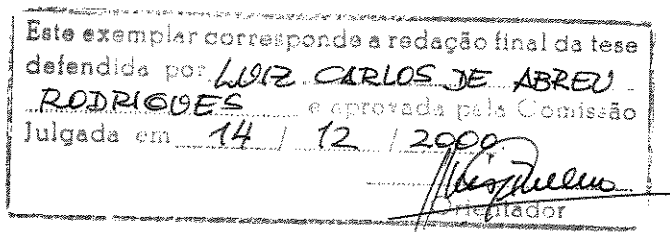


Universidade Estadual de Campinas
Faculdade de Engenharia Elétrica e de Computação
Departamento de Engenharia de Computação e Automação Industrial



Planejamento e programação da produção em plantas multipropósito operando em batelada: Proposta de uma estratégia de decomposição utilizando janelas de tempo

Luiz Carlos de Abreu Rodrigues

Orientador: Prof. Dr. Luis Gimeno Latre

Co-orientadora: Prof. Dr. Maria Teresa Moreira Rodrigues

Banca examinadora:

Prof. Dr. Luis Gimeno Latre. *UNICAMP/FEEC-DCA*

Prof. Dr. Márcio Luiz Andrade Netto. *UNICAMP/FEEC-DCA*

Prof. Dr. Paulo Morelato França. *UNICAMP/FEEC-DENSIS*

Prof. Dr. Luis Puigjaner Corbella. *Universidade Politécnica de Catalunha, Departamento de Engenharia Química*

Prof. Dr. José Maurício Pinto. *Universidade de São Paulo, Departamento de Engenharia Química*

Prof. Dr. Rafael Santos Mendes. *UNICAMP/FEEC-DCA*

Prof. Dr. Roger Josef Zemp. *UNICAMP/FEQ-DESQ*

Tese apresentada à Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas, como parte dos pré-requisitos para obtenção do Título de Doutor em Engenharia Elétrica.

Dezembro de 2000

UNICAMP

BIBLIOTECA CENTRAL
SEÇÃO CIRCULANTE



2000/12

UNIDADE DC
 N.º CHAMADA:
UNICAMP
R.618p
 V. _____ Ex. _____
 TOMBO BC/ 44037
 PROC. 16-892/01
 C ☐ D ☒
 PREÇO R\$ 11,00
 DATA 29/04/01
 N.º CPD _____

CM-00154005-8

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

R618p Rodrigues, Luiz Carlos de Abreu
 Planejamento e programação da produção em plantas
 multipropósito operando em batelada: Proposta de uma
 estratégia de decomposição utilizando janelas de tempo / Luiz
 Carlos de Abreu Rodrigues.--Campinas, SP: [s.n.], 2000.

Orientadores: Luis Gimeno Latre, Maria Teresa Moreira
 Rodrigues.

Tese (doutorado) - Universidade Estadual de Campinas,
 Faculdade de Engenharia Elétrica e de Computação.

1. Planejamento da produção. 2. Otimização matemática. 3.
 Programação inteira. 4. Programação heurística. I. Latre, Luis
 Gimeno. II. Rodrigues, Maria Teresa Moreira. III.
 Universidade Estadual de Campinas. Faculdade de Engenharia
 Elétrica e de Computação. IV. Título.

FINANCIADOR

Agradeço à FAPESP (Fundação de Amparo à Pesquisa do Estado de São Paulo) pela bolsa de doutorado que me foi concedida.

AGRADECIMENTOS

À minha esposa Regina e aos meus filhos Sarah e Gabriel, 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. Luis Gimeno, ao prof. Dr. Luis Puigjaner e à prof. Dra. Maria Teresa Rodrigues, pela orientação e ensino dados.

A todos os (e as) colegas, professores e funcionários da FEQ/UNICAMP e da UPC-Barcelona que sempre me acolheram tão bem.

ÍNDICE

1. Introdução geral	1
1.1. Introdução: Objetivos	1
1.2. Resultados obtidos	3
1.3. Abordagens na literatura para a separação dos problemas de planejamento e <i>scheduling</i> em problemas de curto prazo	4
1.4. Características da solução proposta	10
1.5. Conteúdo	10
2. Sistema de planejamento	11
2.1. Introdução	11
2.2. Estrutura do sistema de planejamento	12
2.3. Representação Rede Estado-Tarefa	13
2.4. Determinação das janelas de tempo para os estados	14
2.4.1. Determinação do número de bateladas a serem produzidas e dos finais das janelas de tempo	14
2.4.2. Determinação dos inícios das janelas de tempo	20
2.4.2.1. Determinação das relações de precedência (<i>pegging</i>) entre bateladas	22
2.4.3. Caso particular: Receita com estados compartilhados	24
2.5. Determinação das janelas de tempo das bateladas das operações	30
2.6. Redução das janelas das bateladas através da propagação das restrições	33
2.6.1. Propagação de restrições disjuntivas	33
2.6.2. Propagação das restrições de armazenagem	39
2.6.3. Propagação por ordenamento de bateladas de uma mesma operação	43
2.6.4. Propagação por balanço de massa	43
2.6.5. Funcionamento conjunto da propagação de restrições	43
2.6.6. Cálculo de crucialidades dos processadores	45
2.6.7. Aspectos da implementação	50
2.7. Exemplos de aplicação	51
2.7.1. Exemplo 1	51
2.7.2. Exemplo 2	57
2.8. Limitação da abordagem em termos da factibilidade das janelas	59
2.9. Conclusões	62
3. Programação da produção usando uma formulação inteira mista (MILP)	65
3.1. Introdução	65
3.2. Modelo MILP para <i>scheduling</i> utilizando janelas de tempo	66
3.3. Uso de subtrens na formulação MILP	72
3.4. Formulação MILP reduzida	77
3.5. Resultados do <i>scheduling</i>	83

3.5.1. Influência da propagação de restrições	83
3.5.2. Utilização do modelo MILP reduzido	86
3.6. Conclusões	95
4. Programação da produção usando <i>simulated annealing</i>	97
4.1. Introdução	97
4.2. Descrição do algoritmo básico do SA	98
4.3. SA com filtragem dos candidatos	99
4.3.1. Detecção de ordenamentos adicionais na fase de planejamento	100
4.3.2. Geração de uma nova solução candidata	101
4.3.3. Alocação e cálculo da função de custo	101
4.3.4. Comentários adicionais	104
4.4. Comparação do SA com e sem a filtragem dos candidatos	105
4.4.1. Caso 1	105
4.4.1.1. Resultados da comparação do SA para as duas abordagens	111
4.4.2. Caso 2	114
4.4.2.1. Resultados da comparação do SA para as duas abordagens	118
4.5. Conclusão	121
5. Conclusões	123
Referências Bibliográficas	125
Anexo 1. Definição dos exemplos estudados	129
Anexo 2. Instâncias do exemplo 2 aplicadas à formulação MILP	133
Anexo 3. Parâmetros usados no SA	141

RESUMO

O problema tratado é o planeamento e programação (*scheduling*) da produção em plantas operando em batelada. Considera-se a situação em que são produzidos diversos produtos finais (planta multiproduto) através de vários estágios de produção (operações), e os processadores são multipropósito, podendo ser utilizados por diversas operações. O objetivo é o estudo de problemas da indústria de processos e, para tanto, são consideradas as restrições de armazenagem típicas desta área. O problema considerado é o chamado de curto prazo, no qual o objetivo do planeamento e *scheduling* da produção é o atendimento de demandas específicas de produtos finais em termos de quantidades e prazos de entrega.

A abordagem proposta é de dois níveis. O nível de planeamento utiliza como dados de entrada a demanda de produtos finais, a disponibilidade de matérias primas e a atribuição de operações a processadores. Através de procedimentos de explosão, determina-se a quantidade de bateladas de cada operação que serão produzidas e a janela de tempo onde deverá ocorrer o processamento de cada batelada. Utilizam-se técnicas de propagação de restrições para analisar o carregamento induzido aos processadores e a factibilidade do plano de produção. O carácter interativo do sistema permite ao usuário modificar os dados de entrada para obter uma situação factível.

O resultado do planeamento, na forma de janelas de tempo, é utilizado no *scheduling* para reduzir a dimensão do problema. Utilizam-se duas abordagens: uma abordagem de programação linear inteira mista (MILP) utilizando discretização uniforme do tempo e a técnica de *simulated annealing*. A formulação MILP utiliza intensivamente a informação dada pelas janelas de tempo, reduzindo a quantidade de variáveis binárias envolvidas e propõe-se uma formulação reduzida que explora as informações de carregamento dos processadores. O algoritmo de *simulated annealing* é acrescido de um processo de filtragem dos candidatos que elimina os candidatos infactíveis, dadas as restrições de ordenamento induzidas pelas janelas de tempo.

ABSTRACT

The problem focused in this thesis is the short term planning and scheduling of multipurpose batch plants. This is an important problem in the process industry and, because of that, intermediate storage limitations are explicitly considered. The main objective is to fulfill specific demands of final products, distributed along the horizon.

It is proposed a two level approach: planning and scheduling. Product's demands, raw material availability plan and operation/equipment assignment are the inputs to the planning level. At this level an exploding procedure is performed in order to determine the number of batches of each operation as well as their processing time windows. Constraint propagation techniques are used to analyze the plant loading and the production plan feasibility. The interactive nature of the proposed approach allows the user to change input data in order to define a feasible scenario.

At the end of the planning level, a set of operations' time windows is released to the scheduling level. Two approaches have been implemented to schedule the operations inside their time windows: a MILP approach based on a uniform time discretization and a simulated annealing approach.

The information given by the time windows is intensively used in the MILP formulation, reducing the number of binary variables in the problem. It is also proposed a reduced MILP model exploiting plant loading information. The simulated annealing algorithm implemented also uses the time windows information to eliminate infeasible candidates through a filtering procedure.

CAPÍTULO 1. INTRODUÇÃO

1.1. Introdução: Objetivos

Esta Tese de Doutorado foi realizada dentro da linha de pesquisa em Planejamento e Programação da Produção (*scheduling*) na indústria de processos, em desenvolvimento na Faculdade de Engenharia Elétrica e de Computação e na Faculdade de Engenharia Química. Os objetivos desta linha de pesquisa são o planejamento e *scheduling* de plantas multipropósito/multiproduto.

No início deste trabalho, em 1997, um dos mais recentes resultados obtidos na citada linha de pesquisa era o desenvolvimento de uma abordagem de horizonte rolante para o problema de *scheduling* (Rodrigues *et al.*, 1996). O problema considerado era o *scheduling* de curto prazo (*short-term*) de plantas multipropósito operando em batelada com restrições de armazenagem e de recursos compartilhados. A solução, através de uma técnica de horizonte rolante, utilizava como dados de entrada as janelas de tempo das bateladas de operações, que identificam o intervalo de processamento destas bateladas. Estas janelas de tempo permitiam a utilização de uma função de previsão (*lookahead*) de gargalos em um horizonte futuro para dirigir a alocação (*scheduling*) de bateladas no horizonte de alocação imediato. Tanto a quantidade de bateladas de cada operação a serem produzidas quanto as respectivas janelas de tempo eram dados de entrada, que deveriam, no futuro, serem saídas de um sistema de planejamento.

Em 1997 o sistema de planejamento estava no início de seu desenvolvimento. Os objetivos para a Tese de Doutorado, incluindo redefinições durante estes quatro anos, foram:

- Participação no desenvolvimento do sistema de planejamento que deveria determinar as janelas de tempo para a fase de *scheduling*;
- Desenvolvimento de formulações de programação linear inteira mista (*Mixed Integer Linear Programming - MILP*) para o problema de *scheduling* utilizando as janelas de tempo;
- Desenvolvimento de metaheurísticas (*simulated annealing*) para o problema de *scheduling* utilizando as janelas de tempo.

Estes objetivos se inseriam no desenvolvimento de um sistema de planejamento e *scheduling* em dois níveis. O nível de planejamento teria a função de: *i*) determinar, a partir de um plano de demanda específico sobre os produtos finais, as necessidades de produtos intermediários e matérias primas em termos de quantidades e prazos de entrega e *ii*) determinar, a partir dos dados de liberação dos processadores, do início do horizonte de planejamento e do plano de disponibilidade de matérias primas, as datas de início mais cedo para a produção dos produtos intermediários. O sistema de planejamento obteria, desta

forma, as janelas de tempo para o nível de *scheduling*. As características básicas do sistema seriam:

- Utilização e desenvolvimento de técnicas que permitissem a análise de capacidade (disputa pela ocupação) dos equipamentos na fase de planejamento a partir das janelas de tempo e isto sem definir a alocação no tempo das bateladas;
- Utilização das janelas de tempo geradas pelo sistema de planejamento para reduzir a dimensão do problema de *scheduling* nas abordagens de programação matemática (MILP) e de técnicas evolutivas (*simulated annealing*).

A análise de capacidade dos equipamentos na fase de planejamento, sem a alocação das bateladas, visava propor uma solução inovadora ao problema principal da abordagem em dois níveis: a necessidade de iterar entre os níveis de planejamento e *scheduling*, dado que os problemas de capacidade são apenas detectados no *scheduling*. A utilização de janelas de tempo na fase de *scheduling* já tinha mostrado ser promissora na técnica de *scheduling* utilizando uma estratégia de horizonte rolante citada anteriormente.

A utilização de janelas de tempo no problema de *scheduling* é uma das características dos sistemas de *scheduling* originados na área de Busca Orientada por Restrições (*Constraint Based Search*) como, por exemplo, nos sistemas Isis (Fox, 1983), Opis (Smith, 1994), Cortes (Fox e Sycara, 1990), MicroBoss (Sadeh, 1991), ILOG Scheduler (ILOG, 1997) e Claire Schedule 1.0 (1999). Esta abordagem tem mostrado bons resultados para problemas de *scheduling* bastante restritos. Todos estes sistemas porém supõem que as janelas de tempo são dados de entrada, sendo necessário um sistema de planejamento prévio.

A obtenção de janelas de tempo na fase de planejamento é uma característica inerente aos sistemas de planejamento MRP (*Manufacturing Resources Planning*) (Orlicky, 1975), que partem de dados de demanda e determinam datas de início para o processamento dos lotes e datas de entrega dos mesmos. A obtenção destas janelas está baseada na utilização de procedimentos de explosão da demanda, para determinar os requisitos sobre os produtos intermediários, e na utilização de *lead times* (dado de entrada do problema definindo o tempo decorrido desde a liberação de uma ordem de produção até a sua conclusão). Estes procedimentos de explosão seguem existindo nos softwares comerciais mais recentes de planejamento e *scheduling*, os denominados APS (*Advanced Planning and Scheduling*) (i2, 1996), como uma forma adequada de determinar os requisitos de produção para atender à demanda em processos de fabricação com múltiplos estágios. Já a utilização de *lead times* grandes, como uma forma de prever antecipadamente os problemas de compartilhamento de recursos, tem sido muito criticada, especialmente dentro do contexto atual de diminuição de prazos de entrega e de maior previsibilidade sobre a operação do chão de fábrica. Os sistemas APS incluem funções de *scheduling*, justamente para resolver o problema de capacidade juntamente com o problema de planejamento.

1.2. Resultados obtidos

- *Planejamento*

O sistema de planejamento atual obtém as janelas de tempo a partir de dois procedimentos de explosão: *i*) para trás (*backward*) a partir das demandas de produtos finais e *ii*) para frente (*forward*) a partir da data de início de planejamento, da ocupação dos equipamentos e da disponibilidade das matérias primas. Para tanto utiliza, como descrição da estrutura de fabricação dos produtos finais, a Rede Estado-Tarefa (State-Task Network) (Kondili *et al.*, 1993), análoga à árvore de produção (*bill of materials*) utilizada em sistemas MRP.

Utilizam-se intensivamente as técnicas de propagação das restrições impostas pelo compartilhamento de equipamentos descritas na literatura de busca orientada por restrições (Caseau e Laburthe, 1995). A propagação das restrições impostas pelos balanços de massa é feita a partir das relações de precedência obtidas nos procedimentos de explosão, não sendo portanto dados de entrada, como é o caso nos sistemas de *scheduling* baseados na busca orientada por restrições. O sistema de planejamento proposto inclui a propagação de restrições originadas por condições de armazenagem limitada, o que não é considerado nos sistemas baseados na busca orientada por restrições.

Propõe-se a utilização de um procedimento de filtragem na ativação dos mecanismos de propagação de restrições, que sabidamente podem ter um custo computacional alto. Este procedimento utiliza o conceito de demanda agregada em um processador, induzida pelas janelas de tempo das bateladas a ele atribuídas (Sadeh, 1991).

As contribuições principais minhas nesta área, além da participação na equipe que desenvolveu o sistema, foram no procedimento de explosão *forward* quando a estrutura de produção envolve operações com múltiplos intermediários de saída (consumo de produtos intermediários compartilhados) e na propagação de restrições originadas pelas condições de armazenagem limitada.

- *Scheduling*

O sistema atual permite três abordagens para o problema de *scheduling*: uma formulação MILP com discretização uniforme do tempo, *simulated annealing* e busca em árvore através de decisões orientadas pelas restrições. As minhas atividades foram concentradas no desenvolvimento das duas primeiras abordagens, através do desenvolvimento e implementação de duas estratégias.

A formulação MILP utiliza intensivamente as informações contidas nas janelas de tempo, permitindo uma redução sensível na dimensão do problema. A disponibilidade da informação sobre o carregamento dos processadores permitiu à proposta de uma formulação MILP reduzida que elimina do modelo matemático os processadores com carregamento baixo, reduzindo a degenerescência na solução do problema.

A técnica de *simulated annealing* foi implementada com um procedimento de filtragem que reduz drasticamente a quantidade de candidatos ineficazes gerados. Este é um problema bem conhecido na utilização de técnicas evolutivas em problemas de *scheduling* com restrições fortes (Löhl *et al.*, 1998), (Mockus e Reklaitis, 1999b). O procedimento de filtragem utiliza uma representação em grafo para analisar a factibilidade dos ordenamentos da solução candidata.

1.3. Abordagens na literatura para a separação dos problemas de planeamento e *scheduling* em problemas de curto prazo

Os problemas de produção de curto prazo (*short-term*) correspondem a uma operação da planta determinada pelos pedidos (*make-to-order*), em contraposição à operação em campanha onde os requisitos de produção são ditados pela reposição de estoques (*make-to-stock*). Neste último caso tem-se interesse em uma operação repetitiva ou cíclica da planta (campanhas de produção), já no primeiro a operação da planta depende totalmente da demanda dos produtos e do *mix* de produção.

Na operação de curto prazo, o problema pode ser definido com sendo o de atender a uma demanda específica, em termos de quantidades e prazos, determinando: *i*) a atribuição de equipamentos a operações, *ii*) os tamanhos de batelada, *iii*) as bateladas necessárias de cada operação e *iv*) a alocação no tempo destas bateladas. O problema estará resolvido quando for obtida uma alocação no tempo satisfatória. O problema é, então, um problema de *scheduling* e a separação em planeamento e *scheduling* deve ser vista apenas como uma forma de decompor o problema, forma esta que obviamente tem o interesse de reproduzir, na técnica de solução, a estrutura organizacional das empresas.

As técnicas utilizadas na indústria tem sempre refletido esta divisão e o conceito do MRP é um exemplo típico. As abordagens académicas tem também investido nesta separação como uma forma de atacar os problemas da dimensão. Basicamente as abordagens académicas tem sido através de programação matemática e através de procedimentos heurísticos.

□ Programação matemática

- Formulações monolíticas**

São bem conhecidos os problemas com a formulação única do problema através de programação matemática, as chamadas formulações monolíticas. A abordagem MILP de Kondili *et al.* (1993) utiliza uma discretização uniforme do tempo. As variáveis envolvidas

na formulação são fundamentalmente: i) $B(i,j,t)$: o tamanho de batelada de cada batelada de cada operação i alocada em um equipamento habilitado j e iniciando no tempo discreto t , ii) $S(s,t)$: o estoque de cada produto intermediário s em cada instante t e iii) $W(i,j,t)$: a variável binária indicando o início de processamento. Para problemas de dimensão média ou grande, a formulação MILP resultante terá uma dimensão excessivamente grande para os recursos computacionais atualmente disponíveis. Por outro lado, esta abordagem leva a uma formulação linear inteira mista (MILP) apenas se o tempo de processamento para cada operação não é dependente do tamanho da batelada. A quantificação do tempo de processamento em função do tamanho de batelada aumenta ainda mais a dimensão do problema.

Para reduzir o problema da dimensão, diversos autores têm utilizado uma representação “contínua” do tempo (Pinto e Grossmann, 1995), (Mockus e Reklaitis, 1997 e 1999a). Nestas formulações o horizonte de tempo é dividido em intervalos de duração variável, sendo que estas durações são variáveis do problema. Os inícios de processamento das operações, chamados eventos, só podem acontecer nos inícios dos intervalos. A duração de cada intervalo é então função da operação a ele associada. A equação que representa esta associação é não linear porque envolve o produto da duração do intervalo pela variável binária representando a alocação da operação ao evento. Os termos bilineares resultantes no modelo podem ser linearizados mas exigem a utilização de dicotomias com constante grandes, o que como é bem conhecido leva a aumentar o *gap* de integralidade do problema (diferença entre a solução relaxada e a solução ótima do problema).

Por outro lado, é importante ressaltar que em todos estes trabalhos o objetivo é maximizar o resultado econômico, ou seja, a produção de produtos finais devidamente ponderados. Para que esta função objetivo seja usada, supõe-se que a inclusão de demandas fixas distribuídas ao longo do horizonte de planejamento não leva a situações infactíveis. Esta suposição não é adotada para outras funções objetivo.

- ***Formulações de programação matemática com separação do planejamento e do scheduling.***

Basset *et al.* (1996), Subrahmanyam *et al.* (1996) e Birewar e Grossmann (1990) propõem a decomposição do problema, utilizando no planejamento intervalos de tempo agregados, similares ao conceito de *time-buckets* do MRP. O objetivo desta fase é determinar quais operações serão realizadas em cada intervalo de tempo agregado em cada processador. A ocupação dos processadores é restringida impondo que, em um intervalo de tempo agregado de um processador, a soma dos tempos de processamento das operações deve ser menor ou igual à capacidade disponível (tempo disponível no intervalo agregado) deste processador.

O modelo utilizado no nível de planejamento é constituído basicamente de três equações:

- *Balanço de massa nos intervalos agregados*

$$S_{s,\tau} = S_{s,\tau-1} + \sum_{i \in out_s} \sum_{j \in hab_i} \rho_{is}^p B_{ij\tau} - \sum_{i \in in_s} \sum_{j \in hab_i} \rho_{is}^c B_{ij\tau} \quad \forall s, \tau \quad [1.1]$$

onde:

- τ Intervalo de tempo agregado.
- s Produtos intermediários ou finais.
- i Operações.
- j Processadores.
- hab_i Processadores habilitados para a operação i .
- out_i Operações que produzem o produto (intermediário ou final) s .
- in_i Operações que consomem o produto intermediário s .
- $S_{s,\tau}$ Estoque do produto (intermediário ou final) s no fim do intervalo τ .
- ρ_{is}^p Fator de produção do produto (intermediário ou final) s pela operação i .
- ρ_{is}^c Fator de consumo do produto (intermediário ou final) s pela operação i .
- $B_{ij\tau}$ Tamanho de batelada da operação i no processador j no intervalo agregado τ .

Esta equação pode incluir aportes ou retiradas externas do produto s no intervalo agregado τ .

- *Capacidade nos intervalos agregados*

Esta equação restringe o tempo disponível do processador j no intervalo agregado τ .

$$\sum_{i \in hab_j} tp_{ij} * y_{ij\tau} \leq H_\tau \quad \forall j, \tau \quad [1.2]$$

onde:

- tp_{ij} Tempo de processamento da operação i no processador j .
- $y_{ij\tau}$ = 1, se a operação i é realizada no processador j no intervalo agregado τ . Caso contrário, $y_{ij\tau} = 0$.
- H_τ Duração de cada intervalo agregado τ .

- *Quantidade processada em um processador nos intervalos agregados*

Esta equação limita a quantidade da operação i realizada no processador j no intervalo agregado τ .

$$B_{ij\tau} \leq m_{ij} * y_{ij\tau} \quad \forall i, j, \tau \quad [1.3]$$

onde:

- m_{ij} Tamanho máximo de batelada para a operação i no processador j .

A função objetivo é a maximização do resultado econômico sobre todo o horizonte de planejamento. A agregação não considera as implicações temporais das restrições de

precedência da receita na equação de balanço de massa quando formula que a produção e o consumo podem acontecer no mesmo intervalo agregado. De fato, o início de uma operação produtora, se a massa produzida for necessária para a operação consumidora, deve preceder o início da operação consumidora de um tempo igual ao tempo de processamento da operação produtora. Estas restrições temporais levam a que a ocupação dos processadores tenha mais restrições que as representadas pela equação 1.2. O resultado é que no nível de *scheduling*, quando se alocam as operações de cada intervalo agregado, a situação pode ser infactível porque o tempo disponível não é suficiente.

Estas abordagens (Basset *et al.*, 1996; Subrahmanyam *et al.*, 1996; Birewar e Grossmann, 1990) diferenciam-se pela forma como tratam a questão da capacidade disponível em um processador em um intervalo de tempo agregado.

Basset *et al.* (1996) propõem três mecanismos para obter uma situação factível: *i*) se a situação é infactível em um intervalo mas é factível em um intervalo imediato, o tempo disponível no segundo intervalo agregado é utilizado alocando-se algumas das operações atribuídas ao primeiro intervalo agregado; *ii*) o planeamento pode dar lugar a algumas bateladas da mesma operação com tamanhos de batelada inferiores ao tamanho máximo de batelada desta operação. Neste caso a junção de bateladas pode ser utilizada para diminuir o tempo de ocupação necessário no processador. Os dois mecanismos têm um alcance limitado e dependem de cada caso particular, razão pela qual incluiu-se um terceiro mecanismo de eliminação das infactibilidades: *iii*) Caso estes procedimentos não sejam suficientes, propõe-se voltar ao nível de planeamento introduzindo-se termos adequados ao lado esquerdo da equação 1.2. Esta equação é substituída pela equação 1.4, que induz uma folga $\Delta_{j\tau}$ no processador j no intervalo agregado τ quando o nível de *scheduling* detecta uma infactibilidade.

$$\Delta_{j\tau} + \sum_{i \mid j \in hab_i} tp_{ij} * y_{ij\tau} \leq H_{\tau} \quad \forall j, \tau \quad [1.4]$$

Os tempos ociosos necessários na fase de *scheduling* podem ser determinados *a priori* em alguns casos particulares. Em (Birewar e Grossmann, 1990), por exemplo, os autores analisam o caso de planta multiproduto (*flow shop*) sem processadores em paralelo e sem limitações de armazenagem. Neste caso, as folgas necessárias podem ser calculadas porque dependem apenas da ordem em que as operações serão processadas. A formulação proposta por estes autores visa maximizar a produção sujeita a níveis de produção mínimos para os produtos e não se aplica ao problema de interesse que é de curto prazo com demandas específicas.

Subrahmanyam *et al.* (1996) propõem também a utilização de um modelo de planeamento considerando intervalos de tempo agregados, mas as eventuais infactibilidades detectadas no nível de *scheduling* são tratadas “desagregando” os intervalos de tempo onde foram detectadas as infactibilidades. De forma resumida o procedimento tem os seguintes passos:

- 1) Resolve o problema agregado obtendo um valor v para a função de custo (supõe-se um procedimento de maximização);
- 2) Testa-se a factibilidade resolvendo o problema de *scheduling* em cada intervalo agregado;
- 3) Se não é detectada qualquer infactibilidade, o problema é encerrado;
- 4) Se um intervalo agregado τ é infactível, resolve-se o problema de planeamento com o modelo agregado para todos os intervalos agregados exceto para o intervalo τ e o modelo desagregado de planeamento e *scheduling* para este último intervalo (este modelo é análogo ao modelo proposto por Kondili *et al.* (1993) (utilizado no capítulo 3). O valor obtido para a função de custo é v_τ ;
- 5) Se $v_\tau < v$, volta-se ao passo 1 incluindo a restrição $v < v_\tau$;
- 6) Se o valor de v_τ não é menor que v , isto significa que a diminuição de v devido à factibilização do intervalo agregado τ foi compensada pelo seu aumento em outros intervalos agregados, o que provavelmente levará a infactibilidades;
- 7) Testa-se a factibilidade dos outros intervalos agregados;
- 8) Se todos intervalos agregados são factíveis, o problema está encerrado;
- 9) Se algum intervalo agregado τ' é infactível, segue-se um procedimento análogo ao passo 4 considerando agora o modelo desagregado para os intervalos τ e τ' .

Como assinalam os próprios autores, o procedimento pode levar a ter que resolver o problema desagregado em todos os intervalos, levando a uma formulação monolítica do tipo citado anteriormente.

É interessante notar que as três abordagens resolvem o problema de uma forma iterativa através da inclusão de folgas na fase de planeamento. A ideia é semelhante ao conceito de *lead time* em sistemas MRP, que através do seu aumento, em relação ao valor mínimo dado pelo tempo de processamento e preparação, inclui uma folga para que a fase de *scheduling* no chão de fábrica possa acomodar os problemas de capacidade.

□ **Procedimentos heurísticos**

Diversos procedimentos heurísticos têm sido propostos para integrar de forma simplificada as restrições de *scheduling* no nível de planeamento, sendo que o objetivo sempre é o de evitar um resultado do planeamento infactível. Provavelmente o início do desenvolvimento destas técnicas foi a introdução do MRP (*Manufacturing Resources Planning*) (Orlicky, 1975) devido à sua hipótese básica de capacidade infinita. Atualmente esta linha tem levado aos softwares comerciais APS (*Advanced Planning and Scheduling*) (i2, 1996) (SAP-APO, 1999) (Preactor, 2000). A característica básica destes sistemas, no que se refere ao problema de capacidade, é a utilização de técnicas heurísticas rápidas de *scheduling* que permitem detectar e corrigir problemas de capacidade nos intervalos agregados com que trabalha o MRP (*time buckets*). A correção dos problemas de capacidade é feita através de ajustes do tipo discutido na seção anterior: deslocamento das

necessidades de produção para intervalos (*time buckets*) vizinhos, podendo levar a antecipações nos inícios de produção, agregação das bateladas e, em último caso, o aumento dos prazos de entrega.

Não cabe aqui uma revisão aprofundada desta área, apenas uma proposta recente será discutida. As referências (Shapiro, 1993) e (Morton e Pentico, 1993) contém diferentes contribuições nesta área.

Em geral, as técnicas heurísticas para a solução do problema de planeamento e *scheduling* atacam o problema através de um sequência de passos com procedimentos de iteração. Em (Mockus e Reklaitis, 1999b) os autores justificam este procedimento como a abordagem natural em processos de decisão. Segundo os autores: “A maioria dos processos de decisão consistem em um certo número de passos. Por exemplo, no *scheduling* de operações em batelada estes passos são selecionar uma operação, selecionar um equipamento habilitado a processar a operação e determinar a quantidade de material a ser processada pela operação. Em cada passo, um objeto é selecionado dentro do conjunto de decisões (por exemplo, selecionar uma operação dentro de um conjunto dado de operações que produzem o produto intermediário ou final necessário). Uma heurística é um conjunto de regras para executar um passo.”

O procedimento heurístico proposto pelos autores é uma repetição aleatória de um conjunto definido de regras (*Bayesian Heuristic Approach*), a aleatoriedade permitindo que as decisões tomadas em cada passo sejam diferentes a cada repetição. Se a função objetivo não melhora após 10 iterações o procedimento é interrompido. Os passos são, resumidamente, os seguintes:

1. Determinar o valor da variável aleatória que será usada para gerar diferentes soluções;
2. Alocar as operações através de:
 - 2a. Escolher o produto final com data de entrega mais cedo;
 - 2b. Para a operação que gera este produto final, escolher o tamanho de batelada e o equipamento habilitado que esteja disponível no intervalo mais próximo da data de entrega;
 - 2c. Alocar a operação de forma a terminar o processamento no fim do intervalo, visando a atualização da ocupação dos equipamentos;
 - 2d. Introduzir na lista as operações necessárias para produzir os intermediários consumidos pela operação alocada;
3. Calcular a função objetivo utilizando as alocações preliminares;
4. Se a função objetivo não melhorou nas 10 últimas iterações, ir para o passo 6
5. Ir para o passo 1
6. Utilizar a formulação por eventos, anteriormente citada (Mockus e Reklaitis, 1999a), fixando as variáveis binárias representando os inícios e finais do processamento das operações i nos equipamentos j nos inícios e fins dos eventos o . A formulação não linear inteira mista (MINLP) se transforma em um problema linear (LP) que determina os instantes de início de alocação das operações e os instantes de recebimento das matérias primas necessárias para ter uma solução factível.

1.4. Características da solução proposta

O sistema de planeamento descrito nesta tese propõe um procedimento de análise dos problemas de capacidade sem exigir a alocação das operações, ou seja, sem entrar na fase de *scheduling*. Para isto, utilizam-se e se ampliam os mecanismos utilizados na área de busca orientada por restrições. Estes mecanismos permitem analisar o carregamento de cada processador induzido pelas janelas de tempo das operações a ele atribuídas. A análise de capacidade não é completa (seção 2.8) mas permite detectar a maior parte das situações infactíveis.

Deste modo, as janelas de tempo, obtidas na fase de planeamento, constituem uma interface adequada entre planeamento e *scheduling*, permitindo, além da análise de capacidade, a visualização pelo usuário do carregamento da planta, e a sua modificação em situações julgadas não convenientes. Por outro lado, em qualquer situação real, o usuário desejará ver satisfeitas restrições diversas que, dificilmente, poderão ser incorporadas ao sistema de planeamento e *scheduling*. Qualquer sistema deve permitir uma interação eficiente do usuário, como por exemplo nos sistemas APS, e as janelas de tempo são um instrumento adequado para este fim.

1.5. Conteúdo

Esta Tese está organizada em cinco capítulos. O Capítulo 2 descreve o sistema de planeamento e os Capítulos 3 e 4 descrevem as abordagens MILP e de *simulated annealing* para o problema de *scheduling*. As conclusões e propostas para trabalhos futuros formam o Capítulo 5. Dois exemplos são utilizados continuamente na Tese, e suas descrições encontram-se no Anexo I.

CAPÍTULO 2. SISTEMA DE PLANEJAMENTO

2.1. Introdução

O objetivo do sistema de planejamento proposto é determinar o número de bateladas de cada operação que deverão ser produzidas, assim como os intervalos, chamados de janelas de tempo, em que estas bateladas deverão ser executadas. A determinação das janelas de tempo das bateladas leva em consideração as disponibilidades de matérias primas e as demandas de produtos finais. Para que as janelas de tempo possam ser calculadas é preciso definir uma atribuição única das operações aos processadores e um tamanho fixo de batelada para cada operação.

O sistema de planejamento realiza a análise do compartilhamento dos processadores, permitindo a definição de um carregamento razoável da planta através da interação com o usuário. A interação do usuário ocorre através da alteração dos dados de entrada do problema: o plano de disponibilidade de matérias primas no tempo, a demanda de produtos finais, a atribuição de processadores a operações e a disponibilidade dos processadores. A análise do compartilhamento dos processadores durante o planejamento utiliza técnicas de propagação de restrições que podem levar a reduções nas janelas de tempo das bateladas. Em situações de carregamento alto, estas reduções podem levar a situações infactíveis quando o tempo de processamento é maior que a janela de tempo remanescente.

A interação do usuário com o planejamento, para gerar e avaliar diversos cenários de produção, é comum na prática industrial pois, em geral, os dados do problemas normalmente não são totalmente conhecidos com antecedência, como apontam Henning e Cerdá (1996). Alterações na demanda de produtos finais, na disponibilidade de matérias primas e na manutenção dos processadores podem afetar as ordens de produção e, consequentemente, o cenário de produção. O usuário responsável pelo planejamento de produção deve sempre avaliar o carregamento da planta imposto pela demanda dos produtos finais, tentando impedir quaisquer atrasos na entrega de pedidos dos clientes e mantendo folgas adequadas de forma a tornar o planejamento de produção robusto frente às perturbações que possam ocorrer durante a operação da planta. Se o cenário gerado é infactível, o sistema de planejamento proposto gera informações sobre o carregamento dos processadores e sobre as janelas de tempo insuficientes. Estas informações permitem que o usuário gere novos cenários de produção a partir do diagnóstico dos problemas encontrados.

A estrutura geral do sistema de planejamento é apresentada na seção 2.2. A representação das receitas de produção da planta é feita através da representação Rede Estado-Tarefa (STN) que é apresentada na seção 2.3. O procedimento de cálculo do número de bateladas de cada operação e das janelas de tempo destas bateladas é apresentado na seção 2.4. O procedimento de modificação das janelas de tempo decorrentes da análise do carregamento dos processadores, da propagação das restrições de armazenagem e da receita da planta (balanço de massa) é apresentado na seção 2.6.

2.2. Estrutura do sistema de planejamento (Rodrigues *et al.*, 2000a)

O planejamento é iniciado com a definição dos dados do problema pelo usuário. A figura 2.1 mostra os dados de entrada (e os mecanismos de interação do usuário) do planejamento: *i*) atribuição de operações a processadores, *ii*) definição da demanda dos produtos finais e *iii*) definição da disponibilidade de matérias primas.

Após a definição dos dados do problema, determina-se o número de bateladas de cada operação que deverá ser produzido e o prazo de conclusão (final das janelas de tempo), ou seja, o instante de término mais tarde (*lft-latest finishing time*) de cada uma das bateladas. Isto é feito através de um procedimento de explosão (propagação da demanda de produtos finais) similar ao utilizado em sistemas MRP, porém, o sistema proposto se baseia no uso de tempos de processamento no lugar dos *lead times*. Assim, o final das janelas de tempo é dado pelo instante mais tarde em que é possível concluir o processamento das bateladas de tal forma que as demandas de produtos finais não sejam comprometidas. O último passo desta fase determina o prazo mais tarde para a disponibilidade das matérias primas.

A introdução de um plano de disponibilidade de matérias primas por parte do usuário permite que se obtenha o início das janelas de tempo das bateladas, ou seja, o instante de início mais cedo (*ebt - earliest beginning time*), através de um procedimento de explosão para frente. Este procedimento encerra a determinação inicial das janelas de tempo.

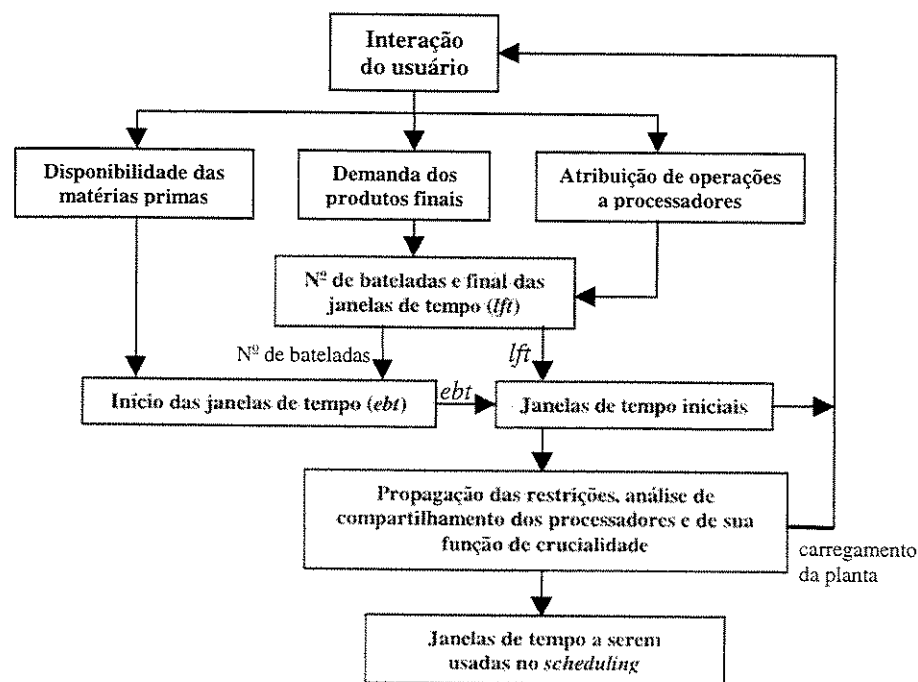


Figura 3.1. Estrutura do sistema de planejamento

A propagação das restrições devido ao balanço de massa e às limitações de armazenagem, e a análise do compartilhamento dos processadores são utilizadas para determinar reduções obrigatórias nas janelas de tempo, resultando em uma indicação mais realista do intervalo em que é possível executar as bateladas. A análise do compartilhamento dos processadores é feita utilizando ferramentas da área de busca orientada por restrições, especificamente, usando os conceitos de intervalos de ocupação obrigatória (*intervals of total reliance* [Sadeh, 1991]) e identificando reduções nas janelas de tempo resultantes do ordenamento entre bateladas induzido pelas suas janelas de tempo (Caseau e Laburthe, 1995; Baptiste e Le Pape, 1995). As reduções nas janelas de tempo originadas pela análise do compartilhamento dos processadores são propagadas de tal forma que tanto o balanço de massa como as restrições de armazenagem sejam satisfeitas pelas janelas de tempo. As funções de crucialidade dos processadores são calculadas a partir das janelas de tempo, dando uma indicação do carregamento em cada instante do horizonte de planejamento para todos os processadores. Através desta informação o usuário pode identificar os gargalos da planta e também saber o intervalo do horizonte de planejamento em que ocorrem estes gargalos.

2.3. Representação Rede Estado-Tarefa

A Representação Rede Estado-Tarefa (STN - *State Task Network*), proposta por Kondili *et al.* (1993), foi desenvolvida originalmente para descrever processos químicos complexos, mas a sua forma clara de representar processos produtivos, permite que ela seja usada na indústria em geral, tanto em processos contínuos, semi-contínuos ou batelada. Nesta representação em forma de grafo existem dois tipos de nós: *i*) os nós Estado (*State*) representam as matérias primas, os produtos intermediários e os produtos finais, e *ii*) os nós Tarefa (*Task*) representam as operações de processamento, as quais transformam um ou mais estados consumidos (que alimentam a operação) em estados gerados (que são produzidos pela operação). Os nós Estado são representados por círculos e os nós Tarefa por retângulos. Esta representação permite que não haja ambigüidades na representação dos processos produtivos, como pode acontecer quando apenas se representam as operações de processamento. As unidades de processamento e as suas conectividades não são mostradas explicitamente neste tipo de representação, assim como nenhum outro recurso disponível é mostrado. As regras para a construção de uma representação STN são:

- Uma tarefa tem a quantidade de estados de entrada (saída) igual ao número de materiais diferentes consumidos (gerados) pela tarefa.
- Duas ou mais entradas (saídas) em um estado são necessariamente de materiais iguais.

Na representação STN assume-se que uma operação consome material dos estados de entrada em proporções fixas e conhecidas, e que esta operação produz material para os estados de saída também em proporções fixas e conhecidas. O tempo de processamento de

cada operação é considerado independente da quantidade de material a ser processado e é previamente conhecido; caso contrário, são criadas diferentes instâncias da operação. Estados (produtos) diferentes gerados por uma mesma operação podem ter tempos de processamento diferentes.

Os estados podem ser associados a quatro restrições de armazenagem diferentes:

1. **UIS (*Unlimited Intermediate Storage*)** - O estado pode ser armazenado em um local de estoque com capacidade infinita, permitindo um estoque ilimitado do estado.
2. **FIS (*Finite Intermediate Storage*)** - O local de armazenamento deste estado tem capacidade limitada.
3. **NIS (*No Intermediate Storage*)** - Para o estado não há um tanque disponível para a armazenagem, sendo que a armazenagem só poderá ser feita no próprio processador, que não estará disponível para processar outra operação.
4. **ZW (*Zero Wait*)** - O estado é consumido imediatamente após a sua produção.

O anexo 1 contém as representações STN dos dois exemplos utilizados na tese.

2.4. Determinação das janelas de tempo para os estados

As janelas de tempo das bateladas de um estado representam os intervalos de tempo em que as bateladas devem ser produzidas. O número de bateladas a serem produzidas e os finais das janelas de tempo dos estados são obtidos a partir da informação sobre as demandas de produtos finais. Os inícios das janelas de tempo dos estados são obtidos a partir da informação sobre as disponibilidades das matérias primas.

2.4.1. Determinação do número de bateladas a serem produzidas e dos finais das janelas de tempo

O primeiro passo na determinação do número de bateladas a serem produzidas é a definição do nível de cada estado da planta, através de um procedimento similar ao *low level coding* utilizado em sistemas MRP. Os níveis dos estados identificam a ordem em que os estados serão tratados durante o cálculo das janelas de tempo. No procedimento de *low level coding* do MRP determinam-se os níveis para os estados de forma que o nível do estado s_c (consumido por uma operação i) seja menor que o nível de qualquer estado s_g (gerado pela operação i). A seguir é apresentado o algoritmo de *low level coding*. A estrutura de grafo da representação STN permite determinar facilmente o nível de cada estado.

Algoritmo de low level coding

```

Para qualquer estado est1 fazer
{
  Tratar(est1) = falso;
  Nível(est1) = 0;
  Se est1 é um produto final então
  {
    Tratar(est1) = verdadeiro;
    Nível(est1) = 1;
  }
}
Segue = verdadeiro;
Enquanto Segue = verdadeiro fazer
{
  Segue = falso
  Para qualquer estado est2 fazer
  {
    Se Tratar(est2) = verdadeiro então
    {
      Tratar(est2) = falso
      Identifica a operação opr que gera est2;
      Identifica o estado est1 consumido por opr;
      Se Nível(est1) < Nível(est2) + 1 então
      {
        Nível(est1) = Nível(est2) + 1;
        Tratar(est1) = verdadeiro;
        Segue = verdadeiro;
      }
    }
  }
}

```

A representação STN é bem adequada para a definição deste ordenamento. Para o exemplo 2, dado no anexo 1, os níveis dos estados são óbvios como mostra a figura 2.2.

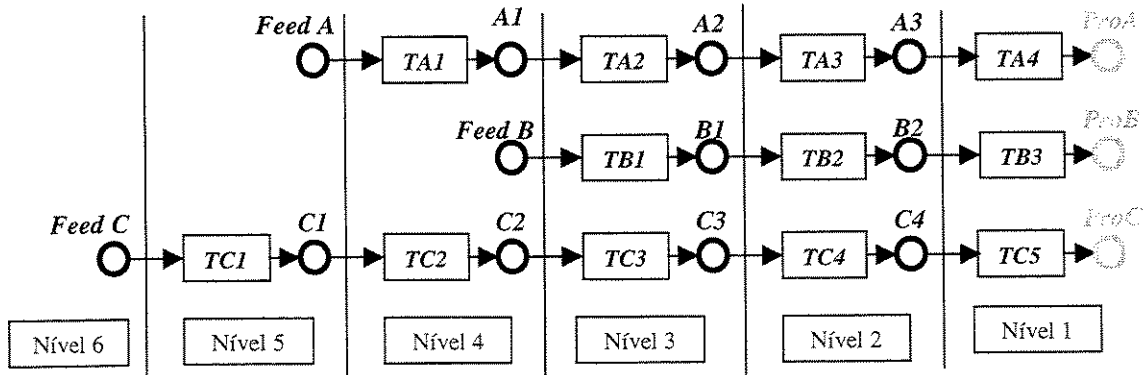


Figura 2.2. Níveis da representação STN para o exemplo 2.

O resultado obtido para o exemplo 1, apresentado no anexo 1, é o indicado na figura 2.3. Esta figura mostra, por exemplo, que o estado *Int2* tem nível 4 e não 3, porque o tratamento do estado *Int2* deve ocorrer após os tratamentos dos estados *S21* e *S40*.

