(SLOT) LT 5 OR ORD(SLOT) GE 43 - TP('H5') - TP('H6'),
W.FX('BAT1','R7','HOLH4',SLOT) = 0;
JANELAS('BAT1','R7','HOLH4',SLOT) = NO;
);
IF(ORD(SLOT) LT 6 OR ORD(SLOT) GT 43 - TP('H5') - TP('H6') OR
(ORD(SLOT) GT 39-TP('H5') AND ORD(SLOT) LE 38),
W.FX('BAT1','FP1','H5',SLOT) = 0;
JANELAS('BAT1','FP1','H5',SLOT) = NO;
);
IF(ORD(SLOT) LT 6 + TP('H5') OR ORD(SLOT) GE 43 - TP('H6'),
W.FX('BAT1','FP1','HOLH5',SLOT) = 0;
JANELAS('BAT1','FP1','HOLH5',SLOT) = NO;
);
IF(ORD(SLOT) LT 6 + TP('H5') OR ORD(SLOT) GT 43 - TP('H6') OR
(ORD(SLOT) GT 39 - TP('H6') AND ORD(SLOT) LE 38),
W.FX('BAT1','TRS','H6',SLOT) = 0;
JANELAS('BAT1','TRS','H6',SLOT) = NO;
);
);
);
* A BATELADA 2 DEVE FICAR PRONTA ATE 09 AGO (SL81)
* NAO PODE HAVER NENHUMA BATELADA EM PROCESSAMENTO NO FIM DE SEMANA (SL39)
* NAO PODE HAVER NENHUMA BATELADA EM PROCESSAMENTO NO FERIADO (SL43)
* NAO PODE HAVER NENHUMA BATELADA EM PROCESSAMENTO NO FIM DE SEMANA (SL64)

```

```

LOOP(SLOT,
  IF(ORD(SLOT) LT 6+T_SET('H1','H1') OR
    (ORD(SLOT) GT 39-TP('H1') AND ORD(SLOT) LE 38) OR
    (ORD(SLOT) GT 43-TP('H1') AND ORD(SLOT) LE 42) OR
    (ORD(SLOT) GT 64-TP('H1') AND ORD(SLOT) LE 63) OR
    ORD(SLOT) GT 81-TP('H1')-TP('H3')-TP('H4')-TP('H5')-TP('H6'),
    W.FX('BAT2','R3','H1',SLOT)=0;
    JANELAS('BAT2','R3','H1',SLOT) = NO;
  );
  IF(ORD(SLOT) LT 6+TP('H1')+T_SET('H1','H1') OR
    ORD(SLOT) GE 81-TP('H3')-TP('H4')-TP('H5')-TP('H6'),
    W.FX('BAT2','R3','HOLH1',SLOT)=0;
    JANELAS('BAT2','R3','HOLH1',SLOT) = NO;
  );
  IF(ORD(SLOT) LT 6 OR
    ORD(SLOT) GT 81-TP('H1')-TP('H3')-TP('H4')-TP('H5')-TP('H6')-
    T_SET('H1','H1'),
    W_SETUP.FX('BAT2','R3','H1','H1',SLOT)=0;
    JAN_SET('BAT2','R3','H1','H1',SLOT) = NO;
  );
  IF(ORD(SLOT) LT 6 OR
    ORD(SLOT) GT 81-TP('H1')-TP('H3')-TP('H4')-TP('H5')-TP('H6')-
    T_SET('E11','H1'),
    W_SETUP.FX('BAT2','R3','E11','H1',SLOT)=0;
    JAN_SET('BAT2','R3','E11','H1',SLOT) = NO;
  );
  IF(ORD(SLOT) LT 6+TP('H1')+T_SET('H1','H1')+T_SET('H4','H3') OR
    (ORD(SLOT) GT 39-TP('H2')-TP('H4') AND ORD(SLOT) LE 38) OR
    (ORD(SLOT) GT 43-TP('H2')-TP('H4') AND ORD(SLOT) LE 42) OR
    (ORD(SLOT) GT 64-TP('H2')-TP('H4') AND ORD(SLOT) LE 63) OR
    ORD(SLOT) GT 81-TP('H2')-TP('H4')-TP('H5')-TP('H6'),
    W.FX('BAT2','R4','H2',SLOT)=0;
    JANELAS('BAT2','R4','H2',SLOT) = NO;
  );
  IF(ORD(SLOT) LT 6 OR
    ORD(SLOT) GT 81-TP('H2')-TP('H4')-TP('H5')-TP('H6')-T_SET('H2','H2'),
    W_SETUP.FX('BAT2','R4','H2','H2',SLOT)=0;
    JAN_SET('BAT2','R4','H2','H2',SLOT) = NO;
  );
  IF(ORD(SLOT) LT 6+TP('H1')+T_SET('H1','H1')+T_SET('H4','H3') OR
    (ORD(SLOT) GT 39-TP('H3')-TP('H4') AND ORD(SLOT) LE 38) OR
    (ORD(SLOT) GT 43-TP('H3')-TP('H4') AND ORD(SLOT) LE 42) OR
    (ORD(SLOT) GT 64-TP('H3')-TP('H4') AND ORD(SLOT) LE 63) OR
    ORD(SLOT) GT 81-TP('H3')-TP('H4')-TP('H5')-TP('H6'),
    W.FX('BAT2','R7','H3',SLOT)=0;
    JANELAS('BAT2','R7','H3',SLOT) = NO;
  );
  IF(ORD(SLOT) LT 6 OR
    ORD(SLOT) GT 81-TP('H3')-TP('H4')-TP('H5')-TP('H6')-T_SET('H4','H3'),
    W_SETUP.FX('BAT2','R7','H4','H3',SLOT)=0;
    JAN_SET('BAT2','R7','H4','H3',SLOT) = NO;
  );
  IF(ORD(SLOT) LT 6+TP('H1')+T_SET('H1','H1')+T_SET('H4','H3')+TP('H3') OR
    (ORD(SLOT) GT 39-TP('H4') AND ORD(SLOT) LE 38) OR
    (ORD(SLOT) GT 43-TP('H4') AND ORD(SLOT) LE 42) OR
    (ORD(SLOT) GT 64-TP('H4') AND ORD(SLOT) LE 63) OR
    ORD(SLOT) GT 81-TP('H4')-TP('H5')-TP('H6'),
    W.FX('BAT2','R7','H4',SLOT)=0;
    JANELAS('BAT2','R7','H4',SLOT) = NO;
  );
  IF(ORD(SLOT) LT 6+TP('H1')+T_SET('H1','H1')+T_SET('H4','H3')+
    TP('H3')+TP('H4') OR
    ORD(SLOT) GE 81-TP('H5')-TP('H6'),
    W.FX('BAT2','R7','HOLH4',SLOT)=0;
    JANELAS('BAT2','R7','HOLH4',SLOT) = NO;
  );
);

```

