



APLICAÇÃO DE INTERFERÊNCIA LÓGICA EM PROBLEMAS DE PROGRAMAÇÃO DE PRODUÇÃO

Edilson de J. Santos

Centro de Ciências Formais e Tecnologia
Universidade Tiradentes (UNIT) – Aracaju – SE
E-mail: conde@unitnet.com.br

M.T.M. Rodrigues

Faculdade de Engenharia Química – UNICAMP

L. Gimeno Latre

Faculdade de Engenharia Elétrica – UNICAMP

Resumo

A Programação de Produção de plantas flexíveis tem merecido crescente atenção no âmbito da Engenharia Química nos últimos anos. O objetivo principal da Programação de Produção é a alocação temporal de recursos, tais como disponibilidade de matéria-prima, utilidades e mão-de-obra, procurando otimizar um critério de desempenho. Este tipo de problema é computacionalmente difícil de ser resolvido e as diferentes abordagens propostas têm se mostrado inadequadas no que se refere ao tempo para a obtenção da solução ótima. No presente trabalho, a abordagem MILP (Mixed Integer Linear Problem) baseada em uma discretização uniforme do tempo de produção é utilizada juntamente com uma abordagem de interferência lógica externa, esta última desenvolvida utilizando relações lógicas envolvendo recursos compartilhados, as quais são implementadas no sistema OSL (Optimization Subroutine Library).

Palavras-chave: *sistemas químicos, processos batelada, programação de produção, programação matemática, inferência lógica.*

1. Introdução

Quando se parte para a formulação matemática de um determinado problema, diferentes

modelos podem surgir, principalmente quando esta formulação exige variáveis inteiras, sendo que cada um dos modelos apresenta uma determinada dificuldade de solução associada.

Neste fato, reside um dos principais objetivos de pesquisadores que atuam na área de modelagem matemática, que é a obtenção de modelos de baixo custo de solução, possibilitando assim a solução de problemas cada vez mais próximos dos problemas reais, que no caso particular da programação de produção, esta realidade é representada pelo elevado número de restrições envolvidas (as quais podem ser de formulação complexa) e/ou pelo grande número de tarefas habilitadas aos equipamentos. Esta última característica é traduzida em um aumento no número de variáveis binárias, fator determinante e incontestável para o aumento do custo computacional de solução dos problemas de programação de produção.

Buscando diminuir o custo de solução de problemas de síntese de processos RAMAN & GROSSMANN (1991) mostram como relações lógicas podem ser obtidas e aplicadas durante a solução dos modelos no sistema OSL (*Optimization Subroutine Library*), diminuindo assim o tempo de solução dos mesmos. Posteriormente, RAMAN & GROSSMANN (1992) mostram como restrições lógicas, na forma de inequações, podem melhorar o custo de solução de problemas de síntese de processo. HOOKER (1994) mostra que a idéia básica da utilização desse conhecimento lógico (*logic cuts*) é a substituição de elementos essenciais e cruciais de otimização por elementos de natureza lógica.

O uso de conhecimento lógico durante a solução do problema implica que de alguma forma o pacote a ser utilizado permita tais interferências. Neste ponto, o projeto de implementação do OSL permite que o usuário, em determinados pontos da busca em árvore, possa interferir e inovar os procedimentos padrões das sub-rotinas do referido pacote (BROOKE *et al.*, 1988). Centrado nesta característica, este trabalho explora interferências lógicas envolvendo recursos compartilhados inerentes aos problemas de programação de produção na tentativa de diminuição do custo de solução.

2. Apresentação da Modelagem Matemática para a Programação de Produção em Sistemas Flexíveis

As estruturas de processamento podem, segundo BAKER (1973), ser classificadas como estruturas multiestágios em série (planta multiproduto) e multiestágios (planta multiproduto), que são as estruturas mais comuns nos processos químicos. Em muitos problemas envolvendo estas estruturas uma das restrições a ser considerada é a oferta limitada de recursos compartilhados, assim como mão-de-obra, água, energia e outras utilidades.

Neste trabalho, a modelagem de sistemas flexíveis de manufatura é desenvolvida utilizando a representação Rede-Estado-Tarefa (STN, *State Task Network*) proposta por KONDILI *et al.* (1993).

A formulação via Rede-Estado-Tarefa é aplicada na programação de produção de curto prazo em estruturas de produção flexíveis. Uma das características dessa formulação é a utilização da representação da estrutura de produção através do detalhamento das receitas de manufatura dos produtos, como mostra a Figura 1.