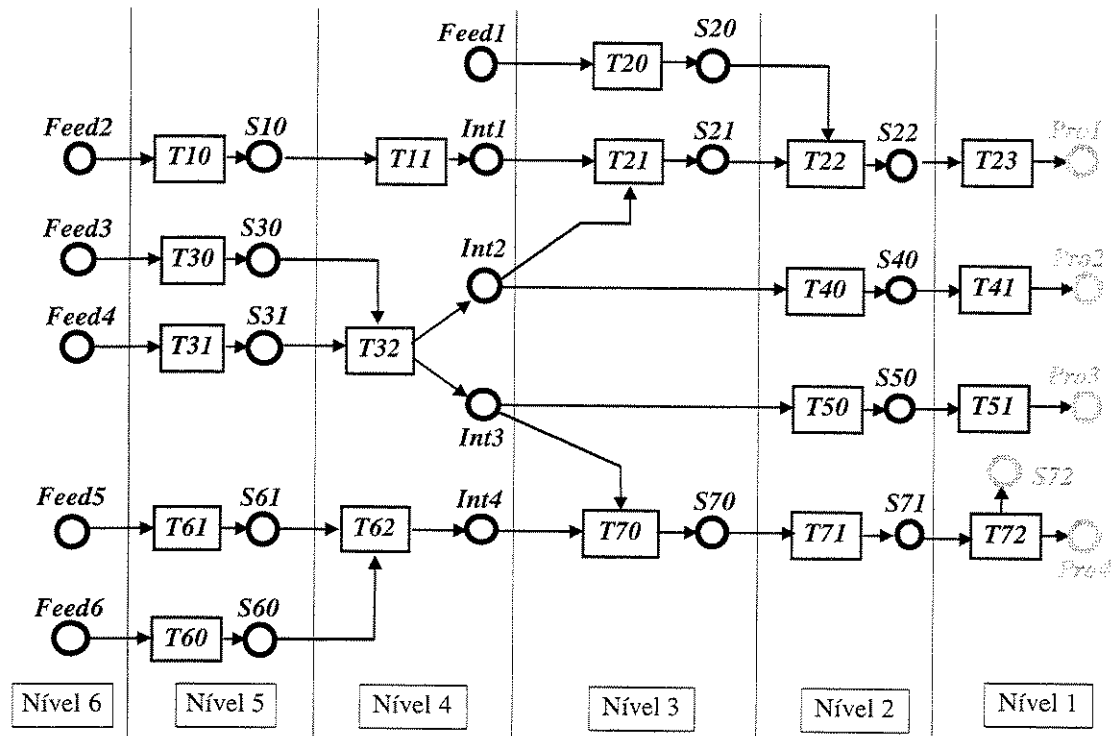


Figura 2.3. Níveis da representação STN para o exemplo 1.

Após a definição dos níveis dos estados, determina-se a quantidade de bateladas que serão produzidas e os finais das janelas de tempo destas bateladas. A seguir apresenta-se o algoritmo para a determinação da quantidade de bateladas que serão produzidas e os respectivos prazos de término. O final das janelas de tempo é dado pelo instante mais tarde em que é possível concluir o processamento das bateladas de tal forma que as demandas de produtos finais não sejam comprometidas. Neste algoritmo os estados são tratados pela ordem crescente dos níveis (da direita para a esquerda nas figuras 2.2 e 2.3) e o contador *N* indica o nível do estado tratado. O procedimento deste algoritmo é chamado de explosão para trás porque os primeiros estados a serem tratados são os produtos finais.

Algoritmo para a determinação do número de bateladas e dos finais das janelas de tempo

```

N = 1;
Enquanto N ≤ valor do nível máximo dos estados fazer
{
  Para qualquer estado est2 fazer
  {
    Se Nível(est2) = N então

```

```

    {
    • Identifica a operação opr que gera o estado est2;
    • Determina o número de bateladas da operação opr que serão produzidas para atender a
      demanda do estado est2;
    • Determina o prazo de conclusão mais tarde possível (finais das janelas) das bateladas de
      opr tal que a demanda do estado est2 seja satisfeita sem atrasos. Os finais das janelas de
      tempo de duas bateladas consecutivas têm um espaçamento mínimo igual ao tempo de
      processamento.
    • Identifica o estado est1 que é consumido pela operação opr.
    • Determina o perfil de demanda do estado est1 a partir do instante mais tarde em que
      cada batelada da operação opr que pode ser iniciada.
    }
  }
  N = N + 1;
}

```

Quando o estado tratado for uma matéria prima, o algoritmo acima define as necessidades de matérias primas e os prazos (*due dates*) de compras, gerando um perfil limite da necessidade de matérias primas. Isto corresponde a um plano de compras provavelmente irrealista no sentido de que serão compras individualizadas por batelada. Certamente a recepção de matérias primas será mais agregada, mas este plano individualizado é um limitante a ser respeitado por qualquer plano de compras que o usuário defina.

Se uma operação gera mais de um estado, estes estados são definidos como estados comuns à uma operação. Se o estado tratado for um estado comum, calcula-se a demanda máxima induzida sobre a operação que gera os estados comuns, ou seja, a produção da operação estará definida pelo estado com restrição mais forte (tanto em termos de quantidade como em prazo).

A figura 2.4 apresenta o resultado do cálculo do número de bateladas a serem produzidas e dos prazos de conclusão das bateladas que geram cada um dos estados para o exemplo 1. A demanda dos produtos finais para este exemplo é dada na tabela 2.1a. No lado esquerdo da figura 2.4 apresentam-se as janelas de tempo para a produção de todos os estados. Por exemplo, para satisfazer a demanda do estado **Pro1** (50 unidades de massa) requer-se a produção de 8 bateladas, já que o tamanho de cada batelada é igual a 7 unidades de massa. Para satisfazer esta demanda até o instante 60, a figura 2.4 mostra que é preciso que a primeira batelada que gera o estado **Pro1** seja concluída até o instante 46. As sete bateladas seguintes são espaçadas de 2 horas de tal forma que a oitava batelada deverá ser concluída até o instante 60. O cálculo do início das janelas de tempo é discutido na seção seguinte. No lado direito superior desta figura apresentam-se os prazos limites para a disponibilidade das matérias primas. Estes prazos indicam a demanda individualizada de cada matéria prima, estabelecendo um limitante para a disponibilidade das matérias primas. O lado direito inferior apresenta um resumo das informações do exemplo analisado, indicando sequencialmente: o tempo de processamento e o tamanho de cada batelada das operações, a restrição de armazenagem imposta sobre cada estado, as operações atribuídas a cada processador e as demandas dos produtos finais.

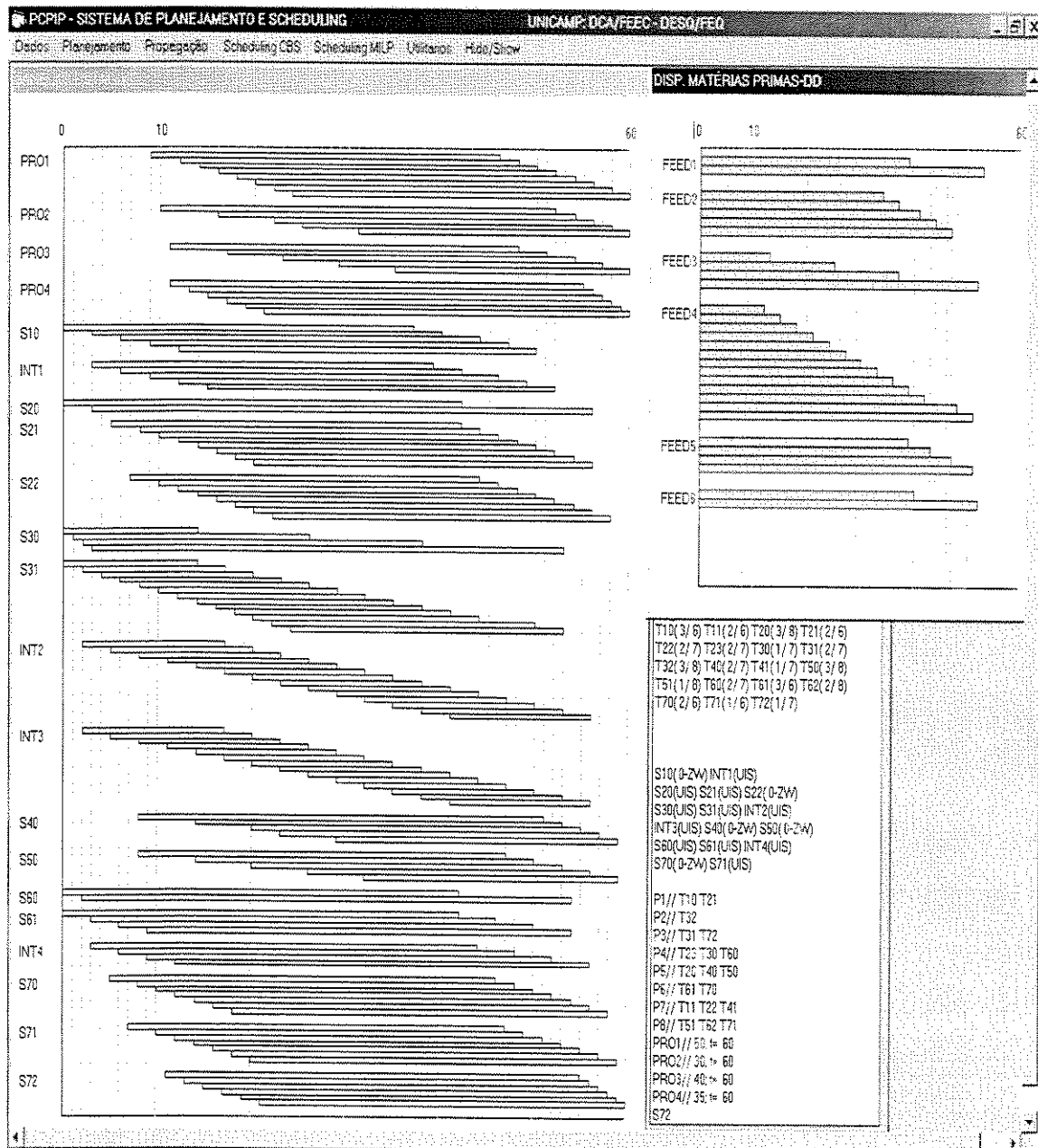


Figura 2.4. Prazos de conclusão dos estados para o exemplo 1.

Tabela 2.1a. Demandas para o exemplo 1.

produto final	demandas	prazo de conclusão
Pro1	50	60
Pro2	30	60
Pro3	40	60
Pro4	35	60

Tabela 2.1b. Tamanhos de batelada do exemplo 1.

operações	tamanho da batelada
T10, T11, T21, T61, T70, T71	6
T22, T23, T30, T31, T40, T41, T60, T72	7
T20, T32, T50, T51, T62	8

A figura 2.5 apresenta o resultado do cálculo do número de bateladas a serem produzidas e dos prazos de conclusão das bateladas para o exemplo 2. A demanda dos produtos finais para este exemplo é dada na tabela 2.2a. No lado esquerdo desta figura apresentam-se as janelas de tempo para a produção de todas as bateladas. No lado direito superior desta figura apresentam-se os prazos limites para a disponibilidade das matérias primas.

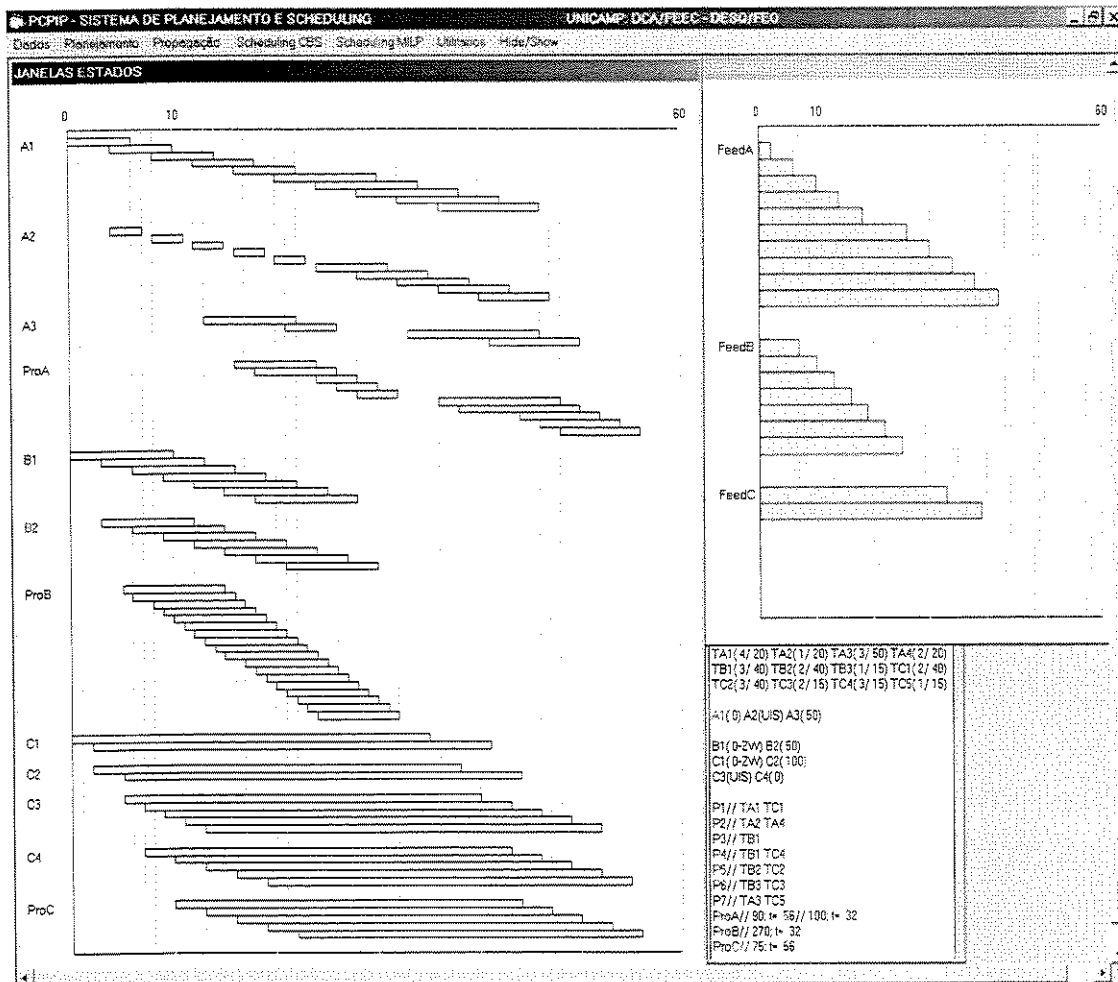


Figura 2.5. Prazos de conclusão dos estados para o exemplo 2.

Tabela 2.2a. Demandas para o exemplo 2.

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

Tabela 2.2b. Tamanhos de bateladas do exemplo 2.

operação	tamanho da batelada
TA1, TA2	20
TA3	50
TA4	20
TB1, TB2	40
TB3	15
TC1, TC2	40
TC3, TC4, TC5	15

2.4.2. Determinação dos inícios das janelas de tempo

Após a determinação do número de bateladas a serem produzidas dos estados e da determinação dos prazos de conclusão (finais das janelas de tempo) destes estados, determinam-se os inícios das janelas de tempo. Os inícios das janelas de tempo das bateladas são obtidos a partir da propagação da disponibilidade de matérias primas, definindo-se o instante mais cedo em que é possível iniciar o processamento das bateladas. Para a determinação dos inícios das janelas de tempo, os estados são tratados pela ordem decrescente dos níveis e o procedimento é chamado de *explosão para frente*. O procedimento de determinação dos inícios das janelas de tempo é apresentado a seguir.

Algoritmo para a determinação dos inícios das janelas de tempo

N = valor do nível máximo dos estados;

Enquanto $N \geq 1$ **fazer**

{
 Para qualquer estado *est1* **fazer**

 {
 Se $\text{Nível}(\text{est1}) = N$ **então**

- {
 • Identifica a operação *opr* que consome o estado *est1*;
 • A partir do perfil de disponibilidade do estado *est1*, determina-se o instante mais cedo (início das janelas) em que pode ser iniciado o processamento de cada batelada de *opr*. Neste procedimento considera-se que toda a massa do estado *est1* está disponível para o consumo pela operação *opr*. Os inícios das janelas de tempo de duas bateladas consecutivas têm um espaçamento mínimo igual ao tempo de processamento.
 • Identifica o estado *est2* gerado pela operação *opr*.
 • Determina-se do perfil de disponibilidade do estado *est2* a partir do instante mais cedo em que o processamento de cada batelada da operação *opr* pode ser concluído.
 }

 }
 N = N - 1;
}

As figuras 2.4 e 2.5 mostram os inícios de janelas de tempo obtidos quando todas as matérias primas estão disponíveis no instante zero. A figura 2.6 apresenta as janelas de tempo para a produção dos estados do exemplo 1 considerando a disponibilidade real de matérias primas dada na tabela 2.3. O lado direito da figura 2.6 mostra a disponibilidade das matérias primas em preto, enquanto que o final de cada intervalo cinza identifica o limite para a disponibilidade das matérias primas, obtido pelo processo de explosão para trás.

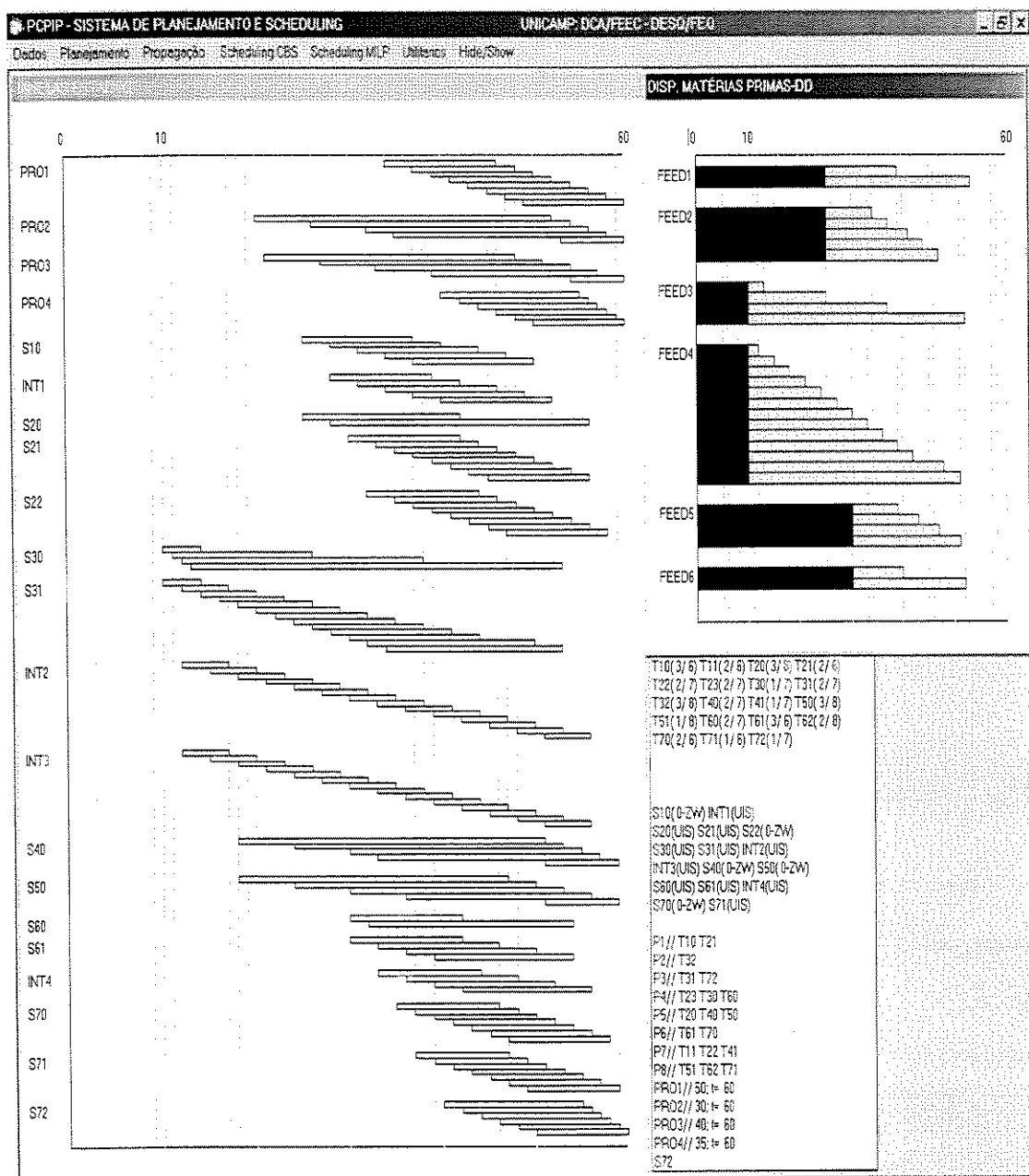


Figura 2.6. Janelas de tempo e disponibilidade das matérias primas para o exemplo 1.

Tabela 2.3. Disponibilidade das matérias primas para o exemplo 1.

Matéria prima	Quantidade	instante
Feed1	16	25
Feed2	30	25
Feed3	28	10
Feed4	91	10
Feed5	24	30
Feed6	14	30

A figura 2.7 apresenta as janelas de tempo iniciais dos estados e a disponibilidade das matérias primas para o exemplo 2. A tabela 2.4 apresenta a disponibilidade de matérias primas para este exemplo.

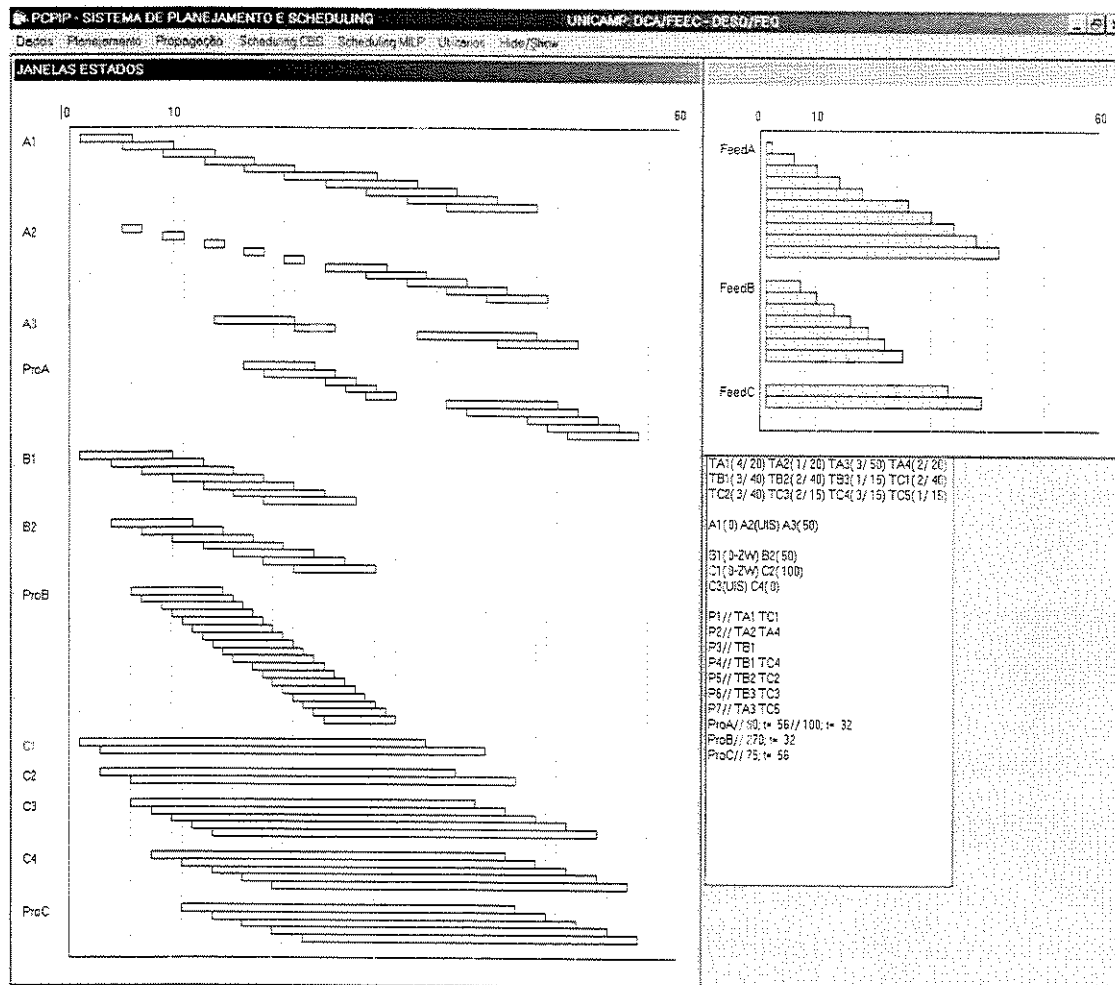


Figura 2.7. Janelas de tempo e disponibilidade das matérias primas para o exemplo 2.

Tabela 2.4. Disponibilidade das matérias primas para o exemplo 2.

Matéria prima	Quantidade	instante
FeedA	200	1
FeedB	280	1
FeedC	80	1

2.4.2.1. Determinação das relações de precedência (*pegging*) entre bateladas

Na fase de planejamento também são determinadas as relações de precedência entre bateladas de estados gerados e consumidos por uma mesma operação (relações de *pegging* em sistemas MRP). Considerando um grafo em que cada batelada é um nó, as relações de

precedência definem arcos fixos representando as relações de dependência decorrentes do balanço de massa. Ou seja, os arcos identificam qual o nó produtor que viabiliza um nó consumidor. Estas relações de precedência serão utilizadas para reduzir a dimensão do problema de scheduling, discutido no capítulo 4.

A figura 2.8 apresenta as relações de precedência decorrentes do balanço de massa para o exemplo 1. A quantidade produzida de cada estado por batelada está indicada nesta figura ao lado do nome do estado. Quando uma batelada produtora viabiliza a execução de mais de uma batelada consumidora, a figura indica apenas o arco dirigido para a primeira batelada consumidora viabilizada. Por exemplo, a primeira batelada do estado **S20** viabiliza as sete primeiras bateladas do estado **S21**, mas apenas o arco com a primeira batelada é representado.

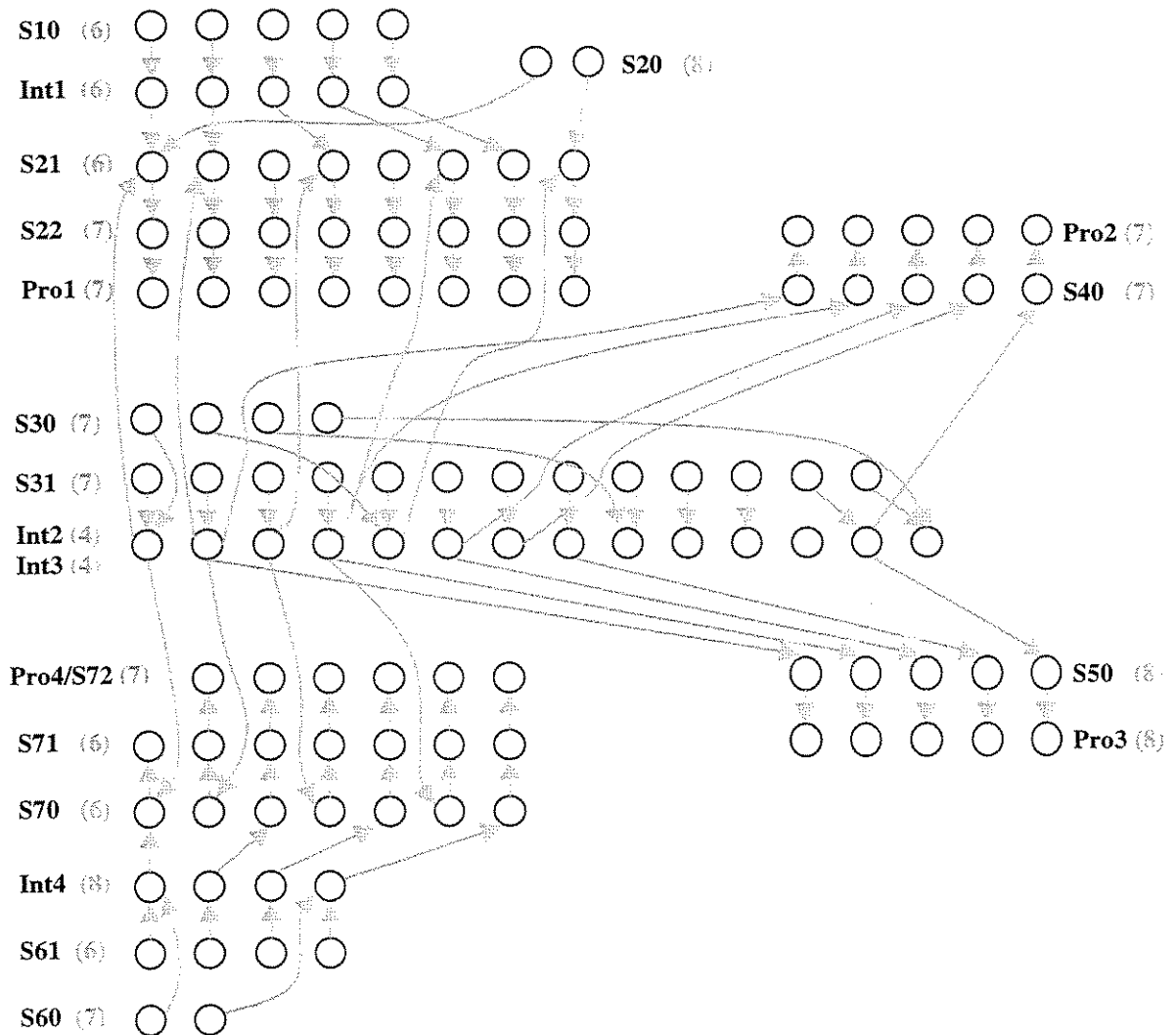


Figura 2.8. Relações de precedência decorrente do balanço de massa para o exemplo 1.

A figura 2.9 apresenta as relações de precedência do exemplo 2.

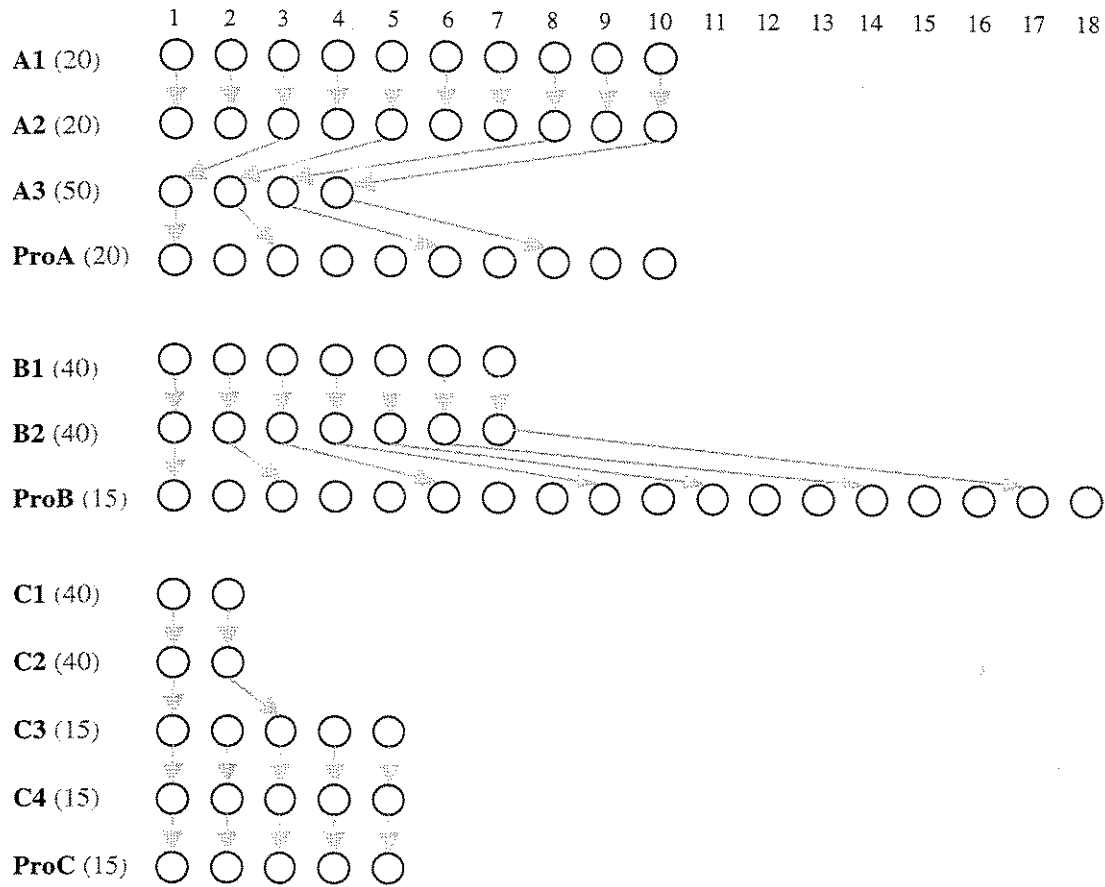


Figura 2.9. Relações de precedência do exemplo 2.

2.4.3. Caso particular: Receita com estados compartilhados

Se duas ou mais operações consomem um mesmo estado, é dito que há um compartilhamento do estado consumido. O exemplo 1 (figura A.1) tem dois estados compartilhados: *Int2* e *Int3*. O cálculo de inícios das janelas de tempo para os estados gerados pelas operações (estados de saída) que consomem estados compartilhados supõe a definição de uma forma de consumo do estado compartilhado. Esta definição é tipicamente uma questão a ser resolvida na fase de *scheduling* e não na fase de planejamento.

Existem duas possibilidades para o tratamento dos estados compartilhados:

- Determinar os inícios das janelas de tempo dos estados de saída supondo que a massa do estado compartilhado está totalmente disponível para cada uma das operações consumidoras;
- Arbitrar uma política de consumo deste estado.

A primeira opção adia corretamente as decisões de consumo do estado compartilhado para a fase de *scheduling*. O inconveniente desta opção é que ela leva a janelas de tempo maiores do que as reais, que podem encobrir uma situação infactível. As janelas de tempo obtidas desta forma serão chamadas de otimistas. A segunda opção implica na definição, na fase de planejamento, de uma política de consumo do estado compartilhado. Esta decisão permitirá o cálculo correto dos inícios de janelas de tempo, mas estes inícios só serão válidos para esta decisão. É importante notar que esta decisão pode ser ignorada na fase de *scheduling*. Basta para isto que a fase de *scheduling* utilize as janelas de tempo otimistas. O fato de se arbitrar uma política de consumo permite que, na fase de planejamento, se analise a factibilidade e o carregamento do plano resultante desta decisão.

Foram implementadas três opções para a definição de inícios das janelas de tempo reduzidas, quando há compartilhamento de estado: *i*) a formulação como um problema de otimização MILP de âmbito reduzido, *ii*) um procedimento heurístico baseado em criticalidades (Keng *et al.*, 1988) e *iii*) a atribuição de prioridades por parte do usuário. Cada um dos procedimentos propostos poderá resultar em janelas de tempo diferentes.

As janelas de tempo para os estados de saída são determinadas inicialmente com a hipótese de que toda a massa do estado compartilhado está disponível para cada uma das operações consumidoras. Estas janelas de tempo otimistas podem ser objeto de uma análise de factibilidade parcial independentemente da política de consumo do estado compartilhado. Isto porque para cada estado de saída *s*, a alocação mais cedo das suas bateladas não pode impedir a alocação de todas as bateladas de todos os outros estados de saída. Se isto ocorrer, algum (alguns) início(s) de janela(s) de tempo da(s) batelada(s) do estado de saída *s* devem ser aumentados. Este procedimento de análise de factibilidade é descrito a seguir:

1. Constrói-se o perfil de disponibilidade mais cedo do estado compartilhado. É o equivalente a simular a produção do estado compartilhado nos instantes de término mais cedo.
2. Constrói-se o perfil de consumo mais tarde do estado compartilhado para cada estado de saída. É equivalente a simular o consumo no instante mais tarde possível das bateladas de cada operação consumidora do estado compartilhado.
3. A análise de factibilidade do início das janelas de tempo de um estado *s* gerado pela operação *i* que consome o estado compartilhado é feita construindo-se o perfil resultante da subtração dos perfis de disponibilidade mais cedo e de consumo mais tarde das operações *k* ($k \neq i$) consumidoras do estado compartilhado. O perfil resultante indica a quantidade do estado compartilhado que está disponível para a execução da operação *i*. Procede-se ao avanço nos inícios das janelas de tempo das bateladas da operação *i* se o perfil resultante não é suficiente para a execução destas bateladas no início de suas janelas de tempo.

Este procedimento é aplicado a seguir ao exemplo 1 que possui dois estados compartilhados (*Int2* e *Int3*). As janelas de tempo otimistas para os estados *S21* e *S40*, que

consomem o estado *Int2*, e as resultantes do procedimento anterior estão apresentadas na tabela 2.5. A figura 2.6 (página 21) mostra as janelas de tempo resultantes com o plano de disponibilidade de matérias primas dado na tabela 2.3. Este plano está indicado em preto no canto direito da figura 2.6. Pode-se observar que o início da janela de tempo da quinta batelada do estado *S40* foi mudado do instante 39 para o instante 51.

Tabela 2.5. Valores de *ebt* para as janelas de tempo otimistas e considerando o compartilhamento de *Int2*.

batelada consumo	<i>ebt</i> otimista	<i>ebt</i> resultante
<i>S21/1</i>	30	30
<i>S21/2</i>	33	33
<i>S21/3</i>	35	35
<i>S21/4</i>	37	37
<i>S21/5</i>	39	39
<i>S21/6</i>	41	41
<i>S21/7</i>	43	43
<i>S21/8</i>	45	45
<i>S40/1</i>	18	18
<i>S40/2</i>	24	24
<i>S40/3</i>	30	30
<i>S40/4</i>	36	36
<i>S40/5</i>	39	51

O procedimento descrito anteriormente é ilustrado nas figuras 2.10a até 2.10d que representam respectivamente: o perfil de produção mais cedo do estado *Int2*, o perfil de consumo mais tarde deste estado pela operação *T21*, o perfil resultante da subtração dos perfis de disponibilidade mais cedo de *Int2* e de consumo mais tarde do estado de *T21*, e o perfil de consumo mais cedo de *Int2* pela operação *T40*.

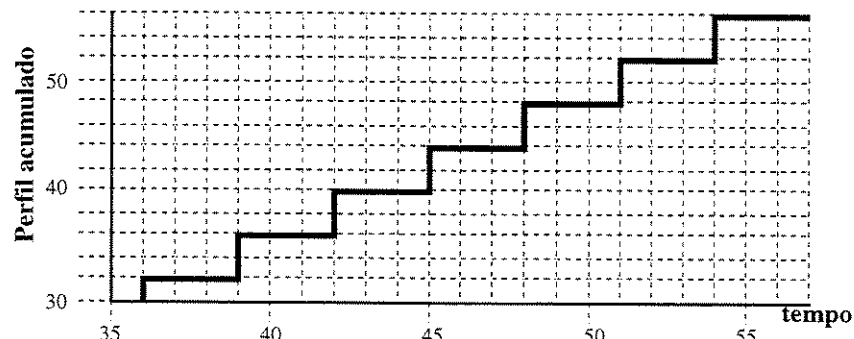


Figura 2.10a. Perfil acumulado de produção mais cedo de *Int2*.

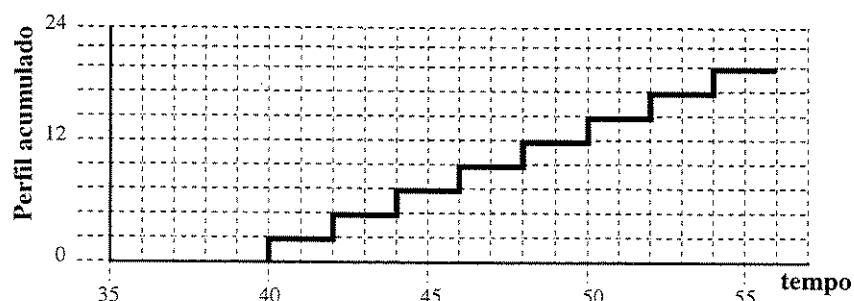


Figura 2.10b. Perfil acumulado de consumo mais tarde de *S21*.

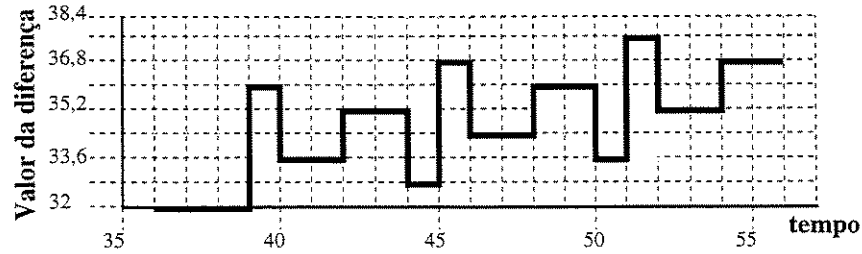


Figura 2.10c. Diferença entre os perfis mais cedo de produção de *Int2* e de consumo mais tarde de *S21*.

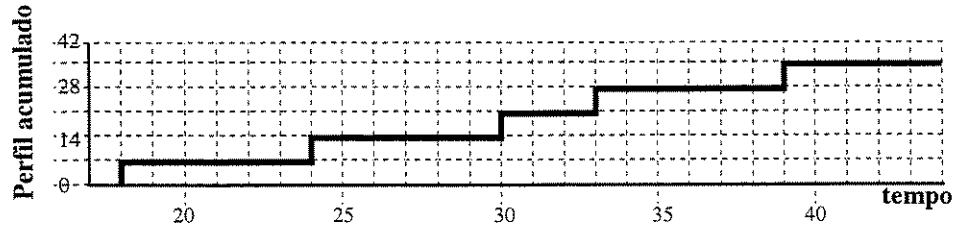


Figura 2.10d. Perfil acumulado de consumo mais cedo de *S40*.

O perfil de consumo mais cedo de *Int2* pela operação *T40* (figura 2.10d) mostra que a produção da quinta batelada desta operação implica em um consumo acumulado de 35 unidades de massa a partir do instante 39. No entanto, o perfil resultante da subtração (figura 2.10c) mostra que a quinta batelada da operação *T40* só pode ser iniciada no instante 51 porque até este instante há diversos intervalos de tempo onde a disponibilidade resultante é inferior a 35 unidades de massa.

Após estas reduções obrigatórias, se o usuário julgar conveniente, ele poderá utilizar um dos métodos apresentados a seguir para considerar uma forma mais “realista” de utilização dos estados compartilhados.

a) Tratamento dos estados compartilhados através de uma proposta de otimização

O ordenamento entre as bateladas consumidoras de um estado compartilhado é feito através de um procedimento de otimização. A função objetivo deste procedimento de otimização é maximizar a soma ponderada do tamanho das janelas de tempo resultantes para os estados de saída. A definição de um valor de ponderação para cada uma das operações na função de custo é feita pelo usuário através da avaliação do carregamento dos processadores utilizados pelas operações consumidoras de um estado compartilhado. Utilizam-se valores maiores para as operações consumidoras que serão executadas em processadores mais carregados. A formulação MILP para o tratamento dos estados compartilhados é dada abaixo e as suas explicações são dadas à continuação.

Função objetivo:

$$\max \left(\sum_{i \in Out} \left\{ \sum_{b \in Outb_i} [lft_{bi} - \sum_{\substack{t \geq ebt_{bit} \\ t \leq lft_{bi} - tp_i + 1}} (ebtm_{bit} * t)] + 1 \right\} * q_i * [tp_i / \overline{tp}] \right) \quad [2.1]$$

Restrições:

$$\sum_{\substack{t \geq ebt_{bi} \\ t \leq lft_{bi} - tp_i + 1}} ebtm_{bit} = 1 \quad \forall i, b \in Out_i \quad [2.2]$$

$$\sum_{\substack{t \geq ebt_{bi} \\ t \leq lft_{bi} - tp_i + 1}} ebtm_{bit} * t \geq \sum_{\substack{t \geq ebt_{b'i} \\ t \leq lft_{b'i} - tp_i + 1}} ebtm_{b'it} * t \quad \forall i \in Cs, b \neq b' / b' = b+1 \quad [2.3]$$

$$S_t = S_{t-1} + P_t - \sum_{i \in Out} \sum_{b \in Outb_i} C_i * ebtm_{bit} \quad \forall t \in H \quad [2.4]$$

onde,

Índices:

i	Operação
b	Batelada
t	Instante de tempo

Variáveis:

$ebtm_{bit}$ = 1 indica que o início da janela de tempo modificado da batelada b da operação i ocorre no instante t . Este início de janela é calculado pela otimização; = 0, caso contrário.

S_t Valor do estoque do estado compartilhado no instante t . (variável positiva)

Parâmetros:

$\overline{tp}$	Média dos tempos de processamento.
tp_i	Tempo de processamento da operação i .
ebt_{bi}	Início da janela de tempo disponível da batelada b da operação i .
lft_{bi}	Final da janela de tempo da batelada b da operação i .
C_i	Quantidade consumida do estado compartilhado de cada operação i .
H	Horizonte de planeamento.
q_i	Peso da janela das bateladas da operação i na função objetivo.
P_t	Acréscimo no estoque do estado compartilhado no instante t baseado no perfil de disponibilidade mais cedo deste estado.

Restrições no âmbito das expressões:

Out_i	Conjunto das bateladas da operação i que consomem o estado compartilhado.
Cs	Conjunto das operações que consomem o estado compartilhado.

A função objetivo deste problema é maximizar o resultado da soma ponderada das janelas de tempo, como mostra a expressão 2.1. A expressão 2.2 define que, para cada batelada b de uma operação i , o início da janela de tempo modificado ($ebtm_{bit}$) deve se encontrar entre o início da janela de tempo otimista (ebt_{bi}) e o último instante dentro da janela de tempo em que é factível iniciar o processamento da batelada b , que é definido pela expressão ($lft_{bi} - tp_i + 1$).

A expressão 2.3 mantém o ordenamento entre bateladas sucessivas de uma mesma operação, reduzindo a degenerescência do problema MILP. Se duas bateladas b e b' são de uma mesma operação e $b < b'$, então a batelada b terá seu início de janela de tempo modificado menor que o início da janela da batelada b' .

A expressão 2.4 realiza o balanço de massa do estado compartilhado, obrigando que os inícios das janelas de tempo modificados ($ebtm_{bit}$) respeitem a disponibilidade do estado compartilhado, para não dar lugar a estoques negativos deste estado.

b) Tratamento dos estados compartilhados através da definição manual de prioridades

Para controlar a disputa pelas operações que consomem os estados compartilhados o

usuário pode estabelecer prioridades entre as bateladas que disputam o mesmo estado. A partir das prioridades das bateladas, utiliza-se o perfil de disponibilidade mais cedo possível dos estados compartilhados para determinar o instante mais cedo em que o processamento de cada batelada pode ser iniciado. O estabelecimento de prioridades pode ser difícil quando há um grande número de bateladas disputando o consumo dos estados compartilhados e os finais das janelas de tempo das bateladas estão relativamente próximos. Nesta situação, o procedimento de otimização pode representar um ponto de partida para o usuário.

c) Utilização de criticalidades como critério heurístico de definição das prioridades no consumo dos estados compartilhados

Nesta seção é apresentada uma proposta para automatizar a definição das prioridades das bateladas baseando-se no conceito de criticalidades das bateladas, proposta por Keng *et al.* (1988). Estes autores propuseram uma medida da possibilidade de ocupação de cada batelada do intervalo dado pela sua janela de tempo. As criticalidades são obtidas a partir da informação dos tempos de processamento e das janelas de tempo das bateladas, como indicado pela expressão 2.5.

$$critk_{bi} = tp_i / dw_{bi} \quad \forall i, b \in Bat_i \quad [2.5]$$

onde:

$critk_{bi}$ criticalidade da batelada b da operação i proposta por Keng *et al.* (1988).
 dw_{bi} tamanho da janela de tempo da batelada b da operação i , dada por: $lft_{bi} - ebt_{bi} + 1$.

Calcula-se o perfil de consumo do estado compartilhado considerando-se o instante mais cedo de consumo dado pelas janelas otimizadas das bateladas consumidoras do estado compartilhado. O perfil de disponibilidade do estado compartilhado é calculado considerando-se o instante mais cedo de produção deste estado. Após a obtenção dos perfis de disponibilidade e de consumo do estado compartilhado, verifica-se se há algum período em que o perfil de consumo é maior que o perfil acumulado de disponibilidade. Quando isto ocorre em um instante t , identificam-se todas as bateladas cujas janelas de tempo contêm este instante ($ebt_{bi} \leq t \leq lft_{bi}$) e calcula-se a criticalidade destas bateladas considerando que o início da janela de tempo de todas estas bateladas ocorra no instante t :

$$crit_{bi} = tp_i / (lft_{bi} - t + 1) \quad \forall i, b \in Bat_i / ebt_{bi} \leq t \leq lft_{bi} \quad [2.6]$$

onde:

$crit_{bi}$ Criticalidade da batelada b da operação i no instante de tempo t
 tp_i tempo de processamento da operação i .
 Bat_i conjunto das bateladas pertencentes à operação i .