```

IF(ORD(SLOT) LT 6+T_SET('H4','H3')+TP('H3') OR
   ORD(SLOT) GT 81-TP('H4')-TP('H5')-TP('H6')-T_SET('H3','H4'),
   W_SETUP.FX('BAT2','R7','H3','H4',SLOT)=0;
   JAN_SET('BAT2','R7','H3','H4',SLOT) = NO;
);
IF(ORD(SLOT) LT 6+TP('H1')+T_SET('H1','H1')+T_SET('H4','H3')+
   TP('H3')+TP('H4') OR
   (ORD(SLOT) GT 39-TP('H5') AND ORD(SLOT) LE 38) OR
   (ORD(SLOT) GT 43-TP('H5') AND ORD(SLOT) LE 42) OR
   (ORD(SLOT) GT 64-TP('H5') AND ORD(SLOT) LE 63) OR
   ORD(SLOT) GT 81-TP('H5')-TP('H6'),
   W.FX('BAT2','FP1','H5',SLOT)=0;
   JANELAS('BAT2','FP1','H5',SLOT) = NO;
);
IF(ORD(SLOT) LT 6+TP('H1')+T_SET('H1','H1')+T_SET('H4','H3')+
   TP('H3')+TP('H4')+TP('H5') OR
   ORD(SLOT) GE 81-TP('H6'),
   W.FX('BAT2','FP1','HOLH5',SLOT)=0;
   JANELAS('BAT2','FP1','HOLH5',SLOT) = NO;
);
IF(ORD(SLOT) LT 6+TP('H5') OR
   ORD(SLOT) GT 81-TP('H5')-TP('H6')-T_SET('H5','H5'),
   W_SETUP.FX('BAT2','FP1','H5','H5',SLOT)=0;
   JAN_SET('BAT2','FP1','H5','H5',SLOT) = NO;
);
IF(ORD(SLOT) LT 6+TP('H5') OR
   ORD(SLOT) GT 81-TP('H5')-TP('H6')-T_SET('E12','H5'),
   W_SETUP.FX('BAT2','FP1','E12','H5',SLOT)=0;
   JAN_SET('BAT2','FP1','E12','H5',SLOT) = NO;
);
IF(ORD(SLOT) LT 6+TP('H1')+T_SET('H1','H1')+T_SET('H4','H3')+TP('H3')+
   TP('H4')+T_SET('H5','H5')+TP('H5')+T_SET('H6','H6') OR
   (ORD(SLOT) GT 39-TP('H4') AND ORD(SLOT) LE 38) OR
   (ORD(SLOT) GT 43-TP('H4') AND ORD(SLOT) LE 42) OR
   (ORD(SLOT) GT 64-TP('H4') AND ORD(SLOT) LE 63) OR
   ORD(SLOT) GT 81-TP('H6'),
   W.FX('BAT2','TRS','H6',SLOT)=0;
   JANELAS('BAT2','TRS','H6',SLOT) = NO;
);
IF(ORD(SLOT) LT 6+TP('H6')+T_SET('H6','H6') OR
   ORD(SLOT) GE 81-TP('H6')-T_SET('H6','H6'),
   W_SETUP.FX('BAT2','TRS','H6','H6',SLOT)=0;
   JAN_SET('BAT2','TRS','H6','H6',SLOT) = NO;
);
IF(ORD(SLOT) LT 6+TP('H6')+T_SET('H6','H6') OR
   ORD(SLOT) GE 81-TP('H6')-T_SET('E13','H6'),
   W_SETUP.FX('BAT2','TRS','E13','H6',SLOT)=0;
   JAN_SET('BAT2','TRS','E13','H6',SLOT) = NO;
);
);

```

```

* A BATELADA 3 DEVE FICAR PRONTA ATE 12 AGO (SL97)
* NAO PODE HAVER NENHUMA BATELADA EM PROCESSAMENTO NO FIM DE SEMANA (SL39)
* NAO PODE HAVER NENHUMA BATELADA EM PROCESSAMENTO NO FERIADO (SL43)
* NAO PODE HAVER NENHUMA BATELADA EM PROCESSAMENTO NO FIM DE SEMANA (SL64)

```

```

LOOP(SLOT,
   IF(ORD(SLOT) LT 6+T_SET('H1','H1')+T_SET('H1','E11')+
      T_SET('E11','H1')+TP('H1') OR
      (ORD(SLOT) GT 39-TP('H1') AND ORD(SLOT) LE 38) OR
      (ORD(SLOT) GT 43-TP('H1') AND ORD(SLOT) LE 42) OR
      (ORD(SLOT) GT 64-TP('H1') AND ORD(SLOT) LE 63) OR
      ORD(SLOT) GT 97-TP('H1')-TP('H3')-TP('H4')-TP('H5')-TP('H6'),
      W.FX('BAT3','R3','H1',SLOT)=0;
      JANELAS('BAT3','R3','H1',SLOT) = NO;

```