A Figura 2 mostra uma representação de sistema flexível de produção. Os nós, que simbolizam os estados, podem representar: matérias-primas, materiais em processo (produtos intermediários) e produtos finais; já os nós simbolizadores das tarefas (os retângulos) identificam a seqüência das tarefas (as quais são habilitadas para serem desenvolvidas em diferentes equipamentos, determinando assim as operações de produção) necessárias para a manufatura dos produtos (intermediários ou finais). Nota-se que por meio desta representação todos os materiais necessários para o desenvolvimento de uma operação ficam perfeitamente definidos.

2.1 Modelagem de Sistemas Flexíveis de Produção via Conceito Rede-Estado-Tarefa

Quando se parte para a solução de problemas de programação de produção via formulação

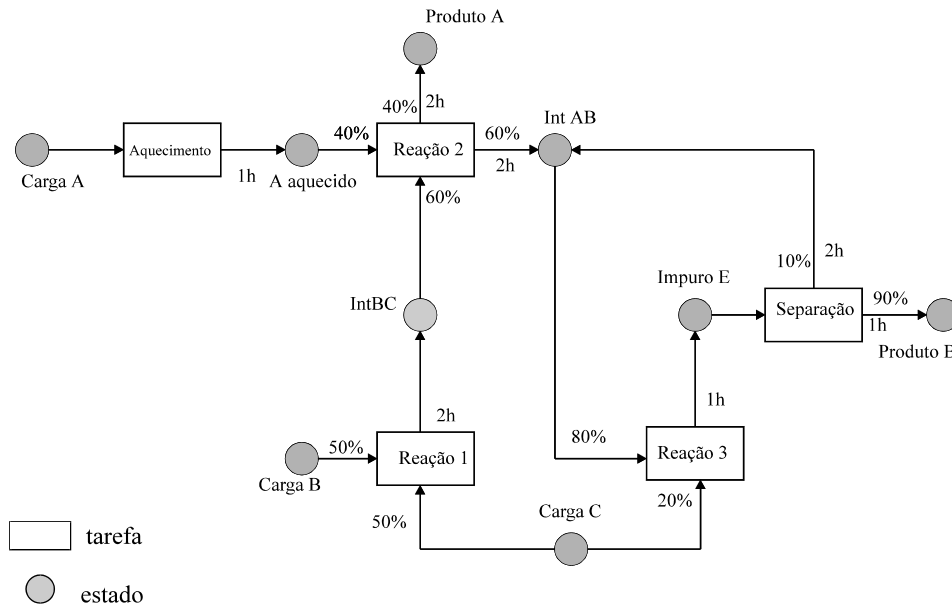


Figura 1 – Representação da Estrutura de Produção via Rede-Estado-Tarefa (KONDILI *et al.*, 1993).

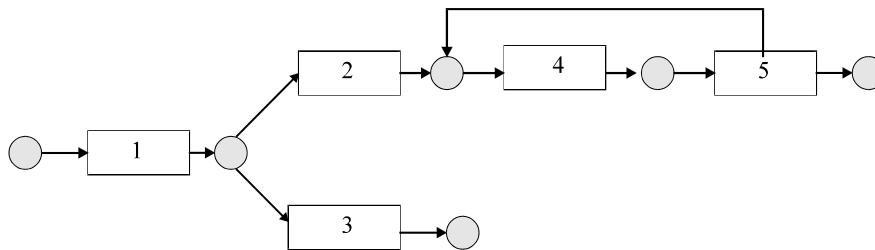


Figura 2 – Representação de processos via Rede-Estado-Tarefa.

matemática, observamos duas restrições são básicas para a modelagem:

- A restrição de ordenação das operações de diferentes tarefas em cada equipamento;
- A restrição de precedência tecnológica estabelece que uma operação de uma determinada tarefa só pode ser desenvolvida depois do término da operação anterior.

A modelagem proposta por KONDILI *et al.* (1993) utiliza a discretização uniforme do tempo previamente fixada. Neste caso, o horizonte de produção (H), estipulado para o término de toda a produção, é dividido em um número de intervalos de igual duração, denominados de período de tempo ou *slot*. O detalhamento da modelagem aqui abordada é apresentada por

KONDILI *et al.* (1993). Assim sendo, neste trabalho são mostradas restrições básicas para solução dos problemas.