As prioridades destas bateladas são definidas seguindo os valores decrescentes da criticalidade. A batelada de maior criticalidade terá prioridade no consumo do estado compartilhado. O objetivo deste procedimento é o de dar prioridade no consumo de um

estado compartilhado à batelada que seria mais prejudicada pelo avanço do início da sua janela de tempo para o instante t .

A partir das prioridades determinam-se as janelas de tempo reduzidas, considerando-se a disponibilidade do estado compartilhado. Este procedimento visa o balanceamento do tamanho das janelas de tempo evitando aumentos de inícios de janelas de tempo que resultem em janelas de tempo muito pequenas.

2.5. Determinação das janelas de tempo das bateladas das operações

As janelas de tempo dos estados são obtidas considerando-se apenas as demandas de produtos finais e as disponibilidades de matérias primas. Uma vez determinadas as janelas de tempo dos estados, a obtenção das janelas de tempo das operações é trivial. A janela de tempo da batelada b de uma operação i é igual à janela de tempo da batelada b do estado gerado pela operação i . O procedimento de obtenção das janelas de tempo dos estados garante que elas são idênticas para todos os estados gerados por uma operação. Estas janelas de tempo são chamadas de janelas de tempo iniciais.

As janelas de tempo iniciais não levam em consideração as restrições decorrentes do compartilhamento dos processadores por operações diferentes. Estas restrições podem impor reduções nas janelas de tempo as quais são importantes porque levam à identificação de intervalos de tempo mais carregados, podendo inclusive revelar infactibilidades das janelas de tempo. Se a situação é factível, reduzem a dimensão do problema de *scheduling*.

As figura 2.11 e 2.12 apresentam as janelas de tempo iniciais dos estados e das bateladas para os dois exemplos analisados. Nestas duas figuras pode-se observar que as janelas de tempo de um estado são iguais às janelas de tempo da operação que gera este estado.

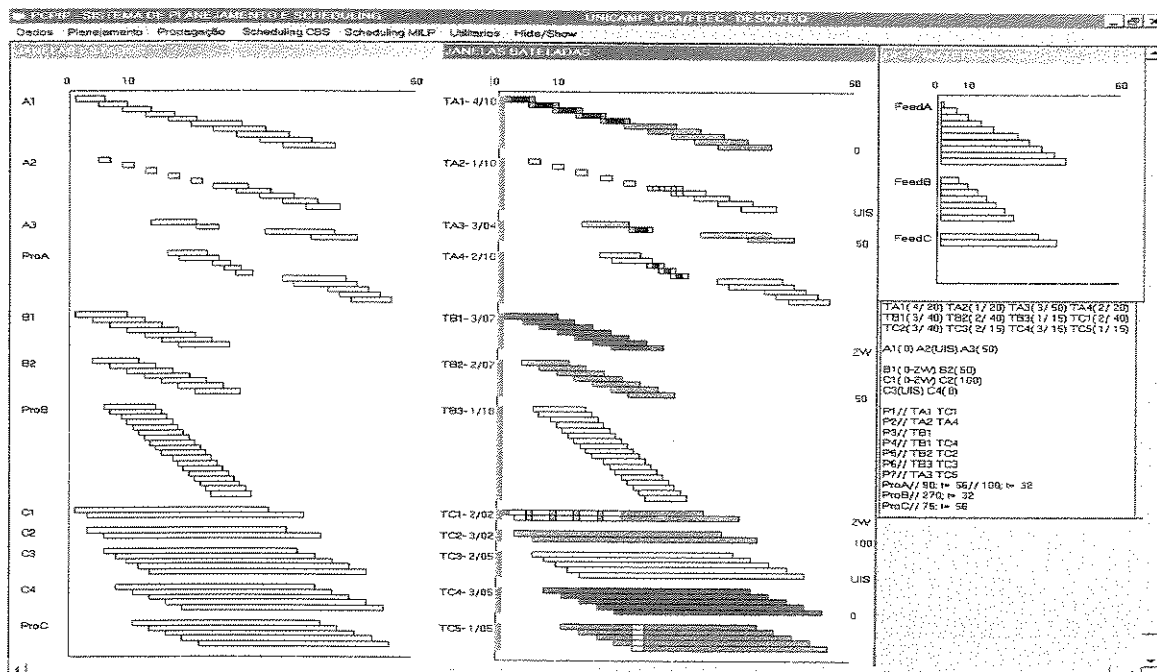


Figura 2.11. Janelas de tempo iniciais dos estados e das bateladas para o exemplo 2.

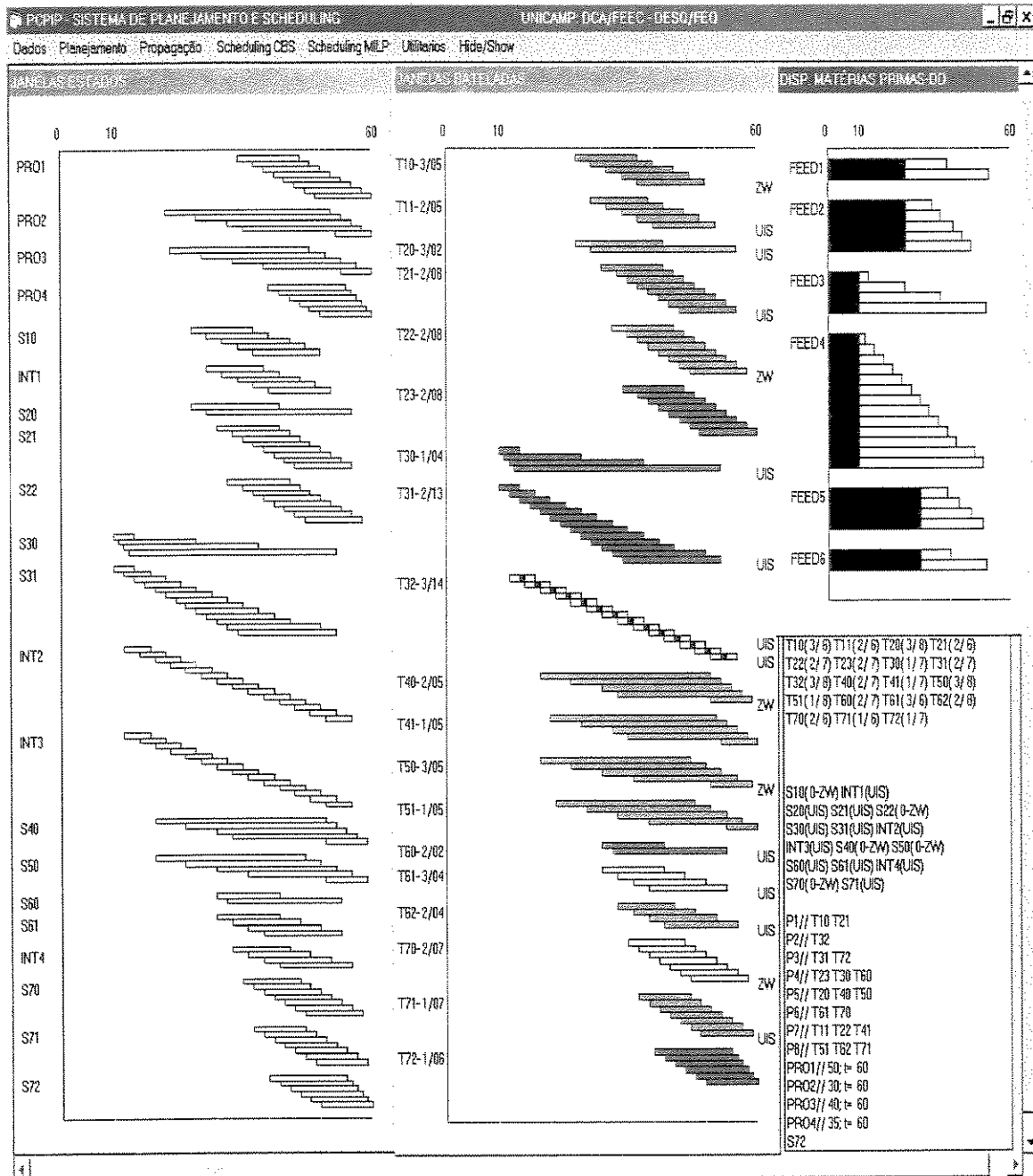


Figura 2.12. Janelas de tempo iniciais dos estados e das bateladas para o exemplo 1.

A figura 2.13 apresenta as relações de precedência decorrentes do balanço de massa para o exemplo 1. Esta figura é similar à figura 2.7, mas desta vez são apresentadas relações de precedência entre as operações. Os tamanhos das bateladas estão também indicados nesta figura ao lado do nome da operação.

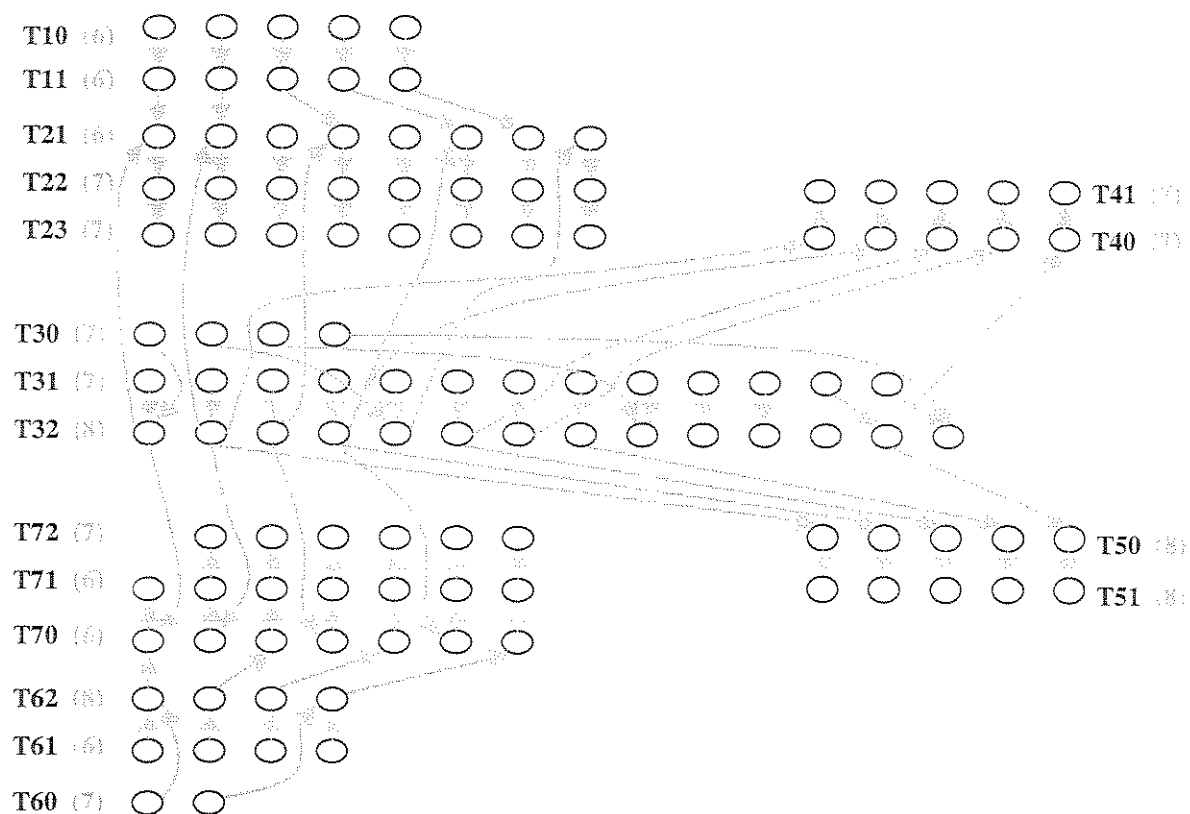


Figura 2.13. Relações de precedência decorrente do balanço de massa para o exemplo 1.

A figura 2.14 apresenta as relações de precedência decorrentes do balanço de massa para o exemplo 2.

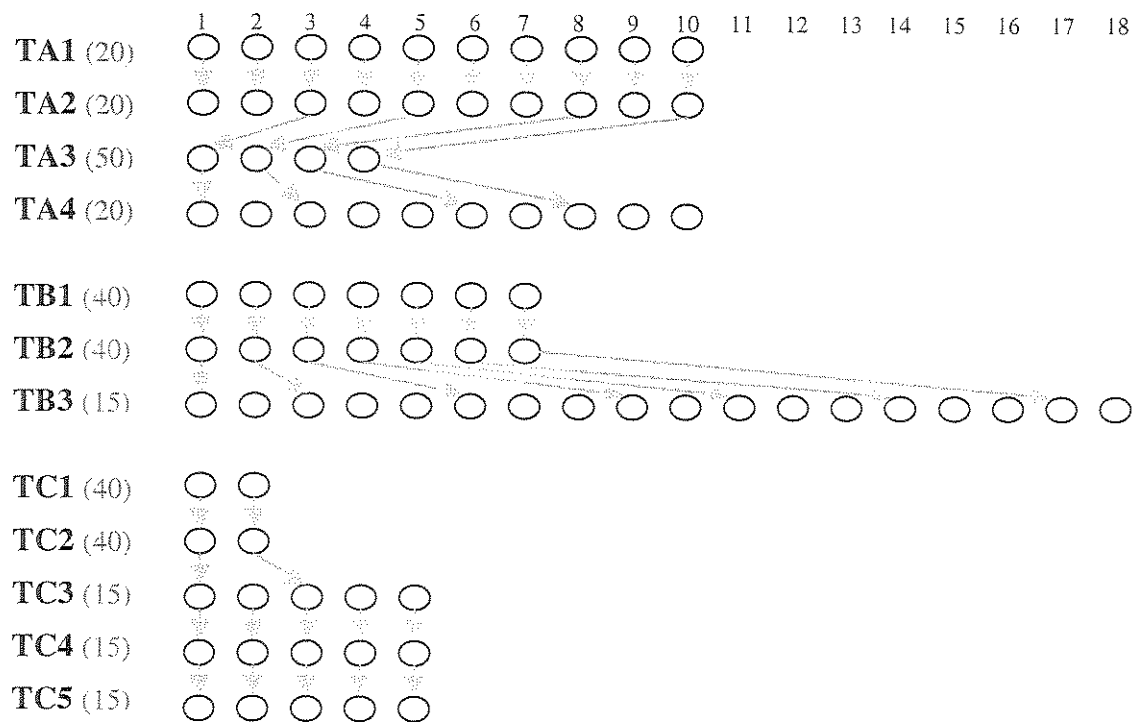


Figura 2.14. Relações de precedência do exemplo 2.

2.6. Redução das janelas das bateladas através da propagação de restrições

Para a análise e eventual redução das janelas de tempo das operações são utilizados métodos de propagação das restrições impostas por:

- Compartilhamento de recursos unitários (processadores);
- Condições de armazenagem;
- Ordenamento de bateladas de uma mesma operação;
- Relações de precedência impostas pelo balanço de massa.

Estas técnicas foram desenvolvidas dentro da área de inteligência artificial no contexto de busca orientada por restrições (*Constrained Based Search*) e visam a redução da dimensão do problema de *scheduling* através da redução das janelas de tempo. Os primeiros trabalhos são de 1976 (Erschler, 1976) e vêm sendo utilizados em diversos sistemas de *scheduling* (Micro-Boss [Sadeh, 1991], ILOG *Scheduler* (1997) etc.). Estes sistemas utilizam as restrições originadas pelo compartilhamento de recursos unitários, também chamadas de restrições disjuntivas, e as restrições de precedência impostas pelo balanço de massa. A redução das janelas de tempo devido a recursos compartilhados cumulativos foi introduzida mais recentemente (Claire *schedule* 1.0, 1999). Não é do conhecimento do autor que tenha sido introduzida a propagação das restrições de armazenagem em algum *software* de *scheduling*.

2.6.1. Propagação de restrições disjuntivas

A propagação de restrições impostas pelo compartilhamento dos processadores é feita através de três mecanismos:

- i) Intervalos de ocupação obrigatória;
- ii) Ordenamentos entre pares de bateladas e
- iii) Ordenamentos impostos por conjuntos de bateladas.

Nesta seção não será utilizado o índice de operação por não ser necessário e facilitar a compreensão.

Os intervalos de ocupação obrigatória são intervalos de tempo que necessariamente serão ocupados por uma batelada *i* de uma operação em um processador *j* e nenhuma outra batelada poderá ser realizada no processador *j* neste intervalo de tempo. A identificação de um intervalo de ocupação obrigatória de uma batelada executada no processador *j* modifica as janelas de tempo das outras bateladas que devem ser processadas em *j* e que contém parcial ou totalmente o intervalo de ocupação obrigatória.

Sadeh (1991) apresenta um procedimento, similar ao proposto por Erschler e Esquirol (1986), onde definem-se os intervalos de ocupação obrigatória (*intervals of total*

reliance) em que um processador é necessariamente ocupado por uma operação. Se o instante mais cedo de término do processamento de uma batelada ($ebt_i + tp_i$) é maior que o instante mais tarde de início do processamento desta mesma batelada ($lft_i - tp_i$), então o intervalo de ocupação obrigatória da batelada i inicia-se no instante ($lft_i - tp_i$) e encerra-se no instante ($ebt_i + tp_i$). Se $ebt_i + tp_i \leq lft_i - tp_i$, então a batelada i não possui intervalo de ocupação obrigatória. Se houver interseção entre os intervalos de ocupação obrigatória de duas bateladas, então uma das janelas de tempo das bateladas será infactível, formando o que Sadeh chamou de *oversubscribed interval*.

O segundo procedimento utilizado de análise do compartilhamento dos processadores baseia-se na identificação de modificações nas janelas de tempo decorrentes da existência de ordenamentos entre pares de bateladas. Considerando-se duas bateladas de operações diferentes, a situação será infactível se a condição 2.7 não for satisfeita.

$$\max(lft_i, lft_j) - \min(ebt_i, ebt_j) \geq tp_i + tp_j \quad [2.7]$$

Identifica-se que a batelada i precede a batelada j se a condição 2.8 é verdadeira.

$$ebt_j + tp_j > lft_i - tp_i \quad [2.8]$$

Se a condição 2.8 é verdadeira e ($ebt_j < ebt_i + tp_i$), então o início da janela de tempo da batelada j deve ser aumentado porque a batelada i necessariamente precede a batelada j .

$$ebt_j = ebt_i + tp_i \quad [2.9]$$

Se a condição 2.8 é verdadeira e ($lft_i > lft_j - tp_j$), então o final da janela de tempo da batelada i deve ser diminuído porque a batelada i necessariamente segue a batelada j .

$$lft_i = lft_j - tp_j \quad [2.10]$$

Podem existir relações de precedência entre uma batelada i e um conjunto de bateladas Ω , executadas no mesmo processador. Caseau e Laburthe (1995) e Baptiste e Le Pape (1995) desenvolveram as condições para detectar e propagar estas restrições. Como consequência da identificação de relações de precedência poderão ser necessárias modificações nas janelas de tempo da batelada i e/ou das bateladas do conjunto Ω . Dado um processador onde serão processadas n bateladas, não é necessário gerar todos os ($2^n - 1$) conjuntos de bateladas. Baptiste e Le Pape (1995) mostraram que é possível considerar apenas n^2 conjuntos para testar todas as precedências. Estes conjuntos de bateladas (Ω) são chamados de intervalos de operações (*task intervals*).

Um par de bateladas (i e j) é usado para definir cada intervalo de operações Ω , podendo a batelada i ser igual a j . O instante de início mais cedo do intervalo de operações (ebt_Ω) é igual ao menor valor de início da janela de tempo entre as bateladas i e j ($\min\{ebt_i, ebt_j\}$). O instante de término mais tarde do intervalo de operações (lft_Ω) é igual ao maior valor de final da janela de tempo entre as bateladas i e j ($\max\{lft_i, lft_j\}$). Toda batelada que estiver contida dentro do intervalo $[ebt_\Omega, lft_\Omega]$ pertencerá ao intervalo de operações. Ou seja, se há uma batelada k tal que $ebt_k \geq ebt_\Omega$ e $lft_k \leq lft_\Omega$, então k pertence ao intervalo de operações gerado por i e j . Define-se o tempo de processamento do intervalo de

operações (tp_{Ω}) como a soma dos tempos de processamento de todas as bateladas pertencentes ao intervalo de operações Ω .

a) Teste de *edge-finding*

O teste de *edge-finding* determina se há alguma relação de precedência entre uma batelada i e um intervalo de operações Ω ao qual a batelada i pertence. Antes de realizar o teste de *edge-finding* realiza-se um teste de factibilidade do intervalo de operações Ω . Se a expressão 2.11

$$lft_{\Omega} - ebt_{\Omega} \geq tp_{\Omega} \quad [2.11]$$

não é satisfeita, então o intervalo de tempo definido para o processamento do conjunto de bateladas Ω é menor que a soma dos tempos de processamento de todas as bateladas pertencentes ao intervalo de operações Ω e a situação é infactível.

O teste de *edge-finding* permite determinar se a batelada i não precede ou não segue o conjunto $\{\Omega - i\}$. Se a expressão 2.12

$$lft_i - ebt_{\Omega} < tp_{\Omega} \quad \forall i \in \Omega \quad [2.12]$$

é satisfeita, então a batelada i não segue as demais bateladas pertencentes ao conjunto Ω , já que o intervalo de tempo entre o início do intervalo de operações Ω e o final da janela de tempo da batelada i é menor que a soma dos tempos de processamento de todas as bateladas pertencentes a Ω (incluindo a batelada i).

Se a batelada i não segue o intervalo de operações Ω , então será preciso garantir que haverá tempo suficiente para o processamento de pelo menos uma batelada pertencente a Ω após o processamento da batelada i . Esta condição é imposta pela expressão 2.13.

$$lft_i = \min(lft_i, \max_{k \in \Omega, k \neq i} (lft_k - tp_k)) \quad \forall i \in \Omega \quad [2.13]$$

O teste de *edge-finding* para verificar se uma batelada i não precede o intervalo de operações Ω é análogo ao teste anterior. Se a expressão 2.14

$$lft_{\Omega} - ebt_i < tp_{\Omega} \quad \forall i \in \Omega \quad [2.14]$$

é satisfeita, então a batelada i não precede as demais bateladas pertencentes ao intervalo de operações Ω , já que não é possível processar todas as bateladas pertencentes a Ω dentro do intervalo de tempo definido entre o início da janela de tempo da batelada i e o final do intervalo de operações Ω . Daí conclui-se que será necessário processar pelo menos uma batelada pertencente ao intervalo de operações Ω antes da batelada i . Esta condição é imposta pela expressão 2.15.

$$ebt_i = \max(ebt_i, \min_{k \in \Omega, k \neq i} (ebt_k + tp_k)) \quad \forall i \in \Omega \quad [2.15]$$

b) Teste de exclusão

O teste de exclusão consiste em verificar se uma batelada i tem alguma relação de precedência com um intervalo de operações Ω , que não contém i . Antes de realizar o teste

de exclusão realiza-se o mesmo teste de factibilidade do intervalo de operações Ω executado no procedimento de *edge-finding*. Se a expressão 2.16 não é satisfeita, então o intervalo de tempo definido para o processamento do conjunto de bateladas Φ ($\Phi = \Omega \cup i$) é menor que a soma dos tempos de processamento de todas as bateladas pertencentes ao conjunto Φ e a situação é infactível.

$$lft_{\Phi} - ebt_{\Phi} \geq tp_{\Phi} \quad [2.16]$$

Para verificar se a batelada i precede o intervalo de operações Ω , testa-se primeiro se a batelada i não segue o intervalo de operações Ω através da expressão 2.17.

$$lft_i - ebt_{\Omega} < tp_{\Omega} + tp_i \quad \forall i \notin \Omega \quad [2.17]$$

A expressão 2.17 é similar à expressão 2.12 usada no teste de *edge-finding*. Se for identificado que a batelada i não segue o intervalo de operações Ω , então é testado se pelo menos uma das expressões 2.18 e 2.19 é satisfeita. Basta que uma das duas expressões seja verdadeira para que se possa declarar que a batelada i precede o intervalo de operações Ω .

$$lft_{\Omega} - ebt_{\Omega} < tp_{\Omega} + tp_i \quad \forall i \notin \Omega \quad [2.18]$$

$$lft_i - tp_i < ebt_k + tp_k \quad \forall k \in \Omega, i \notin \Omega \quad [2.19]$$

A expressão 2.18, quando satisfeita, indica que não há a possibilidade de processar a batelada i dentro do intervalo de operações Ω , já que o tempo total necessário para processar a batelada i e as bateladas pertencentes ao intervalo de operações Ω ($tp_{\Omega} + tp_i$) é maior que o intervalo de tempo disponível para processar todas estas bateladas ($lft_{\Omega} - ebt_{\Omega}$). Como as bateladas pertencentes ao intervalo de operações Ω têm que ser processadas dentro deste intervalo de tempo, restará à batelada i ser processada antes ou depois de todas as bateladas pertencentes ao intervalo de operações Ω . Se a expressão 2.17 foi satisfeita, então pelo menos uma das bateladas pertencentes ao intervalo de operações Ω deverá ser processada depois da batelada i . Se a expressão 2.18 também for satisfeita, então todas as bateladas pertencentes ao intervalo de operações Ω deverão ser processadas depois da batelada i , ou seja, a batelada i precede o intervalo de operações Ω .

A expressão 2.19 indica que há uma relação de precedência entre a batelada i e as bateladas pertencentes ao intervalo de operações Ω , tal que a batelada i necessariamente precederá qualquer batelada k pertencente ao intervalo de operações Ω . Se a expressão 2.19 é satisfeita, então não há como processar a batelada i após a execução de qualquer batelada k .

Se juntamente com a expressão 2.17 pelo menos uma entre as expressões 2.18 e 2.19 for satisfeita, então conclui-se que a batelada i precede o intervalo de operações Ω . Em consequência disto, o final da janela de tempo da batelada i (lft_i) deverá ser menor ou igual ao final do intervalo de operações Ω (lft_{Ω}) menos o tempo total necessário para processar todas as bateladas pertencentes a Ω (tp_{Ω}), como mostra a expressão 2.20.

$$lft_i = \min(lft_i, lft_{\Omega} - tp_{\Omega}) \quad \forall i \notin \Omega \quad [2.20]$$

Se a expressão 2.17 for verdadeira mas as expressões 2.18 e 2.19 forem falsas, então conclui-se que a batelada i não segue o intervalo de operações Ω , ou seja, que há pelo menos uma batelada k , pertencente ao intervalo de operações, que deve ser processada após o processamento da batelada i . Se a batelada i não segue o intervalo de operações Ω , o seu final de janela de tempo (lft_i) será dado pela expressão 2.21.

$$lft_i = \min(lft_i, \max_{k \in \Omega} (lft_k - tp_k)) \quad \forall i \notin \Omega \quad [2.21]$$

O fluxograma a seguir resume os passos dados para testar se uma batelada i ($i \notin \Omega$) precede um intervalo de operações Ω .

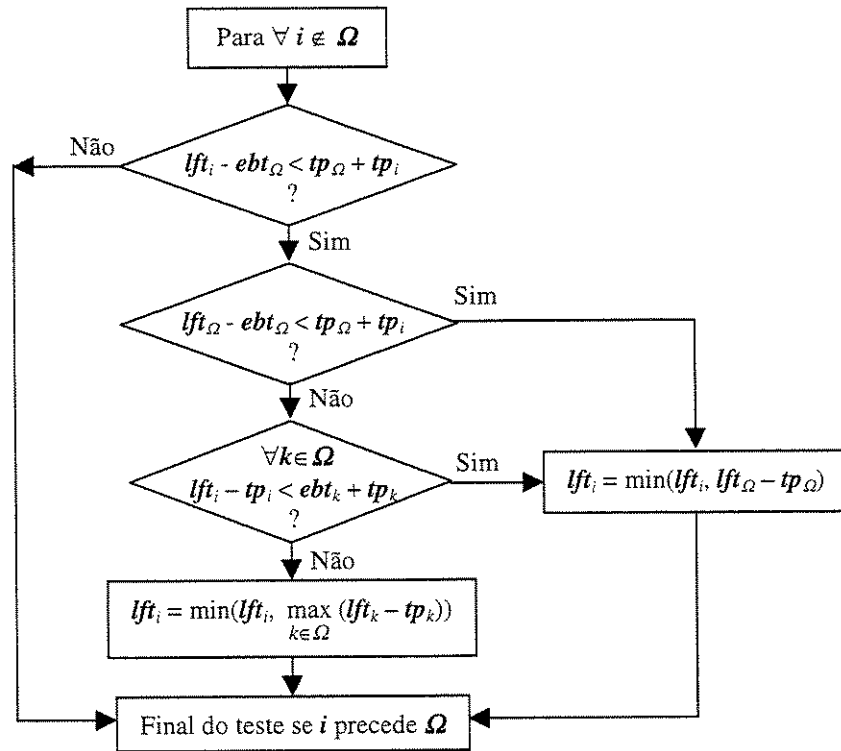


Figura 2.15 – Teste se a batelada i ($i \notin \Omega$) precede Ω .

O procedimento para determinar se uma operação i segue um intervalo de operações Ω é análogo ao que foi apresentado para testar se uma operação i segue um intervalo de operações Ω . O intervalo de operações Ω não segue a batelada i se a expressão 2.22 é verdadeira.

$$lft_\Omega - ebt_i < tp_\Omega + tp_i \quad \forall i \notin \Omega \quad [2.22]$$

Se for identificado que o intervalo de operações Ω não segue a batelada i , então testa-se se pelo menos uma entre expressões 2.23 e 2.24 é satisfeita. Basta que uma das duas expressões seja verdadeira para que se possa declarar que a batelada i segue o intervalo de operações Ω .

$$lft_{\Omega} - ebt_i < tp_{\Omega} + tp_i \quad \forall i \notin \Omega \quad [2.23]$$

$$lft_k - tp_k < ebt_i + tp_i \quad \forall k \in \Omega, i \notin \Omega \quad [2.24]$$

Se a batelada i segue o intervalo de operações Ω , o início da janela de tempo da batelada i (lft_i) deverá ser maior ou igual ao início do intervalo de operações Ω (lft_{Ω}) mais o tempo total necessário para processar todas as bateladas pertencentes a Ω (tp_{Ω}), como mostra a expressão 2.25.

$$ebt_i = \max(ebt_i, ebt_{\Omega} + tp_{\Omega}) \quad \forall i \notin \Omega \quad [2.25]$$

Se a expressão 2.22 for verdadeira mas as expressões 2.23 e 2.24 forem falsas, então conclui-se que o intervalo de operações Ω não segue a batelada i , ou seja, que há pelo menos uma batelada k , pertencente ao intervalo de operações, que deve ser processada antes do processamento da batelada i . Neste caso, o início de janela de tempo (ebt_i) será dado pela expressão 2.26.

$$ebt_i = \max(ebt_i, \min_{k \in \Omega} (ebt_k + tp_k)) \quad \forall i \notin \Omega \quad [2.26]$$

O fluxograma da figura 2.16 resume o teste se uma operação i segue um intervalo de operações Ω .

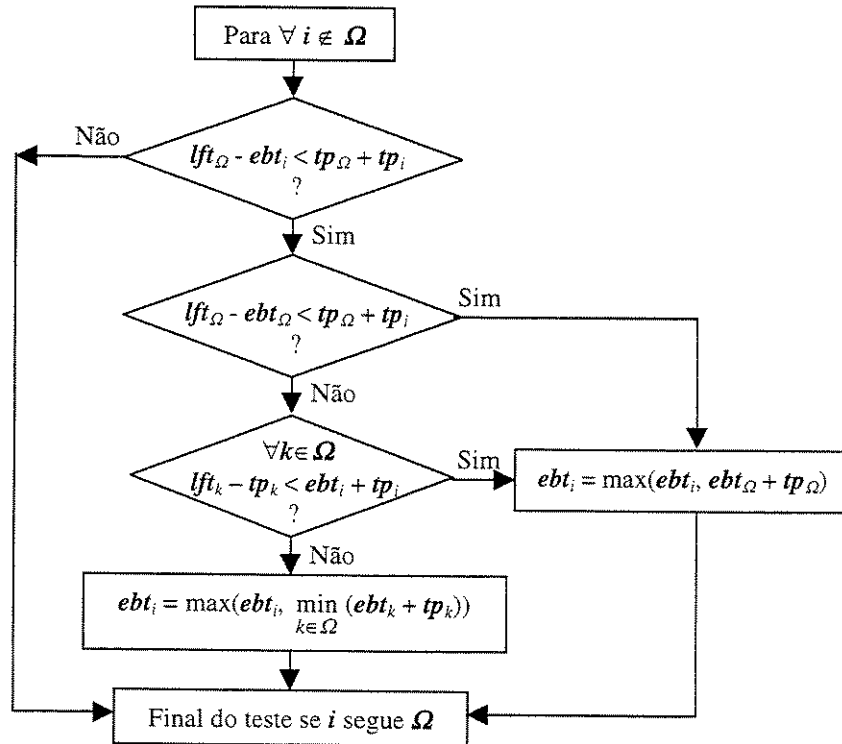


Figura 2.16 – Teste se a batelada i ($i \notin \Omega$) segue Ω .

2.6.2. Propagação das restrições de armazenagem (Gimeno *et al.*, 2000)

Esta fase de propagação de restrições tem como objetivo determinar as reduções nas janelas de tempo das bateladas impostas por condições de armazenagem restritas.

a) Propagação de modificações nas janelas por restrição de armazenagem ZW

Se duas operações i e k , respectivamente, geram e consomem um estado que não pode ser armazenado (ZW), os inícios e finais das janelas de tempo da mesma batelada b das duas operações estão interligados para respeitar a restrição de armazenagem.

As condições a serem satisfeitas pela restrição ZW são:

$$ebt_{bk} = ebt_{bi} + tp_i \quad [2.27]$$

$$lft_{bi} = lft_{bk} - tp_k \quad [2.28]$$

Se qualquer batelada da operação k (i) tem o seu final (ou início) de janela de tempo modificado, procede-se à adequação do final (ou início) de janela da batelada da operação i (k). A figura 2.17 mostra a propagação da antecipação no final da janela de tempo da batelada b da operação k para o final da janela de tempo da batelada b da operação i . As janelas de tempo modificadas são apresentadas em linhas pontilhadas nesta figura.

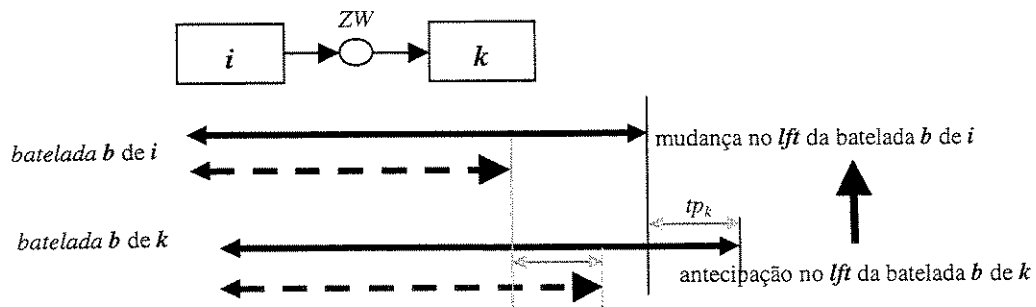


Figura 2.17. Propagação da antecipação do lft da batelada b da operação k .

b) Propagação de modificações nas janelas por restrição de armazenagem NIS

Um estado s com restrição de armazenagem NIS só pode ser armazenado no processador que o gera. Considerando-se que a operação i , atribuída ao processador j , gera o estado s e que a operação k consome este estado, apresentam-se abaixo as quatro possibilidades de modificações nas janelas de tempo:

1. Se houver uma antecipação no final da janela de tempo da batelada b da operação produtora i , o final da janela de tempo de uma batelada b da operação consumidora k estará limitado pela possibilidade de armazenamento do estado gerado. O

armazenamento está limitada pela disponibilidade do processador que, por sua vez, está limitado pelo último instante em que o processamento da batelada $b+1$ da operação i pode ser iniciado ($lbt_{b+1,i}$), dado que todas as bateladas desta operação utilizam o mesmo processador. Portanto, o final da janela de tempo da batelada b da operação k é definido impondo que o lbt_{bk} seja menor ou igual ao $lbt_{b+1,i}$, como mostra a figura 2.18.

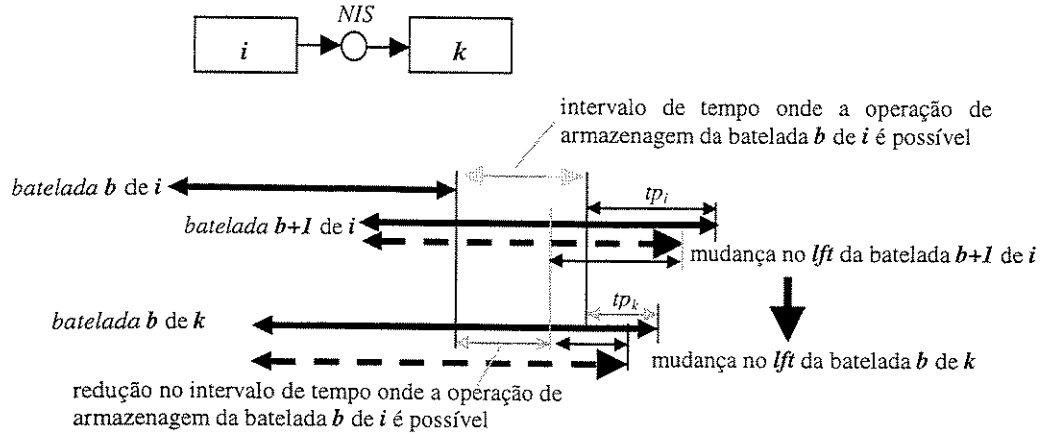


Figura 2.18. Propagação da antecipação no lft da batelada $b+1$ da operação i .

2. A existência de um intervalo de ocupação obrigatória no processador j limita a possibilidade de armazenagem dentro deste processador. Se houver uma antecipação no final da janela de tempo da batelada b da operação produtora i para um instante anterior ao início do intervalo de ocupação obrigatória, então a operação de armazenagem só poderá ocorrer até o instante de início do intervalo de ocupação obrigatória. Com isso, o último instante em que o processamento da batelada b da operação consumidora k pode ser iniciado (lbt_{bk}) deverá ser menor ou igual ao instante de início deste intervalo de ocupação obrigatória, como mostra a figura 2.19.

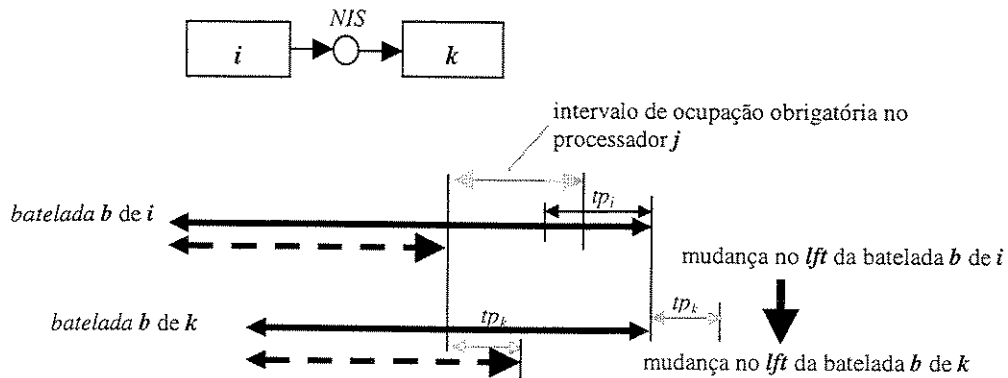


Figura 2.19. Propagação da antecipação no lft da batelada b da operação i .

3. A existência de um intervalo de ocupação obrigatória no processador j e de um aumento do início da janela de tempo da batelada b da operação consumidora k para um instante maior que o instante de início do intervalo de ocupação obrigatória provoca um aumento no início da janela de tempo da batelada b da operação produtora i . A impossibilidade de execução da operação de armazenamento do estado s durante o intervalo de ocupação obrigatória aumenta o início da janela de tempo da batelada b da operação i para o final do intervalo de ocupação obrigatória. Este caso assemelha-se ao caso anterior, analisando as implicações de um intervalo de ocupação obrigatória sobre os inícios das janelas de tempo.
4. Se o final da janela de tempo da batelada b da operação produtora i (lft_{bi}) é menor que o início da janela de tempo da mesma batelada da operação k (ebt_{bk}), haverá a necessidade de armazenamento que necessariamente ocupará o processador j no intervalo de lft_{bi} até ebt_{bk} . Por isso, nenhuma outra batelada poderá ser realizada neste intervalo no processador j porque isto implicaria na impossibilidade de armazenamento. Qualquer batelada executada no processador j terá o início da sua janela de tempo avançado se ele for menor que ebt_{bk} e se o início da janela desta batelada não permite o seu processamento antes de lft_{bi} . Da mesma forma, qualquer batelada executada no processador j terá o final da sua janela de tempo antecipado se ele for maior que lft_{bi} e se não é possível processar esta batelada após ebt_{bk} .

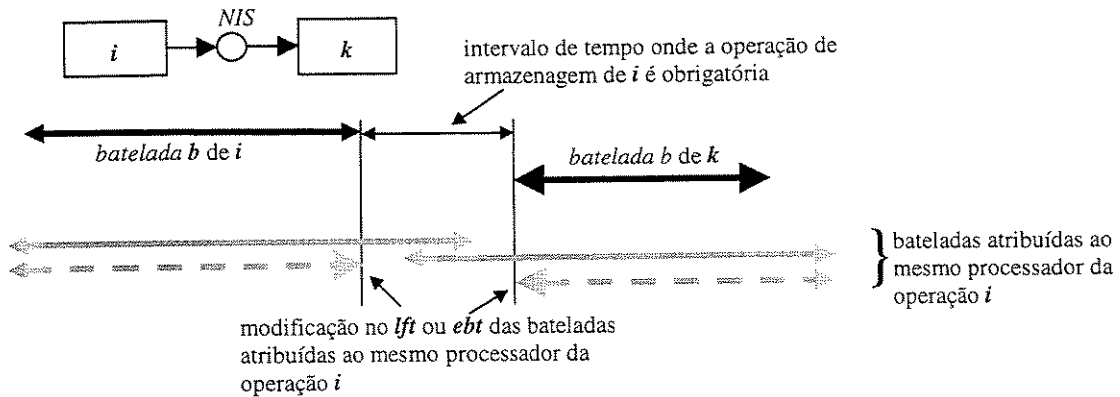


Figura 2.20. Propagação da operação de armazenamento obrigatória.

c) Propagação de modificações nas janelas por restrição de armazenagem FIS

Os inícios e finais das janelas de tempo das bateladas de duas operações que, respectivamente, geram e consomem um estado com restrição de armazenagem FIS devem satisfazer o limite de armazenagem deste estado. As restrições sobre os inícios e finais de janelas de tempo são obtidos estabelecendo as condições de que tanto os perfis de produção e consumo mais cedo como os perfis de produção e consumo mais tarde devem satisfazer a capacidade de armazenagem. Supondo uma situação factível do ponto de vista de armazenagem, a propagação de restrições será ativada por: i) um aumento no início da janela de uma batelada consumidora de um estado com restrição de armazenagem FIS, e/ou

ii) uma antecipação do final da janela de uma batelada produtora de um estado com restrição de armazenagem FIS.

Na primeira situação, o aumento do início da janela de uma batelada consumidora atrasa o perfil de consumo mais cedo, o que pode fazer o limite de capacidade FIS ser ultrapassado. Se isto acontece, o perfil de produção mais cedo deve também ser atrasado, implicando no aumento do início da janela de tempo de uma ou mais bateladas produtoras. Da mesma forma, a antecipação do final de uma janela de tempo de uma batelada produtora provoca uma antecipação no perfil de produção mais tarde, o que pode levar a uma antecipação do final da janela de tempo de uma ou mais bateladas consumidoras.

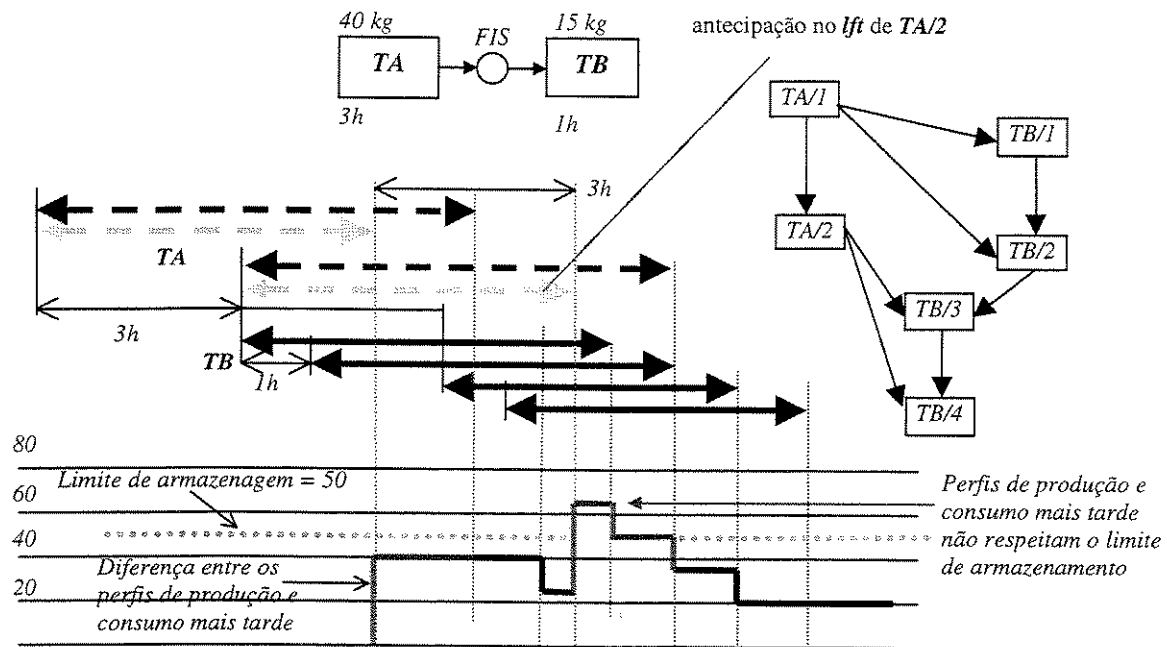


Figura 2.21a. Antecipação no lft da batelada $TA/2$.

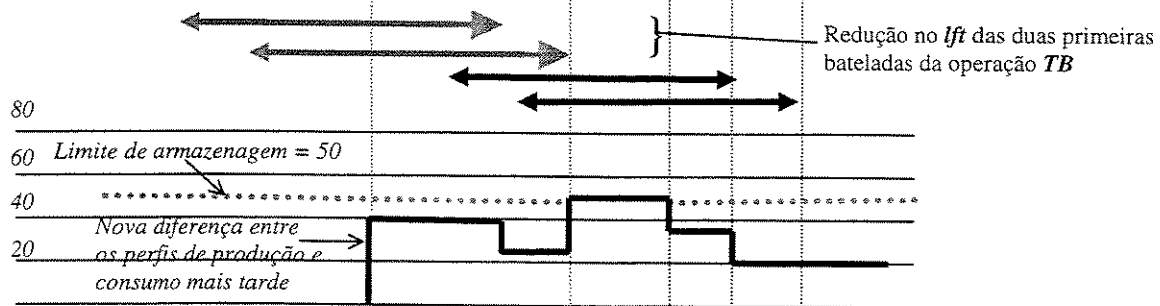


Figura 2.21b. Propagação da modificação no lft da batelada $TA/2$ para $TB/1$ e $TB/2$.