```

);
IF(ORD(SLOT) LT 6+2*TP('H1')+T_SET('H1','H1')+T_SET('H1','E11')+
      T_SET('E11','H1') OR
ORD(SLOT) GE 97-TP('H3')-TP('H4')-TP('H5')-TP('H6'),
W.FX('BAT3','R3','HOLH1',SLOT)=0;
JANELAS('BAT3','R3','HOLH1',SLOT) = NO;
);
IF(ORD(SLOT) LT 6+T_SET('E11','H1')+T_SET('H1','H1')+
      T_SET('H1','E11')+TP('H1') OR
ORD(SLOT) GT 97-TP('H1')-TP('H3')-TP('H4')-TP('H5')-TP('H6')-
      T_SET('H1','H1'),
W_SETUP.FX('BAT3','R3','H1','H1',SLOT)=0;
JAN_SET('BAT3','R3','H1','H1',SLOT) = NO;
);
IF(ORD(SLOT) LT 6+T_SET('E11','H1')+T_SET('H1','H1')+
      T_SET('H1','E11')+TP('H1') OR
ORD(SLOT) GT 97-TP('H1')-TP('H3')-TP('H4')-TP('H5')-TP('H6')-
      T_SET('E11','H1'),
W_SETUP.FX('BAT3','R3','E11','H1',SLOT)=0;
JAN_SET('BAT3','R3','E11','H1',SLOT) = NO;
);
IF(ORD(SLOT) LT 6+TP('E11')+TP('E12')+TP('E13')+T_SET('F2','H3') OR
ORD(SLOT) GT 97-TP('H2')-TP('H4')-TP('H5')-TP('H6'),
W.FX('BAT3','R4','H2',SLOT)=0;
JANELAS('BAT3','R4','H2',SLOT) = NO;
);
IF(ORD(SLOT) LT 6+T_SET('H2','F1')+TP('F1') OR
ORD(SLOT) GT 97-TP('H2')-TP('H4')-TP('H5')-TP('H6')-T_SET('F1','H2'),
W_SETUP.FX('BAT3','R4','F1','H2',SLOT)=0;
JAN_SET('BAT3','R4','F1','H2',SLOT) = NO;
);
IF(ORD(SLOT) LT 6+TP('E11')+TP('E12')+TP('E13')+T_SET('F2','H3') OR
ORD(SLOT) GT 97-TP('H3')-TP('H4')-TP('H5')-TP('H6'),
W.FX('BAT3','R7','H3',SLOT)=0;
JANELAS('BAT3','R7','H3',SLOT) = NO;
);
IF(ORD(SLOT) LT 6+TP('E11')+TP('E12')+TP('E13') OR
ORD(SLOT) GT 97-TP('H3')-TP('H4')-TP('H5')-TP('H6')-T_SET('F2','H3'),
W_SETUP.FX('BAT3','R7','F2','H3',SLOT)=0;
JAN_SET('BAT3','R7','F2','H3',SLOT) = NO;
);
IF(ORD(SLOT) LT 6+TP('E11')+TP('E12')+TP('E13')+T_SET('F2','H3')+
      TP('H3') OR
ORD(SLOT) GT 97-TP('H4')-TP('H5')-TP('H6'),
W.FX('BAT3','R7','H4',SLOT)=0;
JANELAS('BAT3','R7','H4',SLOT) = NO;
);
IF(ORD(SLOT) LT 6+TP('E11')+TP('E12')+TP('E13')+T_SET('F2','H3')+
      TP('H3')+TP('H4') OR
ORD(SLOT) GE 97-TP('H5')-TP('H6'),
W.FX('BAT3','R7','HOLH4',SLOT)=0;
JANELAS('BAT3','R7','HOLH4',SLOT) = NO;
);
IF(ORD(SLOT) LT 6+TP('E11')+TP('E12')+TP('E13')+T_SET('F2','H3')+
      TP('H3') OR
ORD(SLOT) GT 97-TP('H4')-TP('H5')-TP('H6')-T_SET('H3','H4'),
W_SETUP.FX('BAT3','R7','H3','H4',SLOT)=0;
JAN_SET('BAT3','R7','H3','H4',SLOT) = NO;
);
IF(ORD(SLOT) LT 6+TP('E11')+TP('E12')+TP('E13')+T_SET('F2','H3')+
      TP('H3')+TP('H4')+T_SET('H5','H5') OR
ORD(SLOT) GT 97-TP('H5')-TP('H6'),
W.FX('BAT3','FP1','H5',SLOT)=0;
JANELAS('BAT3','FP1','H5',SLOT) = NO;
);
IF(ORD(SLOT) LT 6+TP('E11')+TP('E12')+TP('E13')+T_SET('F2','H3')+

```

```

        TP('H3')+TP('H4')+TP('H5')+T_SET('H5','H3') OR
ORD(SLOT) GE 97-TP('H6'),
W.FX('BAT3','FP1','HOLH5',SLOT)=0;
JANELAS('BAT3','FP1','HOLH5',SLOT) = NO;
);
IF(ORD(SLOT) LT 64+TP('E11')+TP('E12') OR
ORD(SLOT) GT 97-TP('H5')-TP('H6')-T_SET('E12','H5'),
W_SETUP.FX('BAT3','FP1','E12','H5',SLOT)=0;
JAN_SET('BAT3','FP1','E12','H5',SLOT) = NO;
);
IF(ORD(SLOT) LT 64+TP('E21')+TP('E22') OR
ORD(SLOT) GT 97-TP('H5')-TP('H6')-T_SET('E22','H5'),
W_SETUP.FX('BAT3','FP1','E22','H5',SLOT)=0;
JAN_SET('BAT3','FP1','E22','H5',SLOT) = NO;
);
IF(ORD(SLOT) LT 64+TP('E11')+TP('E12')+TP('E13')+T_SET('F2','H3')+
TP('H3')+TP('H4')+TP('H5')+T_SET('F4','H6') OR
ORD(SLOT) GT 97-TP('H6'),
W.FX('BAT3','TRS','H6',SLOT)=0;
JANELAS('BAT3','TRS','H6',SLOT) = NO;
);
IF(ORD(SLOT) LT 64+TP('E11')+TP('E12')+TP('E13')+TP('F3')+TP('F4') OR
ORD(SLOT) GE 97-TP('H6')-T_SET('F4','H6'),
W_SETUP.FX('BAT3','TRS','F4','H6',SLOT)=0;
JAN_SET('BAT3','TRS','F4','H6',SLOT) = NO;
);
);
);

```

Programa Principal (REGBATH2.GMS)