A restrição de não-coexistência de operações em um determinado equipamento é proposta por SHAH *et al.* (1993). Esta restrição envolve variáveis binárias de alocação.

$$\sum_{i \in I_j} \sum_{k=k}^{k-TP_i+1} W_{ijk'} \leq 1 \quad \forall j, k$$

onde:

- i: índice de tarefas;
- j: índice de equipamentos;
- k: índice de período de tempo;
- H: horizonte de tempo;

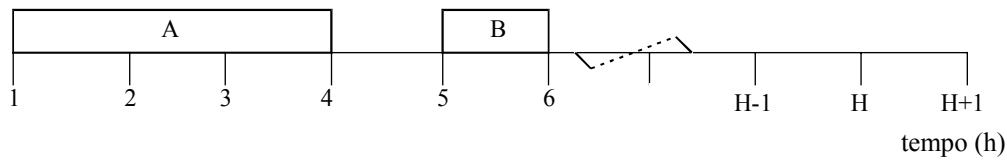


Figura 3 – Representação da alocação das tarefas A e B.

TP_i : tempo de processamento da tarefa i ;
 I_j : conjunto de tarefas habilitadas ao equipamento j .

A restrição de não-coexistência de operações em um determinado equipamento é representada na Figura 3, onde duas operações não podem ser executadas no mesmo período de tempo em um determinado equipamento.

$$W_{ijk} = \begin{cases} 1 & \text{se a operação } (i, j) \text{ inicia} \\ & \text{seu processamento no período} \\ & \text{de tempo } k \\ 0 & \text{caso contrário} \end{cases}$$

A variável binária W_{ijk} determina ou não a alocação de uma tarefa i em um equipamento j no início do período de tempo k .

A restrição de precedência tecnológica, que determina a ordenação das diversas operações de uma tarefa, é garantida nesta formulação por meio de uma equação de balanço de material em torno de cada estado em cada período de tempo, como é mostrado a seguir.

$$S_{sk} = S_{sk-1} + \sum_{i \in To_s} qo_{ik} \sum_{j \in J_i} B_{ij}(k-TP_{is}) - \sum_{i \in Ti_s} qe_{ik} \sum_{j \in J_i} B_{ijk} \quad \forall s, k$$

onde:

s : índice de estados;

S_{sk} : quantidade de material no estado s em estoque no início do período de tempo k ;

B_{ijk} : quantidade de material da operação i em processamento no equipamento j no início do período de tempo k ;

qo_{is} : percentagem de material da operação i que gera material no estado s ;

qe_{is} : percentagem de material da operação i proveniente do estado s ;

TP_{is} : tempo de processamento da operação i que produz material no estado s ;

TP_i : tempo de processamento da operação i ($\max TP_{is}$);

J_i : conjunto de equipamentos habilitados para o desenvolvimento da tarefa i ;

Ti_s : conjunto de tarefas que consomem material no estado s ;

To_s : conjunto de tarefas que geram material no estado s .

A restrição de precedência tecnológica é representada pela Figura 4. Uma operação de uma determinada tarefa só pode ser desenvolvida depois do término das operações anteriormente iniciadas.

As condições iniciais de estoque de cada material intermediário e/ou produtos (S_{s0}), quantidade de cada estado s no início da produção (período de tempo igual a 1) devem ser conhecidas.

A quantidade de material a ser processada em um determinado equipamento deve observar a disponibilidade de armazenagem para o respectivo material. Esta restrição é dada por:

$$0 \leq S_{sk} \leq C_s \quad \forall s, k$$

onde:

C_s : capacidade de armazenagem disponível para o estado s .

O consumo de recursos em um determinado período k é dado por:

$$Q_{kr} = \sum_i \sum_{j \in J_i} \sum_{k'=k-TP_i+1}^k CR_{ijr} W_{ijk} \quad \forall k, r$$

onde:

Q_{kr} : quantidade total do recurso r necessário no período de tempo k ;

CR_{ij} : quantidade de recurso r exigida pela tarefa i quando é desenvolvida no equipamento j .