A figura 2.21a apresenta as janelas de tempo das bateladas de duas operações TA e TB . Ao lado das janelas de tempo são apresentadas as relações de precedência devido ao balanço de massa e o ordenamento de bateladas de uma mesma operação. Abaixo das

janelas de tempo é apresentada a diferença entre os perfis de produção e consumo mais tarde. A antecipação do final da janela de tempo da batelada $TA/2$ (tracejado na figura) impõe a antecipação do final da janela de tempo da batelada $TA/1$, devido à propagação por ordenamento de bateladas de uma mesma operação. Como resultado destas modificações nas janelas de tempo das bateladas de TA , os perfis de produção e consumo mais tarde não respeitam mais o limite de capacidade, forçando a propagação da antecipação do final da janela de tempo da batelada $TA/2$ para os finais das janelas de tempo das duas primeiras bateladas de TB , como mostra a figura 2.21b.

2.6.3. Propagação por ordenamento de bateladas de uma mesma operação

As bateladas sucessivas de uma mesma operação são processadas no mesmo processador. Pode-se exigir então que os instantes de início mais cedo (*ebt*) estejam espaçados de um tempo igual ao tempo de processamento. O mesmo ocorre com os instantes de término mais tarde (*lft*). Este mecanismo de propagação é ativado pelo aumento de um instante de início mais cedo (*ebt*) ou pela antecipação de um instante de término mais tarde das bateladas (*lft*).

2.6.4. Propagação por balanço de massa

As relações de precedência (*pegging*) obtidas na fase de explosão interligam as bateladas produtoras e consumidoras de um mesmo estado. Qualquer aumento do instante de início mais cedo de uma batelada produtora deve ser propagada para o instante de início mais cedo das bateladas consumidoras, ligadas àquela através de relações de precedência, podendo originar um aumento desses últimos. O mesmo ocorre quando há a antecipação do instante de término mais tarde de uma batelada consumidora.

Esta propagação de restrições é implementada nos sistemas comerciais (ILOG, 1997) através das relações de precedência acima mencionadas, que são dados de entrada do problema. No sistema desenvolvido estas relações de precedência são obtidas na fase de explosão.

2.6.5. Funcionamento conjunto dos procedimentos de propagação de restrições

Após a determinação das janelas de tempo iniciais pelos processos de explosão, verifica-se se estas janelas de tempo satisfazem as restrições de disponibilidade dos processadores e as restrições de armazenagem. Sempre que se realiza a propagação das restrições de armazenagem, realiza-se também a propagação devido ao ordenamento entre bateladas sucessivas de uma mesma operação e a propagação por balanço de massa.

A figura 2.22 apresenta o funcionamento conjunto dos mecanismos de propagação das restrições. Neste procedimento, após cada propagação, realiza-se a análise de

factibilidade de toda janelas de tempo modificadas. Os passos condicionais 4, 6, 8 e 9 desta figura só são executados se não for detectada qualquer janela de tempo inactivável.

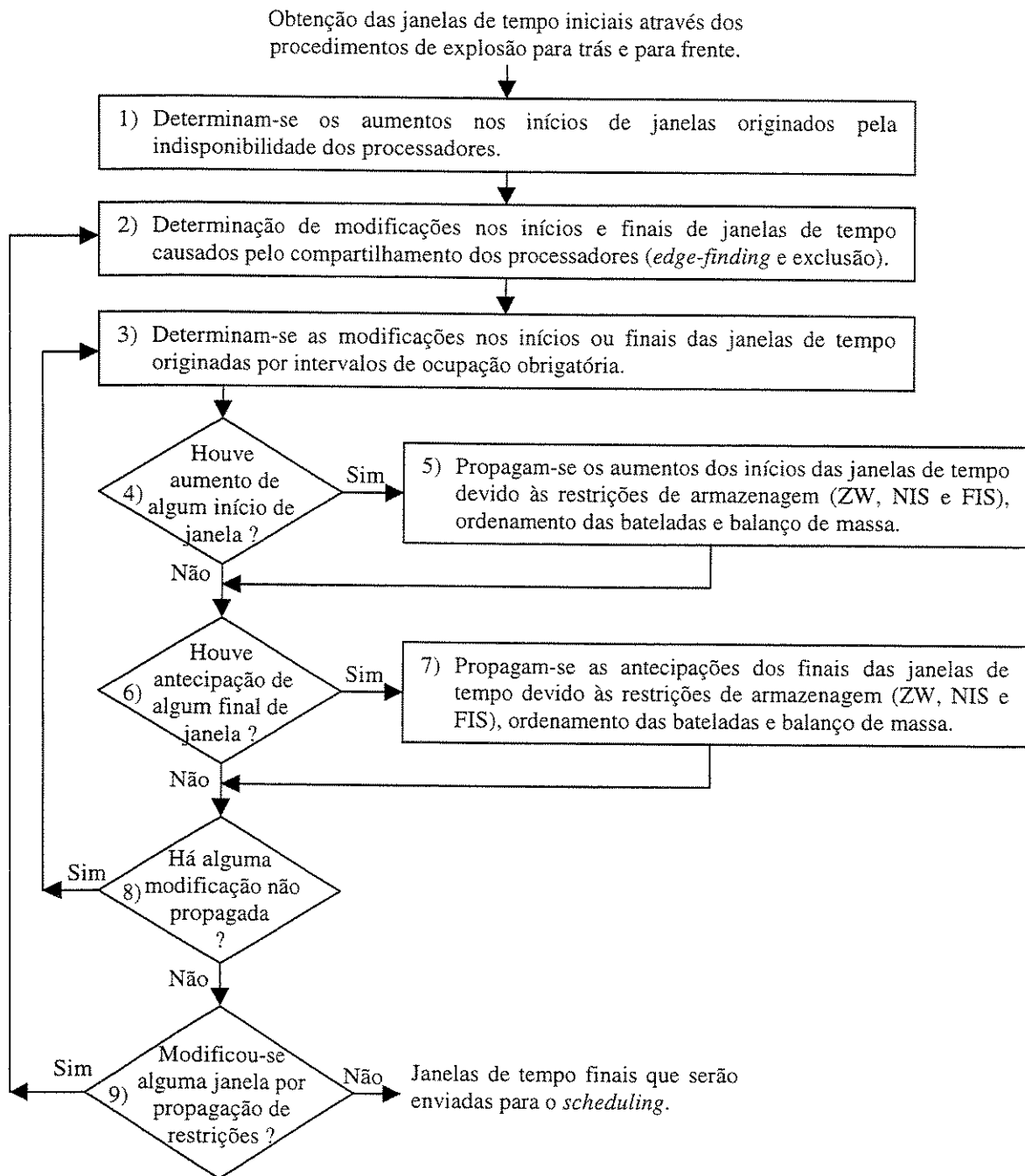


Figura 2.22. Propagação das restrições.

Os passos 1, 2 e 3 são executados inicialmente sobre as janelas de tempo resultantes da fase de explosão. A seguir, as diferentes técnicas de propagação são ativadas pelas modificações nas janelas de tempo. O passo 4 provoca a execução da propagação das modificações de todas as janelas de tempo que tiveram o seu início aumentado. O passo 6

faz o mesmo para todas as janelas de tempo que tiveram o seu final antecipado. O passo 8 provoca a busca por novos intervalos de ocupação obrigatória caso os passos 5 e/ou 7 tenham definido alguma modificação de janela de tempo. Os procedimentos de propagação das modificações nas janelas de tempo são repetidos até que nenhuma nova modificação seja encontrada (passo 9). Os procedimentos de propagação das restrições implementados nos passos 5 e 7 são apresentados na tabela 2.6.

Tabela 2.6. Procedimentos de propagação das restrições implementados.

ativada pelo	tipo de propagação	ativada por	modificação possível
passo 5	restrição de armazenamento ZW	aumento do <i>ebt</i> da batelada consumidora	aumento do <i>ebt</i> da batelada produtora
passo 7	restrição de armazenamento ZW	antecipação do <i>lft</i> da batelada produtora	antecipação do <i>lft</i> da batelada consumidora
passo 7	restrição de armazenamento NIS (caso 1 e 2 nas páginas 39 e 40)	antecipação do <i>lft</i> da batelada produtora	antecipação do <i>lft</i> da batelada consumidora
passo 5	restrição de armazenamento NIS (caso 3 na página 41)	aumento do <i>ebt</i> da batelada consumidora	aumento do <i>ebt</i> da batelada produtora
passo 5 e passo 7	restrição de armazenamento NIS (caso 4 na página 41)	operação obrigatória de armazenamento no processador	aumento do <i>ebt</i> ou antecipação do <i>lft</i> das bateladas que contêm o intervalo de tempo onde ocorre a operação de armazenamento
passo 7	restrição de armazenamento FIS	antecipação do <i>lft</i> da batelada produtora	antecipação do <i>lft</i> da batelada consumidora
passo 5	restrição de armazenamento FIS	aumento do <i>ebt</i> da batelada consumidora	aumento do <i>ebt</i> da batelada produtora
passo 5	ordenamento das bateladas de uma operação	aumento do <i>ebt</i> da batelada <i>b</i>	aumento do <i>ebt</i> das bateladas $b' > b$
passo 7	ordenamento das bateladas de uma operação	antecipação do <i>lft</i> da batelada <i>b</i>	antecipação do <i>lft</i> das bateladas $b' < b$
passo 5	balanço de massa	aumento do <i>ebt</i> da batelada <i>b</i>	aumento do <i>ebt</i> da batelada <i>b'</i> que consome a batelada <i>b</i> (<i>b'</i> e <i>b</i> estão ligadas por relação de precedência)
passo 7	balanço de massa	antecipação do <i>lft</i> da batelada <i>b'</i>	antecipação do <i>lft</i> da batelada <i>b</i> que viabiliza a produção da batelada <i>b'</i> (<i>b'</i> e <i>b</i> estão ligadas por relação de precedência)

2.6.6. Cálculo de crucialidades dos processadores

As crucialidades dos processadores permitem que se obtenha uma medida do carregamento dos processadores, já que permitem a identificação dos instantes em que há uma maior disputa pelos processadores e em quais processadores esta disputa é maior. Keng *et al.* (1988) e Sadeh (1991) apresentaram procedimentos distintos para a determinação das crucialidades (chamada de demanda agregada por Sadeh) dos processadores.

Keng associa a cada batelada *b* de uma operação *i* uma criticalidade, que é uma medida da possibilidade de ocupação pela batelada *b* do intervalo de tempo constituído pela sua janela de tempo. A criticalidade de cada batelada é dada por:

$$\text{critk}_{bi} = \text{tp}_i / \text{dw}_{bi} \quad \forall i, b \in \text{Bat}_i \quad [2.29]$$

onde:

critk_{bi} criticalidade da batelada *b* da operação *i* proposta por Keng *et al.* (1988).
 dw_{bi} tamanho da janela de tempo da batelada *b* da operação *i*, dada por: $\text{lft}_{bi} - \text{ebt}_{bi} + 1$.

O procedimento de cálculo das demandas agregadas proposto por Sadeh inicia-se com a determinação do que o autor chamou de probabilidade subjetiva de cada batelada, que é definida como: $\sigma_{bi} = 1/NAB_{bi}$, onde NAB_{bi} é o número de alocações possíveis da batelada b da operação i dentro da sua janela de tempo. O cálculo do número de alocações possíveis (NAB_{bi}) pressupõe uma discretização uniforme do tempo, onde Δt identifica o intervalo de discretização. O horizonte de planejamento é dividido em um número de intervalos de igual duração, chamados de *slots*.

A partir da probabilidade subjetiva calcula-se a demanda individual de cada batelada proposta por Sadeh. A demanda individual é similar ao conceito de criticalidade e é dada por:

$$crits_{bit} = n_{bit} * \sigma_{bi} \quad \forall i, b \in Bat_i, t \quad [2.30]$$

onde:

- $crits_{bit}$ Demanda individual da batelada b da operação i no *slot* t .
- n_{bit} Número de alocações possíveis dentro da janela de tempo da batelada b da operação i tal que o *slot* t seja ocupado por esta batelada.
- σ_{bi} Probabilidade subjetiva da batelada b da operação i .

O cálculo da função de crucialidade proposta por Keng é similar ao cálculo da demanda agregada proposta por Sadeh. A crucialidade (demanda agregada) de um processador j no *slot* t é calculada identificando-se as bateladas processadas em j cujas janelas de tempo contém este *slot* e somando-se as criticalidades (demandas individuais) destas bateladas. A figura 2.23 apresenta as janelas de tempo de duas bateladas (A e B) usadas para exemplificar os dois procedimentos de cálculo das crucialidades. O tempo de processamento da operação A é igual a 2 e da operação B igual a 3.

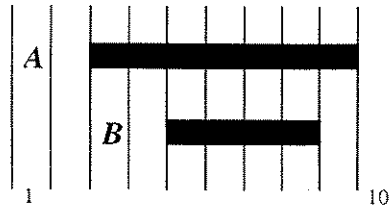


Figura 2.23. Janelas de tempo das bateladas A e B .

Pelo procedimento proposto por Keng, a criticalidade da batelada A é igual a $2/7$ em todos os *slots* pertencentes a janela de tempo desta batelada. Para a batelada B o valor da criticalidade é igual a $3/4$. A figura 2.24 apresenta o valor da crucialidade pela proposta de Keng.

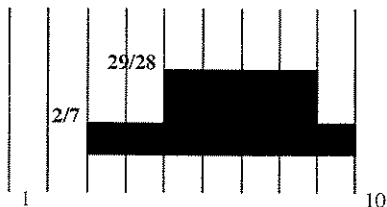


Figura 2.24. Valor da crucialidade (Keng *et al.*, 1988).

A figura 2.25 apresenta as demandas individuais das duas bateladas pela proposta de Sadeh. Estes valores são o resultado das seis possibilidades distintas de alocação da batelada **A** e de duas possibilidades distintas de alocação da batelada **B**, dentro das suas janelas de tempo.

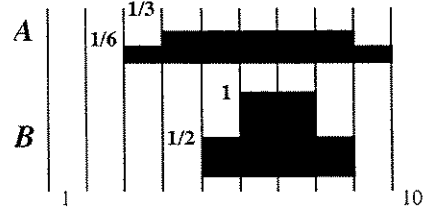


Figura 2.25. Valor da demanda individual (Sadeh, 1991).

A figura 2.26 apresenta o valor da demanda agregada pela proposta de Sadeh. As duas propostas para o cálculo de crucialidade foram implementadas no sistema de planejamento. Mas a comparação entre as figuras 2.24 e 2.26 mostra que a proposta de Sadeh identifica mais claramente em que instantes a disputa pela ocupação de cada processador é maior. Por isso, todas as vezes que se apresentar a medida do carregamento dos processadores, ela será mostrada usando-se a função de demanda agregada proposta de Sadeh.

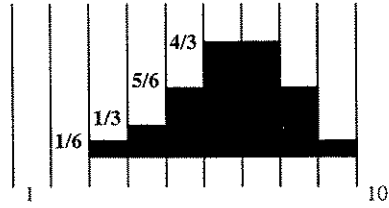


Figura 2.26. Valor das demandas agregadas (Sadeh, 1991).

A estimativa do carregamento do processadores através da demanda agregada implica na discretização do tempo e no cálculo do seu valor para cada *slot*. Considerando-se que o tempo de processamento de cada operação (p_i) é dado em número de *slots*, pode-se diminuir o trabalho de cálculo da demanda agregada utilizando-se apenas quatro pontos da demanda individual de cada batelada, como propõe Beck *et al.* (1997):

- i) No *slot* $[ebt_{bi}, ebt_{bi} + \Delta t)$ a demanda individual é igual à $1/NAB_{bi}$;
- ii) No *slot* $[ebt_{bi} + p_i, ebt_{bi} + p_i + \Delta t)$ a demanda individual é igual à $\min\{1, p_i / NAB_{bi}\}$;
- iii) No *slot* $[lft_{bi} - p_i, lft_{bi} - p_i + \Delta t)$ a demanda individual é igual à $\min\{1, p_i / NAB_{bi}\}$;
- iv) No *slot* $[lft_{bi}, lft_{bi} + \Delta t)$ a demanda individual é igual à $1/NAB_{bi}$.

Esta proposta de Beck permite que a demanda individual só seja calculada em quatro *slots* do horizonte de planejamento, mas ainda implica na discretização uniforme deste horizonte para que se calcule o número de alocações possíveis (NAB_{bi}). Esta

discretização foi eliminada através do procedimento descrito a seguir. O número de alocações possíveis da batelada b da operação i dentro da sua janela de tempo é dado por:

$$NAB_{bi} = (Dw_{bi} / \Delta t) - (tp_i / \Delta t) + 1 \quad [2.31]$$

onde:

Δt Duração de um *slot* em unidades de tempo.
 tp_i Tempo de processamento da operação i em unidades de tempo.
 $Dw_{bi} = lft_{bi} - ebt_{bi}$.

Se $(lft_{bi} - tp_i) > (ebt_{bi} + tp_i)$, o número de alocações possíveis que ocupam o intervalo $[lft_{bi} - tp_i, ebt_{bi} + tp_i]$ é dado por:

$$n_{bi} = tp_i / \Delta t \quad [2.32]$$

Se $(lft_{bi} - tp_i) < (ebt_{bi} + tp_i)$, o intervalo $[lft_{bi} - tp_i, ebt_{bi} + tp_i]$ é um intervalo de ocupação obrigatória e, neste caso:

$$n_{bi} = NAB_{bi} \quad [2.33]$$

Então, a demanda individual é dada por:

$$crits_{bit} = \frac{\frac{tp_i}{\Delta t}}{\frac{Dw_{bi}}{\Delta t} - \frac{tp_i}{\Delta t} + 1} \quad \text{ou} \quad crits_{bit} = 1$$

Na primeira situação é possível calcular o valor limite quando Δt tende a zero, obtendo-se:

$$crits_{bit} = \frac{tp_i}{Dw_{bi} - tp_i} \quad [2.34]$$

Utiliza-se, então, como demanda individual a seguinte função:

- i) No ponto ebt_{bi} a demanda individual é igual à zero;
- ii) No ponto $ebt_{bi} + tp_i$ a demanda individual é igual à $\min\{1, tp_i / (Dw_i - tp_i)\}$;
- iii) No ponto $lft_{bi} - tp_i$ a demanda individual é igual à $\min\{1, tp_i / (Dw_i - tp_i)\}$;
- iv) No ponto lft_{bi} a demanda individual é igual zero.

Com isto a demanda individual dada na figura 2.25 passa a ser indicada como na figura 2.27. Neste caso, as demandas agregadas só são calculadas nos pontos onde há uma alteração nas demandas individuais, como mostra a figura 2.28.

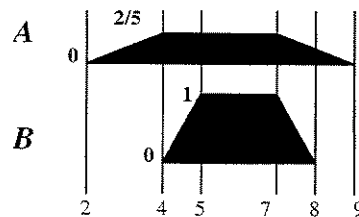


Figura 2.27. Valor das demandas individuais.

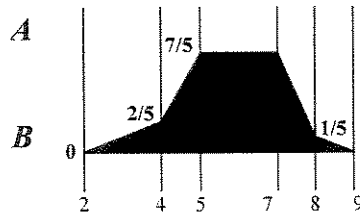


Figura 2.28. Valor das demandas agregadas.

As figuras 2.29 e 2.30 mostram as demandas agregadas para os exemplos 1 e 2, respectivamente, quando a disponibilidade de matérias primas é dada pelas tabelas 2.3 e 2.4, respectivamente.

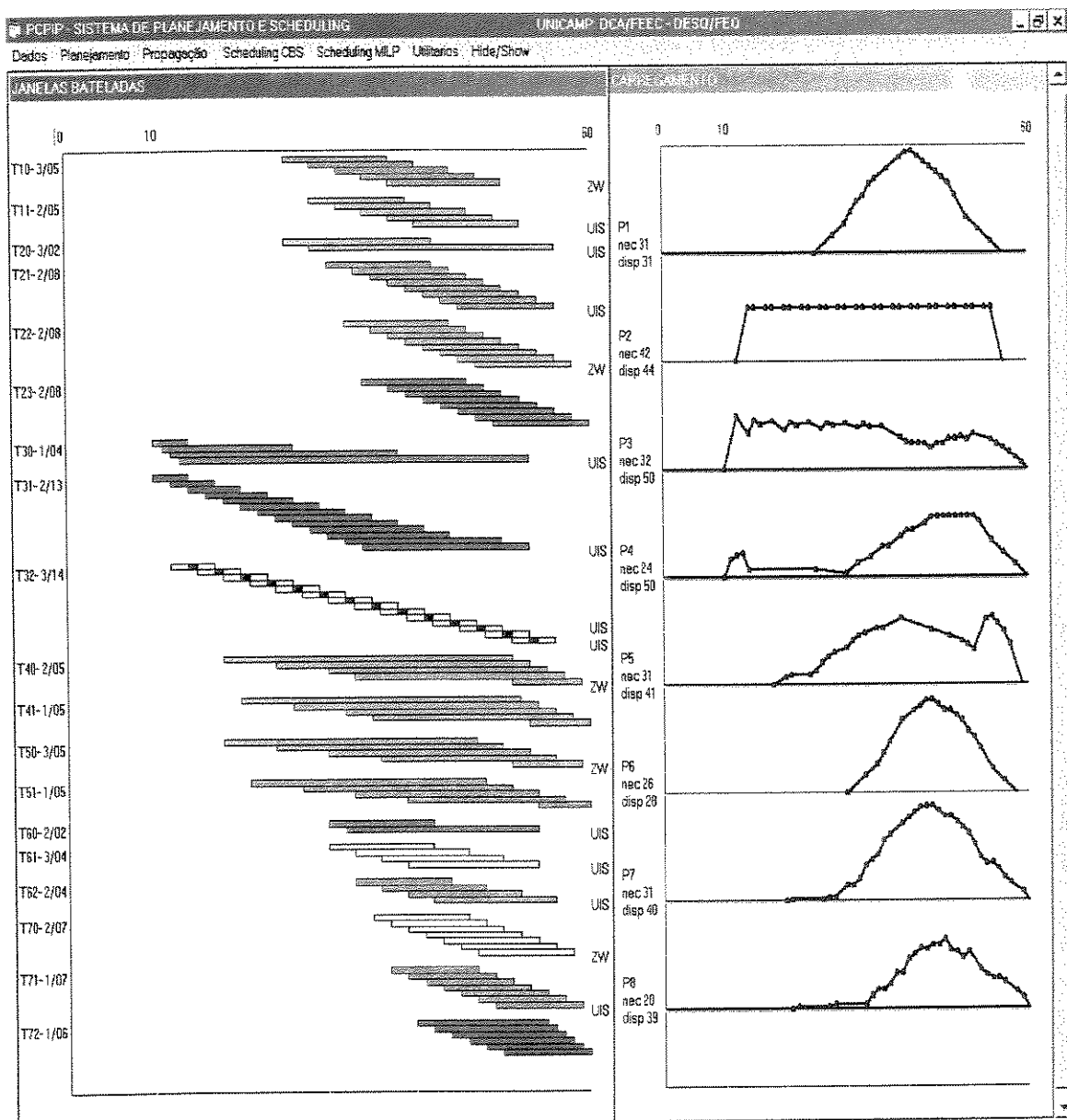


Figura 2.29. Janelas de tempo e demandas agregadas para o exemplo 1.

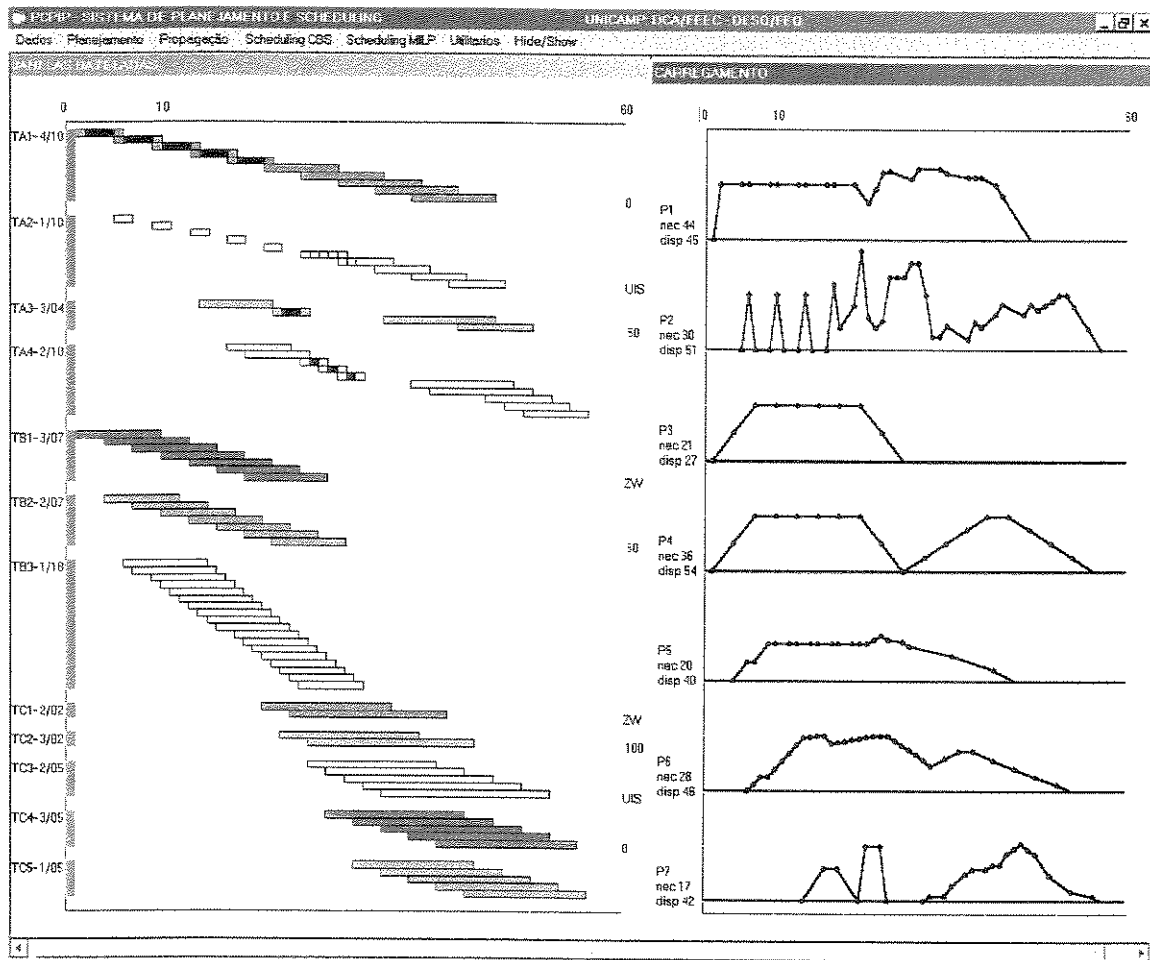


Figura 2.30. Janelas de tempo e demandas agregadas para o exemplo 2.

2.6.7. Aspectos da implementação

A propagação das restrições disjuntivas é repetida sempre que são identificadas novas modificações nas janelas de tempo como mostra a figura 2.17. Esta análise pode ter um tempo de execução significativo. Isto leva, por exemplo, a que tanto o *software* ILOG Scheduler como o *software* Claire Schedule 1.0 permitam que o usuário desabilite a sua execução. No sistema de planejamento proposto foi implementado um teste adicional que só permite a propagação das restrições disjuntivas quando as condições abaixo são satisfeitas. Esta restrição na propagação das restrições é obviamente heurística mas não limita a obtenção da solução ótima. Ela apenas limita a intensidade da propagação de restrições.

1. Utiliza-se um valor de crucialidade mínimo (φ). Os intervalos de operações (na seção 2.6.1) são gerados apenas a partir de pares de janelas de tempo que contenham pontos com crucialidade maior que o valor mínimo. Esta condição é usada para que só haja propagação das restrições disjuntivas se a disputa pela ocupação dos processadores é grande. Isto porque intervalos de operações com baixa crucialidade têm possibilidade baixa de gerar modificações nas janelas de tempo.

2. Utiliza-se um valor mínimo (κ) para a média das criticalidades das bateladas contidas no intervalo de operações. Após a execução do passo 1, verifica-se para cada intervalo de operações, que satisfaz o passo 1, se a média das criticalidades das bateladas que compõem este intervalo de operações é maior que κ . Uma média das criticalidades baixa indica a existência de janelas de tempo grandes em relação ao seu tempo de processamento e, neste caso, as modificações nas janelas de tempo são essencialmente produzidas pela propagação dos intervalos de ocupação obrigatória da(s) batelada(s) que origina(m) a crucialidade alta.
Se a janela de tempo de alguma batelada impõe a alocação desta batelada, a criticalidade terá valor unitário no intervalo definido pela janela de tempo. Por isso, estas bateladas com alocação definida não são incluídas no cálculo da média das criticalidades.
3. Utilização de uma folga máxima relativa (ψ) para o intervalo de operações. A folga do intervalo de operações (*slack*) é obtida a partir da diferença entre o intervalo de tempo definido pelo intervalo de operações e a soma dos tempos de processamento das bateladas contidas neste intervalo. A divisão desta folga pela soma dos tempos de processamento das bateladas contidas neste intervalo fornece a folga relativa do intervalo de operações. Se este valor é alto, pode haver tempo disponível para a execução de outras bateladas dentro do intervalo de operações e a possibilidade de modificação de alguma janela de tempo será pequena. Esta heurística é complementar à utilizada no item 1 permitindo detectar a situação em que a crucialidade é alta pela superposição de uma quantidade grande de bateladas com criticalidade baixa.
4. A propagação das restrições disjuntivas não é executada se as bateladas envolvidas pertencem à mesma operação.

Outros testes para acionar a propagação das restrições disjuntivas poderiam ser usados. A definição de critérios para a execução desta propagação é um tema ainda em aberto e certamente novas contribuições deverão surgir no futuro. Da mesma forma, a definição de valores para o valor mínimo para as crucialidades (ϕ), para o valor mínimo para a média das criticalidades das bateladas contidas no intervalo de operações (κ) e para a folga máxima relativa (ψ) estão ainda em aberto. No sistema de planejamento que foi implementado adotou-se os valores $\phi = 0,8$; $\kappa = 0,2$ e $\psi = 1,5$. Estes valores apresentaram uma redução sensível no tempo de execução para os dois exemplos utilizados.

2.7. Exemplos de aplicação

Nesta seção apresentam-se os resultados obtidos para os dois exemplos do Anexo 1.

2.7.1. Exemplo 1

Considera-se a demanda dada na Tabela 2.1a e, inicialmente, o plano de disponibilidade de matérias primas, a Figura 2.6 é repetida a seguir para facilitar a visualização do exemplo 1.

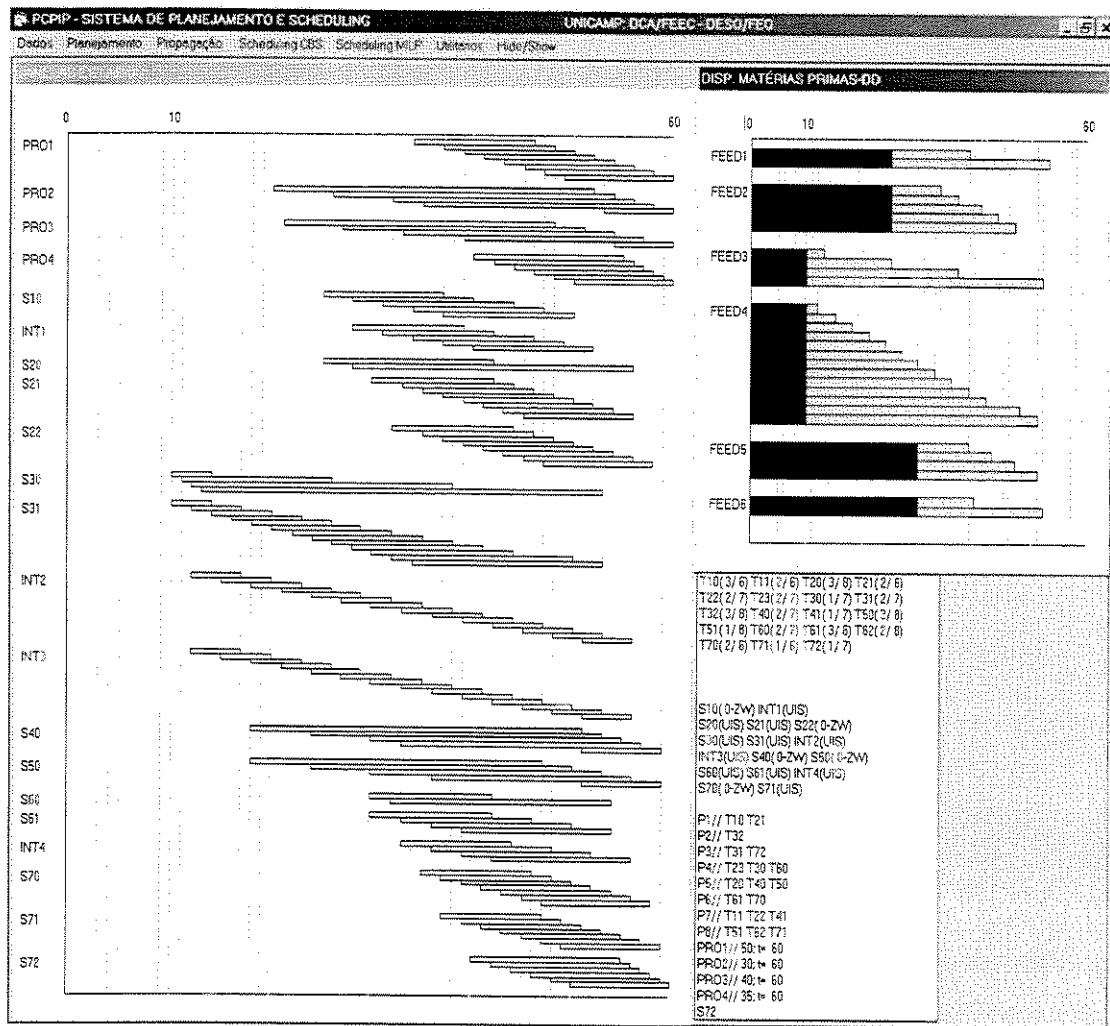


Figura 2.6. Janelas de tempo e disponibilidade das matérias primas para o exemplo 1.

As janelas das bateladas antes da fase de propagação das restrições foram apresentadas na Figura 2.12, repetida na página seguinte. Como resultado da fase de propagação das restrições obtém-se as janelas representadas na Figura 2.31. Se uma batelada possui um intervalo de ocupação obrigatória, os intervalos de tempo da janela de tempo desta batelada que pertencem ao intervalo de ocupação obrigatória são identificados em preto, como ocorre com as bateladas **T10/1** e **2**, **T11/1** e **2**, **T21/6** até **8**, **T22/6** até **8**, **T23/6** até **8**, **T32/1** até **14** e **T40/5**. Quando a janela de tempo de uma batelada *i* contém o intervalo de ocupação obrigatória de outra batelada, os intervalos desta janela de tempo, onde ocorre o intervalo de ocupação obrigatória, são apresentados em branco para indicar que eles não estão disponíveis para o processamento da batelada *i*. Isto pode ser observado nas bateladas **T41/1** e **2**.

O efeito da propagação das restrições para o exemplo 1 pode ser visto comparando-se as janelas de tempo antes (figura 2.29) e depois (figura 2.31) da propagação. A tabela 2.7 indica o número total de possíveis alocações para todas as bateladas de cada operação. A sua redução é um indicativo da redução da dimensão do problema na fase de *scheduling*.

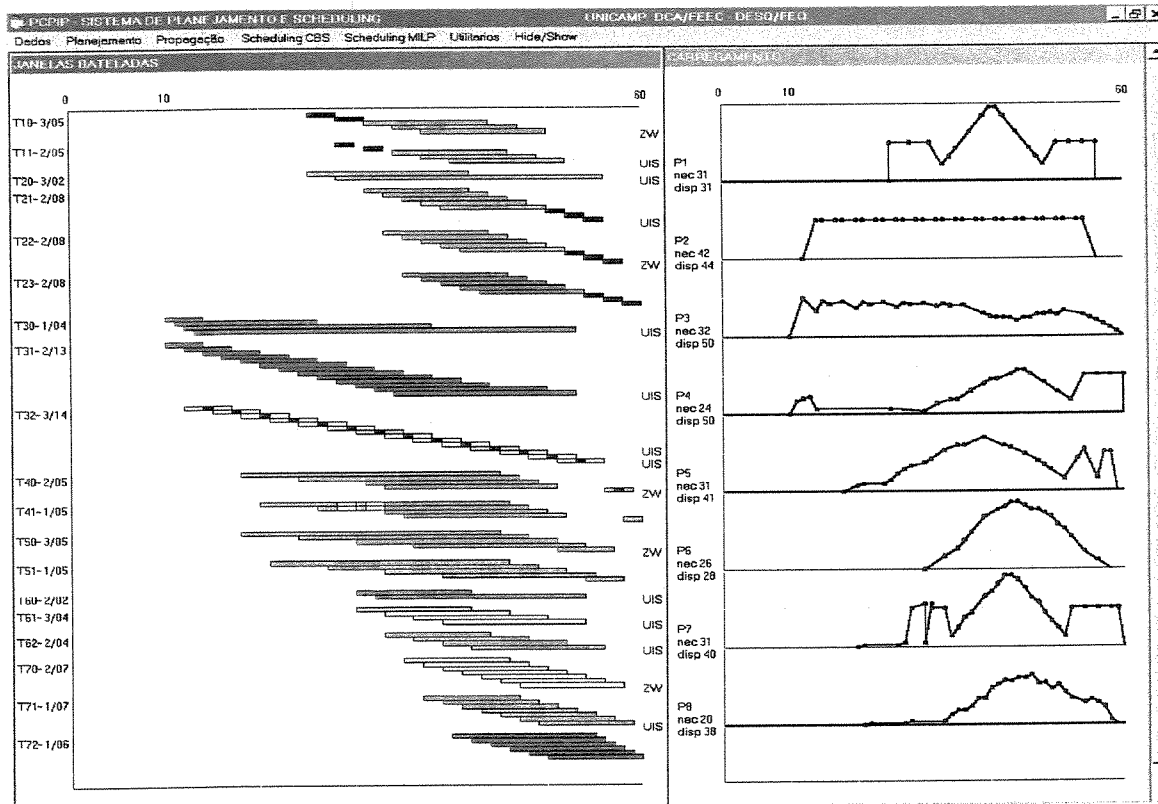
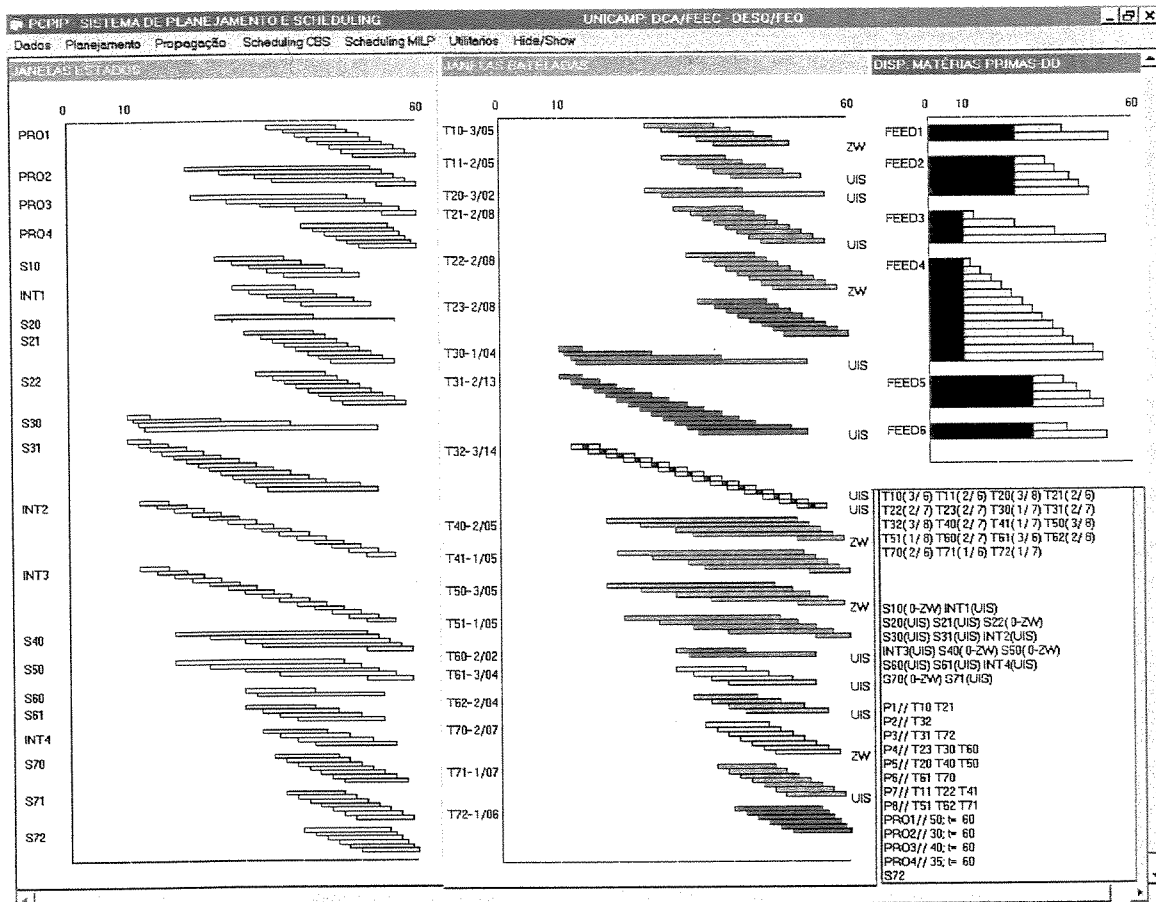


Tabela 2.7. Número total de possíveis alocações antes e depois da propagação das restrições.

batelada	nº poss. alocações antes	nº poss. alocações depois	dimi- nuição (%)	batelada	nº poss. alocações antes	nº poss. alocações depois	dimi- nuição (%)
T10/1	10	1	90	T32/6	3	3	0
T10/2	10	1	90	T32/7	3	3	0
T10/3	11	11	0	T32/8	3	3	0
T10/4	11	11	0	T32/9	3	3	0
T10/5	11	11	0	T32/10	3	3	0
T11/1	10	1	90	T32/11	3	3	0
T11/2	10	1	90	T32/12	3	3	0
T11/3	11	11	0	T32/13	3	3	0
T11/4	11	11	0	T32/14	3	3	0
T11/5	11	11	0	T40/1	32	22	31,25
T20/1	15	15	0	T40/2	28	18	35,71
T20/2	26	26	0	T40/3	24	17	29,17
T21/1	11	10	9,09	T40/4	23	17	26,09
T21/2	10	10	0	T40/5	7	2	71,43
T21/3	10	10	0	T41/1	32	22	31,25
T21/4	10	10	0	T41/2	28	18	35,71
T21/5	10	10	0	T41/3	24	17	29,17
T21/6	10	1	90	T41/4	23	17	26,09
T21/7	10	1	90	T41/5	7	2	71,43
T21/8	10	1	90	T50/1	27	25	7,41
T22/1	11	10	9,09	T50/2	24	22	8,33
T22/2	10	10	0	T50/3	21	19	9,52
T22/3	10	10	0	T50/4	18	16	11,11
T22/4	10	10	0	T50/5	6	4	33,33
T22/5	10	10	0	T51/1	27	25	7,41
T22/6	10	1	90	T51/2	24	22	8,33
T22/7	10	1	90	T51/3	21	19	9,52
T22/8	10	1	90	T51/4	18	16	11,11
T23/1	11	10	9,09	T51/5	6	4	33,33
T23/2	10	10	0	T60/1	11	11	0
T23/3	10	10	0	T60/2	21	21	0
T23/4	10	10	0	T61/1	10	10	0
T23/5	10	10	0	T61/2	11	11	0
T23/6	10	1	90	T61/3	12	12	0
T23/7	10	1	90	T61/4	13	13	0
T23/8	10	1	90	T62/1	10	10	0
T30/1	4	4	0	T62/2	11	11	0
T30/2	15	15	0	T62/3	12	12	0
T30/3	26	26	0	T62/4	13	13	0
T30/4	40	40	0	T70/1	10	10	0
T31/1	3	3	0	T70/2	10	10	0
T31/2	4	4	0	T70/3	10	10	0
T31/3	5	5	0	T70/4	10	10	0
T31/4	6	6	0	T70/5	10	10	0
T31/5	7	7	0	T70/6	10	10	0
T31/6	8	8	0	T70/7	10	10	0
T31/7	9	9	0	T71/1	10	10	0
T31/8	10	10	0	T71/2	10	10	0
T31/9	11	11	0	T71/3	10	10	0
T31/10	12	12	0	T71/4	10	10	0
T31/11	13	13	0	T71/5	10	10	0
T31/12	17	17	0	T71/6	10	10	0
T31/13	18	18	0	T71/7	10	10	0
T32/1	3	3	0	T72/1	15	15	0
T32/2	3	3	0	T72/2	14	14	0
T32/3	3	3	0	T72/3	13	13	0
T32/4	3	3	0	T72/4	12	12	0
T32/5	3	3	0	T72/5	11	11	0
				T72/6	10	10	0
				total	1399	1183	15,44

Para exemplificar os diferentes mecanismos de propagação das restrições, a seguir, inclui-se uma listagem sequencial das modificações nas janelas de tempo geradas por estes mecanismos.

Tabela 2.8. Resultado da análise de intervalos de ocupação obrigatória e da propagação das restrições.

batelada	modificação realizada	motivo da modificação na janela
T21/6	<i>ebt</i> mudado de 41 para 50	Ω precede T21/6, onde $\Omega = \{T10/1 \text{ até } 5; T21/1 \text{ até } 5\}$
T21/7	<i>ebt</i> mudado de 43 para 52	Ω precede T21/7, onde $\Omega = \{T10/1 \text{ até } 5; T21/1 \text{ até } 6\}$
T21/8	<i>ebt</i> mudado de 45 para 54	Ω precede T21/8, onde $\Omega = \{T10/1 \text{ até } 5; T21/1 \text{ até } 7\}$
T21/1	<i>ebt</i> mudado de 30 para 31	T21/1 não precede Ω , onde $\Omega = \{T10/2 \text{ até } 5; T21/1 \text{ até } 5\}$
T10/1	<i>lft</i> mudado de 37 para 28	T10/1 precede Ω , onde $\Omega = \{T10/2 \text{ até } 5; T21/1 \text{ até } 5\}$
T10/2	<i>lft</i> mudado de 40 para 31	T10/2 precede Ω , onde $\Omega = \{T10/3 \text{ até } 5; T21/1 \text{ até } 5\}$
T22/6	<i>ebt</i> mudado de 43 para 52	propagação por balanço de massa de T21/6 em T22/6
T22/7	<i>ebt</i> mudado de 45 para 54	propagação por balanço de massa de T21/7 em T22/7
T22/8	<i>ebt</i> mudado de 47 para 56	propagação por balanço de massa de T21/8 em T22/8
T23/1	<i>ebt</i> mudado de 34 para 35	propagação por balanço de massa de T22/1 em T23/1
T23/6	<i>ebt</i> mudado de 45 para 54	propagação por balanço de massa de T22/6 em T23/6
T23/7	<i>ebt</i> mudado de 47 para 56	propagação por balanço de massa de T22/7 em T23/7
T23/8	<i>ebt</i> mudado de 48 para 58	propagação por balanço de massa de T22/8 em T23/8
T11/1	<i>lft</i> mudado de 39 para 30	propagação por restrição de armazenagem ZW de T10/1 em T11/1
T11/2	<i>lft</i> mudado de 42 para 33	propagação por restrição de armazenagem ZW de T10/2 em T11/2
T41/3	<i>ebt</i> mudado de 32 para 33	intervalo de ocupação obrigatória
T41/2	<i>lft</i> mudado de 54 para 52	intervalo de ocupação obrigatória
T41/3	<i>lft</i> mudado de 56 para 52	intervalo de ocupação obrigatória
T41/4	<i>lft</i> mudado de 58 para 52	intervalo de ocupação obrigatória
T41/5	<i>ebt</i> mudado de 53 para 58	intervalo de ocupação obrigatória
T40/3	<i>ebt</i> mudado de 30 para 31	propagação por restrição de armazenagem ZW de T41/3 em T40/3
T40/5	<i>ebt</i> mudado de 51 para 56	propagação por restrição de armazenagem ZW de T41/5 em T40/5
T40/4	<i>lft</i> mudado de 57 para 51	propagação por balanço de massa de T41/4 em T40/4
T40/3	<i>lft</i> mudado de 55 para 49	propagação entre bateladas da mesma operação de T40/4 em T40/3
T40/2	<i>lft</i> mudado de 53 para 47	propagação entre bateladas da mesma operação de T40/3 em T40/2
T40/1	<i>lft</i> mudado de 51 para 45	propagação entre bateladas da mesma operação de T40/2 em T40/1
T41/1	<i>lft</i> mudado de 51 para 46	propagação por restrição de armazenagem ZW de T40/1 em T41/1
T41/2	<i>lft</i> mudado de 52 para 48	propagação por restrição de armazenagem ZW de T40/2 em T41/2
T41/3	<i>lft</i> mudado de 52 para 50	propagação por restrição de armazenagem ZW de T40/2 em T41/2
T50/5	<i>lft</i> mudado de 59 para 57	intervalo de ocupação obrigatória
T50/4	<i>lft</i> mudado de 56 para 54	propagação entre bateladas da mesma operação de T50/5 em T50/4
T50/3	<i>lft</i> mudado de 53 para 51	propagação entre bateladas da mesma operação de T50/4 em T50/3
T50/2	<i>lft</i> mudado de 50 para 48	propagação entre bateladas da mesma operação de T50/3 em T50/2
T50/1	<i>lft</i> mudado de 47 para 45	propagação entre bateladas da mesma operação de T50/2 em T50/1
T51/5	<i>lft</i> mudado de 60 para 58	propagação por restrição de armazenagem ZW de T50/5 em T51/5
T51/4	<i>lft</i> mudado de 57 para 55	propagação por restrição de armazenagem ZW de T50/4 em T51/4
T51/3	<i>lft</i> mudado de 54 para 52	propagação por restrição de armazenagem ZW de T50/3 em T51/3
T51/2	<i>lft</i> mudado de 51 para 49	propagação por restrição de armazenagem ZW de T50/2 em T51/2
T51/1	<i>lft</i> mudado de 48 para 46	propagação por restrição de armazenagem ZW de T50/1 em T51/1

A figura 2.31 mostra também, no seu lado direito, o carregamento induzido nos processadores pelas janelas de tempo, representado através das demandas agregadas (seção 2.6.6). Adicionalmente estão indicadas duas informações que podem caracterizar a folga global em cada processador: *i*) a soma dos tempos de processamento das bateladas atribuídas a cada processador (necessidade) e *ii*) o intervalo de tempo contido pelas janelas de tempo (disponibilidade). Observa-se um carregamento próximo do limite para os processadores **P1**, **P2** e **P6**. É interessante notar a complementaridade entre as informações dadas pela demanda agregada e a folga global. Por exemplo, algumas das janelas de tempo

das bateladas de *T10* e *T21*, no processador *P1*, são bem maiores que os respectivos tempos de processamento, porém, a folga total é nula.