```

$TITLE Artigo Egli(1986)/
$OFFUPPER
$OFFSYMXREF
$OFFSYMLIST

```

```

$INCLUDE 'REGBATH2.INC'

```

```

VARIABLES

```

```

STOCK(ESTADOS,SLOT) Estoque de ESTADOS no inicio de SLOT
W(BAT,PROC,OPR,SLOT) Inicio de BAT em PROC SLOT
W_SETUP(BAT,PROC,OPR,OPRL,SLOT) Inicio de Setup de OPRL apos OPR
ULT_OPR(BAT,PROC,OPR) Indica ultima OPR em PROC
V(ESTADOS,SLOT) Venda de ESTADOS no inicio de SLOT
QEL(SLOT) Consumo de energia eletrica no SLOT
QVAP(SLOT) Consumo de vapor no SLOT
COST Custo;

```

```

POSITIVE VARIABLES STOCK,V,QEL,QVAP;
BINARY VARIABLES W, W_SETUP, ULT_OPR;

```

```

$INCLUDE 'JAN_TUDO.INC'

```

```

EQUATIONS

```

```

ORDEN_BAT1(BATL1,PROC,OPR1) Ordena as bateladas com SETUP
ORDEN_BAT2(BATL2,PROC,OPR2) Ordena as bateladas com SETUP
ORDEN_BAT3(BAT,PROC,OPR) Ordena as bateladas sem SETUP
SOMA_BAT1(BAT,PROC,OPR) Existencia de apenas uma batelada em BAT de OPR
SOMA_BAT2(PROC,OPR)
SOMA_BAT3
SOMA_BAT4
SOMA_BAT5

```

```

EXI_SET1(BAT,PROC,OPR2) Existencia de setup em PROC
EXI_SET2
EXI_SET3
EXI_SET4
EXI_SET5
EXI_SET6
ULTIMO_SET(PROC) Existencia da ultima OPR em PROC
BAL_BAT(BAT,ESTADOS,SLOT) Balanco de massa das bateladas
BAL_MASSA(ESTADOS,SLOT) Balanco de massa de ESTADOS em SLOT
RESTR_COEX(PROC,SLOT) Nao coexistencia de OPR em PROC SLOT e outras
CONSUMO_EL,SLOT) Consumo de eletricidade no SLOT
CONSUMO_VP,SLOT) Consumo de vapor no SLOT
VENDAS(ESTADOS,DIA) Equacao das vendas efetuadas
CUSTO;

ORDEN_BAT1(BATL1,PROC,OPR1) $(HOR_BAT(BATL1,OPR1)*NOT_HOLD(OPR1)*
SET_PROC(PROC)*PROC_OPR(PROC,OPR1))..
SUM((BATL2,OPR2) $(PREC_SET(BATL1,BATL2,PROC,OPR1,OPR2)),
SUM(SLOT $(JAN_SET(BATL2,PROC,OPR1,OPR2,SLOT)),
W_SETUP(BATL2,PROC,OPR1,OPR2,SLOT)*ORD(SLOT)) +
SUM(SLOT $(CAND_ULT(BATL1,PROC,OPR1)),
ULT_OPR(BATL1,PROC,OPR1)) =G=
SUM(SLOT $(JANELAS(BATL1,PROC,OPR1,SLOT)),
W(BATL1,PROC,OPR1,SLOT)*(ORD(SLOT)+TP(OPR1)));

ORDEN_BAT2(BATL2,PROC,OPR2)
$(HOR_SET(BATL2,OPR2)*SET_PROC(PROC)*PROC_OPR(PROC,OPR2))..
SUM(SLOT,
SUM(OPR1 $(JAN_SET(BATL2,PROC,OPR1,OPR2,SLOT)),
W_SETUP(BATL2,PROC,OPR1,OPR2,SLOT)*(ORD(SLOT)+T_SET(OPR1,OPR2))))=L=
SUM(SLOT $(JANELAS(BATL2,PROC,OPR2,SLOT)),
W(BATL2,PROC,OPR2,SLOT)*ORD(SLOT));

ORDEN_BAT3(BAT,PROC,OPR) $(PROC_OPR(PROC,OPR)*NOT_HOLD(OPR)*
(NOT(SET_PROC(PROC)))*HOR_BAT(BAT,OPR)*ORD_BAT(BAT))..
SUM(SLOT $(JANELAS(BAT-1,PROC,OPR,SLOT)),
W(BAT-1,PROC,OPR,SLOT)*(ORD(SLOT)+TP(OPR))) =L=
SUM(SLOT $(JANELAS(BAT,PROC,OPR,SLOT)),
W(BAT,PROC,OPR,SLOT)*ORD(SLOT));

SOMA_BAT1(BAT,PROC,OPR)
$(PROC_OPR(PROC,OPR)*NOT_HOLD(OPR)*HOR_BAT(BAT,OPR)*
(NOT(PROD_OPR('E1',OPR)*HOR_BAT('BAT2',OPR)))*(NOT(PROD_OPR('E2',OPR))))..
SUM(SLOTL $(JANELAS(BAT,PROC,OPR,SLOTL)),
W(BAT,PROC,OPR,SLOTL)) =E= 1;