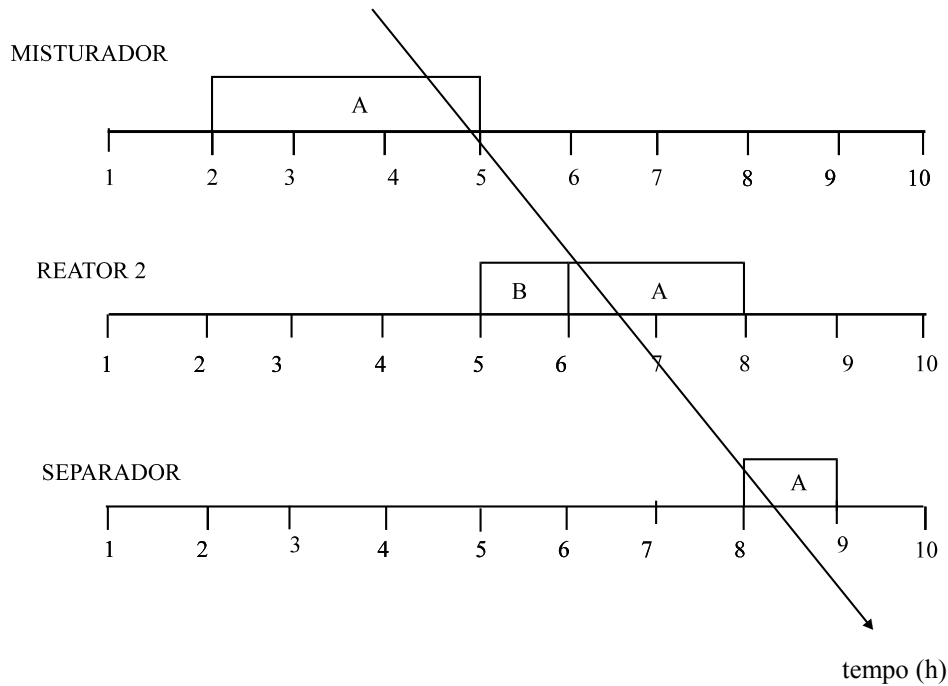


Figura 4 – Restrição de precedência tecnológica.

A limitação de consumo é dada pela expressão abaixo:

$$Q_{kr} \leq Of_r \quad \forall k,r$$

onde:

Of_r : quantidade de recurso r disponível na planta.

Esta restrição garante que a limitação de recurso não seja violada, impedindo assim a configuração mostrada na Figura 5, na qual a oferta de recurso é 10 u.r. (unidades de recurso). No período 18 a oferta do determinado recurso encontra-se violada.

As restrições que contemplam o consumo de recursos compartilhados contribuem de modo significativo para o aumento de complexidade de solução dos problemas.

No caso de estruturas de produção com processadores em paralelo e que nem todas as tarefas sejam habilitadas a todos equipamentos, é necessário algumas reformulações nas restrições vistas anteriormente. Em cada estágio de produção (que pode comportar vários equipa-

mentos em paralelo), cada operação só pode ser processada em um dos equipamentos. Esta característica de produção é dada por:

$$\sum_{j \in E_j} \sum_{k \in J_i} W_{ijk} = 1 \quad \forall i, E$$

onde:

E : índice de estágios de processamento;
 E_j : conjunto de equipamentos em paralelo no estágio E .

Adotando-se o tempo de execução total das tarefas (*makespan*), como função objetivo, tem-se

$$Mk_i \geq \sum_{j \in E_j} \sum_{k \in J_i} W_{ijk} \cdot k + TP_{ij} - 1 \quad \forall i, E = NE$$

onde:

NE : número de estágios de processamento;
 Mk_i : tempo de execução da tarefa i no último equipamento.

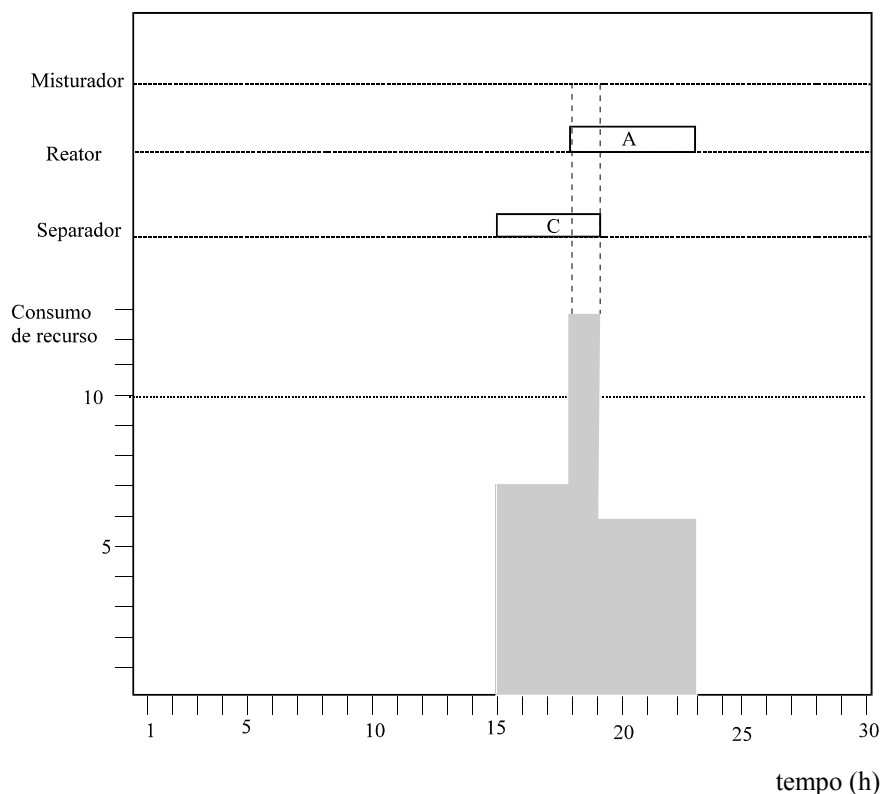


Figura 5 – Programa parcial de produção infactível.