As janelas finais na Figura 2.31 mostram diversos casos em que a janela de tempo é igual ao tempo de processamento. É o caso das bateladas *T10/1,2*, *T11/1,2*, *T21/6,7,8*, *T22/6,7,8* e *T23/6,7,8*. Nesta condição não existe qualquer flexibilidade de alocação na fase de *scheduling* para estas bateladas. O usuário pode não aceitar esta situação pouco robusta frente a perturbação na operação da planta. Neste caso, para aumentar a flexibilidade, o usuário pode redefinir, por exemplo, a disponibilidade das matérias primas *Feed1* e *Feed2*. A Figura 2.32 mostra as janelas finais obtidas antecipando em cinco unidades de tempo as disponibilidades das materiais primas acima citadas.

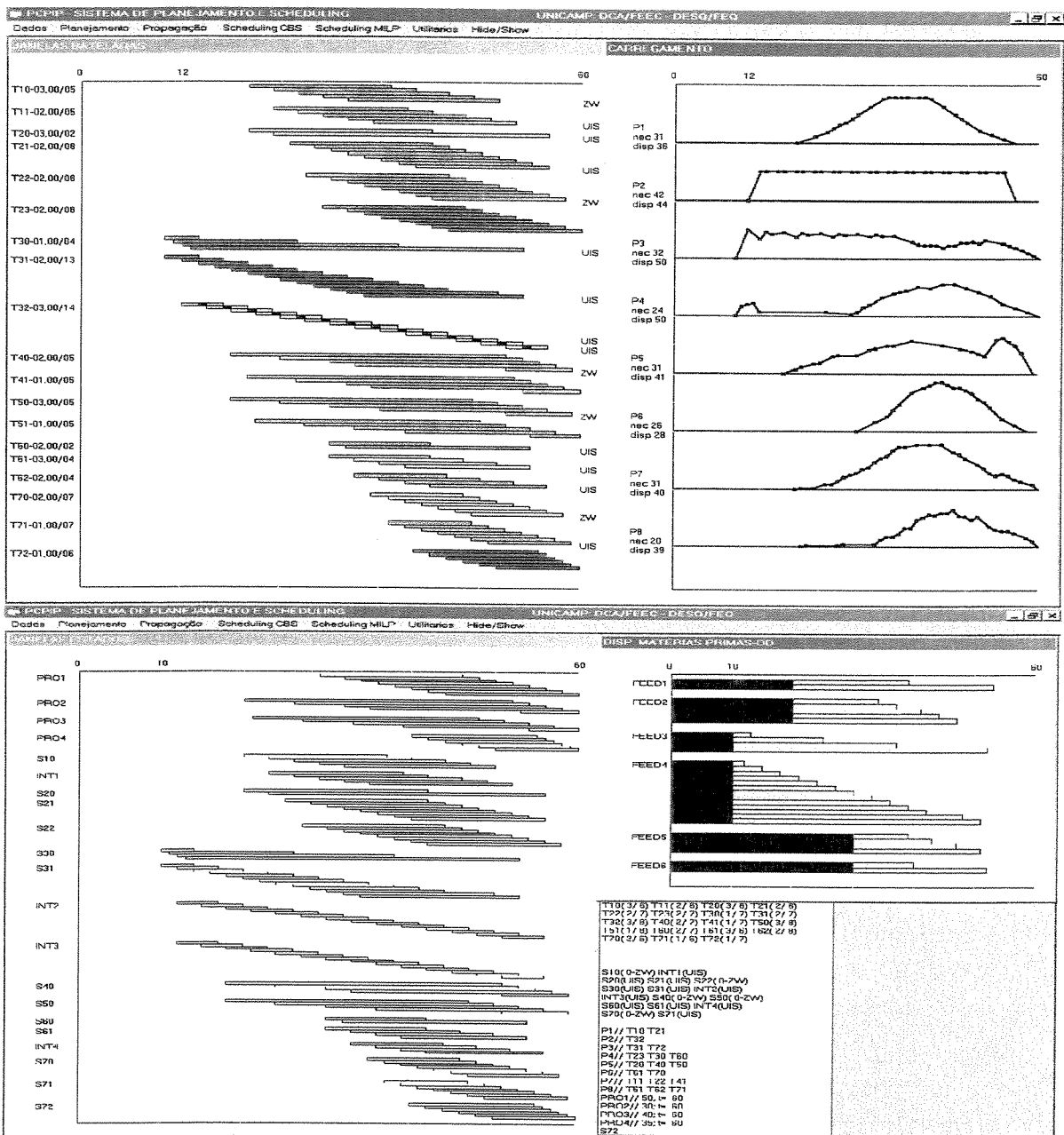


Figura 2.32. Janelas de tempo para o exemplo 1 (*Feed1* e *Feed2* estão disponíveis no instante 20).

Um trabalho em andamento tem como objetivos determinar, a partir de aumentos de janelas introduzidos pelo usuário, as materiais primas que devem ter sua disponibilidade antecipada. Isto é feito utilizando as relações de precedência decorrentes do balanço de massa, descritas na seção 2.4.2.1. Este procedimento pode ser complexo quando a redução de janelas de tempo é devido essencialmente a problemas de capacidade (ocupação do processador). A antecipação da disponibilidade das matérias primas diretamente envolvidas pelas relações de precedência decorrentes do balanço de massa pode ter que ser excessiva dado que as janelas de tempo são fundamentalmente reduzidas por intervalos de ocupação obrigatória de outras bateladas competindo pelo mesmo processador. Neste caso, será necessário antecipar a disponibilidade de outras matérias primas.

2.7.2. Exemplo 2

Considera-se a demanda dada na Tabela 2.2a e, inicialmente, o plano de disponibilidade de matérias primas, a Figura 2.7 é repetida abaixo para facilitar a visualização do exemplo 2.

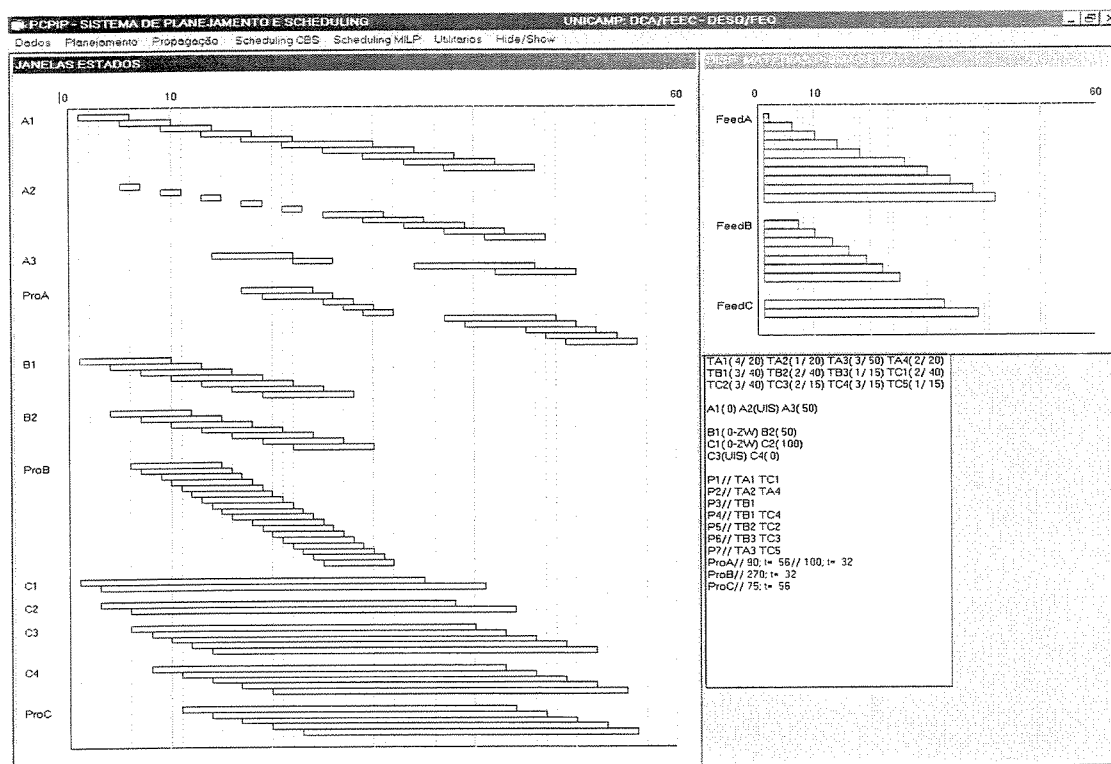


Figura 2.7. Janelas iniciais dos estados e disponibilidade das matérias primas para o exemplo 2.

As janelas das bateladas antes da fase de propagação das restrições foram apresentadas na Figura 2.11, repetida na página seguinte. A figura 2.33 apresenta as janelas de tempo e as demandas agregadas dos processadores resultantes da propagação das restrições para o exemplo 2.

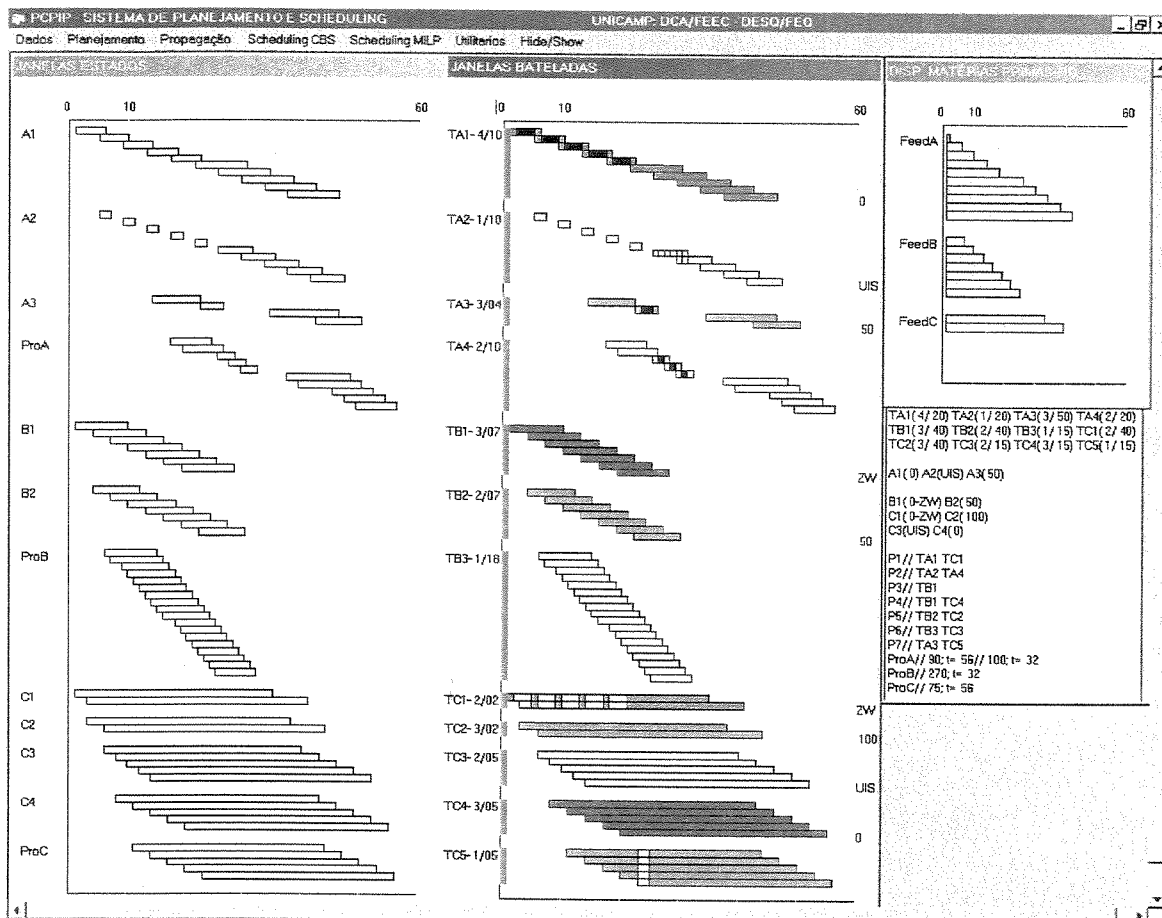


Figura 2.11. Janelas de tempo iniciais dos estados e das bateladas para o exemplo 2.

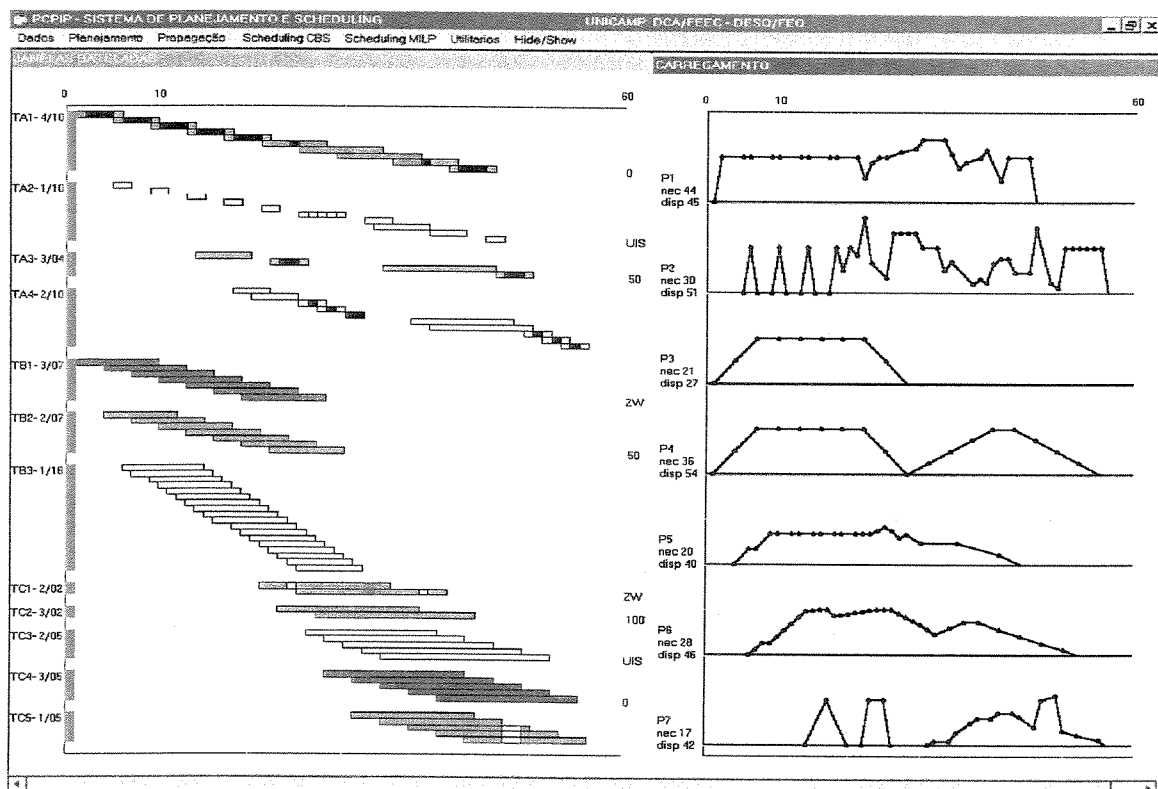


Figura 2.33. Janelas de tempo e demandas agregadas dos processadores para o exemplo 2.

O efeito da propagação das restrições para o exemplo 2 pode ser visto comparando-se as janelas de tempo antes (figura 2.30) e depois (figura 2.33) da propagação. A tabela 2.9 indica o número total de possíveis alocações para todas as bateladas de cada operação. A sua redução é um indicativo da redução da dimensão do problema na fase de *scheduling*.

Tabela 2.9. Número total de possíveis alocações antes e depois da propagação das restrições.

batelada	nº poss. alocações antes	nº poss. alocações depois	diminuição (%)	batelada	nº poss. alocações antes	nº poss. alocações depois	diminuição (%)
TA1/1	2	2	0	TB1/1	7	7	0
TA1/2	2	2	0	TB1/2	7	7	0
TA1/3	2	2	0	TB1/3	7	7	0
TA1/4	2	2	0	TB1/4	7	7	0
TA1/5	2	2	0	TB1/5	7	7	0
TA1/6	5	4	20	TB1/6	7	7	0
TA1/7	6	6	0	TB1/7	7	7	0
TA1/8	6	6	0	TB2/1	7	7	0
TA1/9	6	4	33,33	TB2/2	7	7	0
TA1/10	6	2	66,67	TB2/3	7	7	0
TA2/1	2	2	0	TB2/4	7	7	0
TA2/2	2	2	0	TB2/5	7	7	0
TA2/3	2	2	0	TB2/6	7	7	0
TA2/4	2	2	0	TB2/7	7	7	0
TA2/5	2	2	0	TB3/1	9	9	0
TA2/6	3	3	0	TB3/2	9	9	0
TA2/7	5	3	40	TB3/3	8	8	0
TA2/8	6	6	0	TB3/4	8	8	0
TA2/9	6	4	33,33	TB3/5	8	8	0
TA2/10	6	2	66,67	TB3/6	8	8	0
TA3/1	6	4	33,33	TB3/7	8	8	0
TA3/2	2	2	0	TB3/8	8	8	0
TA3/3	10	10	0	TB3/9	8	8	0
TA3/4	6	2	66,67	TB3/10	8	8	0
TA4/1	6	3	50	TB3/11	7	7	0
TA4/2	6	6	0	TB3/12	7	7	0
TA4/3	2	2	0	TB3/13	7	7	0
TA4/4	2	2	0	TB3/14	7	7	0
TA4/5	2	1	50	TB3/15	7	7	0
TA4/6	10	10	0	TB3/16	7	7	0
TA4/7	10	10	0	TB3/17	7	7	0
TA4/8	6	2	66,67	TB3/18	7	7	0
TA4/9	6	2	66,67	TC4/1	13	13	0
TA4/10	6	2	66,67	TC4/2	13	13	0
TC1/1	13	11	15,38	TC4/3	13	13	0
TC1/2	16	13	18,75	TC4/4	13	13	0
TC2/1	13	13	0	TC4/5	13	13	0
TC2/2	16	15	6,25	TC5/1	13	13	0
TC3/1	13	13	0	TC5/2	13	13	0
TC3/2	14	14	0	TC5/3	13	11	15,38
TC3/3	15	15	0	TC5/4	13	11	15,38
TC3/4	16	16	0	TC5/5	13	11	15,38
TC3/5	17	17	0				
				total	654	605	7,49

2.8. Limitações da abordagem em termos de factibilidade das janelas

Não é possível garantir que as janelas de tempo resultantes do planeamento sejam factíveis na fase de *scheduling* em situações em que o impacto das restrições é forte. A razão básica para isto é que a análise de capacidade é feita sobre as janelas de tempo e não sobre as alocações. Uma janela de tempo maior que o tempo de processamento significa que a batelada em questão tem diversas possibilidades de alocação. A propagação das restrições garante, para bateladas interligadas por estas restrições, que os respectivos inícios

e finais de janelas são “compatíveis” com as restrições (balanço de massa, armazenamento e capacidade). Em outras palavras, dadas duas bateladas interligadas por uma restrição, a alocação de uma delas em qualquer das alocações possíveis para ela deixa uma ou mais possibilidades de alocação para a outra batelada. Este resultado não pode ser garantido em dois casos:

- Se a restrição não é considerada, por exemplo no caso de recursos compartilhados cumulativos, na versão atual. A alocação de uma batelada em um intervalo de tempo utilizando um dado recurso pode inviabilizar as alocações, de outras bateladas em outros processadores utilizando este recurso, que impliquem em sobreposição no tempo com aquela alocação. As reduções nas suas janelas de tempo podem, eventualmente, levar a uma situação infactível.
- Se a restrição interligando duas bateladas, mesmo considerada, tem implicações sobre as alocações propriamente ditas. Por exemplo, em uma restrição de armazenagem ZW a alocação da batelada produtora implica na alocação da batelada consumidora. A batelada consumidora deixa de ter as possibilidades de alocação que poderia ter se a sua janela de tempo (após a alocação da produtora) fosse maior que o seu tempo de processamento. Esta alocação “forçada” pode implicar na redução da janela de tempo de uma batelada competindo pelo mesmo processador e levar a uma situação infactível. Esta situação não seria detectada porque de fato a janela remanescente da batelada consumidora está condicionada à alocação da produtora.

2.9. Aspectos operacionais

Nesta seção são apresentados os aspectos operacionais do sistema de planeamento, que foi totalmente desenvolvido utilizando o pacote Visual Basic. Neste sistema, a fase de planeamento foi separada em três passos: *i)* Dados, *ii)* Planeamento e *iii)* Propagação. O primeiro passo na utilização deste sistema é definir os dados do problema que será tratado. A execução deste primeiro passo resulta no cálculo do número de bateladas que serão produzidas e na determinação dos finais das janelas de tempo (descrito na seção 2.4.1). A figura 2.34 apresenta as opções disponíveis para a definição dos dados do problema. O sistema de planeamento permite a utilização de um problema definido anteriormente, a modificação de um problema definido anteriormente e a definição de um novo problema.

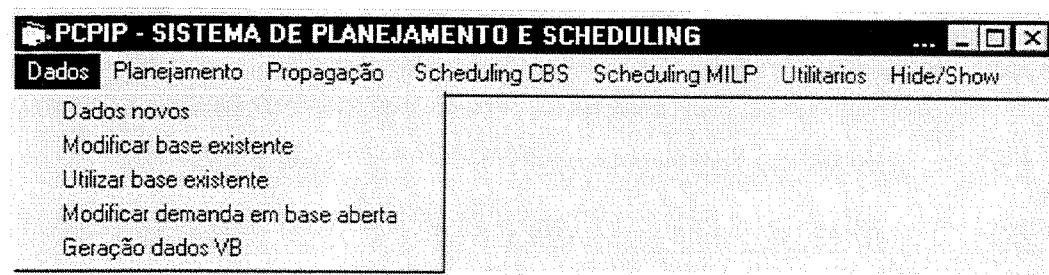


Figura 2.34. Dados do problema.

O cálculo do início das janelas de tempo é resultado da execução do segundo passo (identificado como “Planejamento” na figura 2.35), onde o usuário define a disponibilidade para ocupação dos processadores e a disponibilidade de matérias primas. O passo de “Planejamento” também permite a visualização das janelas de tempo iniciais e dos perfis de estoque (disponibilidade) dos estados imposto pelas janelas de tempo iniciais.

O usuário pode interagir livremente com o sistema de planejamento. A modificação na atribuição das operações aos processadores e a alteração nas demandas dos produtos finais é feita através do procedimento de definição dos dados do problema, apresentado na figura 2.34. A modificação na disponibilidade para ocupação dos processadores e na disponibilidade de matérias primas é feita repetindo-se o passo de “Planejamento”, apresentado na figura 2.35.

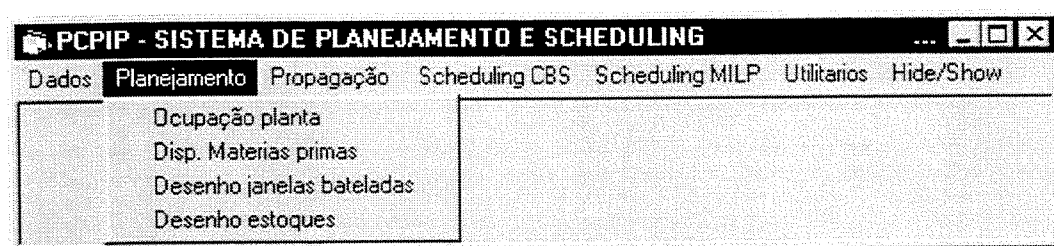


Figura 2.35. Definição da ocupação da planta e da disponibilidade de matérias primas.

O último passo da fase de planejamento consiste na propagação das restrições. Este procedimento poderia ter sido automatizado de tal forma que, após a definição dos dados do problema, as janelas de tempo iniciais e a propagação de restrições fossem executadas pelo sistema. Esta automação dos passos da fase de planejamento não foi feita para que se pudesse acompanhar as implicações de cada um dos passos desta fase sobre as janelas de tempo. Com isso, a propagação de restrições (descrita na seção 2.6) é executada através de duas etapas, como mostra a figura 2.36. Primeiro propagam-se os cortes decorrentes da definição da disponibilidade para ocupação dos processadores e depois considera-se o compartilhamento dos processadores.

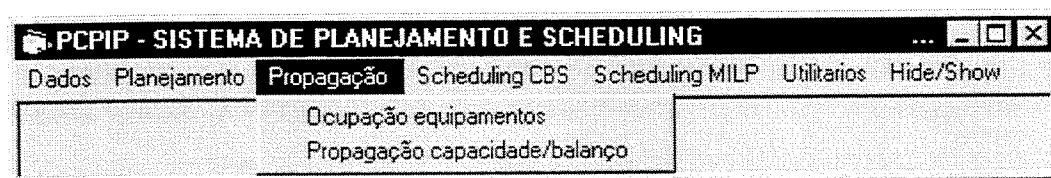


Figura 2.36. Acionamento da propagação de restrições.

A propagação de restrições encerra a fase de planejamento. Se nenhuma infactibilidade for encontrada e se o usuário considerar o cenário gerado pelo planejamento adequado, pode-se executar o *scheduling* do problema. O sistema implementado permite a execução do *scheduling* através de três procedimentos: *i*) programação linear inteira mista

(apresentado no capítulo 3), *ii) simulated annealing* (apresentado no capítulo 4) e *iii) busca orientada por restrições (constrained based search)* que é o tema de mestrado de Wesley Alvarenga).

Todos os resultados da fase de planejamento e de *scheduling* (tal como janelas de tempo, carregamento dos processadores *etc.*) podem ter as suas escalas definidas pelo usuário, como mostra a figura 2.37. Além disso, o usuário também pode definir quais resultados estarão visíveis através do passo “Hide/Show”, apresentado na figura 2.38.

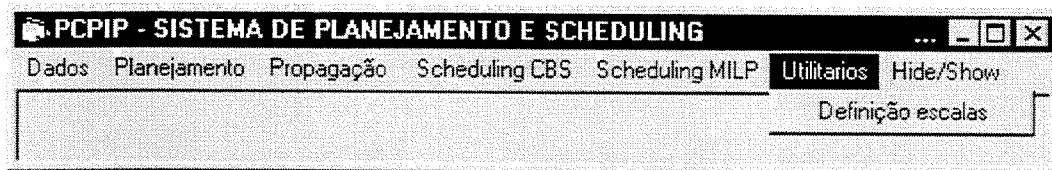


Figura 2.37. Definição da escala dos resultados apresentados.

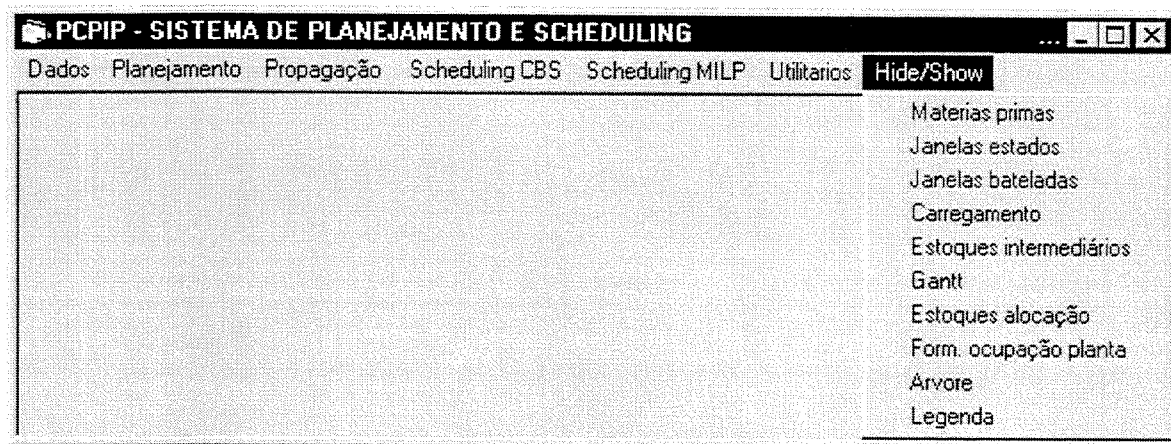


Figura 2.38. Definição dos resultados e dados do problema que estarão visíveis/ocultos.

2.10. Conclusões

O sistema de planejamento implementado permite que o usuário tenha informação sobre a factibilidade do cenário de produção gerado e sobre o carregamento dos processadores decorrente deste cenário. A possibilidade de interação do usuário com o sistema permite que novos cenários de produção sejam gerados sempre que for detectada alguma infactibilidade ou quando o usuário achar conveniente.

Os pontos essenciais no sistema de planejamento proposto são:

- Utilização de técnicas de explosão para frente e para trás para a obtenção das janelas de processamento;
- Utilização de técnicas de propagação de restrições para obter as janelas de tempo reais decorrentes das restrições do problema;

- Utilização dos conceitos de crucialidade/demanda agregada para visualizar o carregamento dos processadores.

As técnicas de propagação de restrições utilizadas em sistemas de busca orientada por restrições foram ampliadas para incluir a propagação de restrições de armazenagem. Utilizam-se em conjunto dois mecanismos originados na área de busca orientada por restrições: a propagação de restrições e as funções de crucialidade. Esta abordagem nova levou a uma proposta para automatizar a ativação dos mecanismos de propagação de restrições apenas nas situações mais promissoras.

O interesse principal do sistema de planejamento é o de gerar um cenário reduzido para a fase de *scheduling* e de fornecer aos usuários informações sobre o carregamento dos processadores induzido pelas janelas de tempo.

CAPÍTULO 3. PROGRAMAÇÃO DA PRODUÇÃO USANDO UMA FORMULAÇÃO INTEIRA MISTA (MILP)

3.1. Introdução

Neste capítulo apresenta-se uma formulação linear inteira mista (MILP) para resolver o problema de programação da produção (*scheduling*) utilizando-se a informação das janelas de tempo obtidas na fase de planejamento. A formulação utiliza a representação Rede Estado-Tarefa (STN), apresentada na seção 2.3, para descrever os processos de fabricação (receita).

Uma consideração básica em qualquer formulação do problema de *scheduling* refere-se à representação do tempo. Neste trabalho utiliza-se uma representação discreta do tempo definida *a priori*. O horizonte de tempo de interesse é dividido em um número de intervalos de igual duração, chamados *slots*. A duração de um *slot* é igual ao máximo divisor comum de todos os tempo de processamento e de preparação dos processadores envolvidos no problema. 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 que competem por recursos compartilhados são posicionadas. Eventos de qualquer tipo, tal como início ou fim de processamento de uma operação, mudanças na disponibilidade de um processador e de outros recursos, etc., só são permitidos nos extremos desses *slots*.

Na formulação proposta por Kondili *et al.* (1993), a alocação de uma operação i em um processador j com início de processamento no *slot* t é representada por uma variável binária, W_{ijt} . A medida que se aumenta a dimensão do problema a ser resolvido a dificuldade de resolução aumenta devido à elevação no número de variáveis binárias. Esta é a principal desvantagem da formulação: a discretização pré-fixada do tempo deve ser suficientemente fina para permitir representar todos os eventos de interesse. O interesse deste trabalho é que a discretização uniforme do tempo permite representar as restrições temporais impostas pelas janelas de tempo das bateladas.

Para se reduzir a dimensão da formulação proposta por Kondili, foram implementadas três alterações nesta formulação:

- Uso das janelas de tempo que identificam o intervalo de tempo disponível para a execução das bateladas das operações;
- Uso de subtrens na formulação. A receita pode incluir estados intermediários sujeitos a restrições de armazenagem ZW (zero wait). Os subtrens englobam as operações que devem ser alocadas sem espera porque, respectivamente, geram e consomem estados com restrição de armazenagem ZW (*zero wait*). Basta conhecer o início do processamento da primeira operação pertencente ao subtrem para que se conheça os inícios de processamento de todas as operações pertencentes ao subtrem. As variáveis binárias serão definidas por subtrens e não por operações;
- Uso de uma formulação reduzida, eliminando processadores dedicados ou pouco carregados.

As implicações do uso das janelas de tempo na formulação de *scheduling* são discutidas na seção 3.2. A inclusão de subtrens na formulação é discutida na seção 3.3. A seção 3.4 apresenta a formulação para problemas reduzidos obtidos através da eliminação de processadores pouco carregados ou dedicados. A seção 3.5 apresenta a comparação entre a formulação geral apresentada na seção 3.3 e a formulação reduzida apresentada na seção 3.4.

3.2. Modelo MILP para *scheduling* utilizando janelas de tempo (Rodrigues *et al.*, 2000b)

A obtenção das janelas de tempo na fase de planejamento (capítulo 2) implica na determinação de atribuições únicas de operações a processadores e na determinação do tamanho de batelada de cada operação. Isto simplifica a formulação proposta por Kondili porque elimina a determinação (na resolução do problema MILP) da atribuição de processadores e dos tamanhos das bateladas. No entanto, torna-se necessário introduzir o índice batelada na variável de alocação das operações porque cada batelada tem uma janela de tempo diferente.

A formulação para o problema de *scheduling* total (não reduzido) utilizando a informação das janelas de tempo é apresentada a seguir. São incluídas todas as expressões matemáticas e a sua derivação, a partir da formulação de Kondili, é descrita à continuação.

a) Balanço de massa

$$S_{st} = S_{s,t-1} + \sum_{i \in T_s} M'_{is} \sum_{\substack{b \in Bat_i \\ t \geq ebt_{bi} + p_{is} \\ t \leq lft_{bi} + p_{is} - tp_i + 1}} W_{b,i,t-p_{is}} - \sum_{i \in T_s} M_{is} \sum_{\substack{b \in Bat_i \\ t \geq ebt_{bi} \\ t \leq lft_{bi} - tp_i + 1}} W_{bit} + R_{st} - D_{st} \quad \forall s, t \in Te_s \quad [3.1a]$$

b) Alocação das bateladas resultantes do planejamento

$$\sum_{t=ebt_{bi}}^{lft_{bi}-tp_i+1} W_{bit} = 1 \quad \forall i \notin Ap, b \in Bat_i \quad [3.2]$$

c) Utilização dos processadores

$$\sum_{i \in I_j} \sum_{b \in Bat_i} \sum_{\substack{t'=t-tp_i+1 \\ t' \geq ebt_{bi} \\ t' \leq lft_{bi}-tp_i+1}}^t W_{bit'} \leq 1 \quad \forall j, t \in Ec_j \quad [3.3a]$$

d) Ordenamento de bateladas de uma mesma operação

$$\sum_{t=ebt_{b'i}}^{lft_{b'i}-tp_i+1} W_{b'it} * (t + tp_i) \leq \sum_{t=ebt_{bi}}^{lft_{bi}-tp_i+1} W_{bit} * t \quad \forall i, b' \in Bat_i, b \in Bat_i / b' < b \quad [3.4]$$

e) **Capacidade de armazenamento**

$$S_{st} \leq C_s \quad \forall s, t \in Te_s \quad [3.5]$$

f) **Operação de armazenamento dentro do processador**

$$W_{b^*,i^*,t} \leq W_{b^*,i^*,t-1} + \sum_{s \in T_s'} \sum_{i^* \in H_{is}} W_{b^*,i^*,t-p_{is}} \quad [3.6]$$

$$\forall i^* \in Ap, b^* \in Bat_{i^*}, t / ebt_{b^*,i^*} \leq t \leq lft_{b^*,i^*}$$

g) **Função objetivo: min {obj}**

$$obj = \sum_{i \in Prod} \sum_{b \in Bat_i} \{lft_{bi} - tp_i + 1 - (\sum_{t \geq ebt_{bi}}^{lft_{bi} - tp_i + 1} W_{bit} * t)\} \quad [3.7]$$

onde,

Índices:

i	Operação.
b	Batelada de uma operação.
j	Processador.
t	Slot.
s	Estado gerado ou consumido por uma operação.

Variáveis:

W_{bit}	= 1 indica o início do processamento da batelada b da operação i no slot t . Caso contrário, $W_{bit} = 0$.
S_{st}	Massa (ou volume) do estado s armazenada no slot t .

Parâmetros:

M'_{is}	Massa (ou volume) de saída da operação i na forma de estado s .
M_{is}	Massa (ou volume) de entrada na operação i na forma de estado s .
C_s	Capacidade de armazenamento do estado s .
p_{is}	Tempo decorrido desde o início do processamento da operação i até que esta operação produza o estado s .
tp_i	Tempo total de processamento da operação i .
ebt_{bi}	Início da janela de tempo da batelada b da operação i .
lft_{bi}	Final da janela de tempo da batelada b da operação i .
R_{st}	Quantidade de produto s recebida por fornecimento externo à planta no slot t .
D_{st}	Quantidade de produto s retirada no slot t para ser entregue a um cliente.

Restrições no âmbito das expressões:

H_{is}	Operação de armazenamento dentro do processador do estado s gerado pela operação i .
Sa_{st}	Slot "anterior" ao slot t para fins de balanço de massa do estado s .
Ec_j	Conjunto dos slots onde há disputa pela ocupação do processador j .
Ap	Conjunto das operações que realizam armazenamento dentro de um processador.
I_j	Conjunto das operações atribuídas ao processador j .
$Prod$	Conjunto das operações geradoras de um produto final.
T'_s	Conjunto das operações produzindo o estado s .
T_s	Conjunto das operações consumindo o estado s .
Te_s	Conjunto dos slots em que pode haver consumo ou produção do estado s .

a) Balanço de massa (expressão 3.1a)

A expressão 3.1a é obtida a partir da expressão de balanço de massa inicialmente proposta por Kondili, indicada na expressão 3.1b.

$$S_{st} = S_{s,t-1} + \sum_{i \in T_s'} \rho'_{is} \sum_{j \in J_i} B_{i,j,t-p_{is}} - \sum_{i \in T_s} \rho_{is} \sum_{j \in J_i} B_{ijt} + R_{st} - D_{st} \quad \forall s, t \quad [3.1b]$$

onde:

B_{ijt}	Massa da operação i no processador j com início de processamento no <i>slot</i> t .
ρ'_{is}	Fração mássica de saída da operação i na forma de estado s .
ρ_{is}	Fração mássica de entrada na operação i na forma de estado s .
J_i	Conjunto dos processadores que podem executar a operação i .

Esta expressão indica que a variação no estoque de material ($S_{s,t} - S_{s,t-1}$) do estado s , no *slot* t , é dado pela diferença entre a quantidade produzida e a quantidade consumida deste estado. Os somatórios na expressão 3.1b servem para discriminar os termos que devem ser considerados no balanço de massa. Assim, o somatório de $(\rho_{is} * B_{ijt})$ indica a quantidade de material que está sendo retirada do estado s para ser consumida pela operação i iniciada no *slot* t no processador j . O somatório de $(\rho'_{is} * B_{i,j,t-p_{is}})$ indica a quantidade de material que está sendo recebida no estado s proveniente da operação i no *slot* t , se o processamento desta operação é feito no processador j iniciou-se no *slot* $t - p_{is}$.

Admitindo que a atribuição de operações a processadores é definida e é única, o conjunto J_i e o índice j podem ser eliminados das expressões matemáticas, obtendo-se a expressão 3.1c.

$$S_{st} = S_{s,t-1} + \sum_{i \in T_s'} \rho'_{is} * B_{i,t-p_{is}} - \sum_{i \in T_s} \rho_{is} * B_{it} + R_{st} - D_{st} \quad \forall s, t \quad [3.1c]$$

O tamanho da batelada da operação i é fixado na fase de planejamento. Supondo um valor V_i para o tamanho da batelada que se inicia em um *slot* t , B_{it} pode ser escrito como: $B_{it} = W_{it} * V_i$. Definindo-se: $M'_{is} = \rho'_{is} * V_i$ e $M_{is} = \rho_{is} * V_i$, obtém-se a expressão 3.1d.

$$S_{st} = S_{s,t-1} + \sum_{i \in T_s'} M'_{is} * W_{i,t-p_{is}} - \sum_{i \in T_s} M_{is} * W_{it} + R_{st} - D_{st} \quad \forall s, t \quad [3.1d]$$

Dado que existe uma janela de tempo para cada batelada, torna-se necessário introduzir o índice batelada (b) na variável binária de alocação das operações (W_{it}), que passará a ser uma variável de alocação das bateladas das operações (W_{bit}). Como o processamento de uma batelada b de uma operação i só pode ocorrer dentro da sua janela de tempo, o primeiro *slot* em que é possível iniciar o processamento desta batelada é o *slot* ebt_{bi} e o último *slot* é o *slot* $lft_{bi} - tp_i + 1$. A expressão de balanço de massa com o uso de

variáveis de alocação (W_{bit}) e janelas de tempo das bateladas passa a ser escrita como indicado abaixo

$$S_{st} = S_{s,t-1} + \sum_{i \in T'_s} M'_{is} \sum_{\substack{b \in Bat_i \\ t \geq ebt_{bi} + p_{is} \\ t \leq lft_{bi} + p_{is} - tp_i + 1}} W_{b,i,t-p_{is}} - \sum_{i \in T_s} M_{is} \sum_{\substack{b \in Bat_i \\ t \geq ebt_{bi} \\ t \leq lft_{bi} - tp_i + 1}} W_{bit} + R_{st} - D_{st}$$

$\forall s, t$ [3.1e]

A utilização de janelas de tempo das bateladas torna possível restringir o domínio da expressão 3.1e apenas aos slots em que é possível haver a produção ou consumo do estado s , obtendo-se, finalmente, a expressão 3.1a. Esta restrição é implementada através do conjunto Te_s definido como segue. Definem-se inicialmente os conjuntos Tg_s e Tc_s como:

- O conjunto Tg_s é o conjunto de *slots* onde pode haver geração do estado s .

$$Tg_s = \{t / ebt_{bi} + p_{is} \leq t \leq lft_{bi} + p_{is} - tp_i + 1, i \in T'_s\}.$$

- O conjunto Tc_s compreende os *slots* onde pode haver consumo do estado s .

$$Tc_s = \{t / ebt_{bi} \leq t \leq lft_{bi} - tp_i + 1, i \in T_s\}.$$

Dados os conjuntos Tg_s e Tc_s , pode-se identificar o conjunto Te_s dos *slots* em que é possível haver a produção ou consumo do estado s ($Te_s = Tg_s \cup Tc_s$). A expressão de balanço de massa não precisa ser escrita nos *slots* não pertencentes ao conjunto Te_s . Isto porém coloca o problema de determinação dos *slots* sucessivos na recorrência representada pelo balanço de massa. Os *slots* sucessivos são determinados facilmente a partir do conjunto Te_s , definindo-se Sa_{st} como o *slot* “anterior” ao *slot* t para fins de balanço de massa do estado s . O exemplo a seguir ilustra a determinação destes conjuntos.

A figura 3.1 apresenta um exemplo com as janelas de tempo das bateladas de duas operações (A e B) ligadas pelo estado *int*. As bateladas destas duas operações possuem o mesmo tamanho e o mesmo tempo de processamento (3 horas). As janelas de tempo das bateladas destas operações são indicadas pelos retângulos em preto e abaixo de cada retângulo é apresentada uma seta cinza, identificando o intervalo definido pelas janelas de tempo onde poderá haver a produção ou consumo do estado *int*. Os *slots* dos conjuntos Tg_{int} , Tc_{int} e Te_{int} são identificados nesta figura por asteriscos. Entre os *slots* 1 até 6, entre os *slots* 20 até 26 e após o *slot* 37, a expressão de balanço de massa não precisa ser escrita porque não há nenhuma possibilidade de o estoque do estado *int* ser alterado. No *slot* 27, Sa_{st} indicará o *slot* 19 como sendo o último *slot* em que a expressão de balanço de massa foi escrita para o estado *int*. Com isso, o estoque inicial $S_{s,t-1}$ na expressão de balanço do *slot* 27 é igual ao estoque final no *slot* 19. Como o *slot* 7 é o primeiro *slot* em que a expressão de balanço de massa é escrita, Sa_{st} será igual a zero e o valor do estoque $S_{s,t-1}$ será igual ao estoque inicial do estado *int* no horizonte de planejamento.