SOMA_BAT2(PROC,OPR)
$(PROC_OPR(PROC,OPR)*NOT_HOLD(OPR)*PROD_OPR('E1',OPR))..
SUM(SLOTL $(JANELAS('BAT1',PROC,OPR,SLOTL)),
W('BAT1',PROC,OPR,SLOTL)) =E= 1;

SOMA_BAT3 ..
SUM(SLOTL,
W('BAT2','R3','E11',SLOTL) $(JANELAS('BAT2','R3','E11',SLOTL)) +
W('BAT1','R4','E21',SLOTL) $(JANELAS('BAT1','R4','E21',SLOTL)))
=E= 1;

SOMA_BAT4 ..
SUM(SLOTL,
W('BAT2','FP1','E12',SLOTL) $(JANELAS('BAT2','FP1','E12',SLOTL)) +
W('BAT1','FP1','E22',SLOTL) $(JANELAS('BAT1','FP1','E22',SLOTL)))
=E= 1;

SOMA_BAT5 ..

```

```

SUM(SLOTL,
  W('BAT2', 'TRS', 'E13', SLOTL) $(JANELAS('BAT2', 'TRS', 'E13', SLOTL)) +
  W('BAT1', 'TRS', 'E23', SLOTL) $(JANELAS('BAT1', 'TRS', 'E23', SLOTL)))
  =E= 1;

EXI_SET1(BATL2, PROC, OPR2) $(PROC_OPR(PROC, OPR2)*HOR_SET(BATL2, OPR2)*
(NOT(PROD_OPR('E1', OPR2)*HOR_BAT('BAT2', OPR2)))*(NOT(PROD_OPR('E2', OPR2))))..
  SUM(OPR1,
    SUM(SLOT $(JAN_SET(BATL2, PROC, OPR1, OPR2, SLOT)),
      W_SETUP(BATL2, PROC, OPR1, OPR2, SLOT))) =E= 1;

EXI_SET2 ..
  SUM((OPR1, SLOT),
    W_SETUP('BAT2', 'R3', OPR1, 'E11', SLOT)
      $(JAN_SET('BAT2', 'R3', OPR1, 'E11', SLOT)) +
    W_SETUP('BAT1', 'R4', OPR1, 'E21', SLOT)
      $(JAN_SET('BAT1', 'R4', OPR1, 'E21', SLOT)))
  =E= 1;

EXI_SET3 ..
  SUM((OPR1, SLOT),
    W_SETUP('BAT2', 'FP1', OPR1, 'E12', SLOT)
      $(JAN_SET('BAT2', 'FP1', OPR1, 'E12', SLOT)) +
    W_SETUP('BAT1', 'FP1', OPR1, 'E22', SLOT)
      $(JAN_SET('BAT1', 'FP1', OPR1, 'E22', SLOT)))
  =E= 1;

EXI_SET4 ..
  SUM((OPR1, SLOT),
    W_SETUP('BAT2', 'TRS', OPR1, 'E13', SLOT)
      $(JAN_SET('BAT2', 'TRS', OPR1, 'E13', SLOT)) +
    W_SETUP('BAT1', 'TRS', OPR1, 'E23', SLOT)
      $(JAN_SET('BAT1', 'TRS', OPR1, 'E23', SLOT)))
  =E= 1;

EXI_SET5 ..
  SUM(SLOT,
    W_SETUP('BAT2', 'R4', 'H2', 'F1', SLOT)
      $(JAN_SET('BAT2', 'R4', 'H2', 'F1', SLOT)) +
    W_SETUP('BAT1', 'R4', 'E21', 'F1', SLOT)
      $(JAN_SET('BAT1', 'R4', 'E21', 'F1', SLOT)))
  =E= 1;

EXI_SET6 ..
  SUM(SLOT,
    W_SETUP('BAT2', 'TRS', 'E13', 'F4', SLOT)
      $(JAN_SET('BAT2', 'TRS', 'E13', 'F4', SLOT)) +
    W_SETUP('BAT1', 'TRS', 'E23', 'F4', SLOT)
      $(JAN_SET('BAT1', 'TRS', 'E23', 'F4', SLOT)))
  =E= 1;

ULTIMO_SET(PROC) $(SET_PROC(PROC))..
  SUM((BAT, OPR)
    $(PROC_OPR(PROC, OPR)*CAND_ULT(BAT, PROC, OPR)),
    ULT_OPR(BAT, PROC, OPR)) =E= 1;