3. Relações Lógicas Envolvendo Recursos Compartilhados

As relações lógicas que envolvem a quantidade de recurso r consumido em um determinado período de tempo são desenvolvidas abaixo.

$$(of_r - QC_r) < CR_{i,j,r} \Rightarrow \neg W_{i,j,k} \\ \forall (i', j') \in P, k, r$$

onde:

$$P = \left\{ (i', j') : CR_{i', j', r} + QC_k > of_r, \forall (i', j') \in NA, k, r \right\}$$

P: conjunto de operações que não podem ser alocadas em k devido à falta de recurso r ;

A: conjunto de operações (i, j) alocadas;

NA: conjunto de operações não alocadas;

W_{ijk} : variável binária de alocação da operação (i, j) no período k ;

Cr_{ijr} : quantidade do recurso r exigida pela operação (i, j) ;

QC_{kr} : quantidade consumida de recurso r no período de tempo k das operações alocadas.

Esta quantidade de consumo é avaliada pelas operações alocadas anteriormente a um determinado ponto da árvore de busca. Isto é possível, pois o OSL permite o acesso a seu processo de busca *branch and bound*.

A restrição lógica dada pela expressão impede que novas operações sejam alocadas durante o desenvolvimento da árvore de *branch and bound* se não houver recursos disponíveis para a execução dessas operações, evitando assim violação de restrições e conseqüentemente nós da árvore de busca inexoráveis. Logo, o sistema de inferência lógica desenvolvido fixa as variáveis binárias (W_{ijk}) em 0 (zero) quando não existe recurso disponível em um determinado período de tempo k , esta análise é feita durante o processo de busca em árvore.

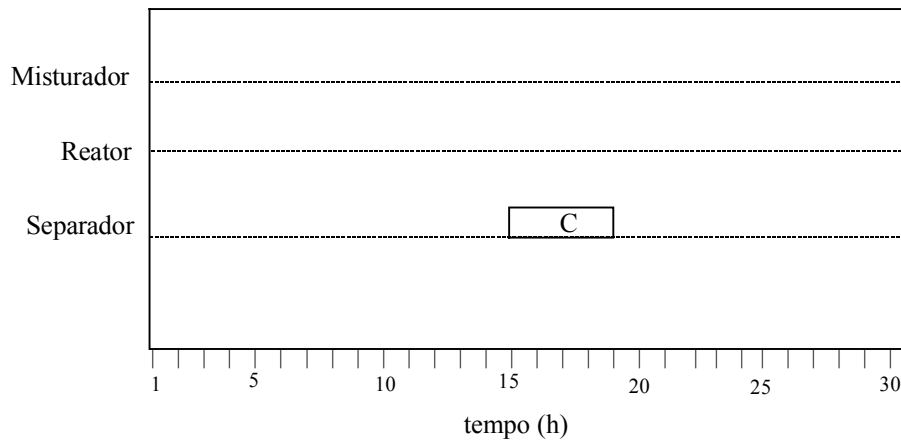


Figura 6 – Situação da estrutura de produção com a alocação da operação (C, Separador) no período 15.

A seguir as expressões lógicas provenientes de toda a análise feita anteriormente envolvendo o compartilhamento de recursos são incorporadas à metodologia *branch and bound*. O OSL é um sistema que permite interferências durante o desenvolvimento da árvore de busca, para tanto são fornecidas as chamadas sub-rotinas *user exit* que possibilitam que o usuário altere seus procedimentos. Dentre as sub-rotinas *user exit* utilizadas neste trabalho estão:

- EKKBRNU: permite que regras de seleção da variável a ser ramificada sejam implementadas pelo usuário.
- EKKCHNU: permite que a regra padrão de escolha do nó do sistema OSL seja modificada.
- EKKEVNU: permite que sejam feitas análises em cada nó da árvore *branch and bound*. Nesta sub-rotina é implementada a expressão lógica que fixa variáveis binárias em 0, quando estas não puderem mais assumir o valor 1, ou seja, quando operações não puderem mais ser alocadas por falta de recursos.