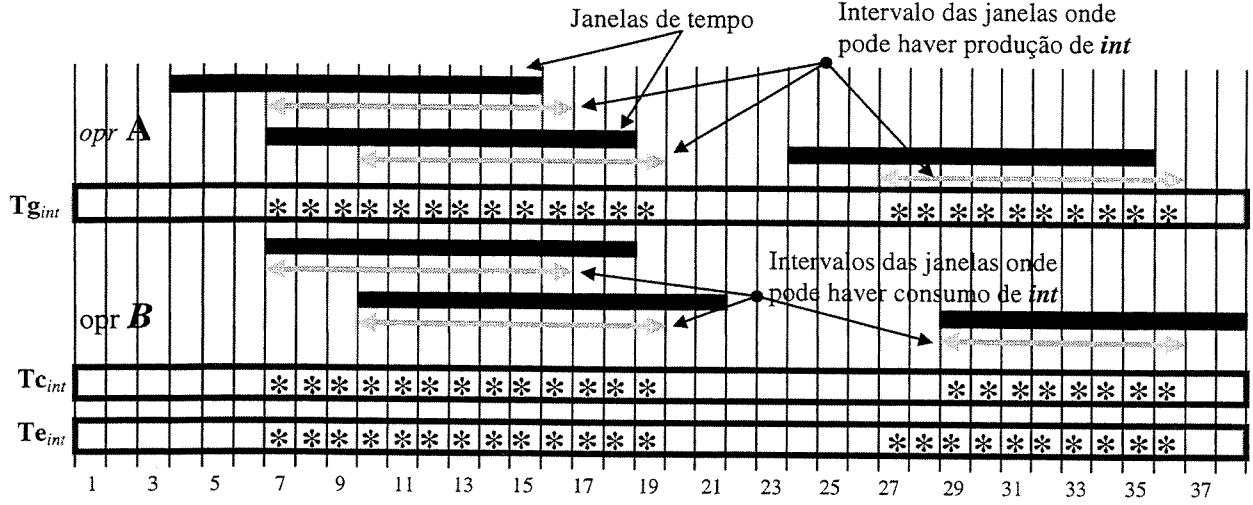


Figura 3.1 – Janelas de tempo das operações A e B.

b) Existência das bateladas (expressão 3.2)

A expressão 3.2 obriga que cada batelada seja obrigatoriamente processada uma vez e dentro da sua janela de tempo. Para que a batelada b de uma operação i seja processada dentro de sua janela de tempo, o slot de início do processamento desta batelada deverá ocorrer dentro do intervalo: $ebt_{bi} \leq t \leq lft_{bi} - tp_i + 1$. A expressão 3.2 só não é gerada para as operações de armazenamento dentro dos processadores porque a execução destas operações não é obrigatória.

c) Utilização dos processadores (expressão 3.3a)

A expressão 3.3a faz com que em qualquer *slot* t haja no máximo uma operação sendo processada em qualquer um dos processadores. A expressão modificada proposta por Shah *et al.* (1993) é dada abaixo.

$$\sum_{i \in I_j} \sum_{t'=t-tp_i+1}^t W_{ijt'} \leq 1 \quad \forall j, t \quad [3.3b]$$

A expressão 3.3c é obtida a partir da expressão anterior, introduzindo-se o índice batelada e restringindo-se o somatório aos *slots* dentro da janela de tempo.

$$\sum_{i \in I_j} \sum_{b \in Bat_i} \sum_{\substack{t'=t-tp_i+1 \\ t' \geq ebt_{bi} \\ t' \leq lft_{bi}-tp_i+1}}^t W_{bit'} \leq 1 \quad \forall j, t \quad [3.3c]$$

O resultado do planejamento pode levar a *slots* dos processadores onde há apenas uma batelada podendo ocupar tal *slot* ou onde o *slot* não vai ser utilizado. A expressão 3.3c não precisa ser escrita nestas situações. O conjunto Ec_j é definido como contendo apenas os *slots* onde há duas ou mais bateladas disputando a ocupação do *slot*. A expressão final 3.3a representa a expressão de utilização dos processadores com a inclusão do conjunto Ec_j .

A figura 3.2 apresenta um exemplo com as janelas de tempo das bateladas de duas operações (*B* e *D*). Nesta figura, os *slots* do conjunto Ec_j são identificados por um asterisco. Para este exemplo, pode-se constatar que é possível obter uma redução do número de expressões matemáticas geradas (de 29 para 18 expressões) através do uso do conjunto Ec_j .

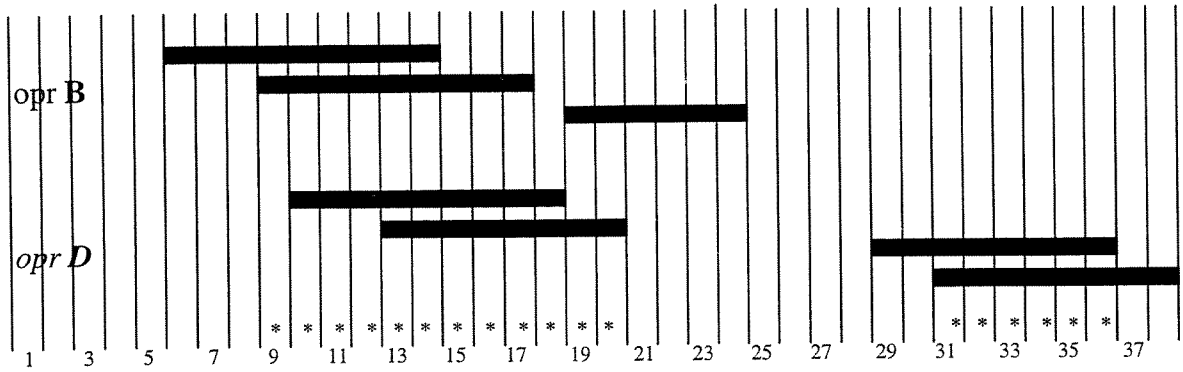


Figura 3.2 – Janelas de tempo das duas operações que ocupam o mesmo processador.

d) Ordenamento das bateladas de uma mesma operação (expressão 3.4)

O propósito da definição do ordenamento entre as bateladas de uma mesma operação é o de impor uma condição de monotonicidade das bateladas, restringindo a árvore de busca das soluções. Dadas duas bateladas b' e b de uma mesma operação i , onde $b' < b$, então a batelada b' deverá ser processada antes da batelada b . A expressão proposta para a definição de um ordenamento entre as bateladas de uma mesma operação é a expressão 3.4. Esta expressão impõe que, se o processamento da batelada b' for iniciado no *slot* t' , o processamento da batelada b só poderá ser iniciado no *slot* t , tal que: $t \geq t' + tp_i$.

e) Capacidade de armazenamento (expressão 3.5)

A quantidade de massa armazenada em um estado s não deve exceder a capacidade de armazenamento deste estado (C_s) em nenhum instante de tempo. A expressão 3.5 impõe o cumprimento à limitação de armazenamento, sendo gerada para todo *slot* t onde pode haver consumo ou produção do estado s .

f) Operação de armazenamento dentro dos processadores (expressão 3.6)

Um estado s gerado pela operação i está sujeito a uma restrição de armazenagem NIS (*no intermediate storage*) se não existe capacidade de armazenagem para o estado s .

Neste caso, é possível que o processador habilitado para realizar a operação i também possa realizar uma operação de armazenamento i^* . A operação i que produz o estado s terá associada uma operação de armazenamento i^* , identificada por H_{is} . O tempo de processamento da operação de armazenamento é igual a um *slot*. A expressão 3.6 estabelece que a execução da batelada b da operação de armazenamento i^* só poderá ocorrer no *slot* t se o processamento da batelada b da operação i for iniciado no *slot* $t-p_{i^*,s}$ ou se a batelada b da operação de armazenamento i^* já estava sendo executada no *slot* $t-1$. A figura 3.3 mostra a forma como a restrição NIS é apresentada na representação STN.

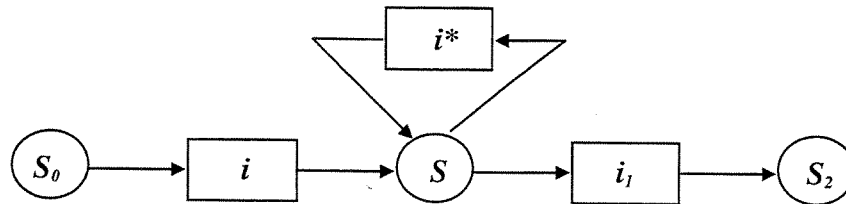


Figura 3.3 - Representação STN da operação de armazenamento i^* .

g) Função objetivo (*earliness*) (expressão 3.7)

Como deseja-se que as bateladas sejam alocadas dentro das suas janelas de tempo e que as demandas dos produtos finais sejam atendidas o mais próximo possível dos seus prazos de conclusão (*due dates*), a função objetivo adotada é de minimização da soma de *earliness* das bateladas das operações que geram produtos finais. A função objetivo é apresentada na expressão 3.7. Se uma batelada b de uma operação i gera um produto final, esta expressão calculará a diferença entre o valor do último *slot* da janela de tempo ($lft_{bi}-tp_i+1$) em que pode ocorrer o início do processamento desta batelada e o *slot* t em que ocorre a alocação do início desta batelada.

3.3. Uso de subtrens na formulação MILP

A receita de produção pode incluir estados intermediários instáveis que não podem ser armazenados, ou seja, sujeitos a restrições de armazenagem ZW (*zero wait*). Isto permite a definição de subtrens que englobam as operações que devem ser alocadas sem espera. Uma implicação da definição de um subtrem é que basta conhecer o início do processamento da primeira operação pertencente ao subtrem para que se conheça os inícios de processamento de todas as operações pertencentes ao subtrem. É desta constatação que surgiu a proposta de se definir, na formulação de *scheduling*, o início do processamento dos subtrens em lugar de se definir o início de cada uma das operações. A substituição das operações pelo uso dos subtrens visa reduzir a dimensão do problema MILP quando há um ou mais estados de produtos intermediários com restrição ZW.

Com o uso dos subtrens na formulação de *scheduling*, o índice i usado até agora para identificar as operações é substituído pelo índice k que identifica os subtrens. A

formulação utilizada para o problema de scheduling (não reduzido) em termos de subtrens é apresentada abaixo e as modificações em relação à formulação anterior são discutidas.

a) Balanço de massa

$$S_{st} = S_{s,t-1} + \sum_{k \in T'_s} M'_{ks} \sum_{\substack{b \in \text{Bat}_k \\ t \geq \text{ebt}_{bk} + p_{ks} \\ t \leq \text{lft}_{bk} + p_{ks} - tp_k + 1}} W_{b,k,t-p_{ks}} - \sum_{k \in T_s} M_{ks} \sum_{\substack{b \in \text{Bat}_k \\ t \geq \text{ebt}_{bk} + \delta_{ks} \\ t \leq \text{lft}_{bk} + \delta_{ks} - tp_k + 1}} W_{b,k,t-\delta_{ks}} + R_{st} - D_{st} \quad \forall s, t \in \text{Te}_s \quad [3.8]$$

b) Alocação das bateladas resultantes do planejamento

$$\sum_{t=\text{ebt}_{bk}}^{\text{lft}_{bk}-tp_k+1} W_{bkt} = 1 \quad \forall k \notin \text{Ap}, b \in \text{Bat}_k \quad [3.9]$$

c) Utilização de processadores

$$\sum_{k \in I_j} \sum_{b \in \text{Bat}_k} \sum_{\substack{t'=\text{t}-\phi_{kj}-\lambda_{kj}+1 \\ t' \geq \text{ebt}_{bk} \\ t' \leq \text{lft}_{bk}-tp_k+1}}^{t-\phi_{kj}} W_{bkt'} \leq 1 \quad \forall j, t \in \text{Ec}_j \quad [3.10]$$

d) Ordenamento de bateladas de um mesmo subtrem

$$\sum_{t=\text{ebt}_{b'k}}^{\text{lft}_{b'k}-tp_k+1} W_{b'kt} * (t + tp_k) \leq \sum_{t=\text{ebt}_{bk}}^{\text{lft}_{bk}-tp_k+1} W_{bkt} * t \quad \forall k, b' \in \text{Bat}_k, b \in \text{Bat}_k / b' < b \quad [3.11]$$

e) Capacidade de armazenamento

$$S_{st} \leq C_s \quad \forall s, t \in \text{Te}_s \quad [\text{idêntica à expressão 3.4}]$$

f) Operação de armazenamento dentro do processador

$$W_{b'k't} \leq W_{b',k',t-1} + \sum_{s'} \sum_{k' \in H_{ks}} W_{b',k,t-p_{ks}} \quad \forall k', b' \in \text{Bat}_{k'}, t / \text{ebt}_{b'k'} \leq t \leq \text{lft}_{b'k'} \quad [3.12]$$

g) Função objetivo: $\min \{obj\}$

$$obj = \sum_{k \in Prod} \sum_{b \in Bat_k} \{lft_{bk} - tp_k + 1 - (\sum_{t \geq ebt_{bk}}^{lft_{bk} - tp_k + 1} W_{bkt} * t)\} \quad [3.13]$$

onde,

Variáveis:

W_{bkt} = 1 indica o início do processamento da batelada b do subtrem k no *slot* t . Caso contrário, W_{bkt} = 0.

S_{st} Massa (ou volume) do estado s armazenada no *slot* t .

Parâmetros:

M'_{ks} Massa (ou volume) de saída da subtrem k na forma de estado s .

M_{ks} Massa (ou volume) de entrada na subtrem k na forma de estado s .

C_s Capacidade de armazenamento do estado s .

p_{ks} Tempo decorrido desde o início do processamento da operação i até que esta operação produza o estado s .

tp_k Tempo total de processamento do subtrem k .

ebt_{bk} Início da janela de tempo da batelada b do subtrem k .

lft_{bk} Final da janela de tempo da batelada b do subtrem k .

R_{st} Quantidade de produto s recebida por fornecimento externo à planta no *slot* t .

D_{st} Quantidade de produto s retirada no *slot* t para ser entregue a um cliente.

Restrições no âmbito das expressões:

H_{ks} Operação de armazenamento dentro do processador do estado s gerado pela subtrem k .

Sa_{st} Slot “anterior” ao *slot* t para fins de balanço de massa do estado s .

Ec_j Conjunto dos *slots* onde há disputa pela ocupação do processador j .

Ap Conjunto dos subtrens que realizam armazenamento dentro de um processador.

I_j Conjunto dos subtrens atribuídos ao processador j .

$Prod$ Conjunto dos subtrens geradores de um produto final.

T'_s Conjunto dos subtrens produzindo o estado s .

T_s Conjunto dos subtrens consumindo o estado s .

Te_s Conjunto dos *slots* em que pode haver consumo ou produção do estado s .

Parâmetros novos incluídos nesta formulação:

δ_{ks} Tempo decorrido desde o início do processamento do subtrem k até o instante de consumo do estado s .

λ_{kj} Tempo de processamento do subtrem k no processador j . λ_{kj} é igual ao tempo de processamento da operação i (tp_i) pertencente ao subtrem k no processador j .

ϕ_{kj} Tempo decorrido entre o início do processamento do subtrem k e o início da ocupação do processador j . ϕ_{kj} é igual ao tempo decorrido entre o início do processamento do subtrem k e o início do processamento da operação i (que pertence ao subtrem k) e que ocupa o processador j .

As únicas expressões matemáticas desta formulação que sofreram mudanças significativas em relação à formulação apresentada usando as janelas de tempo, foram as expressões de balanço de massa (expressão 3.8) e de utilização dos processadores (expressão 3.10). A expressão de capacidade de armazenamento (expressão 3.4) permaneceu inalterada e as demais expressões matemáticas tiveram apenas a troca do índice i pelo índice k .

A expressão 3.8 apresenta a expressão de balanço de massa com o uso de subtrens. A expressão: $M'_{ks} * W_{b,k,t-p_{ks}}$ define que serão geradas M'_{ks} unidades do estado s no instante t se o subtrem começar a ser processado no *slot* $t-p_{ks}$. A restrição:

$ebt_{bk}+p_{ks} \leq t \leq lft_{bk}+p_{ks}-tp_k+1$, força que a expressão $M'_{ks} * W_{b,k,t-p_{ks}}$ só figure na expressão de balanço de massa se as janelas de tempo permitirem que o processamento da batelada b se inicie no *slot* $t-p_{ks}$.

A expressão $M_{ks} * W_{b,k,t-\delta_{ks}}$ define que serão consumidas M_{ks} unidades do estado s no instante t se o subtrein começar a ser processado no *slot* $t-\delta_{ks}$. A restrição: $ebt_{bk}+\delta_{ks} \leq t \leq lft_{bk}+\delta_{ks}-tp_k+1$, serve para forçar que a expressão $M_{ks} * W_{b,k,t-\delta_{ks}}$ só seja escrita na expressão de balanço de massa se as janelas de tempo permitirem que o processamento da batelada b se inicie no *slot* $t-\delta_{ks}$.

A expressão 3.10 apresenta a expressão de utilização de processadores com o uso de subtreins. Quando a expressão de utilização de processadores é tratada por operações, diz-se que um processador j pode ser ocupado em qualquer *slot* t por no máximo uma operação i . Quando esta formulação é tratada por subtreins, diz-se o mesmo, apenas tomando-se o cuidado de representar corretamente como se dá a ocupação de um processador j por um subtrein k . As variáveis de alocação W_{bkt} , só são escritas nesta expressão se a janela de tempo da batelada b do subtrein k permitem que o processamento desta batelada seja iniciada no *slot* t' . Isto é expressado por: $ebt_{bk} \leq t' \leq lft_{bk}-tp_k+1$. Além disso, as variáveis binárias de alocação das bateladas dos subtreins serão escritas na expressão 3.10 se o início do processamento da batelada b do subtrein k no *slot* t' implica na ocupação do processador j no *slot* t por esta batelada. Isto é expressado por: $t-\phi_{kj}-\lambda_{kj}+1 \leq t' \leq t-\phi_{kj}$.

O exemplo a seguir ilustra a utilização dos parâmetros δ_{ks} , λ_{kj} e ϕ_{kj} . A figura 3.4 representa dois subtreins, que estão delimitados por retângulos tracejados. Cada subtrein é formado pelas operações ligadas por estados sem capacidade de armazenamento (ZW). Neste exemplo, os estados com capacidade de armazenamento são indicados por um círculo cinza, enquanto que os estados sem capacidade de armazenamento (com restrição de armazenagem ZW – zero wait) são indicados por um círculo branco. No exemplo abaixo, as operações A, B e C pertencem ao mesmo subtrein (ST1) e a operação D compõe sozinha outro subtrein (ST2).

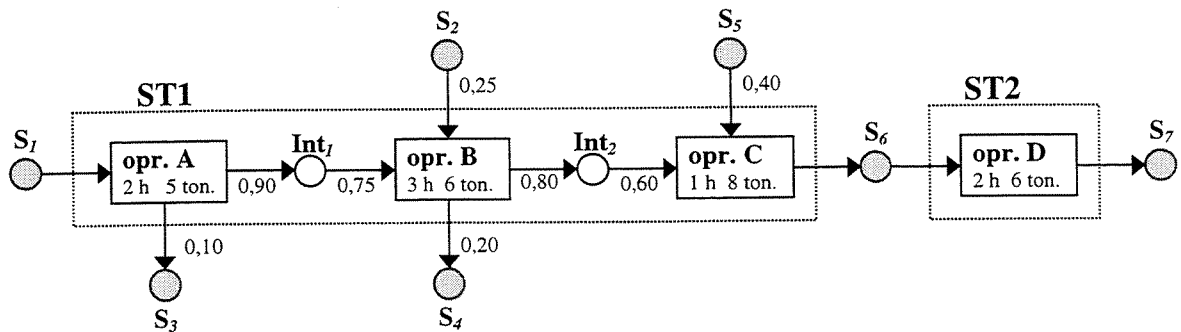


Figura 3.4 – Receita com dois subtreins, ST1 e ST2.

A tabela 3.1 apresenta os valores de δ_{ks} de todos os estados consumidos e os valores de p_{ks} de todos os estados produzidos pelos dois subtreins dados no exemplo da figura 3.4.

Tabela 3.1 – Valores de M'_{ks} , M_{ks} , p_{ks} e δ_{ks} encontrados no exemplo da figura 3.4.

Subtrem	Estado gerado	M'_{ks}	p_{ks}	Subtrem	Estado consumido	M_{ks}	δ_{ks}
ST1	S3	0,5	2	ST1	S1	5	0
ST1	S4	1,2	5	ST1	S2	1,5	2
ST1	S6	8	6	ST1	S5	3,2	5
ST2	S7	6	2	ST2	S6	6	0

Na tabela 3.2 são dados os valores de ϕ_{kj} e de λ_{kj} para o processador (do exemplo da figura 3.4) onde são realizadas as operações B e D , pertencentes aos subtrens $ST1$ e $ST2$, respectivamente.

Tabela 3.2 - Valores de ϕ_{kj} e de λ_{kj} no processador que realiza as operações B e D .

Subtrem	ϕ_{kj}	λ_{kj}
ST1	2	3
ST2	0	2

Na figura 3.5 são apresentadas todas as possibilidades de alocação de duas bateladas (do subtrem $ST1$ e do subtrem $ST2$) no *slot* 19, no processador onde são realizadas as operações B e D . As possíveis ocupações deste processador pelos dois subtrens são indicadas em preto. Como o tempo de processamento do subtrem $ST1$ neste processador é igual a 3 horas ($\lambda_{ST1,j} = 3$), há 3 possibilidades de ocupação do *slot* 19 por este subtrem. O tempo decorrido entre o início do processamento do subtrem $ST1$ e o início da operação B é igual a 2 ($\phi_{ST1,j} = 2$). O *slot* $t - \phi_{ST1,j}$ (igual a 17 no exemplo) da expressão 3.10 identifica quando o processamento do subtrem $ST1$ deverá ser iniciado para que a operação B deste subtrem comece a ser executada no *slot* 19. O *slot* $t - \phi_{kj} - \lambda_{kj} + 1$ (igual a 15 no exemplo) identifica quando o processamento do subtrem $ST1$ deverá ser iniciado para que o processamento da operação B deste subtrem seja encerrado no *slot* 19. Por isso, a expressão 3.10 do *slot* 19 escreverá as variáveis de alocação do subtrem $ST1$ para os *slots* 15, 16 e 17. Por sua vez, o subtrem $ST2$ poderá ocupar o *slot* 19 se o seu processamento for iniciado nos *slots* 18 ou 19.

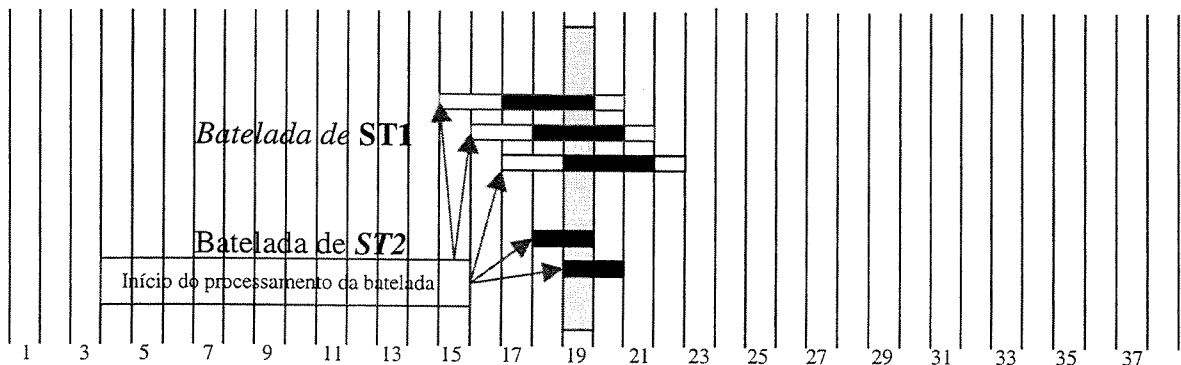


Figura 3.5 – Disputa entre duas bateladas pelo processamento no *slot* 19.

3.4. Formulação MILP reduzida

Em problemas de média e de grande dimensão, a resolução do MILP pode exigir a abertura de um grande número de nós, e portanto, levando a tempos de execução elevados. Uma das principais razões para este problema das formulações MILP é a chamada degenerescência. A degenerescência ocorre quando uma grande quantidade de variáveis binárias podem tomar valores iguais a 0 ou 1 sem que isto provoque grande impacto sobre a função objetivo. No problema de *scheduling* minimizando a antecipação (*earliness*) dos produtos finais, isto pode ocorrer se existem processadores dedicados (atribuídos a apenas uma operação) ou pouco carregados atribuídos a operações que geram estados intermediários. Neste caso, podem existir muitas possibilidades de alocação das bateladas destas operações e que podem ter pequeno impacto sobre a função objetivo. O impacto será principalmente sobre as restrições de balanço de massa, mas estas podem ter pouca influência sobre o resultado da função objetivo se existe capacidade de armazenagem entre as operações que geram estados intermediários e as operações que geram produtos finais.

A proposta de uma formulação MILP reduzida consiste na eliminação das situações responsáveis pela degenerescência do problema, resultando na diminuição do número de variáveis binárias. Basicamente eliminam-se:

- Os subtrens utilizando processadores dedicados;
- Os subtrens que compartilham um processador, se a união das janelas de tempo das suas bateladas abrange intervalos de tempo com carregamento baixo.

O procedimento proposto pode ser visto como uma evolução em relação às técnicas de *scheduling* orientadas pelos gargalos (p. ex.: Morton e Pentico, 1993). Estas técnicas estão baseadas na detecção sequencial dos gargalos e na sua solução. A crítica principal é que, após resolver um gargalo, podem aparecer outros gargalos (*wandering bottlenecks*) e no fim é questionável se a ordem em que foram tratados os gargalos foi a mais adequada. A proposta da formulação reduzida não escolhe um gargalo. Esta proposta elimina do problema de *scheduling* os subtrens não envolvidos em gargalos, tratando conjuntamente o resto. Este tratamento conjunto dos gargalos é possível através da formulação MILP.

Na formulação MILP reduzida, a solução é obtida em dois passos. No primeiro passo obtém-se a solução do problema reduzido, sem alocar os subtrens eliminados. No segundo passo alocam-se as bateladas dos subtrens eliminados, mantendo-se as alocações obtidas no primeiro passo. Obviamente este procedimento não garante a obtenção da solução ótima, mas possui uma visível justificativa prática.

A escolha dos subtrens eliminados é feita pelo usuário utilizando as informações de carregamento dadas pelas curvas de demanda agregada (crucialidades) dos processadores e baseando-se no conhecimento da receita da planta (principalmente, as restrições de armazenagem). A informação do carregamento dos processadores permite a eliminação dos subtrens atribuídos a processadores pouco carregados. Porém, deve-se considerar também as restrições de armazenagem porque pode haver uma forte interdependência nas alocações de dois subtrens ligados por um estado com armazenagem limitada. Ainda que um dos

subtrens esteja atribuído a um processador pouco carregado, as alocações das bateladas deste subtrem podem estar bastante restringidas pela limitação de armazenagem. A eliminação de subtrens sujeitos a restrições de armazenagem pode resultar na infactibilidade da solução no segundo passo.

Para a execução do primeiro passo da formulação MILP reduzida é necessário introduzir o conceito de subtrens ativos, de estados ativos e semi-ativos. Os subtrens não eliminados pelo usuário são definidos como ativos. Um estado é ativo se ele é gerado por um subtrem ativo e consumido por um subtrem ativo. Um estado é semi-ativo se o subtrem gerador (consumidor) deste estado é ativo e o subtrem consumidor (gerador) deste estado foi eliminado. As alocações dos subtrens têm que satisfazer as relações de precedência decorrentes do balanço de massa. Se um estado é ativo, a expressão de balanço de massa deste estado garante a satisfação das relações de precedência entre as bateladas dos subtrens ativos ligados a este estado. Se um subtrem ativo consome um estado semi-ativo, é preciso garantir que a alocação deste subtrem satisfaz as relações de precedência implicadas pelos subtrens eliminados. Como no primeiro passo da formulação reduzida não há a alocação dos subtrens inativos, é preciso garantir que haja um intervalo de tempo suficiente para a alocação no segundo passo das bateladas dos subtrens eliminados no primeiro passo. Para tanto, é introduzida uma expressão no primeiro passo da formulação MILP reduzida que impõe um espaçamento mínimo entre as bateladas dos subtrens ativos ligados a estados semi-ativos.

Este espaçamento mínimo ($gap_{bkb'k'}$ na expressão 3.15) é calculado antes do primeiro passo da formulação MILP reduzida, após o usuário definir os subtrens ativos. Para tanto, utilizam-se as relações de precedência entre bateladas de operações (subtrens) obtidas na fase de planejamento. Identificam-se os estados semi-ativos com subtrens consumidores eliminados. Para cada estado semi-ativo com subtrens consumidores eliminados, percorre-se “para frente” (sentido *forward*) o diagrama STN enquanto os subtrens encontrados forem inativos (eliminados). O procedimento é finalizado quando se encontram estados semi-ativos com subtrens consumidores ativos. Se o subtrem k gera um estado semi-ativo e o subtrem k' consome um estado semi-ativo, o parâmetro $gap_{bkb'k'}$ representará o intervalo de tempo mínimo que deve existir entre o início do processamento da batelada b do subtrem k e o início do processamento da batelada b' do subtrem k' . O valor é obtido através de um procedimento que adiciona os tempos de processamento das bateladas dos subtrens encontrados no percurso do STN e que estão interligados por relações de precedência.

A figura 3.6 apresenta um exemplo do cálculo do parâmetro $gap_{bkb'k'}$ (usado na expressão 3.15), onde os subtrens **B** e **C** foram eliminados e os subtrens **A** e **D** estão ativos. O tempo de processamento de cada subtrem é apresentado acima do respectivo subtrem no diagrama da receita de produção (STN) na figura 3.6a. O tamanho das bateladas de cada subtrem é apresentado abaixo da sua representação no diagrama STN. A figura 3.6a apresenta as relações de precedência (*pegging*) decorrentes do balanço de massa entre as bateladas do exemplo. Os subtrens e estados ativos são apresentados em cinza, os estados semi-ativos (**SA** e **SC**) são apresentados em preto e o estado inativo (**SB**) é apresentado em

branco nesta figura.

O parâmetro $gap_{bkb'k'}$ é definido em um grafo a partir do caminho mais longo entre a batelada b do subtrem k e a batelada b' do subtrem k' . A figura 3.6b apresenta os caminhos mais longos, através de arcos pontilhados, que impõem um intervalo mínimo entre o início de produção da batelada $A/1$ e o início das demais bateladas do exemplo, sendo que o valor deste intervalo mínimo é apresentado ao lado de cada batelada. A figura 3.6c apresenta os caminhos mais longos (intervalos mínimos) entre o início de produção da batelada $A/2$ e o início das demais bateladas do exemplo. O intervalo mínimo entre o início da produção das bateladas $A/1$ e $D/1$ ($gap_{A,1,D,1}$) é igual a 16. Este intervalo ($gap_{A,1,D,1}$) garante o tempo mínimo necessário para a produção das bateladas $B/1$, $B/2$, $B/3$, $C/1$ e $C/2$, que foram eliminadas do passo 1, entre as bateladas $A/1$ e $D/1$. A figura 3.6b mostra que é preciso impor um intervalo mínimo entre as bateladas $A/1$ e $D/1$, e também entre as bateladas $A/1$ e $D/2$. Impor um intervalo mínimo, através da expressão 3.15, somente entre as bateladas $A/1$ e $D/1$ não garantirá que haverá o intervalo mínimo (de 21 unidades de tempo) necessário entre os inícios de produção das bateladas $A/1$ e $D/2$. Para que haja um intervalo suficiente para o processamento das bateladas dos subtrems eliminados (B , C), a expressão 3.15 deverá ser escrita para os pares de bateladas: $A/1$ e $D/1$; $A/1$ e $D/2$; $A/2$ e $D/1$; e $A/2$ e $D/2$.

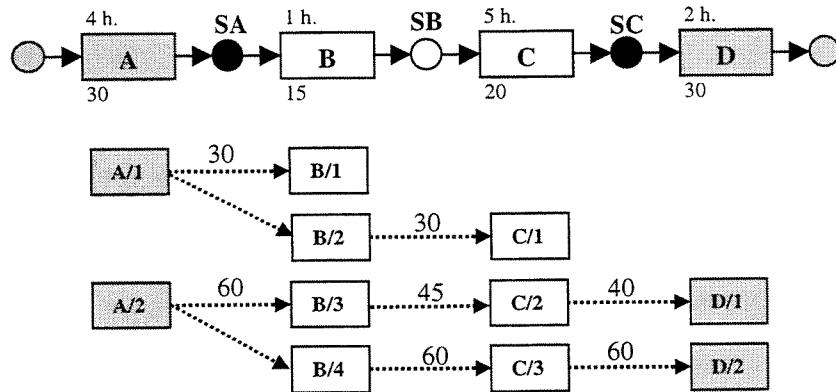


Figura 3.6a. Representação STN e relações de *pegging*.

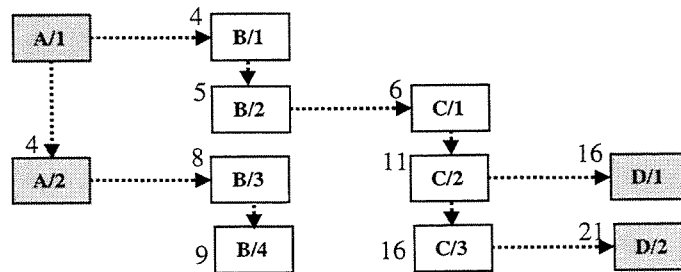


Figura 3.6b. Intervalo mínimo entre $A/1$ e as bateladas que a seguem.

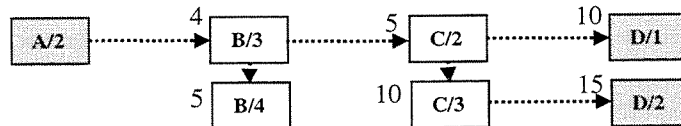


Figura 3.6c. Intervalo mínimo entre $A/2$ e as bateladas que a seguem.

As expressões matemáticas da formulação MILP reduzida são apresentadas a seguir e as modificações em relação à formulação anterior são discutidas.

a) Balanço de massa

$$S_{st} = S_{s,t-1} + \sum_{k \in T_s} M'_{ks} \sum_{\substack{b \in Bat_k \\ t \geq ebt_{bk} + p_{ks} \\ t \leq lft_{bk} + p_{ks} - tp_k + 1}} W_{b,k,t-p_{ks}} - \sum_{k \in T_s} M_{ks} \sum_{\substack{b \in Bat_k \\ t \geq ebt_{bk} + \delta_{ks} \\ t \leq lft_{bk} + \delta_{ks} - tp_k + 1}} W_{b,k,t-\delta_{ks}} + R_{st} - D_{st} \quad \forall s \in As, t \in Te_s \quad [3.14]$$

b) Espaçamento entre bateladas de subtrens ativos

$$\sum_{t=ebt_{bk}}^{lft_{bk}-tp_k+1} W_{bkt} * (t + gap_{bb'k'}) \leq \sum_{t'=ebt_{b'k'}}^{lft_{b'k'}-tp_{k'}+1} W_{b'k't'} * t' \quad \forall k \in Ak, k' \in Ak, b \in Bat_k, b' \in Bat_{k'} / peg_{bb'k'} = true \quad [3.15]$$

c) Alocação das bateladas resultantes do planejamento

$$\sum_{t=ebt_{bk}}^{lft_{bk}-tp_k+1} W_{bkt} = 1 \quad \forall k, b \in Bat_k / k \in Ak \therefore k \notin Ap \quad [3.16]$$

d) Utilização de processadores

$$\sum_{k \in I_j} \sum_{b \in Bat_k} \sum_{\substack{t' = t - \phi_{kj} - \lambda_{kj} + 1 \\ t \geq ebt_{bk} \\ t' \leq lft_{bk} - tp_k + 1}} W_{bkt'} \leq 1 \quad \forall j, t \in Ec_j \quad [\text{idêntica à expressão 3.10}]$$

e) Ordenamento de bateladas de um mesmo subtrem

$$\sum_{t=ebt_{b'k}}^{lft_{b'k}-tp_k+1} W_{b'kt} * (t + tp_k) \leq \sum_{t=ebt_{bk}}^{lft_{bk}-tp_k+1} W_{bkt} * t \quad \forall k \in Ak, b' \in Bat_k, b \in Bat_k / b' < b \quad [3.17]$$

f) Capacidade de armazenamento

$$S_{st} \leq C_s \quad \forall s \in As, t \in Te_s \quad [3.18]$$

g) Operação de armazenamento dentro do processador

$$W_{b'k't} \leq W_{b',k',t-1} + \sum_{s \in T_s'} \sum_{k' \in H_{ks}} W_{b',k,t-p_{ks}}$$

$$\forall k' \in Ak, b' \in Bat_{k'}, t / ebt_{b'k'} \leq t \leq lft_{b'k'} \quad [3.19]$$

h) Função objetivo: $\min \{obj\}$

$$obj = \sum_{k \in Prod} \sum_{b \in Bat_k} \{lft_{bk} - tp_k + 1 - (\sum_{t \geq ebt_{bk}}^{lft_{bk} - tp_k + 1} W_{bkt} * t)\} \quad [\text{idêntica à expressão 3.13}]$$

onde,

Variáveis:

W_{bkt} = 1 indica o início do processamento da batelada b do subtrem k no *slot* t . Caso contrário, W_{bkt} = 0.
 S_{st} Massa (ou volume) do estado s armazenada no *slot* t .

Parâmetros:

M'_{ks} Massa (ou volume) de saída da subtrem k na forma de estado s .
 M_{ks} Massa (ou volume) de entrada na subtrem k na forma de estado s .
 C_s Capacidade de armazenamento do estado s .
 p_{ks} Tempo decorrido desde o início do processamento da operação i até que esta operação produza o estado s .
 tp_k Tempo total de processamento do subtrem k .
 δ_{ks} Tempo decorrido desde o início do processamento do subtrem k até o instante de consumo do estado s .
 λ_{kj} Tempo de processamento do subtrem k no processador j . λ_{kj} é igual ao tempo de processamento da operação i (tp_i) pertencente ao subtrem k no processador j .
 ϕ_{kj} Tempo decorrido entre o início do processamento do subtrem k e o início da ocupação do processador j . ϕ_{kj} é igual ao tempo decorrido entre o início do processamento do subtrem k e o início do processamento da operação i (que pertence ao subtrem k) e que ocupa o processador j .
 ebt_{bk} Início da janela de tempo da batelada b do subtrem k .
 lft_{bk} Final da janela de tempo da batelada b do subtrem k .
 R_{st} Quantidade de produto s recebida por fornecimento externo à planta no *slot* t .
 D_{st} Quantidade de produto s retirada no *slot* t para ser entregue a um cliente.

Restrições no âmbito das expressões:

H_{ks} Operação de armazenamento dentro do processador do estado s gerado pela subtrem k .
 Sa_{st} Slot “anterior” ao *slot* t para fins de balanço de massa do estado s .
 Ec_j Conjunto dos *slots* onde há disputa pela ocupação do processador j .
 Ap Conjunto dos subtrens que realizam armazenamento dentro de um processador.
 I_j Conjunto dos subtrens atribuídos ao processador j .
 $Prod$ Conjunto dos subtrens geradores de um produto final.
 T'_s Conjunto dos subtrens produzindo o estado s .
 T_s Conjunto dos subtrens consumindo o estado s .
 Te_s Conjunto dos *slots* em que pode haver consumo ou produção do estado s .

Parâmetros e restrições novas decorrentes do uso da formulação reduzida:

As Conjunto dos estados ativos na formulação MILP reduzida.
 Ak Conjunto dos subtrens ativos na formulação MILP reduzida.
 $gap_{bkb'k'}$ Espaçamento mínimo necessário entre o processamento da batelada b do subtrem k e a batelada b' do subtrem k' .
 $peg_{bkb'k'}$ $=true$ indica que a batelada b do subtrem k e a batelada b' do subtrem k' estão ligadas por relações de precedência decorrentes do balanço de massa.

A expressão de balanço de massa (expressão 3.14) só será escrita para os estados ativos, ou seja, para os estados (pertencentes ao conjunto A_s) que são gerados e consumidos por subtrens ativos. A expressão suplementar (expressão 3.15) estabelece que deve haver um espaçamento mínimo, igual a $gap_{bkb'k'}$, entre o processamento da batelada b do subtrem k e o processamento da batelada b' do subtrem k' . Os subtrens ativos k e k' (pertencentes ao conjunto A_k), respectivamente, geram e consomem estados semi-ativos e estão ligados por relações de precedência decorrentes do balanço de massa, indicadas por $peg_{bkb'k'}=true$. Na fase de planejamento são identificadas relações de precedência entre as bateladas dos subtrens a partir do procedimento de explosão para frente, apresentado na seção 2.3.2. Este mesmo procedimento de explosão permite que se calcule o espaçamento mínimo ($gap_{bkb'k'}$) entre o processamento de duas bateladas de subtrens ligados por relações de precedência.

A expressão de utilização dos processadores (expressão 3.10) e a função objetivo (expressão 3.13) são as mesmas usadas na formulação apresentada na seção anterior. O conjunto dos *slots* onde há disputa pela ocupação de um processador j (Ec_j), presente na expressão de utilização dos processadores, é afetado pela definição dos subtrens ativos. Por isso, as expressões referentes à utilização dos processadores nas duas formulações, para um mesmo problema, podem ser diferentes. O conjunto *Prod*, usado na função objetivo para definir os subtrens produtores de produtos finais também é afetado pela definição dos subtrens ativos. Um subtrem que gera um produto final pode estar inativo na primeira fase da solução da formulação reduzida. Neste caso, caberá ao usuário incluir outro subtrem ativo no conjunto *Prod* para que a função objetivo leve em consideração o último subtrem ativo na receita deste produto. Com isso, as funções objetivo geradas pelas duas formulações, para um mesmo problema, também poderão ser diferentes.

Assim como a expressão de balanço de massa (expressão 3.14), a expressão de capacidade de armazenamento (expressão 3.18) será escrita somente para os estados ativos na formulação reduzida. As expressões de alocação das bateladas resultantes do planejamento (expressão 3.16), de ordenamento de bateladas de uma mesma operação (expressão 3.17) e de operação de armazenamento dentro do processador (expressão 3.19) passam a ser escritas somente para os estados ativos na formulação reduzida.

O primeiro passo da formulação MILP reduzida pode dar lugar a alocações que tornem infactível o problema no segundo passo, mesmo que o problema global seja factível. Isto não acontece quando eliminam-se subtrens utilizando processadores dedicados a uma única operação mas é possível quando eliminam-se subtrens envolvidos em processadores com baixo compartilhamento. A razão disso é que o primeiro passo vai ignorar a alocação destes subtrens eliminados e no segundo passo estas alocações podem ser impossíveis dentro das janelas de tempo, devido às outras alocações feitas no primeiro passo. Esta infactibilidade pode ser detectada através da: i) determinação das janelas de tempo para as bateladas não alocadas resultantes das alocações no primeiro passo e da ii) propagação das restrições originadas por estas janelas de tempo (que necessariamente serão menores ou iguais às janelas de tempo originais do planejamento).

3.5. Resultados do *scheduling*

Nesta seção apresentam-se alguns exemplos para ilustrar a influência da propagação de restrições na diminuição da dimensão do problema de *scheduling* e a utilização da formulação MILP reduzida. Nas soluções de *scheduling* apresentadas foi utilizada uma discretização uniforme do tempo com intervalo unitário. Os problemas MILP foram resolvidos com o software GAMS/OSL (GAMS, 1998).

3.5.1. Influência da propagação de restrições.

A propagação de restrições reduz as janelas de tempo das bateladas diminuindo a dimensão da formulação MILP. As figuras 3.7a e 3.7b mostram os resultados do planejamento para o exemplo 1 com o plano de disponibilidade de matérias primas indicado nas figuras, com e sem propagação de restrições. Utiliza-se como demanda de produtos finais, nos dois casos, a demanda dada na tabela 2.1a (seção 2.4.1). O início do planejamento nos dois casos ocorre no instante 10. Observa-se uma redução nas janelas de tempo de diversas bateladas quando se inclui a propagação de restrições no planejamento (figura 3.7a).

A formulação MILP global, apresentada na seção 3.3., tem as características dadas na tabela 3.3. A tabela inclui os dados referentes às duas formulações MILP em termos de: solução relaxada, solução inteira final, número de iterações, número de nós e tempo de execução em segundos (Pentium III-600). As figuras 3.8a e 3.8b apresentam as cartas de Gantt resultantes do *scheduling* nos dois casos.

Tabela 3.3. Comparação entre as formulações MILP com e sem propagação de restrições.

com propagação	equações	variáveis contínuas	variáveis binárias	não zeros	solução relaxada	solução inteira final	número de iterações	número de nós	tempo (s.)
sim	2372	595	871	10645	54	56	5039	149	16
não	2661	627	1612	12781	88	90	13313	339	75

A tabela 3.3 mostra uma redução de 46 % no número de variáveis binárias e de 10,9 % no número de equações, levando a uma redução sensível no esforço computacional. As duas soluções fornecem valores diferentes para a função de custo, ou seja a antecipação (*earliness*) em relação aos prazos de entrega de produtos finais. Isto se deve à forma como ele é calculado, já que a antecipação é calculada, para cada batelada, em relação ao final da sua janela de tempo. Quando se utiliza a propagação das restrições, diversos fins de janela das bateladas produzindo produtos finais são antecipados, levando a um valor menor da função de custo. De fato, o valor da função de custo neste caso (56) deve ser acrescido da antecipação inevitável de 34 unidades de tempo devida à antecipação dos finais de janelas originais (sem propagação) das bateladas $T41/1,2,3,4$ (6 unidades cada) e $T51/1,2,3,4,5$ (2 unidades cada), conduzindo a um valor de 90 como no segundo caso. As figuras 3.8a e 3.8b

mostram que a alocação das bateladas produzindo produtos finais é a mesma nas duas cartas de Gantt.

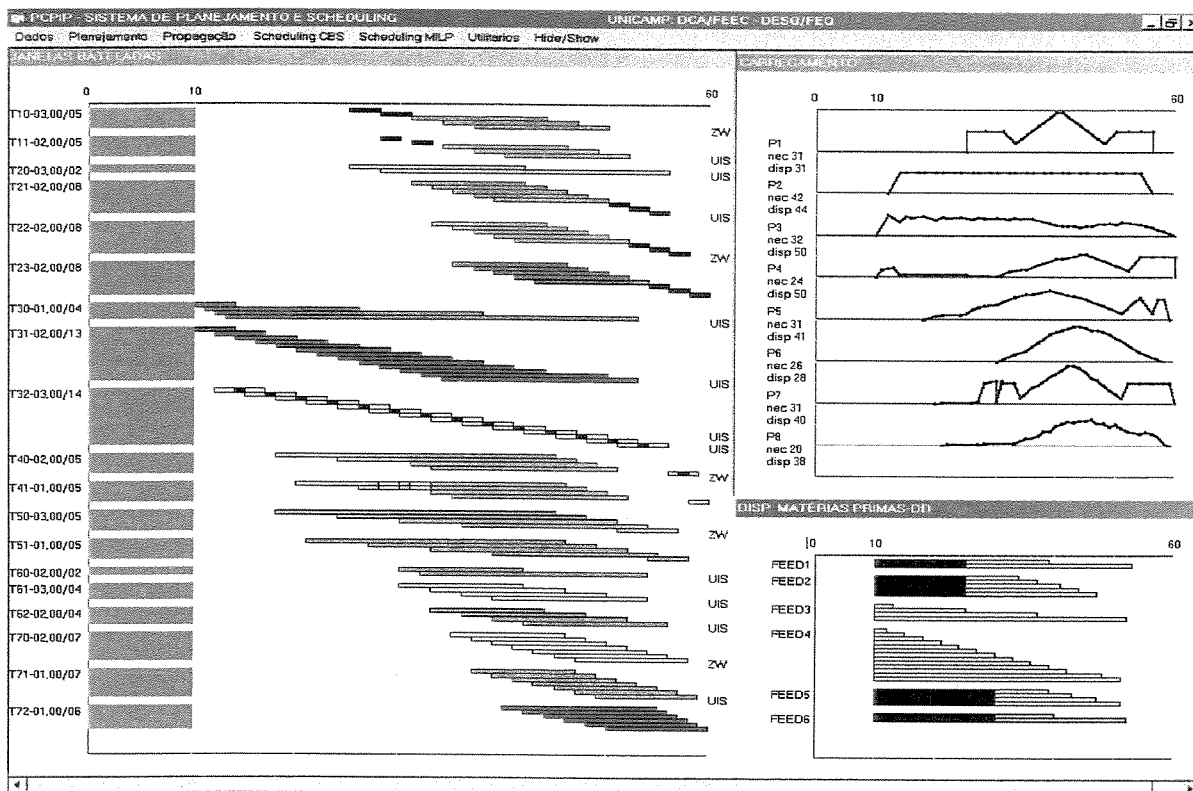


Figura 3.7a. Solução do planejamento com propagação.