BAL_BAT(BAT, ESTADOS, SLOT) $(HOR_INI(SLOT)*EST_NULO(ESTADOS))..
  SUM(OPR $(OPR_G_PROD(OPR, ESTADOS)),
    PCOPR_PROD(OPR, ESTADOS)*
    SUM(PROC $(PROC_OPR(PROC, OPR)*JANELAS(BAT, PROC, OPR, SLOT-TP(OPR))),
      MAT(OPR)*W(BAT, PROC, OPR, SLOT-TP(OPR)))) =E=
  SUM(OPR $(OPR_C_PROD(OPR, ESTADOS)),
    PCPROD_OPR(OPR, ESTADOS)*
    SUM(PROC $(PROC_OPR(PROC, OPR)*JANELAS(BAT, PROC, OPR, SLOT)),
      MAT(OPR)*W(BAT, PROC, OPR, SLOT)));

```

BAL_MASSA(ESTADOS, SLOT) \$(HOR_INI(SLOT)*PRODUTOS(ESTADOS))..
 STOCK(ESTADOS, SLOT) =E= STOCK(ESTADOS, SLOT-1) +
 SUM((BAT, OPR) \$(OPR_G_PROD(OPR, ESTADOS)),
 PCOPR_PROD(OPR, ESTADOS) *
 SUM(PROC \$(PROC_OPR(PROC, OPR)*JANELAS(BAT, PROC, OPR, SLOT-TP(OPR))),
 MAT(OPR)*W(BAT, PROC, OPR, SLOT-TP(OPR))))-
 SUM((BAT, OPR) \$(OPR_C_PROD(OPR, ESTADOS)),
 PCPROD_OPR(OPR, ESTADOS) *
 SUM(PROC \$(PROC_OPR(PROC, OPR)*JANELAS(BAT, PROC, OPR, SLOT)),
 MAT(OPR)*W(BAT, PROC, OPR, SLOT)))-
 V(ESTADOS, SLOT);

RESTR_COEX(PROC, SLOT) \$(HORIZ(SLOT)*HOR_INI(SLOT))..
 SUM((BAT, OPR) \$(PROC_OPR(PROC, OPR)),
 SUM(SLOTL \$(ORD(SLOTL) LE ORD(SLOT) AND
 ORD(SLOTL) GE ORD(SLOT)-TP(OPR)+1),
 W(BAT, PROC, OPR, SLOTL) \$(JANELAS(BAT, PROC, OPR, SLOTL)))) +
 SUM((BATL2, OPR1, OPR2),
 SUM(SLOTL \$(ORD(SLOTL) LE ORD(SLOT) AND
 ORD(SLOTL) GE ORD(SLOT)-T_SET(OPR1, OPR2)+1),
 W_SETUP(BATL2, PROC, OPR1, OPR2, SLOTL)
 \$(JAN_SET(BATL2, PROC, OPR1, OPR2, SLOTL)))) =L= 1;

CONSUMO_EL(SLOT) \$(HORIZ(SLOT)*HOR_INI(SLOT))..
 QEL(SLOT)=E=
 SUM((BAT, OPR),
 SUM(PROC \$(PROC_OPR(PROC, OPR)),
 SUM(TETA \$(ORD(TETA) GE 1 AND ORD(TETA) LE TP(OPR) AND
 ORD(TETA) LE ORD(SLOT)),
 CREL(OPR, TETA)*W(BAT, PROC, OPR, SLOT-(ORD(TETA)-1))
 \$(JANELAS(BAT, PROC, OPR, SLOT-(ORD(TETA)-1))))));

CONSUMO_VP(SLOT) \$(HORIZ(SLOT)*HOR_INI(SLOT))..
 QVAP(SLOT)=E=
 SUM((BAT, OPR),
 SUM(PROC \$(PROC_OPR(PROC, OPR)),
 SUM(TETA \$(ORD(TETA) GE 1 AND ORD(TETA) LE TP(OPR) AND
 ORD(TETA) LE ORD(SLOT)),
 CRVAP(OPR, TETA)*W(BAT, PROC, OPR, SLOT-(ORD(TETA)-1))
 \$(JANELAS(BAT, PROC, OPR, SLOT-(ORD(TETA)-1))))));

VENDAS(ESTADOS, DIA) \$(PRODUTOS(ESTADOS)*HOR_DIA(DIA))..
 ENTREGA(ESTADOS, DIA) =E=
 SUM(SLOT \$(SLOT_DIA(SLOT, DIA)),
 V(ESTADOS, SLOT));

CUSTO..COST=E=
 SUM(ESTADOS \$(PRODUTOS(ESTADOS)), COST_EST(ESTADOS)*
 (STOCK(ESTADOS, 'SL11') + STOCK(ESTADOS, 'SL17') +
 STOCK(ESTADOS, 'SL23') + STOCK(ESTADOS, 'SL29') +
 STOCK(ESTADOS, 'SL35') + 2*STOCK(ESTADOS, 'SL39') +
 2*STOCK(ESTADOS, 'SL43') + STOCK(ESTADOS, 'SL48') +
 STOCK(ESTADOS, 'SL54') + STOCK(ESTADOS, 'SL60') +
 2*STOCK(ESTADOS, 'SL64') + STOCK(ESTADOS, 'SL69') +
 STOCK(ESTADOS, 'SL75') + STOCK(ESTADOS, 'SL81') +
 STOCK(ESTADOS, 'SL87') + STOCK(ESTADOS, 'SL93') +
 STOCK(ESTADOS, 'SL97')) +
 SUM((BAT, PROC, OPR, SLOT)
 \$(JANELAS(BAT, PROC, OPR, SLOT)*PROC_OPR(PROC, OPR)),
 COST_BAT(OPR)*W(BAT, PROC, OPR, SLOT) +
 SUM((BATL2, PROC, OPR1, OPR2, SLOT)
 \$(JAN_SET(BATL2, PROC, OPR1, OPR2, SLOT)),
 COST_SET(OPR1, OPR2)*W_SETUP(BATL2, PROC, OPR1, OPR2, SLOT));

```
MODEL EGLI/ALL/;
OPTION LIMROW=0;
OPTION LIMCOL=0;
OPTION SOLPRINT=OFF;
*EGLI.OPTFILE=1;
OPTION RESLIM=8000;
OPTION ITERLIM=100000;
OPTION OPTCR=0.15;

SOLVE EGLI USING MIP MINIMIZING COST;

DISPLAY EGLI.NODUSD, STOCK.L, V.L, W.L, W_SETUP.L, QEL.L, QVAP.L,
        ULT_OPR.L;