Na identificação de variáveis binárias que quando fixadas em 1 podem gerar nós infactíveis devido a não-satisfação da restrição de limitação de oferta de recurso, dois conjuntos de períodos de tempo que podem levar a esta situação de infactibilidade são estabelecidos. Para estabelecer o primeiro conjunto (conjunto I) de períodos infactíveis considere a situação em que a

operação (C, separador) seja alocada no período 15 e que esta operação necessite de 5 unidades de recursos (u.r).

a) Estabelecimento do conjunto I de variáveis não candidatas à ramificação.

Analisando a Figura 6, nos períodos 15, 16, 17, 18, por exemplo, a operação (A, reator) não pode ser alocada, considerando que esta necessita de 8 u.r, pois a operação (C, separador) encontra-se consumindo 5 u.r. Logo é perfeitamente possível eliminar do processo de busca as variáveis binárias que representam a alocação da operação (A, reator) nos períodos de 15 a 18. Considerando a Tabela 1, por exemplo, nota-se que as variáveis binárias representantes da alocação da tarefa A no reator nos períodos de 15 a 18 apresentam valores fracionários e, portanto são candidatas à ramificação (as variáveis da Tabela 1 podem ser fixadas em 1), isto porque não há como o resolvidor (que neste trabalho corresponde ao OSL) detectar que tal escolha viola oferta de recurso.

Digamos, por exemplo, que a variável $W_{A(2)(18)}$ seja escolhida pelo OSL para gerar um novo nó, isto leva o sistema de produção à situação mostrada na Figura 5.

No período 18 o consumo de recurso extrapola a oferta estipulada, conseqüentemente a

Tabela 1 – Valores das variáveis binárias da operação (A, reator) em um determinado ponto da árvore.

Períodos	Variáveis binárias de alocação	Valores das variáveis
15	$W_{A(2)15}$	0.0
16	$W_{A(2)16}$	0.052
17	$W_{A(2)17}$	0.035
18	$W_{A(2)18}$	0.012

operação (A, reator) não deve ser alocada nos períodos 15, 16, 17 e 18, significando que as variáveis binárias de alocação dadas na Tabela 1 devem ser fixadas em 0, evitando assim a fixação em 1.

Considerando a alocação dada pela Figura 5, em que a operação (C, separador) foi alocada no período 15, a determinação de períodos inactíveis é mostrada a seguir.

$$S1_{i',j'} = \{k, k+1, \dots, k+TP_{ij} - 1\} \quad \forall i' \neq i, j' \neq j$$

onde:

i: tarefa alocada;

i': tarefa não alocada;

j: processador onde i foi alocada;

j': processador conveniente ao processamento da tarefa i';

k: período de alocação;

$S1_{i',j'}$: conjunto de períodos de alocação inactíveis para alocação das operações (i', j');

TP_{ij} : tempo de processamento de i em j.

Considerando o tempo de processamento da operação (C, separador) igual a 4 h, tem-se que o conjunto I de períodos inactíveis é dado por:

$$S1_{A(2)} = \{15, 16, 17, 18\}$$

As variáveis binárias de alocação correspondentes à alocação da operação (A, reator) nestes períodos de tempo podem então ser eliminadas do processo de busca, ou seja, podem ser fixadas em 0.

A Figura 7 mostra a fixação destas variáveis em 0. A mesma análise é feita para as demais operações que tem seu início de processamento

devido à alocação da operação da tarefa C no separador.

b) Estabelecimento do conjunto II de variáveis não candidatas à ramificação.

De acordo com as características de processamento da planta, onde as operações são desenvolvidas sem interrupção uma vez iniciado o seu processamento, pode-se a partir de um período de alocação de alguma operação, determinar um conjunto de períodos onde operações que consomem quantidades de recurso, além da oferta, não podem ser iniciadas. Este conjunto de períodos inactíveis para uma determinada operação (i', j') é dado abaixo.

$$S2_{i',j'} = \{k - TP_{i'j'} + 1, \dots, k - 1\} \quad \forall i' \neq i, j' < j$$

onde:

$S2_{i',j'}$: conjunto de períodos de alocação inactíveis para a operação (i', j');

i: tarefa alocada;

i': tarefa ainda não alocada;

j: processador onde i foi alocada;

j': processador onde i' será alocada;

k: período de alocação;

$TP_{i'j'}$: tempo de processamento da tarefa i' no processador j'.

A Figura 8 mostra as variáveis sendo fixadas a partir do conjunto II. Estas variáveis representam o início da operação (A, reator) em períodos de tempo que podem levar a execução simultânea com a operação (C, separador), a qual se encontra alocada.