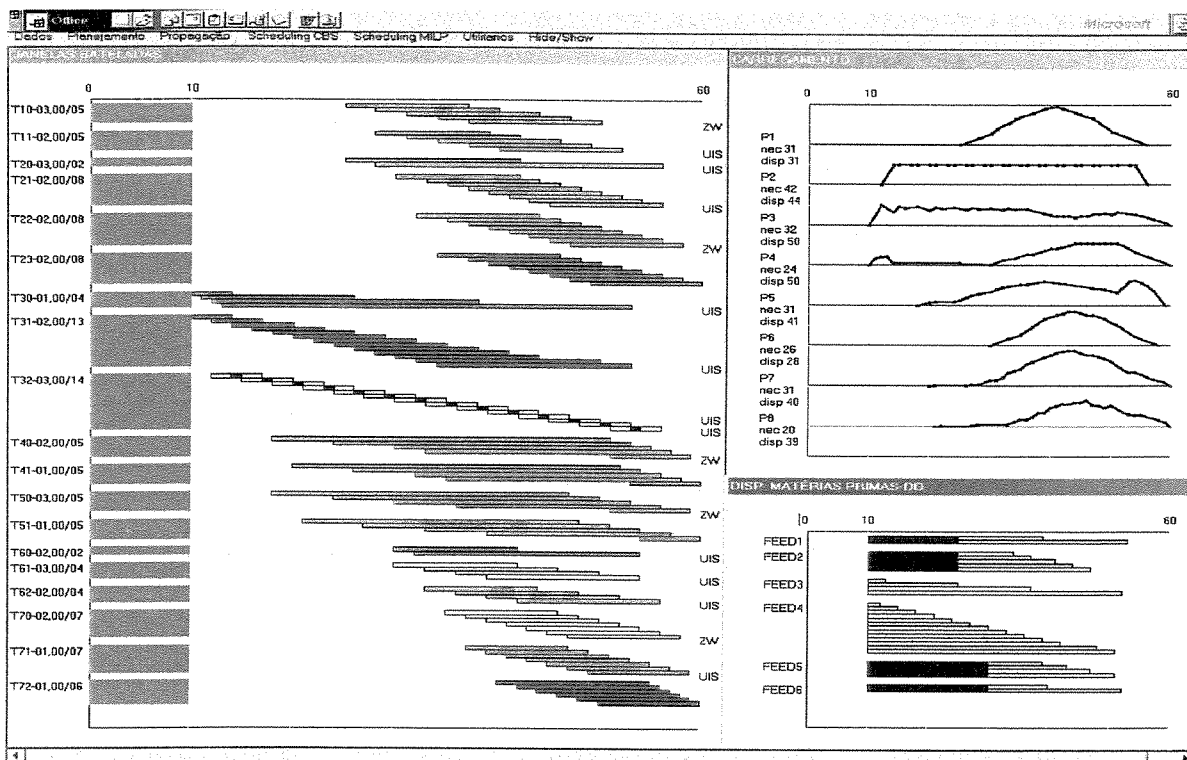


Figura 3.7b. Solução do planejamento sem propagação.

O código de cores utilizado nas cartas de Gantt é o seguinte: operações com estados gerados ou consumidos com restrição de armazenagem ZW são representadas em vermelho, utiliza-se o amarelo para casos de armazenagem NIS, marrom para casos de armazenagem FIS e cinza para casos de armazenagem UIS. Nas situações em que as restrições de armazenagem são diferentes nas entradas e nas saídas a cor utilizada é a da restrição mais forte (pela ordem: ZW, NIS, FIS, UIS).

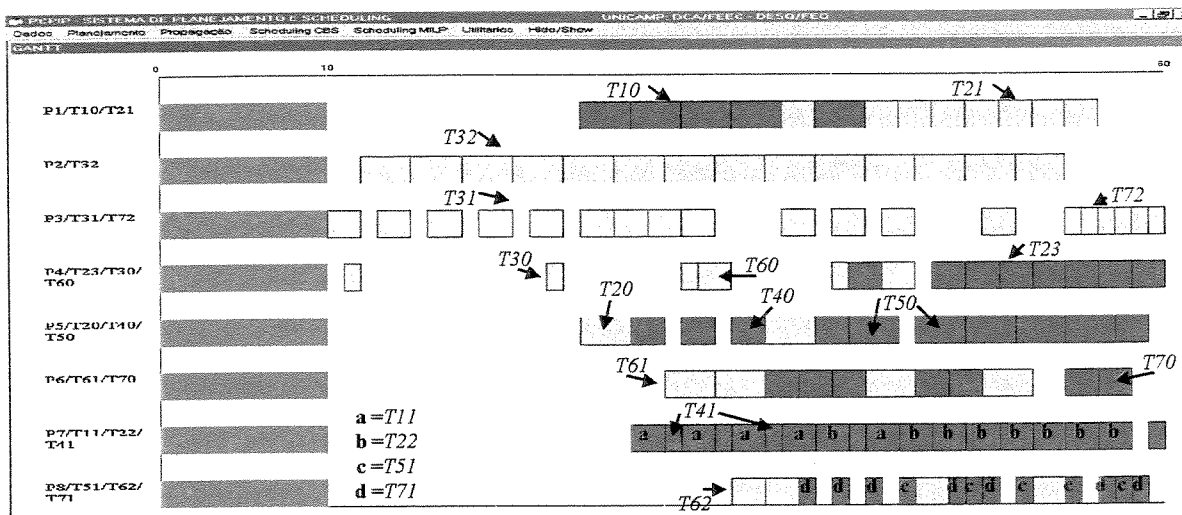


Figura 3.8a. Solução do *scheduling* utilizando a formulação MILP global com propagação.

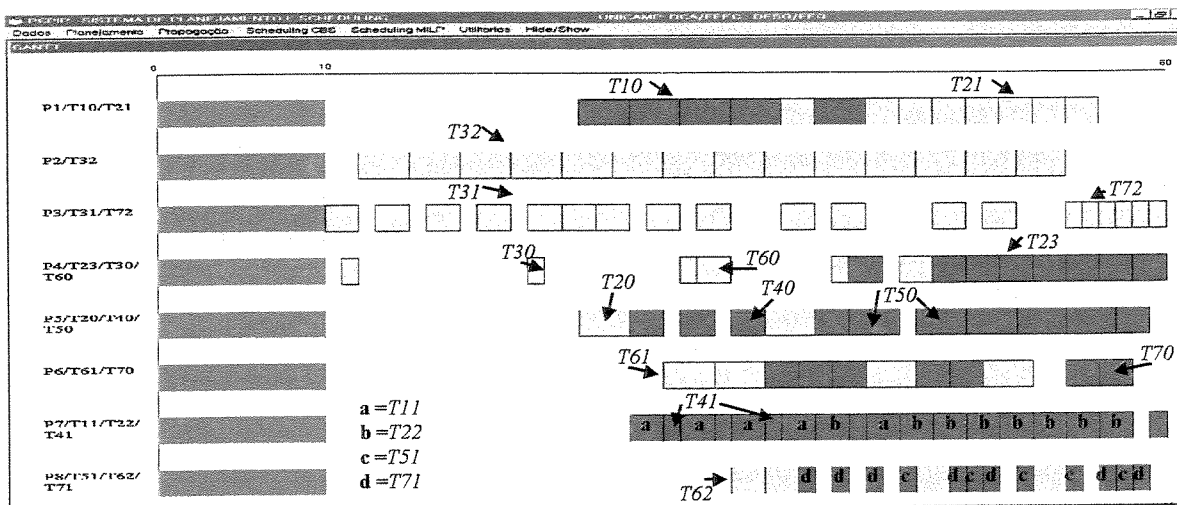


Figura 3.8b. Solução do *scheduling* utilizando a formulação MILP global sem propagação.

A influência da propagação de restrições depende da duração das janelas resultantes da fase de explosão no planejamento. Para o mesmo exemplo, se a disponibilidade das matérias primas *Feed1* e *Feed2* é antecipada para o instante 20, como mostrado na figura 3.9, a propagação de restrições não gera qualquer redução de janela de tempo. As

características do problema MILP resultante estão na tabela 3.4 e a formulação MILP não obteve uma solução inteira em 400000 iterações e 6761 nós.

Tabela 3.4. Características da formulação MILP

equações	variáveis contínuas	variáveis binárias	não zeros
3046	657	1100	15763

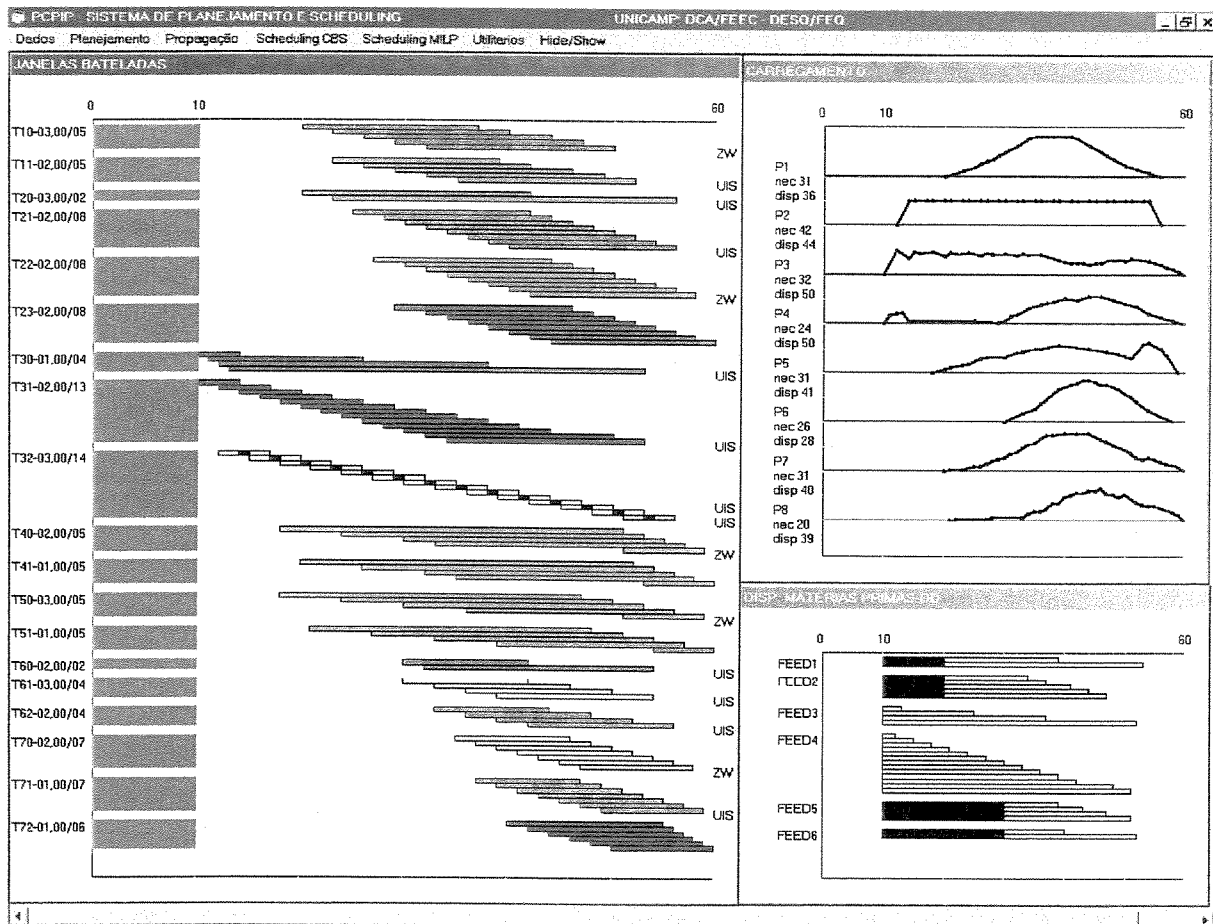


Figura 3.9. Solução do planejamento com antecipação na disponibilidade de *Feed1* e *Feed2*.

3.5.2. Utilização do modelo MILP reduzido

• Exemplo 1

Este exemplo inclui uma situação típica para utilização do modelo MILP reduzido proposto na seção 3.4. A operação *T32* é crítica do ponto de vista da receita (STN) porque ela produz dois produtos intermediários necessários para todos os produtos finais. O tamanho de batelada desta operação (8) faz com que o número de bateladas necessárias para atender todas as demandas destes produtos intermediários seja grande (14), levando a

uma necessidade mínima de 42 unidades de tempo (u.t.) para o seu processamento (14 bateladas x 3 u.t./batelada). O horizonte de planejamento é de 50 (60 – 10), indicando que o processador **P2** onde esta operação deve ser realizada terá um carregamento alto.

As possibilidades de atribuição de operações a processadores (habilitação de processadores) dadas na referência que apresenta este problema (Papageorgiu e Pantelides, 1996) estão indicadas na tabela 3.5.

Tabela 3.5. Habilitação de processadores

processadores	operações	processadores	operações
P1	T10,T21	P5	T20,T40,T50
P2	T32,T41,T51	P6	T61,T70
P3	T31,T72	P7	T11,T22,T41
P4	T23,T30,T60	P8	T62,T71,T51,T72

O sistema de planejamento, como discutido no Capítulo 2, utiliza atribuição única de cada operação a processadores. Dado o carregamento alto de **P2** induzido por **T32** as operações **T41** e **T51** foram atribuídas aos processadores **P7** e **P8** respectivamente, deixando **P2** dedicado para a operação **T32**. Observa-se mesmo assim, nas curvas de carregamento das figuras 3.7a e 3.7b, uma folga total pequena (44 unidades de tempo disponíveis versus 42 necessárias).

A operação **T31** tem também um número de bateladas grande (13) mas o seu tempo de processamento é menor. Esta operação, assim como da operação **T72**, foram atribuídas ao processador **P3** porque a demanda por parte destas duas operações é bastante separada no tempo (**T31** é uma operação do início da receita e **T72** produz um produto final). A folga total no processador **P3** não é pequena (50 u.t. disponíveis versus 32 necessárias). Um raciocínio análogo mostra que o compartilhamento do processador **P4** pelas operações **T30**, **T23** e **T60** tem duas regiões de disputa, uma inicial em que **P4** apenas será ocupado por **T30** e uma intermediária/final em que as janelas das bateladas das outras duas operações se sobrepõem.

Tem-se portanto uma situação em que não existe concorrência pelo processador **P2**, a concorrência é baixa para o processador **P3** e a concorrência para **P4** é limitada numa região. Isto leva a propor um MILP reduzido em que as operações **T30**, **T31** e **T32** são eliminadas no primeiro passo. O resultado é mostrado nas figuras 3.10a e 3.10b para as duas situações de disponibilidade das matéria primas **Feed1** e **Feed2**. Neste exemplo a expressão 3.15 não será escrita para nenhum par de bateladas porque não há estados semi-ativos consumidos por um dos subtrens eliminados.

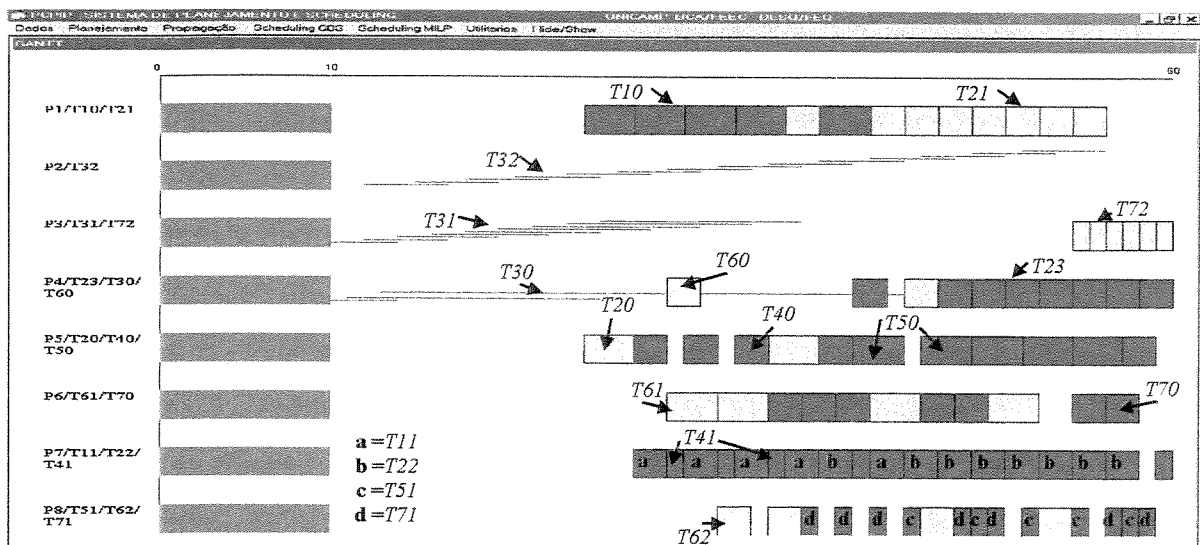


Figura 3.10a. Carta de Gantt para o MILP reduzido (*Feed1* e *Feed2* em 25).

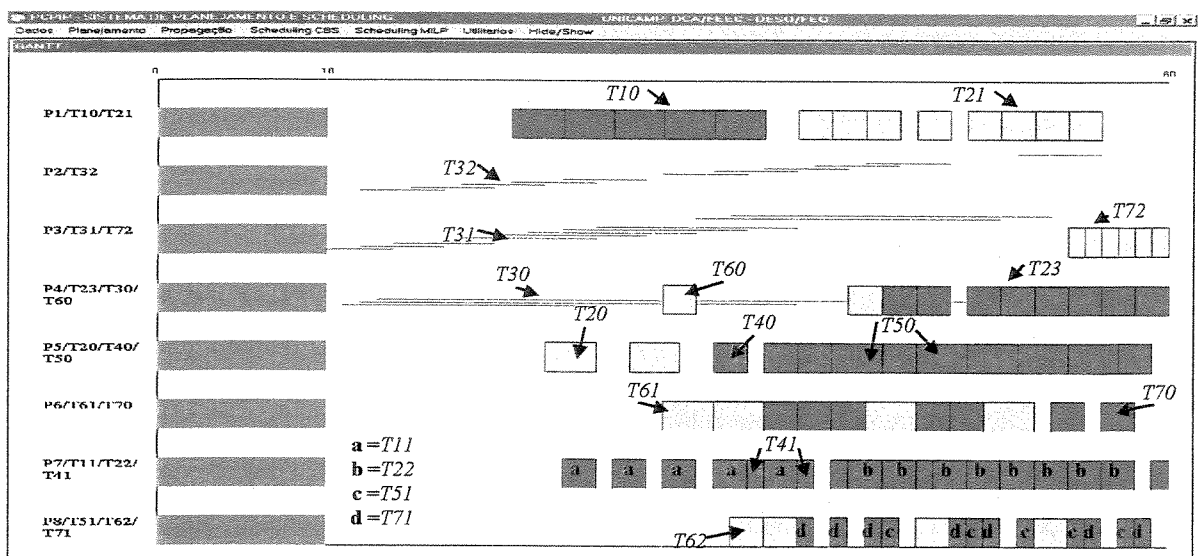


Figura 3.10b. Carta de Gantt para o MILP reduzido (*Feed1* e *Feed2* em 20).

As figuras 3.10a e 3.109b apresentam também as janelas de tempo remanescentes para as bateladas das operações eliminadas, após a alocação das demais operações resultantes do primeiro passo. A tabela 3.6 mostra as características dos problemas MILP (“a” para *Feed1* e *Feed2* em 25; “b” para *Feed1* e *Feed2* em 20).

Tabela 3.6. Características dos problemas MILP reduzidos

	equações	variáveis contínuas	variáveis binárias	não zeros	número de iterações	número de nós	solução inteira	tempo (s.)
a	1713	425	621	7765	3755	97	56	9.6
b	2389	487	850	12777	15521	245	72	50

No segundo passo obtém-se as cartas de Gantt das figuras 3.11a e 3.11b.

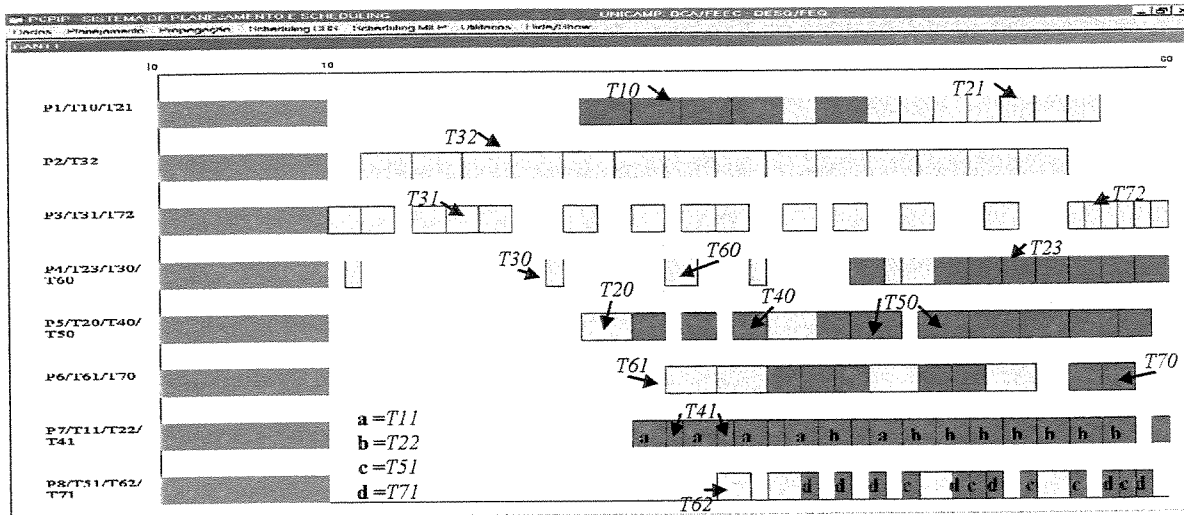


Figura 3.11a. Carta de Gantt para o segundo passo (*Feed1* e *Feed2* em 25)

Comparando as soluções global e em dois passos para a disponibilidade no instante 25 observa-se que as únicas diferenças referem-se a bateladas de intermediários que não tem impacto sobre o valor da função de custo.

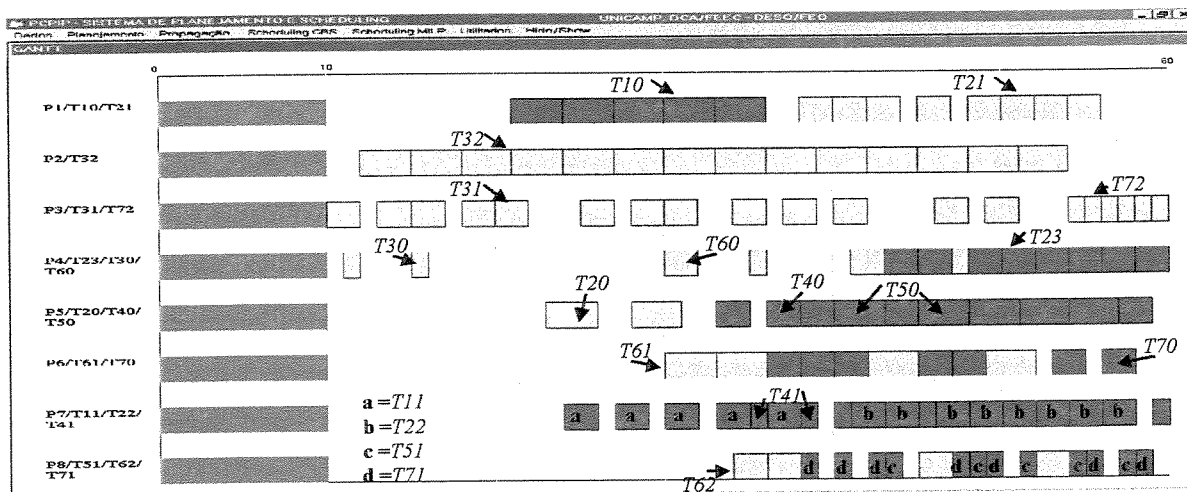


Figura 3.11b. Carta de Gantt para o segundo passo (*Feed1* e *Feed2* em 20)

A solução global para este caso (*Feed1* e *Feed2* em 20) não foi obtida em 400000 iterações. Na abordagem em dois passos obteve-se a solução do primeiro passo em 50 segundos com 15521 iterações. A alocação no segundo passo é trivial. Foi obtida através de uma formulação MILP análoga à do primeiro passo, mas poderia ser utilizada qualquer heurística simples que alocasse as bateladas dentro das janelas.

A abordagem em dois passos mostra-se adequada para eliminar ou reduzir a degenerescência associada às diferentes possibilidades de alocação das operações *T30*, *T31* e *T32* sem impacto na função de custo.

- **Exemplo2.**

Para o exemplo 2, a figura 2.33, reproduzida abaixo, mostra as janelas de tempo e o carregamento dos processadores resultantes da propagação de restrições.

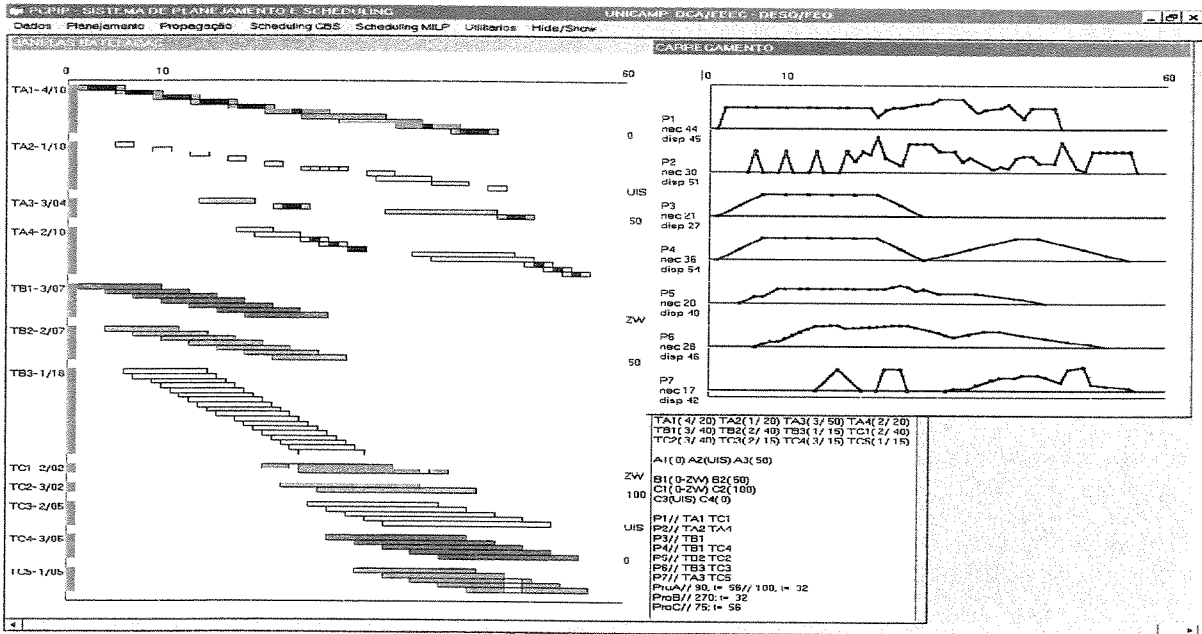


Figura 2.33. Janelas de tempo e demandas agregadas dos processadores para o exemplo 2.

A definição de problemas reduzidos neste exemplo precisa considerar as condições de armazenagem restritas. Na seção 2.8 foi levantada a possibilidade de as janelas de tempo serem ineficazes na fase de *scheduling* quando as restrições de armazenagem são fortes. Da mesma forma, na definição do problema reduzido, esta situação deve ser tratada cuidadosamente visto que o primeiro passo não considera a alocação de algumas bateladas e suas janelas de tempo remanescentes, para o segundo passo, são obtidas a partir das alocações do primeiro passo. No caso em que algumas das operações alocadas no primeiro passo estejam interligadas a operações eliminadas através de condições de armazenagem muito restritas, as janelas remanescentes podem ser ineficazes.

O processador *P6* tem um carregamento baixo (figura 2.33) e as operações atribuídas a ele, *TB3* e *TC3* (em amarelo na representação STN reproduzida abaixo), estão interligadas com as outras operações através de restrições de armazenagem FIS ou UIS. As tarefas *TB3* e *TC3* são assim candidatas a serem eliminadas no primeiro passo. As soluções do *scheduling* global e em dois passos são mostradas nas figuras 3.12, 3.13a e 3.13b abaixo.

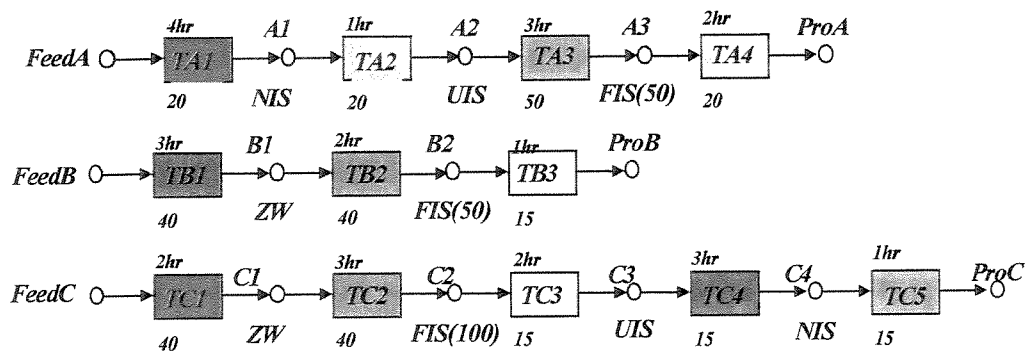


Figura A2. representação STN do exemplo 2.

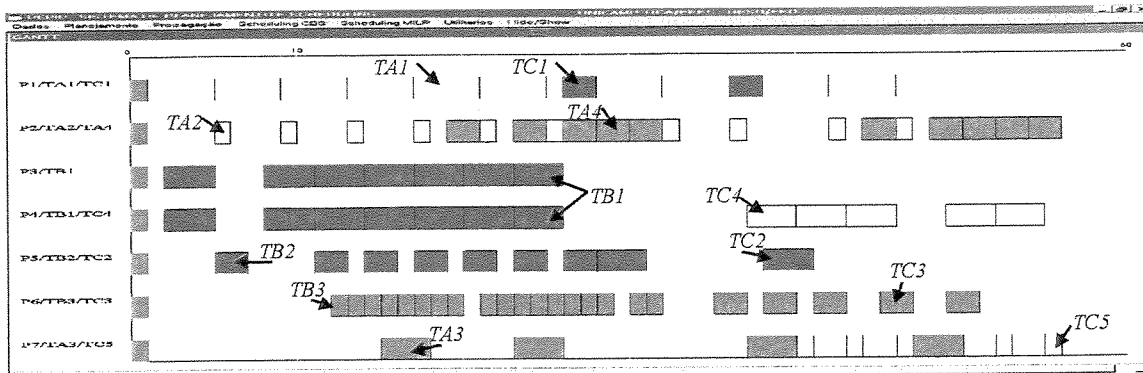


Figura 3.12. Solução global

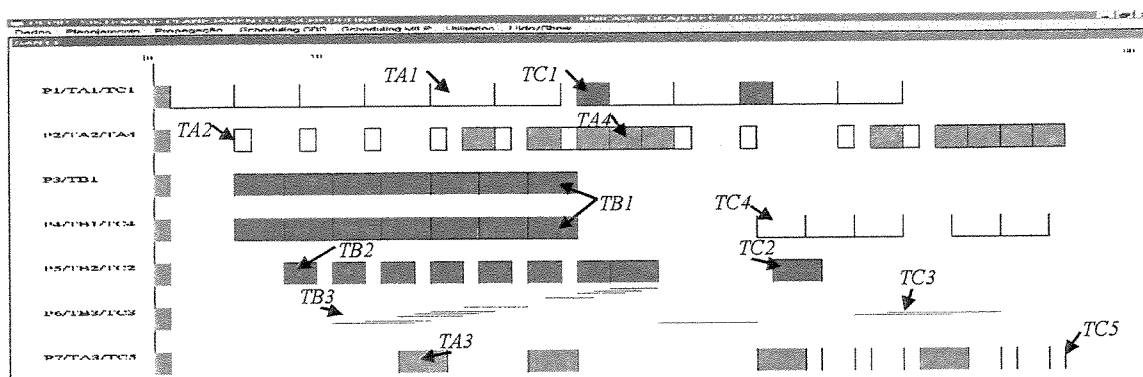


Figura 3.13a. Solução do primeiro passo do MILP reduzido.

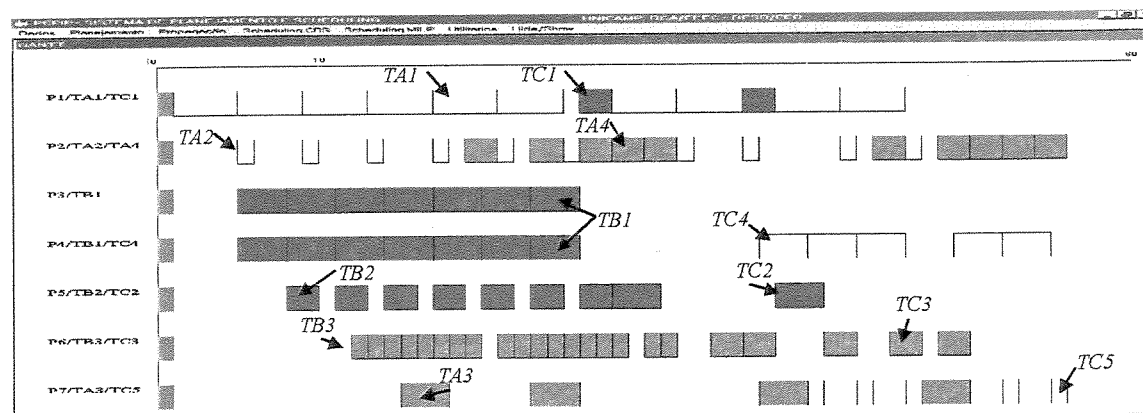


Figura 3.13b. Solução do segundo passo do MILP reduzido.

A utilização da formulação reduzida implica na determinação do parâmetro $gap_{bkb'k'}$, que define o intervalo mínimo entre o início do processamento da batelada b do subtrem k e o início do processamento da batelada b' do subtrem k' . Como o subtrem produz um produto final, não há nenhum subtrem consumindo o produto de $TB3$, e por isso, não há a definição de intervalos mínimos para este subtrem. A definição de intervalos mínimos se restringe à consideração do subtrem $TC3$ que foi eliminado. A figura 3.14 apresenta as relações de precedência entre as operações $TC1$, $TC2$, $TC3$ e $TC4$. Como o estado CI tem restrição de armazenagem ZW , as operações $TC1$ e $TC2$ pertencem ao mesmo subtrem $STC1$. Assim, os intervalos mínimos são definidos entre o subtrem $STC1$ e o subtrem $STC4$, que contém a operação $TC4$. A tabela 3.7 apresenta os intervalos mínimos definidos a partir das relações de precedência da figura 3.14. Nesta tabela são apresentados em cinza os pares de bateladas que serão utilizados na geração de equações da expressão 3.15. Para este exemplo bastaria considerar o intervalo mínimo entre $STC1/1$ e $STC4/1$, também entre $STC1/2$ e $STC4/3$ porque o ordenamento entre bateladas do subtrem $STC4$ (expressão 3.17) impõe um espaçamento igual a 3 unidades de tempo.

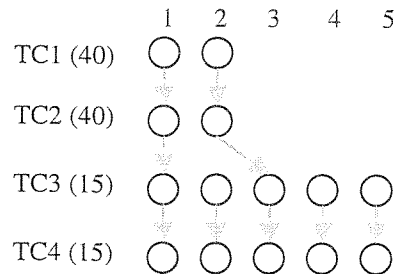


Figura 3.14. Relações de precedência entre as bateladas do exemplo 2.

Tabela 3.7. Intervalo mínimo entre o início das bateladas.

	STC4/1	STC4/2	STC4/3	STC4/4	STC4/5
STC1/1	7	10	13	16	19
STC1/2			7	10	12

A tabela 3.8 apresenta o resultado da execução do problema MILP global e do MILP reduzido. Esta tabela mostra o quanto a redução do problema MILP pode contribuir para a redução na dimensão do problema, sendo que para este caso o MILP reduzido foi capaz de encontrar a solução ótima (36).

Tabela 3.8. Características dos problemas MILP global e reduzido

	equações	variáveis contínuas	variáveis binárias	não zeros	número de iterações	número de nós	solução inteira	tempo (s.)
global	1564	314	538	6110	216990	25567	36	679,3
reduz. passo 1	869	212	325	3442	521	11	36(*)	0.4

(*) valor da função de custo obtida no passo 2

Foram criadas várias instâncias deste exemplo para analisar o desempenho da formulação reduzida. Em todas elas o problema reduzido exclui as operações *TB3* e *TC3*. A tabela 3.9 descreve as modificações e o Anexo 2 contém os resultados do planejamento e scheduling.

Tabela 3.9. Instâncias do exemplo 2.

caso	particularidade do caso
2.1	a situação anterior sem modificações
2.2	caso 2.1 com disponibilidade de matérias primas em 0
2.3	caso 2.1 com estado <i>A1</i> com restrição <i>ZW</i>
2.4	caso 2.2 com estado <i>A1</i> com restrição <i>ZW</i>
2.5	caso 2.1 com prazos de entrega de <i>ProB</i> em 52 (em lugar de 32). Isto leva a janelas de <i>TB3</i> com maior sobreposição com as janelas de <i>TC3</i>
2.6	caso 2.5 com disponibilidade de <i>FeedB</i> em 10 (em lugar de 1). O carregamento em <i>P6</i> induzido pela sobreposição aumenta porque as janelas de <i>TB3</i> são menores
2.7	caso 2.5 com disponibilidade de <i>FeedB</i> em 15 (em lugar de 1). O carregamento em <i>P6</i> aumenta

Em todos os casos o MILP global foi interrompido, se necessário, com 200000 iterações ou 900 segundos de tempo de execução. A precisão absoluta (optca no GAMS/OSL) foi fixada em 0.9. Dado que se minimiza a antecipação (earliness), e esta é medida em *slots* (de duração unitária), a solução inteira obtida quando o GAMS/OSL interrompe a execução atingindo o valor 0.9 é a solução ótima. A tabela 3.10 contém as informações principais para cada caso e a figura 3.15 mostra o tempo de execução em função do número de variáveis binárias em cada caso.

Tabela 3.10. Resultados da formulação MILP global e reduzida para as sete instâncias testadas.

caso		equ.	var. cont	var. bin.	não zeros	iter.	nós	sol.int.	tempo	primeira sol.int	tempo
2.1	global	1564	314	538	6110	216990	25567	36	679	85	530
	passo1	869	212	325	3442	521	11	36(*)	0.4		
2.2	global	2712	448	993	21422	122805	2893	13	731	39	390
	passo1	1722	302	657	15693	9080	291	13(*)	33.9		
2.3	global	1496	295	486	5783	74172	4481	36	158	44	22
	passo1	790	193	273	2988	610	13	36(*)	0.5		
2.4	global	2690	448	943	21248	94967	2006	13	544	38	93
	passo1	1700	302	607	15519	8136	277	13(*)	29.3		
2.5	global	4935	374	1038	48531	68333	461	351	900	351	119
	passo1	1314	232	465	8354	3097	64	106(*)	5.9		
2.6	global	3279	347	813	21186	153081	1377	111	900	142	139
	passo1	1080	223	402	5237	1325	47	104(*)	1.9		
2.7	global	2388	332	688	12032	200000	3021	111	672	145	74
	passo1	959	218	367	4203	1116	41	106(*)	1.4		

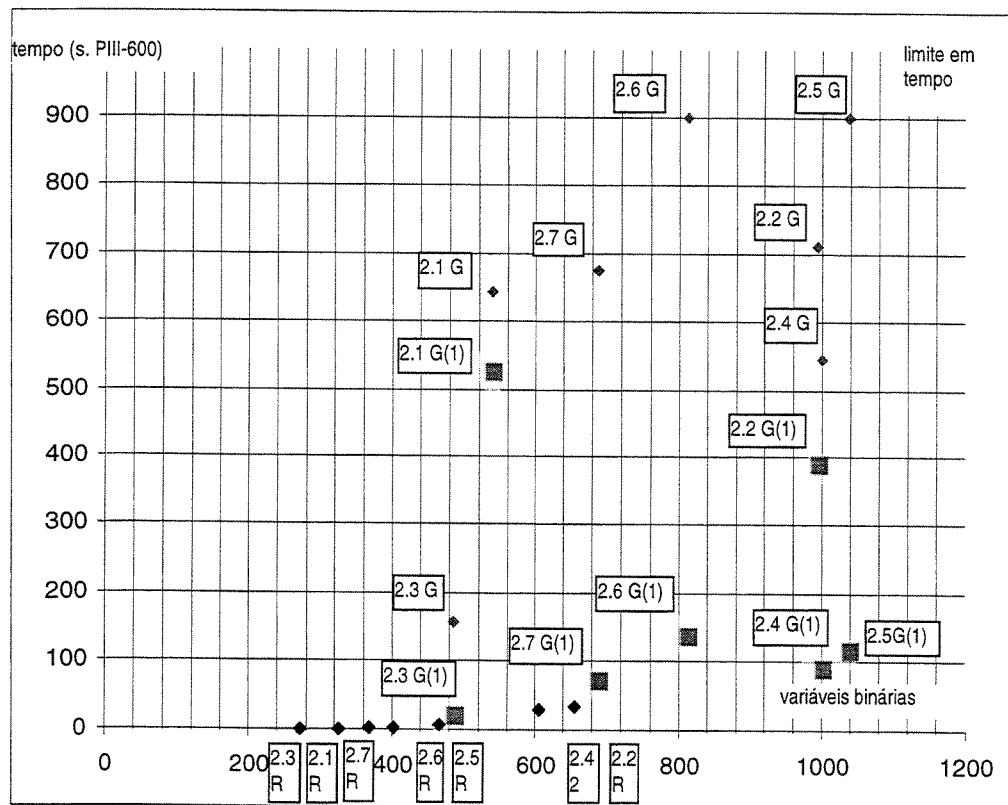


Figura 3.15. Dispersão dos tempos de execução com o número de variáveis.

onde:

- 2.X G Solução da formulação MILP global do caso 2.X
- 2.X G(1) Primeira solução inteira da formulação MILP global do caso 2.X
- 2.X R Solução da formulação MILP reduzida do caso 2.X

Comentários:

- Os resultados mostram uma redução sensível no tempo de execução e, mais importante, uma menor dispersão nos tempos de execução para o MILP reduzido, visível na figura 3.15. A degenerescência do MILP global leva a tempos no OSL muito variáveis, provavelmente devido a forma como ocorre a busca em árvore em cada caso.
- As soluções ótimas (104) para os casos 2.6 e 2.7 foram obtidas após 6538 segundos (1 hora e 49 minutos) e 5627 segundos (1 hora e 34 minutos), respectivamente.
- Os casos 2.1, 2.2, 2.5, 2.6, e 2.7 foram escolhidos para que hajam vários cenários de carregamento no processador **P6** através do deslocamento das janelas de tempo da operação **TB3** e da sua duração. As janelas estão nas figuras no Anexo 2 e as curvas de carregamento estão agrupadas na figura 3.16. Nos casos 2.1 e 2.2 a formulação MILP reduzida leva à mesma solução que o MILP global (que é a solução ótima). Nos casos 2.5, 2.6 e 2.7 a solução ótima é 104 e o MILP reduzido só chega a esta solução no caso 2.6. Isto é devido a que algumas das alocações no primeiro passo em 2.5 e 2.7 são diferentes das alocações em 2.6, mesmo levando a valores idênticos na função de custo do primeiro passo. Apenas as alocações em 2.6 levam à solução ótima (104), nos outros

dois casos obtém-se um valor de 106. Este valor é próximo à solução ótima mas obviamente não pode ser garantido que isto acontecerá sempre. O problema se deve a que o primeiro passo tem diversas soluções (no mínimo duas) levando à mesma função de custo neste passo, mas a situação de carregamento no processador **P6** não é suficientemente baixa para permitir achar a solução ótima independentemente da alocação no primeiro passo. Isto leva a uma análise mais detalhada da informação contida na curva de carregamento. Um trabalho atualmente em desenvolvimento está utilizando a curva de carregamento para dar ao usuário uma informação de carregamento semelhante aos gráficos de carga utilizados em sistemas MRP, que representam a carga induzida em um *time bucket* de um processador pelas bateladas alocadas nele. Um *time bucket* pode ser visto como um intervalo de tempo agregado e a carga total neste intervalo pode ser calculada como a integral da curva de carregamento neste intervalo. Em relação a um sistema MRP, o interesse desta abordagem é, obviamente, o fato de não requerer a alocação das bateladas, dado que a curva de carregamento é obtida a partir das janelas de tempo. A figura 3.16 mostra estas informações para os casos 2.1 e 2.5 quando o intervalo de integração é de 5 unidades de tempo. Observa-se que no caso 2.1 nenhum intervalo no processador **P6** tem excesso de carga enquanto que, no caso 2.5, nos intervalos (30-35) e (35-40) o excesso é respectivamente de 0.8 e 1.2. Em outras palavras, no caso 2.1 a concorrência pelo processador **P6** está menos concentrada do que no caso 2.5 e a abordagem em dois passos tem menos possibilidades de perder a solução ótima.

3.6. Conclusões

A utilização de janelas de tempo das bateladas, na formulação MILP do problema de *scheduling*, permite reduzir sensivelmente a dimensão do problema. Esta redução é maior para situações com carregamento alto porque este carregamento implica em janelas de tempo menores. A redução na dimensão do problema é conseguida basicamente através da redução do número de variáveis binárias.

A formulação MILP reduzida foi desenvolvida com o intuito de eliminar do problema de *scheduling* as bateladas que não envolvem problemas de alocação difíceis, por exemplo, por estarem atribuídas a processadores dedicados ou com carregamento baixo. Estas situações levam geralmente a degenerescência na solução do modelo MILP com os softwares comerciais hoje existentes. Ao que tudo indica, isto ocorre porque o procedimento de busca em árvore (*branch and bound*) não é capaz de focalizar o esforço sobre as variáveis binárias relevantes. Os exemplos apresentados mostram a importância da formulação reduzida. A definição do problema reduzido é feita a partir das informações de carregamento resultantes da fase de planejamento, mas deve levar em consideração também as restrições de armazenagem. Isto porque restrições de armazenagem fortes estabelecem interdependências entre as operações envolvidas, não sendo correta a sua alocação de forma separada, ou seja, no segundo passo. É interessante observar que as duas considerações utilizadas para reduzir o problema MILP têm uma correspondência com a abordagem na

prática, onde o usuário desconsidera, em uma primeira instância, o *scheduling* de equipamentos pouco carregados e de operações isoladas do resto da receita por restrições de armazenagem altamente flexíveis.

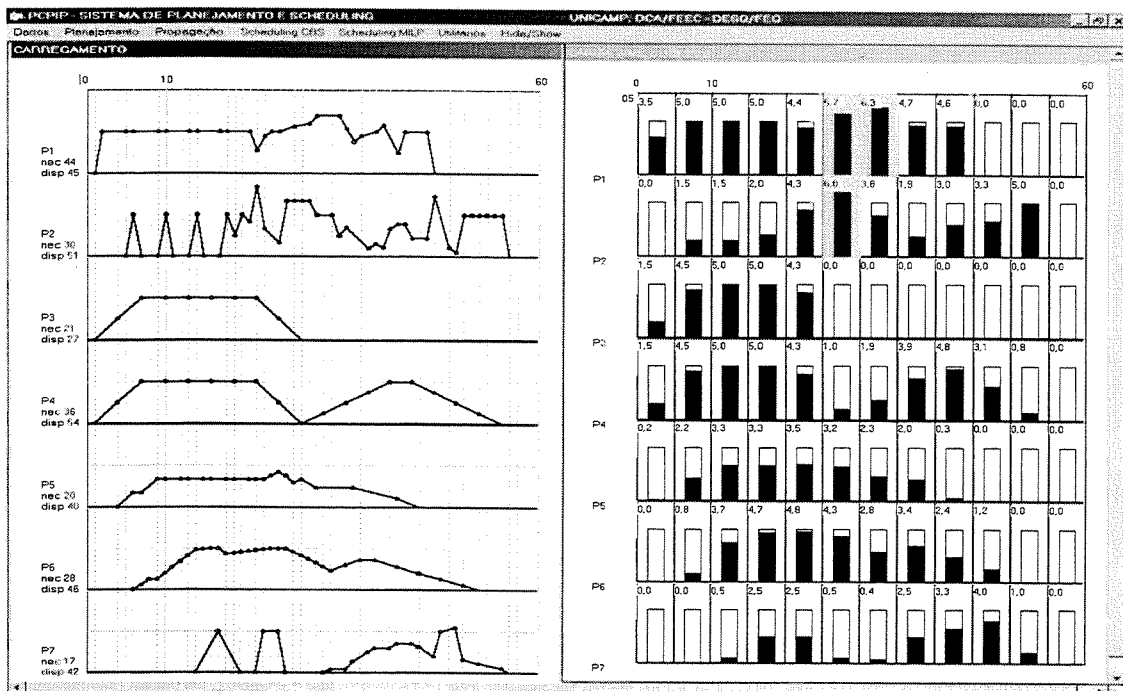


Figura 3.16a. Carregamento dos processadores para o caso 2.1 (*time bucket* igual a 5).