$INCLUDE 'SAIREGH2.INC'
```

Anexo 2 - Nomenclatura utilizada no GAMS/OSL

Índices utilizados:

b	BAT
b_1	BATL1
b_2	BATL2
d	DIA
i	OPR
i_1	OPR1
i_2	OPR2
j	PROC
s	ESTADOS
t	SLOT

Variáveis das equações:

D_{st}	V(ESTADOS,SLOT)	
S_{st}	STOCK(ESTADOS,SLOT)	
$U_{ut} \Rightarrow U_{el,t}$	QEL(SLOT)	$u = el$ (eletricidade)
$U_{ut} \Rightarrow U_{vap,t}$	QVAP(SLOT)	$u = vap$ (vapor)
W_{bjit}	W(BAT,PROC,OPR,SLOT)	
$WS_{b_2,j,i_1,i_2,t}$	W_SETUP(BATL2,PROC,OPR1,OPR2,SLOT)	
Y_{bji}	ULT_OPR(BAT,PROC,OPR)	

Dados das equações:

E_{sd}	ENTREGA(ESTADOS,DIA)	
M_i	MAT(OPR)	
p_i	TP(OPR)	
$U_{u,t}^{max} \Rightarrow U_{el,t}^{max}$	QEL.UP(SLOT)	$u = el$ (eletricidade)
$U_{u,t}^{max} \Rightarrow U_{vap,t}^{max}$	QVAP.UP(SLOT)	$u = vap$ (vapor)
$\alpha_{ui\theta} \Rightarrow \alpha_{el,i\theta}$	CREL(OPR,TETA)	$u = el$ (eletricidade)
$\alpha_{ui\theta} \Rightarrow \alpha_{vap,i\theta}$	CRVAP(OPR,TETA)	$u = vap$ (vapor)
ρ'_{is}	PCOPR_PROD(OPR,ESTADOS)	
ρ_{is}	PCPROD_OPR(OPR,ESTADOS)	
τ_{i_1,i_2}	T_SET(OPR1,OPR2)	
θ	TETA	

Conjuntos utilizados na limitação do domínio das equações:

$b \in C_{ji}$	CAND_ULT(BAT,PROC,OPR)
$s \in EN$	EST_NULO(ESTADOS)
$t \in H$	HORIZ(SLOT)
$b \in HB_i$	HOR_BAT(BAT,OPR)
$d \in HD$	HOR_DIA(DIA)
$t \in HI$	HOR_INI(SLOT)
$b \in HS_i$	HOR_SET(BAT,OPR)
$i \in I_j$	PROC_OPR(PROC,OPR)
$j \in J_i$	PROC_OPR(PROC,OPR)
$t \in JB_{bji}$	JANELAS(BAT,PROC,OPR,SLOT)
$i \in NH$	NOT_HOLD(OPR)
$i \in OC_s$	OPR_C_PROD(OPR,ESTADOS)
$i \in OG_s$	OPR_G_PROD(OPR,ESTADOS)
$s \in P$	PRODUTOS(ESTADOS)
$b_2 \in PS_{b_1,j_1,j_2}$	PREC_SET(BATL1,BATL2,PROC,OPR1,OPR2)
$t \in SB_{b_2,j_1,j_2}$	JAN_SET(BATL2,PROC,OPR1,OPR2,SLOT)
$j \in SP$	SET_PROC(PROC)
$t \in TD_d$	SLOT_DIA(SLOT,DIA)

Anexo 3 - Modelamento da entrada e saída contínua de produtos

Até o momento foi assumido que todas as matérias primas são transferidas para dentro do processador antes do início do processamento das tarefas, e que todos os produtos são retirados do processador após o fim do processamento das tarefas. No entanto, é possível estender esta formulação para os casos de alimentação e produção contínuas de matérias primas e produtos, respectivamente. A seguir é apresentada a formulação de Kondili *et al.* (1993).

A equação de balanço de massa [eq. 2.1] é reescrita como [eq. a.1], usando os dados $\rho'_{is\theta}$ e $\rho_{is\theta}$ dependentes do tempo de processamento θ , para permitir uma taxa variável de alimentação de material e de saída de produto da tarefa i .

$$S_{st} = S_{s,t-1} + \sum_{i \in OC_s} \sum_{\theta=0}^{p_i} \rho'_{is\theta} * \sum_{j \in J_i} M_{j,i,t-\theta} - \sum_{i \in OC_s} \sum_{\theta=0}^{p_i} \rho_{is\theta} * \sum_{j \in J_i} M_{j,i,t-\theta} + R_{st} - D_{st}$$

,para $\forall s$ e t . [a.1].

Tabela A.1. Nomenclatura das equações [a.1] a [a.3]

Dados das equações:

$\rho'_{is\theta}$	Proporção da batelada que sai da tarefa i para o estado s no início do intervalo de tempo θ ($\theta = 0, \dots, p_i$) relativo ao início do processamento da tarefa.
$\rho_{is\theta}$	Proporção da batelada que entra na tarefa i vindo do estado s no início do intervalo de tempo θ ($\theta = 0, \dots, p_i$) relativo ao início do processamento da tarefa.
λ_i	Fração máxima da batelada da tarefa i que pode estar em processamento em qualquer instante de tempo.

É possível que haja saída de produto da tarefa i antes que termine a entrada de matéria prima para esta tarefa. Neste caso, sempre haverá apenas uma fração da batelada dentro do processador durante o processamento desta tarefa. A existência de uma fração máxima de uma batelada da tarefa i (λ_i) que pode estar em processamento em um instante qualquer, força uma alteração na equação da limitação da capacidade de processamento (eq. [2.7]). É dada abaixo a equação [a.2], que substitui a equação [2.7] no caso de haver alimentação e produção contínuas.

$$W_{jt} * V_{ji}^{min} \leq \lambda_i * M_{jt} \leq W_{jt} * V_{ji}^{max} \quad , \text{ para } \forall i, j \in J, t \in H. \quad [a.2].$$

A fração máxima de uma batelada da tarefa i (λ_i) que pode estar em processamento em um instante qualquer, é uma característica da tarefa i e pode ser calculada *a priori*, como é mostrado na equação [a.3].

$$\lambda_i \equiv \max_{0 \leq \theta \leq \rho_i} \sum_{\theta'=0}^{\theta} \left(\sum_{s \in S_i} \rho_{is\theta'} - \sum_{s \in S_i'} \rho'_{is\theta'} \right) \quad , \text{ para } \forall i. \quad [a.3].$$

Anexo 4 - Modelamento da manutenção preventiva

A maioria dos processadores necessita de manutenção preventiva que pode ser apenas uma limpeza, ou a troca de peças do equipamento ou a troca dos ferramentais de produção que se desgastam com o uso. Para estes processadores Kondili *et al.*(1993), estabeleceu uma formulação para as operações de manutenção que são feitas periodicamente e que também podem ser planejadas juntamente com o planejamento do processamento das tarefas.