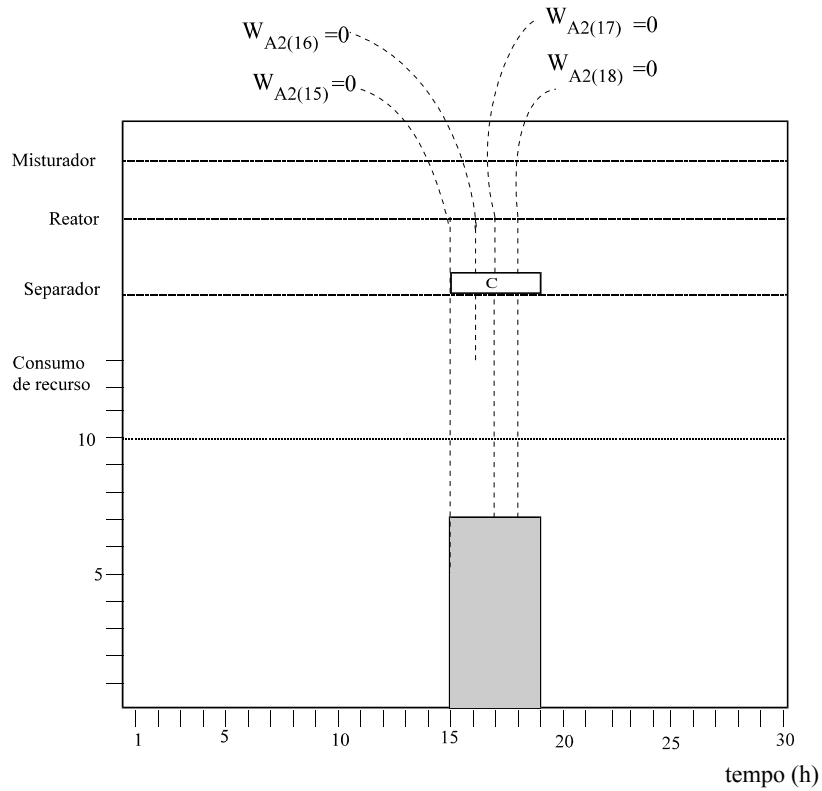


Figura 7 – Fixação de variáveis binárias em 0 pela análise do conjunto $S2_{A1}$.

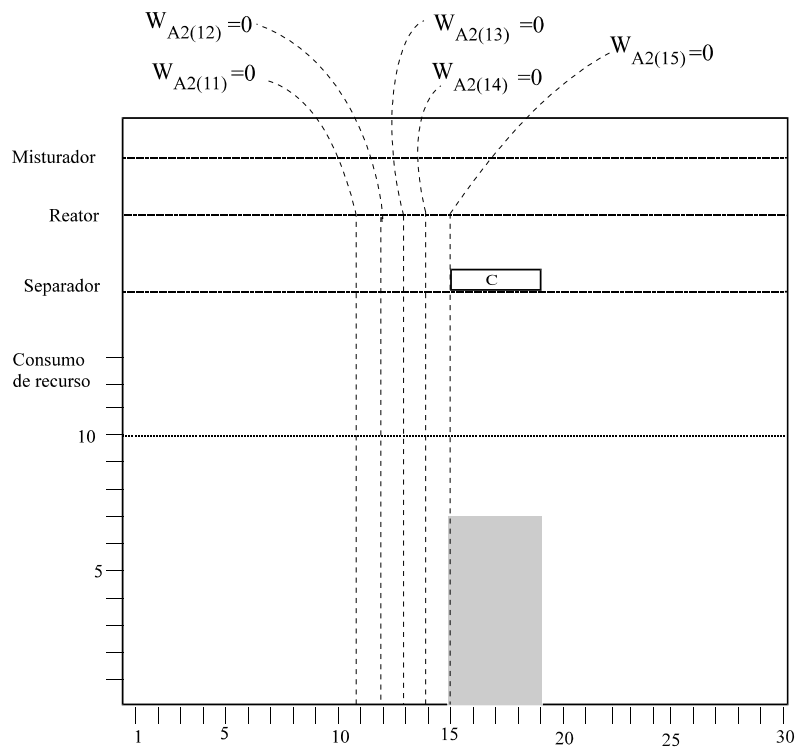


Figura 8 – Fixação de variáveis binárias em 0 pela análise do conjunto $S2_{A2}$.

Tabela 2 – Resultados em função do número de nós gerados.