A necessidade das operações de manutenção em um processador é expressa em termos da frequência da sua utilização. Por exemplo, um reator pode precisar de limpeza após ser usado no máximo m_j vezes, a fim de evitar o aumento da quantidade de certas impurezas no reator. As equações [a.4] e [a.5], abaixo, garantem que haverá uma operação de manutenção em um processador j após o processamento de no máximo m_j tarefas (que é o limite máximo de tarefas processáveis sem uma operação de manutenção).

$$0 \leq \Pi_{jt} \leq m_j (1 - WM_{jt}) \quad , \text{ para } \forall j, t \in H. \quad [a.4].$$

$$- m_j * WM_{jt} \leq \Pi_{jt} - \Pi_{j,t-1} - \sum_{i \in I_j} W_{jit} \leq WM_{jt} \quad , \text{ para } \forall j, t \in H. \quad [a.5].$$

Tabela A.2. Nomenclatura das equações [a.4] a [a.6].

Variáveis das equações:

WM_{jt} = 1, Indica que a operação de manutenção do processador j é iniciada no *slot* t .
= 0, Caso contrário.

Π_{jt} Indica o número de tarefas que já foram processadas no processador j no *slot* t desde a última operação de manutenção.

Dados das equações:

m_j Limite máximo de tarefas que podem ser processadas antes que uma operação de manutenção seja necessária. Indica com que frequência as operações de manutenção devem ser feitas.

ξ_j Tempo de duração da operação de manutenção preventiva no processador j .

Nas equações [a.4] e [a.5] é preciso garantir que o número de tarefas que já foram processadas no processador j no *slot* t desde a última operação de manutenção (Π_{jt}), cumpram as seguintes restrições:

- a) Π_{jt} jamais exceda o limite máximo de tarefas que podem ser processadas antes que uma operação de manutenção seja necessária, m_j .
- b) Π_{jt} deve ser zerado se $WM_{jt} = 1$.
- c) Π_{jt} deve ser elevado em 1 se $W_{jit} = 1$, para $i \in I_j$.

Se $WM_{jt} = 0$, a equação [a.4] garante que Π_{jt} não excederá o limite máximo de tarefas que podem ser processadas antes que uma operação de manutenção seja necessária (m_j). Além disso, se $WM_{jt} = 0$, a equação [a.5] garante que Π_{jt} será elevado em 1 se $W_{jit} = 1$, para $i \in I_j$.

Se $WM_{jt} = 1$, a equação [a.4] garante que Π_{jt} será zerado, enquanto a equação [a.5] se transforma na expressão $-m_j \leq -\Pi_{j,t-1} \leq 1$ que não impõe nenhuma restrição ao problema.

Se for usada a operação de manutenção preventiva no planejamento de produção, será necessário alterar a equação [2.15], equação de não coexistência. A equação [2.15] é substituída pela equação [a.6], que também leva em consideração as operações de manutenção preventiva. Note que ou o processador j está processando uma tarefa $i \in I_j$ ou o processador está realizando uma operação de preparação ou o processador está realizando uma operação de manutenção preventiva ou o processador está desocupado.

$$\sum_{i \in I_j} \sum_{t'=t}^{t-p_i+1} W_{j,i,t'} + \sum_{k_1} \sum_{k_2} \sum_{t'=t}^{t-\tau_{j,k_1,k_2}+1} WS_{j,k_1,k_2,t'} + \sum_{t'=t}^{t-\xi_j+1} WM_{jt'} \leq 1$$

, para $\forall j, t \in H$. [a.6].

Referências Bibliográficas

- Brooke, A. *et al.* (1992), "A user's guide (release 2.25)", The Scientific Press, San Francisco, 1992.
- Corrêa L.H. e Giansesi I.G.N. (1996), "Just in Time, MRP II e OPT: Um Enfoque Estratégico", editora Atlas, 1996.
- Dilworth J.B. (1992), "Operations Management: Design, Planning, and Control for Manufacturing and Services", editora McGraw-Hill, 1992.
- Egli U.M., Rippin D.W.T. (1986), "Short-Term Scheduling for Multiproduct Batch Chemical Plants", *Computers and Chemical Engineering*, 10 (4), pp. 303-325, 1986.
- IBM-OSL (1991), "Guide and Reference (release 2)", Kingston, New York, 1991.
- Kondili E., Pantelides C.C., Sargent R.W.H. (1993), "A General Algorithm for Short-Term Scheduling of Batch Operations - I. A MILP Formulation", *Computers and Chemical Engineering*, 17 (2), pp. 211-227, 1993.
- Medeiros A. C. G. (1995), "Estratégia de Simulação para Alocação de Operações em Plantas Multipropósito", Tese de Mestrado (DESQ, FEQ, Unicamp), 1995.
- Rapacoulias C., Shah N., Pantelides C.C. (1991), "Optimal Scheduling of Order-Driven Batch Chemical Plants", *Computer-Oriented Process Engineering*, 145-150, 1991.
- Rippin D.W.T. (1991), "Batch Process Planning", *Chemical Engineering*, (5), pp. 100-107, 1991.
- Shah N., Pantelides C.C. (1991), "Scheduling of Lubricants Blending Operations - A Case Study in Optimal Scheduling", *AIChE Annual Meeting*, Los Angeles, 1991.
- Shah N., Pantelides C.C., Sargent R.W.H. (1993), "A General Algorithm for Short-Term Scheduling of Batch Operations - II. Computational Issues", *Computers and Chemical Engineering*, 17 (2), pp. 229-244, 1993.