Problema	GAMS/OSL	OSL com Interferência Lógica
Problema 1		
Tarefas	4	4
Processadores	3	3
Restrições	611	611
Variáveis	876	876
Variáveis Binárias	420	420
Não zeros	5377	5377
Nós	4007	1776
CPU (s)	2239.0	1929,6
Problema 2		
Tarefas	8	8
Processadores	5	5
Restrições	1233	1233
Variáveis	1981	1981
Variáveis Binárias	960	960
Não zeros	8735	8735
Nós	2400	411
CPU (s)	3342.5	1602.5
Problema 3		
Tarefas	8	8
Processadores	5	5
Restrições	1233	1233
Variáveis	1981	1981
Variáveis Binárias	960	960
Não zeros	8735	8735
Nós	19257	10977
CPU (s)	50 000.0	33203.0

(+) SPARC/SOLARIS (*) RISC/6000

4. Discussão de Resultados e Conclusão

Este trabalho tem como principal objetivo mostrar que entre a solução devidamente apresentada, dentro de um critério de satisfação admissível por parte do usuário e o processo que se desenvolve para se alcançar esta solução, existem pontos que podem e devem ser explorados. Este trabalho utiliza a modelagem via Rede-Estado-Tarefa como uma forma fácil de representação das características de processo em sistemas flexíveis, como por exemplo, as que envolvem recursos compartilhados estabelecimento.

No entanto, mesmo apresentando esta facilidade de modelagem o estudo detalhado do processo de solução relativo à árvore de busca

inserta nos pacotes destinados à solução desses modelos, como é o caso do OSL, mostra deficiências oriundas da junção de elementos matemáticos (representados pela função objetivo e pelas restrições que garantem a fenomenologia do problema) com o elemento de inteligência artificial (representado pela metodológica *branch and bound*, que busca satisfazer as restrições de integralidade das variáveis binárias, as quais devem apresentar valores 0 ou 1 na solução). Se tais deficiências não forem devidamente eliminadas do processo de busca, tem-se um aumento excessivo na dificuldade computacional e muitas vezes impossibilita a solução do problema.

Como apresentado na Tabela 2, em todos os problemas analisados, o sistema que utiliza o

conhecimento lógico, gerou menos nós que o GAMS (*General Algebraic Modelling System*). Este pacote foi utilizado para solucionar os problemas sem inferência lógica.

Nota-se que a diminuição do número de nós, quando se utiliza o sistema com inferência

lógica, é significativa e esta diminuição vai depender, por exemplo, da limitação do recurso.

Quando se tem muitas operações que competem pelo mesmo recurso, a possibilidade de fixar variáveis durante a busca será maior se a oferta do mesmo for pequena.

Referências Bibliográficas

BAKER, K.R.: *A Comparative Study of Flow-Shop Algorithms*, 1973.

BROOKE, A.; KENDRICK, D. & MEERAUS, A.: "GAMS – A User's Guide", Release 2.25, *The Scientific Press Series*, Boyd & Fraser Publishing Company, Danvers, Massachusetts, 1988.

HOOVER J.N.: "Logic-Based Methods For Optimization: A Tutorial". *ORSA Computer Science Technical Section Conference*, Williamsburg, VA, USA, January 1994.

KONDILI, E.; PANTELIDES, C.C. & SARGENT, W.H.: "A General Algorithm For Short-Term Scheduling Of Batch Operations – I: MILP Formulation", *Computers Chem. Eng.*, 17, pp.211-227, 1993.

RAMAN, R. & GROSSMANN, I.E.: "Relation Between MILP Modelling and Logical Inference for Chemical Process Synthesis", *Computers Chem. Eng.*, Vol. 15, No. 2, pp.73-84, 1991.

RAMAN, R. & GROSSMANN, I.E.: "Integration of Logic and Heuristic knowledge in MILP Optimization for Process Synthesis", *Computers Chem. Eng.*, Vol. 6, No. 3, pp.155-171, 1992.

SHAH, N.; PANTELIDES, C.C. & SARGENT, W.H.: "A General Algorithm for Short-Term Scheduling of Batch Operations – II: Computational Issues", *Computers Chem. Eng.*, 17, pp.229-244, 1993.

APPLICATION OF LOGICAL INFERENCE PRODUCTION SCHEDULING PROBLEMS

Abstract

Production scheduling of flexible plants has receiving growing interest in the last years because of its economical importance. The main objective is to allocate the shared resources as availability of equipments, utilities and manpower, in order to minimize some performance criteria as, for example, makespan, tardiness etc. That problem is classified as computationally hard to handle, and different approaches proposed in the literature are unable to solve large problems. This paper introduces another approach using a Mixed Integer Linear Problem (MILP) formulation based on the State Task Network (STN) representation and the solution is obtained by using logical inference based on shared resources.

Key words: *chemical systems, batch processes, production scheduling, mathematical programming, logical inference.*