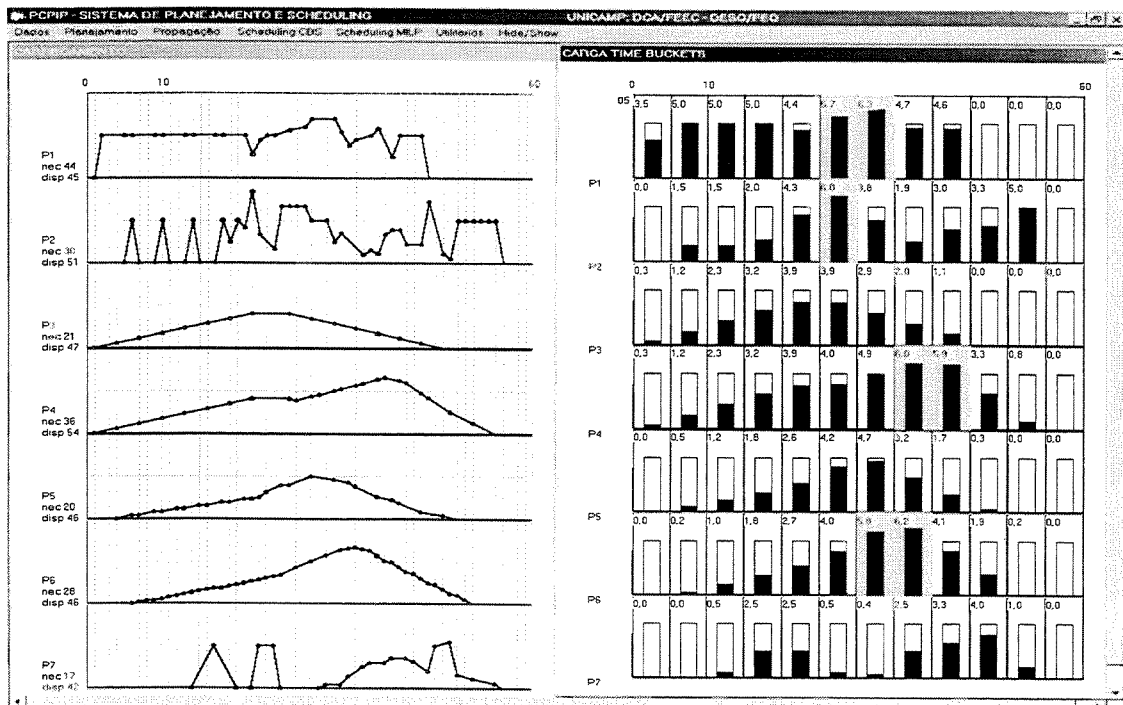


Figura 3.16b. Carregamento dos processadores para o caso 2.5 (*time bucket* igual a 5).

CAPÍTULO 4. PROGRAMAÇÃO DA PRODUÇÃO USANDO SIMULATED ANNEALING

4.1. Introdução

Neste capítulo apresenta-se o algoritmo de *simulated annealing* (SA) para resolver o problema de programação da produção (*scheduling*) usando a informação das janelas de tempo, obtidas na fase de planejamento. O SA é um algoritmo metaheurístico de busca da solução ótima e que vem sendo utilizado em um extenso número de problemas de otimização combinatória. Entre outros trabalhos, van Laarhoven e Aarts (1987) apresentam vários exemplos de aplicações do SA. Vários autores propuseram aplicações do SA a problemas de *scheduling*, entre eles: Das *et al.* (1990), Ku e Karimi (1991) e Graells *et al.* (1996).

Metropolis *et al.* (1953) propôs um algoritmo para a simulação da evolução de um sólido até o equilíbrio térmico, simulando o fenômeno físico da mudança de fase dos materiais em um processo de recozimento. Kirkpatrick *et al.* (1982) e Cerny (1985) constataram que há uma analogia entre a minimização da função de custo de um problema de otimização combinatorial e o resfriamento lento de um sólido até que ele atinja o seu estado de energia mais baixo. Substituindo-se os valores da energia pelos valores da função de custo e executando-se o algoritmo de Metropolis em uma seqüência de valores de temperatura, lentamente reduzidos, Kirkpatrick *et al.* (1982) e Cerny (1985) obtiveram um algoritmo de otimização combinatorial que foi chamado de *simulated annealing* (SA). Apesar de não se tratar mais do fenômeno físico da mudança de fase dos materiais em um processo de recozimento, o termo temperatura é mantido no SA. O algoritmo do SA não dispõe de meios para identificar quando se está no ótimo global e nem de quão longe se está do ótimo global. Portanto, cabe ao usuário definir o seu compromisso entre buscar soluções cada vez mais próximas do ótimo global e consumir mais tempo de busca.

A implementação de técnicas de SA aos problemas de *scheduling* é feita, em geral, em dois passos. No primeiro passo são definidas decisões de ordenamento, e no segundo passo são definidos os instantes de alocação, com o objetivo de calcular a função de custo (normalmente o *makespan*). A complexidade do cálculo do valor da função de custo para problemas de *scheduling* está associada ao cálculo dos instantes de início do processamento de cada batelada. Em problemas em que estão presentes relações de precedência entre bateladas (decorrentes do balanço de massa e de ordenamentos impostos pelas janelas de tempo) a representação adequada destas restrições pode reduzir o esforço e o espaço de soluções do problema. Neste trabalho é proposta a representação do problema através do ordenamento entre as bateladas de operações que competem pelo mesmo processador.

É bem conhecido o fato de que algoritmos heurísticos podem gerar um grande número de candidatos ineficazes em situações com restrições, porque os mecanismos tradicionais de geração de soluções não levam em consideração as restrições. Um candidato é representado por um ordenamento entre bateladas e pode não ser fácil ou possível analisar a satisfação das restrições (de recursos compartilhados, por exemplo) através de um

ordenamento. Neste trabalho, as restrições consideradas são restrições temporais dadas pelas janelas de tempo, de forma que a solução obtida deve estar dentro das janelas de tempo. Neste caso, é possível analisar parcialmente a factibilidade do candidato através das relações de ordenamento impostas pelas janelas de tempo. O que se propõe neste capítulo é desenvolver um algoritmo SA novo onde a informação das janelas de tempo é usada para filtrar o SA, restringindo a geração de candidatos inactíveis.

A seção 4.2 apresenta o algoritmo básico do SA. A seção 4.3 apresenta as alterações feitas no SA para que ele possa utilizar a informação das janelas de tempo, constituindo-se em um SA com filtragem dos candidatos. A seção 4.4 apresenta a comparação do desempenho das duas abordagens do SA para dois casos analisados.

4.2. Descrição do algoritmo básico do SA

O SA é iniciado com uma solução factível. Esta solução será a solução vigente inicial. Sempre que se encontra um candidato melhor que a solução vigente, este candidato é definido como a nova solução vigente. A repetição contínua deste procedimento pode levar a solução vigente a um ponto de ótimo local. Um ponto de ótimo local é um ponto cuja solução é pior que o ótimo global, mas onde todas as soluções vizinhas têm o valor da função objetivo pior que o valor da função objetivo no ótimo local. A fim de que a solução vigente não fique presa em um ponto de ótimo local, o SA se utiliza de um critério estocástico para a aceitação de soluções piores que a vigente. A seguir são apresentados os passos dados na execução do algoritmo SA.

1. Definição dos parâmetros iniciais do SA. Os parâmetros são a temperatura inicial (T_0), o fator de redução da temperatura e a solução inicial do problema. O fator de redução da temperatura determina a forma como se dará a redução da temperatura,. O contador de iterações é zerado neste passo ($k = 0$).
2. Atualização do contador de iterações ($k = k + 1$).
3. Geração de um candidato (S') a partir da solução vigente (S). Isto é feito sorteando-se duas bateladas distintas e depois permutando-se a posição da primeira batelada sorteada com a segunda batelada sorteada.
4. Cálculo do valor de Δ . O valor de Δ é dado pela diferença entre os valores da função objetivo da solução vigente (S) e do candidato (S').
5. Se o valor da função objetivo do candidato (S') é melhor ou igual ao valor da solução vigente (S), segue-se para o passo 6. Caso contrário, segue-se para o passo 7.
6. Quando o candidato é melhor ou igual a solução vigente, define-se o candidato como a nova solução vigente ($S = S'$). Também verifica-se neste passo se a nova solução vigente é melhor que a melhor solução encontrada (S^*). Este teste é necessário para manter na memória a melhor solução encontrada, dado que o SA pode ter soluções vigentes com valores piores que S^* . Deste passo segue-se para o passo 8.
7. Quando o candidato é pior que a solução vigente, escolhe-se um número aleatório r entre 0 e 1. Se r for menor ou igual ao exponencial de $(-\Delta/T_k)$, onde T_k é a temperatura do SA na atual iteração (iteração k), então o candidato (S') é definido como a nova

solução vigente (S), apesar de piorar o valor da função objetivo da solução vigente. Este procedimento é a forma que o SA tem para escapar dos pontos de ótimo local. Quanto maior o valor da temperatura utilizada, maior a probabilidade de aceitação de uma solução ruim.

8. Determinar o valor da temperatura para a próxima iteração (T_{k+1}). Há diversas propostas para o cálculo desta temperatura, sendo que no anexo 3 são apresentadas algumas das propostas existentes na literatura para o cálculo da temperatura. A proposta adotada foi de só alterar o valor da temperatura a cada 100 iterações. O procedimento de redução da temperatura adotado é o proposto por Kirkpatrick *et al.* (1982), onde: $T_{k+1} = \alpha T_k$, sendo que a temperatura só será alterada se k for múltiplo de 100. O valor adotado para o fator de redução da temperatura (α) foi 0,9.
9. No último passo de cada iteração do SA verifica-se se o critério de parada foi atingido. Caso contrário, segue-se para o passo 2. O critério de parada pode ser uma temperatura mínima ou um número limite de iterações ou o número limite de iterações depois da última alteração no valor da função objetivo ou algum outro critério.

4.3. SA com filtragem dos candidatos (Rodrigues *et al.*, 2000c)

A fase de planejamento determina o número de bateladas que serão produzidas de cada uma das operações, assim como a janela de tempo em que cada uma destas bateladas deverá ser produzida. Os ordenamentos impostos pelas janelas de tempo, assim como os ordenamentos exigidos pelas relações de precedência (i.e., balanço de massa e armazenagem), podem ser bastante complexos. O que se propõe é restringir o sorteio do SA de forma que o ordenamento das bateladas do candidato seja factível. Ou seja, que os ordenamentos representando o candidato não estejam em contradição com algum dos ordenamentos obrigatórios resultantes da fase de planejamento. Para esta análise mostrou-se conveniente a representação do problema na forma de um grafo porque permite a detecção de infactibilidades no candidato através de um algoritmo de detecção de ciclos (Cantón, 2000). Os nós no grafo representam as bateladas que serão produzidas.

A fase de planejamento define restrições de ordenamento que qualquer candidato gerado pelo SA deve obedecer. Através do procedimento de análise do compartilhamento dos processadores, identificam-se na fase de planejamento quais as relações de ordenamento existentes entre as bateladas de um mesmo processador. São geradas as relações de precedência entre bateladas de operações que geram e consomem o mesmo estado. Estas relações de precedência garantem que os ordenamentos decorrentes das restrições de armazenagem ZW e NIS serão respeitados. No entanto, as restrições de armazenagem FIS impõem ordenamentos entre bateladas que deverão ser respeitados para que os limites de armazenagem não sejam excedidos. Por fim, são definidos, na fase de planejamento, ordenamentos entre os nós representando as bateladas da mesma operação. Todas estas relações de ordenamento originam arcos direcionados no grafo que resumindo têm as seguintes origens:

- Relações de ordenamento no mesmo processador impostas pelas janelas de tempo,

- Relações de ordenamento entre bateladas da mesma operação,
- Relações de ordenamento devido às relações de precedência decorrentes do balaço de massa e
- Relações de ordenamento originadas pelas restrições de armazenagem.

Um arco disjuntivo é definido quando não há um ordenamento definido entre dois nós. O grafo resultante pode ser visto como um grafo em que inicialmente todos os nós no mesmo processador são ligados por arcos disjuntivos. As restrições de ordenamento do planeamento transformam alguns destes arcos disjuntivos em arcos orientados. Os arcos disjuntivos remanescentes definem as opções de sorteio para o SA.

A solução vigente no SA é representada pelo ordenamento de todos os nós alocados em cada processador e a geração de um novo candidato é feita através da troca no ordenamento de dois nós da solução vigente em um processador (permutação). O objetivo do processo de filtragem na geração do candidato é o de impedir que o ordenamento do candidato seja inactível. Isto acontecerá se o ordenamento novo (em relação à solução vigente) não é compatível com os ordenamentos existentes, que são decorrentes do planeamento e da solução vigente. Os ordenamentos da solução vigente também influem na factibilidade do candidato porque o grafo representando esta solução vigente contém outros arcos orientados, além dos resultantes do planeamento. Não é óbvio nesta situação identificar quais são os arcos disjuntivos remanescentes habilitados para a permutação. Isto é feito a partir do grafo usando-se um teste de detecção de ciclos.

4.3.1. Detecção de ordenamentos adicionais na fase de planeamento

As relações de ordenamento identificadas na fase de planeamento geram arcos direcionados no grafo. É possível que conjuntamente elas impliquem em ordenamentos adicionais. A figura 4.1 apresenta o algoritmo que permite a determinação destes ordenamentos adicionais e a identificação dos arcos disjuntivos remanescentes na fase de planeamento.

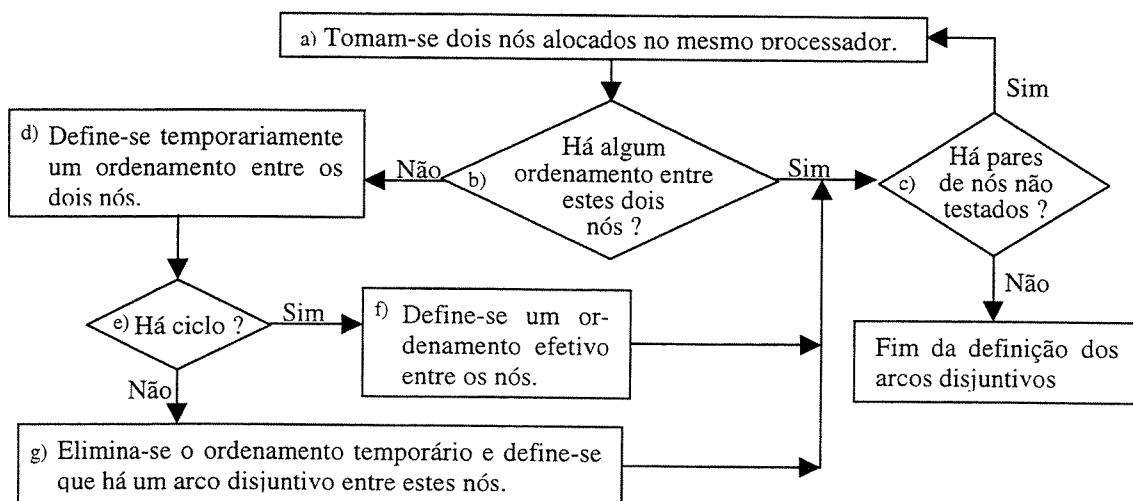


Fig. 4.1. Identificação de ordenamentos adicionais e dos arcos disjuntivos remanescentes.

4.3.2. Geração de uma nova solução candidata

O procedimento de geração dos candidatos utilizando-se de um procedimento de filtragem dos nós habilitados para serem sorteados é descrito a seguir.

1. A lista dos nós habilitados para permutar só tem nós da solução vigente que possuem arcos disjuntivos. Seleciona-se aleatoriamente um dos nós desta lista para ser o primeiro nó sorteado.
2. Após o sorteio do primeiro nó, eliminam-se os ordenamentos impostos pela solução vigente ligados ao primeiro nó sorteado. Somente os ordenamentos definidos na fase de planejamento e os ordenamentos da solução vigente entre os demais nós são mantidos.
3. Identificam-se os arcos disjuntivos remanescentes ligados ao primeiro nó sorteado através do teste de detecção de ciclos, que é idêntico ao algoritmo da figura 4.1. O propósito deste passo é identificar o conjunto de nós habilitados para serem o segundo nó sorteado. Este conjunto será em geral menor que o conjunto dos arcos disjuntivos iniciais (do planejamento) ligados a este nó, porque a solução vigente no SA contém ordenamentos entre nós que não estavam definidos pelos arcos orientados resultantes da fase de planejamento.

Quando um nó sorteado não possui nenhum arco disjuntivo remanescente, ele não pode ser usado para gerar a solução candidata, sendo eliminado da lista dos nós habilitados para o primeiro sorteio. Neste caso, o procedimento retorna para o passo 1 (sorteio do primeiro nó), onde o nó sorteado sairá da lista atualizada dos nós habilitados para o primeiro sorteio. Se existem arcos disjuntivos remanescentes prossegue-se para o passo 4.

4. Sorteia-se o segundo nó para a permutação. Ou seja, seleciona-se um dos arcos disjuntivos remanescentes (identificados no passo 3) para gerar uma nova solução candidata através da permutação dos dois nós sorteados.
5. Conclui-se a geração do novo candidato com a permutação dos dois nós sorteados e a execução do procedimento de temporização, que define o instante de início de cada batelada. O procedimento de temporização pode ser realizado usando programação linear (PL), como mostrado na seção 4.3.3, ou através de uma heurística (se houver uma para a função objetivo utilizada).

Se o problema tratado é bastante restrungido, o procedimento de filtragem dos candidatos reduzirá bastante a dimensão do problema devido à redução no número de trocas (nos ordenamentos dos nós) permitidas. Obviamente se o problema é pouco restrungido, serão identificados poucos ordenamentos e, com poucos arcos orientados, o procedimento de filtragem será incapaz de reduzir significativamente a dimensão do problema.

4.3.3. Alocação e cálculo da função de custo

Como a representação de um candidato do SA é feita através de uma lista ordenada das bateladas por processador, é preciso que após a permutação se defina a alocação de

cada uma das bateladas do candidato, o que permitirá avaliar a função de custo. Uma possibilidade para a definição da alocação (temporização) das bateladas no SA é através da programação matemática, como um problema de otimização MILP em que as variáveis binárias sejam de ordenamento, como por exemplo a formulação apresentada por Rich e Prokopakis (1986). Dado que os ordenamentos no SA estão pré-definidos, a formulação de programação linear inteira mista é transformada em um problema de programação linear (PL), porque as variáveis binárias (de ordenamento) estão todas fixadas.

Como deseja-se que as bateladas sejam alocadas dentro das suas janelas de tempo e que as demandas dos produtos finais sejam atendidas o mais próximo possível dos seus prazos de conclusão (*due dates*), a função objetivo utilizada nesta seção é a minimização de *earliness*. As demais expressões do PL expressam as restrições de ordenamento e de armazenagem existentes no problema.

A expressão 4.1 define que o instante final de processamento de uma batelada é igual à soma do instante de início de processamento desta batelada mais o seu tempo de processamento. As expressões 4.2 e 4.3 forçam a alocação das bateladas dentro das janelas de tempo.

$$TE_{i,b,j} = TS_{i,b,j} + tp_i \quad \forall i, b, j \quad [4.1]$$

$$TS_{i,b,j} \geq ebt_{i,b} \quad \forall i, b, j \quad [4.2]$$

$$TE_{i,b,j} \leq lft_{i,b} \quad \forall i, b, j \quad [4.3]$$

onde:

variáveis

$TS_{i,b,j}$ Instante de início de processamento da batelada b da operação i no processador j .

$TE_{i,b,j}$ Instante final de processamento da batelada b da operação i no processador j .

dados

tp_i Tempo de processamento da operação i .

$ebt_{i,b}$ Instante de início da janela de tempo da batelada b da operação i .

$lft_{i,b}$ Instante final da janela de tempo da batelada b da operação i .

A expressão 4.4 define o valor da antecipação no final do processamento de cada batelada em relação ao final da sua janela de tempo. A expressão 4.5 impõe restrições sobre os processamentos das bateladas devido aos ordenamentos da solução candidata; se a batelada b da operação i precede a batelada b' da operação i' , então o início do processamento da batelada b' só pode ocorrer a partir do final do processamento da batelada b .

$$TE_{i,b,j} = lft_{i,b} - earl_{i,b} \quad \forall i, b, j \quad [4.4]$$

$$TS_{i',b',j} \geq TE_{i,b,j} \quad \forall j, i, b, i', b' / prec_{i,b,i',b'} = \text{true} \quad [4.5]$$

onde:

$earl_{i,b}$ Valor da antecipação no final do processamento da batelada b da operação i em relação ao final de sua janela de tempo.

$prec_{i,b,i',b'}$ = true indica que a batelada b da operação i precede a batelada b' da operação i' .

A formulação matemática do problema de *scheduling* não precisa introduzir as expressões de balanço de massa porque as restrições de precedência (ordenamentos) determinados na fase de planejamento já o garantem. As expressões 4.6 a 4.8 impõem as restrições de armazenamento nas situações ZW, NIS e FIS. A expressão 4.6 estabelece que, se as operações i e i' estão ligadas por um estado com restrição de armazenagem ZW, então o início do processamento da batelada b da operação i' coincide com o final do processamento da batelada b da operação i .

Um estado com restrição de armazenagem NIS somente permite a armazenagem dentro do processador onde a operação (que gera o estado) é realizada. Enquanto um processador está sendo usado para armazenagem, nenhuma outra operação poderá ocupar este processador. Para estados com restrição de armazenagem NIS, a expressão 4.7 é necessária para tratar situações como a representada na figura 4.2. Se no candidato a alocar a batelada (i, b) precede a batelada (i'', b') no processador $P1$, a alocação da batelada (i', b) no processador $P2$ deve ser anterior à alocação de (i'', b') em $P1$. Caso contrário, o processador $P1$ não poderia armazenar o estado NIS até o início da batelada (i', b) no processador $P2$.

$$TS_{i',b,j'} = TE_{i,b,j} \quad \forall j, j', i, b, i' / ZW_{i,i'} = \text{true} \quad [4.6]$$

$$TS_{i',b',j'} \leq TS_{i'',b'',j} \quad \forall j, j', i, i', b', i'', b'' / NIS_{i,i'} = \text{true} \text{ e } prec_{i,b',i'',b''} = \text{true} \quad [4.7]$$

onde:

$ZW_{i,i'}$ = true indica que as operações i e i' estão ligadas por um estado com restrição de armazenagem ZW.
 $NIS_{i,i'}$ = true indica que as operações i e i' estão ligadas por um estado com restrição de armazenagem NIS.

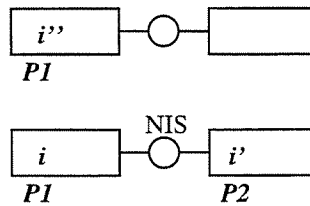


Fig. 4.2. Restrição à armazenagem NIS imposta pela operação i'' .

Um estado com restrição de armazenagem FIS possui capacidade limitada de armazenagem. A expressão 4.8 estabelece que, se a batelada consumidora (i, b) deve preceder a batelada produtora (i', b') para que não seja excedida a capacidade de armazenagem, então o início do processamento da batelada b da operação i deve ocorrer até o final do processamento da batelada b' da operação i' . Este ordenamento entre bateladas consumidoras e produtoras é determinado facilmente a partir dos tamanhos de bateladas e a limitação de armazenagem. Finalmente, a expressão 4.9 apresenta a função objetivo do PL.

$$TE_{i',b',j'} \geq TS_{i,b,j} \quad \forall j, j', i, b, i', b' / FIS_{i,b,i',b'} = \text{true} \quad [4.8]$$

$$\text{Função objetivo:} \quad \min \sum_{i \in \text{prod}, b} earl_{i,b} \quad [4.9]$$

onde:

$FIS_{i,b,i',b'}$ = true indica que a batelada b' da operação i' deve ser executada antes do início do processamento da batelada b da operação i para que a restrição de armazenagem FIS entre as duas operações seja satisfeita.

A utilização de uma técnica de otimização como a programação linear para a alocação e cálculo da função de custo tem um custo alto em termos de tempo computacional. Especialmente porque isto será feito para cada candidato e o SA pode gerar um grande número de candidatos. Provavelmente esta é a razão que leva a maioria das aplicações de *scheduling* usando SA na literatura a terem como objetivo a minimização do *makespan*, principalmente para problemas sem restrições de armazenagem. Nesta situação o problema de alocação pode ser resolvido de forma ótima através de uma heurística de alocação das bateladas no instante mais cedo possível. Graells (1996) estendeu esta abordagem para a situação em que existem restrições de armazenagem acrescentando para tanto um balanço de massa no processo de simulação que permite detectar candidatos infactíveis, por não respeitarem as restrições de armazenagem.

Cabe então a questão se há alguma heurística que garanta a solução ótima para o problema da alocação das bateladas minimizando *earliness*, quando os ordenamentos estão fixados. Não é do conhecimento do autor a existência de uma tal heurística. Foi testada uma heurística simétrica à de alocação no instante mais cedo, alocando as bateladas no instante mais tarde possível. Em um número reduzido de exemplos, os resultados foram idênticos aos obtidos com a programação linear, mas obviamente isto não é conclusivo.

Uma outra postura frente a este problema consistiria em postular que a alocação deve ser sempre no instante mais cedo porque o usuário não vai estar interessado em uma solução que deixe a planta ociosa, devido aos intervalos de tempo não utilizados, como pode aparecer na solução ótima minimizando *earliness*. Uma abordagem deste tipo subverte todo o problema, e em particular o critério de *earliness* pode não ser mais adequado para esta situação. Claramente a alocação no instante mais cedo ocupando ao máximo a planta será adequado para uma situação infactível onde é certeza que existirão atrasos em relação aos prazos de conclusão das bateladas (*due dates*) e onde, portanto, o usuário não deseja tempos ociosos.

4.3.4. Comentários adicionais

A análise de factibilidade através da detecção de ciclos em grafos não é completa e, por isso, não garante que a solução candidata gerada é factível. No procedimento de geração de um candidato realiza-se, após o sorteio do primeiro nó, a identificação dos arcos disjuntivos remanescentes. Isto porque alguns dos ordenamentos da solução vigente são mantidos e, portanto, os arcos disjuntivos remanescentes dependerão destes ordenamentos e dos ordenamentos definidos no planeamento. A existência de ordenamentos definidos pela

solução vigente justificaria a execução do procedimento de análise do compartilhamento dos processadores, com a conseqüente propagação dos cortes nas janelas de tempo gerados pelos ordenamentos da solução vigente. Este procedimento poderia identificar novos ordenamentos ainda não definidos e, com isso, a identificação dos arcos disjuntivos remanescentes seria mais completa. No entanto, o procedimento de propagação dos cortes nas janelas de tempo devido aos ordenamentos da solução vigente não é realizado devido ao seu tempo de execução que pode tornar o SA mais lento. Trata-se de um compromisso entre o tempo gasto na geração de soluções infactíveis e o tempo gasto eliminando-as.

De qualquer modo, satisfazer todas as relações de ordenamento não implica na garantia de alocação das bateladas dentro das janelas. Satisfazer todas as relações de ordenamento não elimina a possibilidade de soluções infactíveis na alocação porque, por exemplo, as restrições de armazenagem podem levar a dependências entre a alocação de bateladas produtoras e consumidoras que não permitem explorar a liberdade de alocação dada pelas respectivas janelas de tempo. Estas alocações forçadas pelas restrições de armazenagem podem levar a infactibilidades na alocação de outras bateladas.

4.4. Comparação do SA com e sem a filtragem dos candidatos

Para comparar os resultados do SA com e sem filtragem dos candidatos utilizou-se dois casos do problema proposto por Papageorgiou e Pantelides (1993), apresentado no anexo 1. Estes dois casos foram criados com o propósito de gerar uma situação de carregamento dos processadores mais restringida (caso 1) e uma situação menos restringida (caso 2), para o mesmo problema.

4.4.1. Caso 1

O caso 1 corresponde à demanda e prazos de conclusão dos produtos finais dados na tabela 4.1. A execução da fase de planejamento para o caso 1 gerou as janelas de tempo apresentadas na figura 4.3.

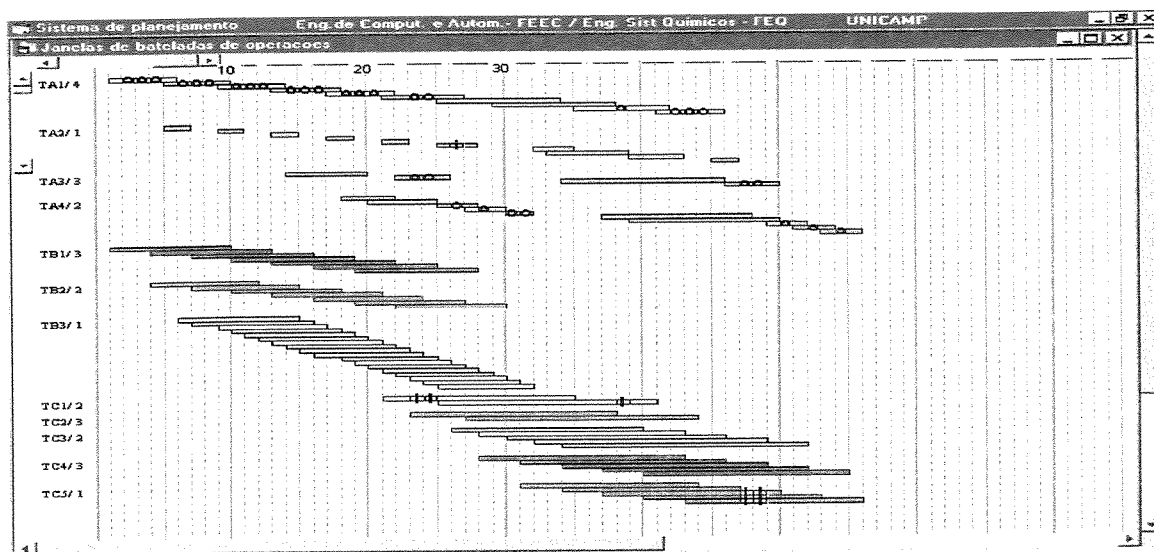


Figura 4.3. Janelas de tempo das bateladas.

Tabela 4.1. Demanda dos produtos finais para o caso 1.

Produto final	Demanda	Prazo (due date)
ProA	100	32
	90	56
ProB	270	32
ProC	75	56

A figura 4.4 apresenta em cinza todos os arcos disjuntivos remanescentes identificados no final da fase de planejamento. Os arcos dirigidos são apresentados em linhas pontilhadas. Os arcos dirigidos obtidos a partir das janelas de tempo são indicados em azul. Os arcos dirigidos entre bateladas de uma mesma operação são indicados em preto. Os arcos dirigidos obtidos a partir das relações de precedência decorrentes do balanço de massa e das relações de armazenamento são indicados em vermelho.

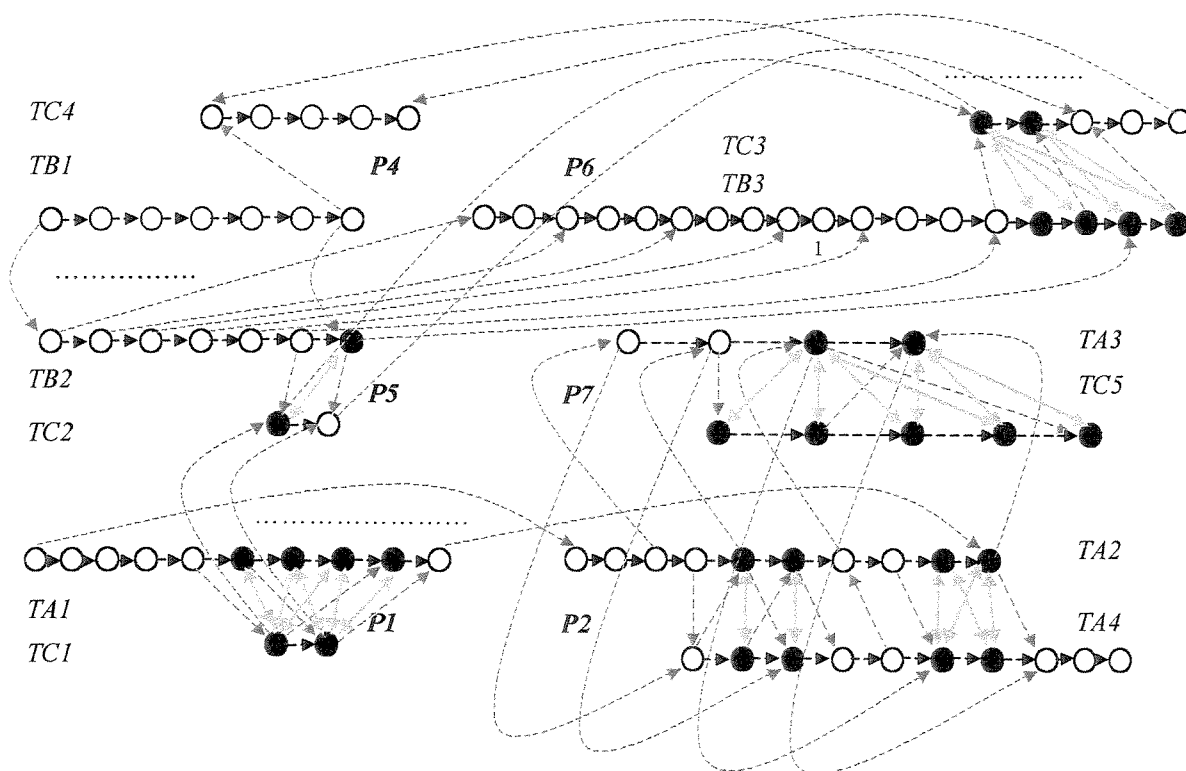


Figura 4.4. arcos dirigidos e disjuntivos identificados na fase de planejamento.

A seguir são dados alguns exemplos de arcos dirigidos obtidos a partir das janelas de tempo. A figura 4.3 identifica que o final da janela de tempo da batelada $TB1/7$ coincide com o início da janela de tempo da batelada $TC4/1$ e, por isso, $TB1/7$ necessariamente precederá $TC4/1$ no processador $P4$, como mostra o arco dirigido na figura 4.4. No processador $P1$, as janelas de tempo das bateladas $TA1/5$ e $TC1/1$ mostram que $TA1/5$ necessariamente precederá $TC1/1$. O mesmo ocorre com os arcos dirigidos entre as bateladas: $TA1/6$ e $TC1/2$, $TC1/1$ e $TA1/9$, e $TC1/2$ e $TA1/10$.

Nem todas as relações de precedência decorrentes do balanço de massa são apresentadas na figura abaixo. Quando duas operações consecutivas possuem o mesmo tamanho de batelada, será produzido um mesmo número de bateladas das duas operações e, portanto, as relações de precedência devido ao balanço de massa destas operações serão óbvias. Entre as operações TA1 e TA2 são apresentados na figura apenas os arcos dirigidos ligando a primeira batelada das duas operações e a última batelada das duas operações, sendo omitidos os demais arcos dirigidos. Entre as operações TB1 e TB2, e TC3 e TC4, foi feito o mesmo procedimento. Entre as operações TC4 e TC5 não é apresentado nenhum arco dirigido decorrente do balanço de massa para não prejudicar a clareza da figura. Os nós ligados por arcos disjuntivos estão habilitados para serem o primeiro nó sorteado no SA e são apresentados em preto na figura 4.4.

A tabela 4.2 apresenta a lista dos nós habilitados para serem o primeiro nó sorteado para o caso 1. O procedimento de filtragem no final da fase de planejamento permitiu reduzir o número de bateladas habilitadas para o primeiro sorteio de 85 para 29 bateladas.

Tabela 4.2. Lista dos nós habilitados para serem o primeiro nó sorteado.

processadores	nós habilitados para serem o primeiro nó sorteado
P1	TA1/6, TA1/7, TA1/8, TA1/9, TC1/1 e TC1/2
P2	TA2/5, TA2/6, TA2/9, TA2/10, TA4/2, TA4/3, TA4/6 e TA4/7
P3	-
P4	-
P5	TB2/7 e TC2/1
P6	TB3/15, TB3/16, TB3/17, TB3/18, TC3/1 e TC3/2,
P7	TA3/3, TA3/4, TC5/1, TC5/2, TC5/3, TC5/4 e TC5/5

A figura 4.5 apresenta os ordenamentos da solução inicial adotada. A figura 4.6 apresenta a carta de Gantt para a solução inicial (*earliness* = 53).

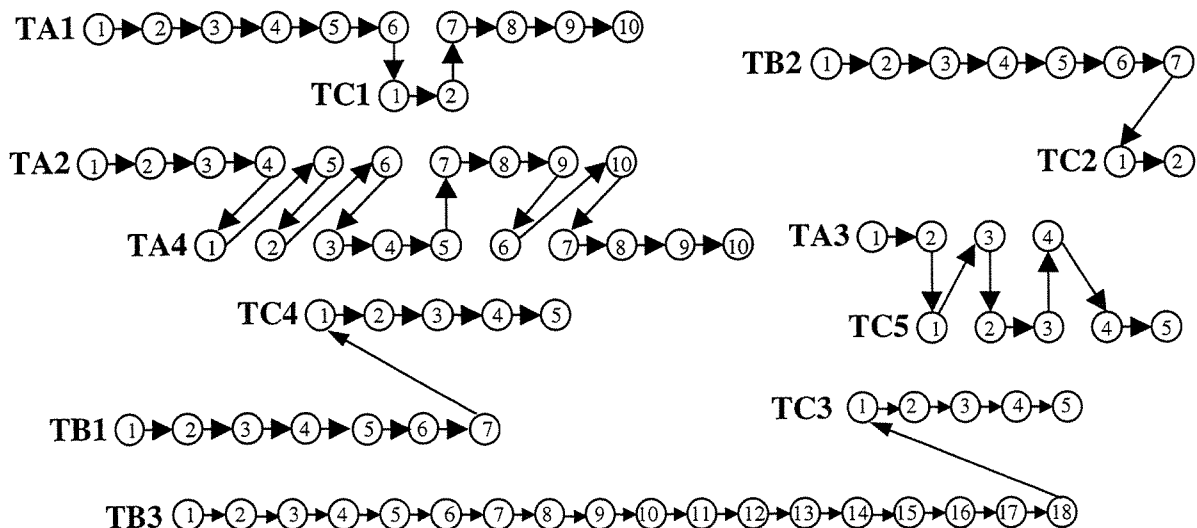
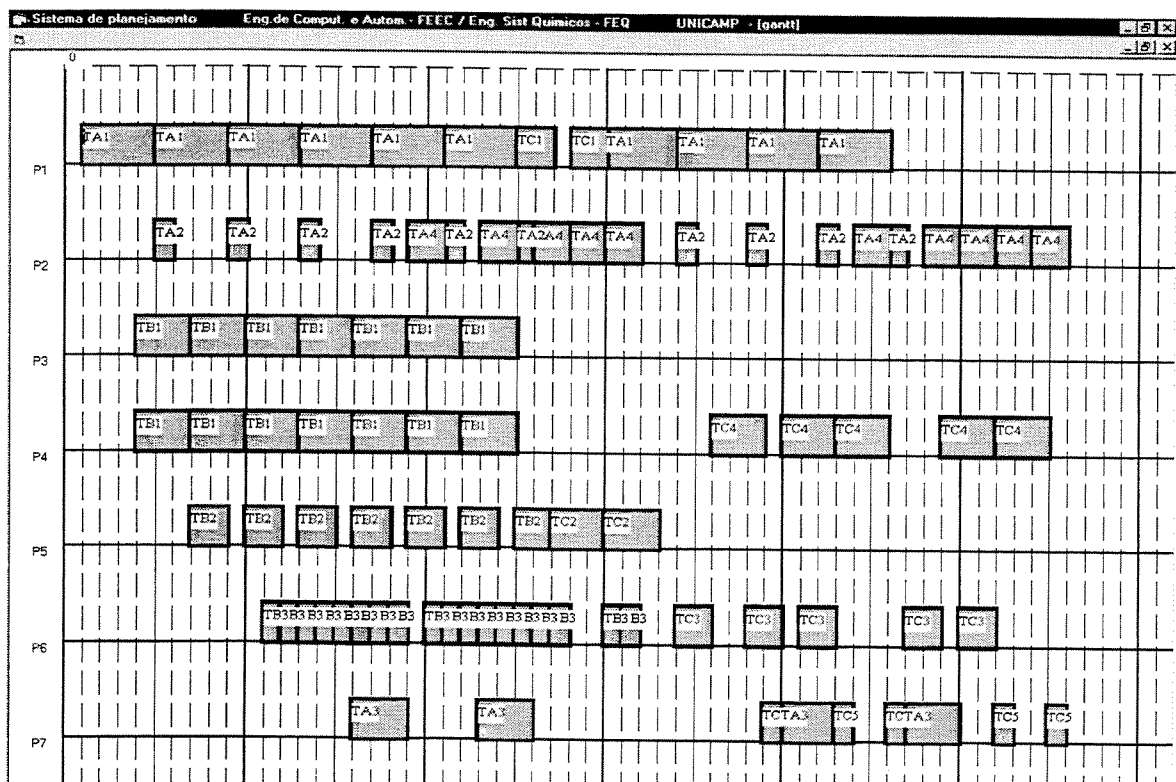
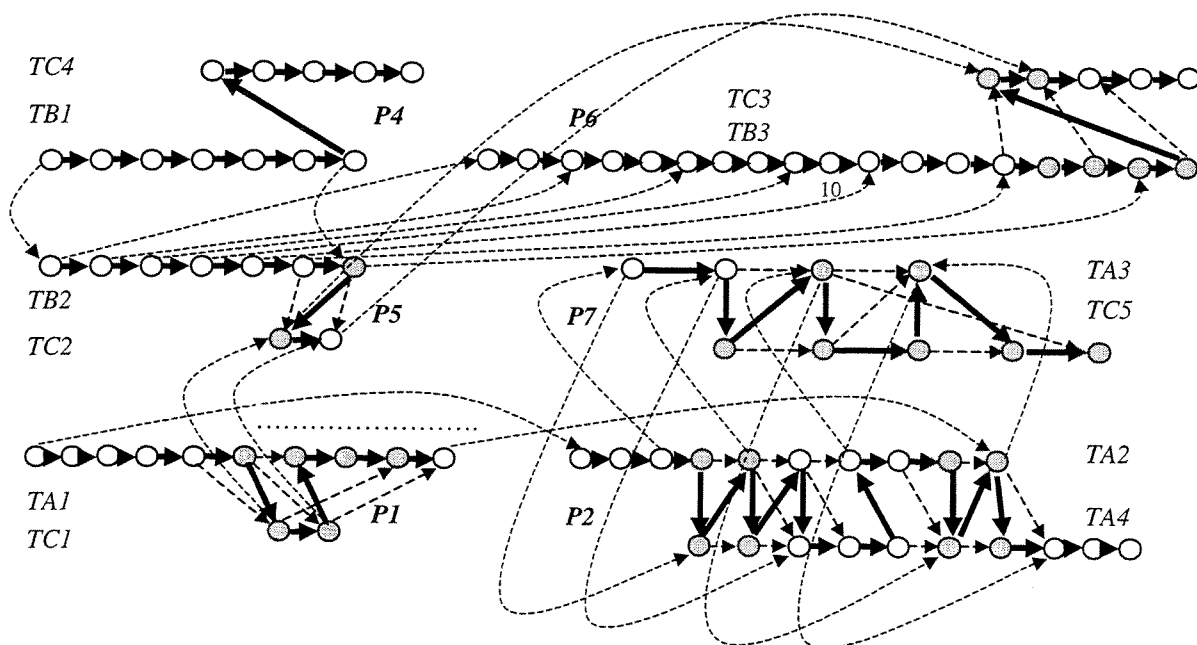


Figura 4.5. Ordenamentos da solução inicial no caso 2.



A figura 4.7 acrescenta os arcos do planejamento ao grafo da solução vigente, sendo que os nós habilitados para serem sorteados são destacados.



A fim de exemplificar o procedimento de geração de uma solução candidata, apresenta-se aqui a situação onde a solução inicial é a solução vigente. Neste passo são realizados os procedimentos descritos na seção 4.3.2. Considerando-se que o primeiro nó sorteado foi o nó TC1/1, a figura 4.8 apresenta o grafo da solução vigente no processador P1. Após o sorteio do primeiro nó, eliminam-se os arcos dirigidos definidos pela solução vigente envolvendo o primeiro nó sorteado - TC1/1, como mostra a figura 4.9.

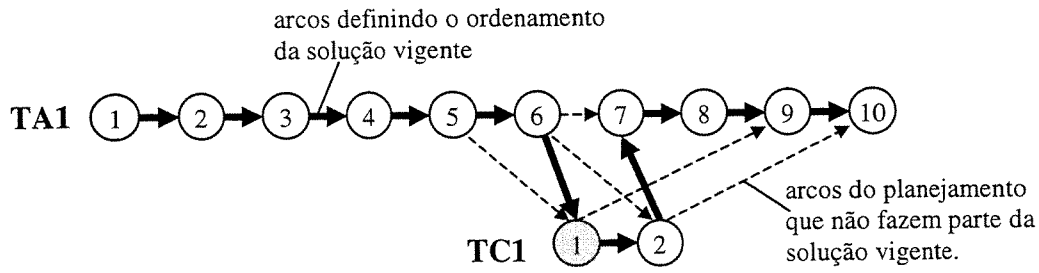


Figura 4.8. Solução inicial no processador 1.

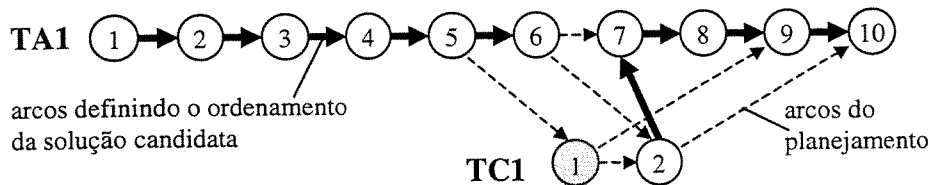


Figura 4.9. Eliminação dos ordenamentos da solução vigente no nó TC1/1.

A seguir, identificam-se os arcos disjuntivos remanescentes ligados a este nó através do teste de ciclos. O propósito deste passo é identificar os nós habilitados para serem o segundo nó sorteado. Na fase de planejamento foi identificado que os nós TA1/6, TA1/7 e TA1/8 tinham arcos disjuntivos com o nó TC1/1 (figura 4.4). Porém, apenas o arco entre os nós TA1/6 e TC1/1 permanece sendo disjuntivo, sendo apresentado em cinza na figura 4.10. Os outros dois arcos (TA1/7 - TC1/1 e TA1/8 - TC1/1), disjuntivos na fase de planejamento transformaram-se em arcos dirigidos quando se consideram os arcos dirigidos introduzidos pela solução vigente. Estes dois arcos têm o seu ordenamento forçado levando a $TC1/1 < TA1/7$ e $TC1/1 < TA1/8$.

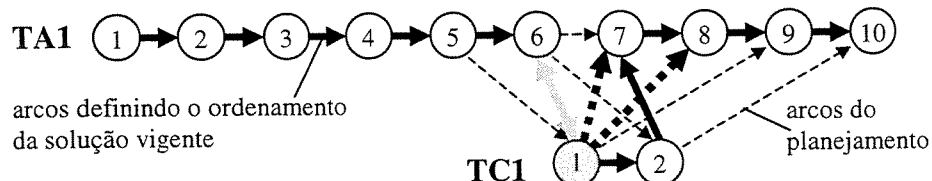


Figura 4.10. Arcos disjuntivos do nó TC1/1.

A tabela 4.3 mostra para cada possível primeiro sorteio a quantidade de arcos disjuntivos remanescentes e a quantidade inicial de arcos disjuntivos (planejamento). A tabela 4.3 mostra que há uma redução no número de arcos disjuntivos remanescentes quando são considerados os arcos do planejamento e os arcos definidos pela solução vigente. Por exemplo, devido à consideração dos arcos dirigidos da solução vigente, o número de arcos disjuntivos ligados ao nó *TC1/1* foi reduzido de três arcos para apenas um, entre os nós *TC1/1* e *TA1/6*, como pode ser visto na figura 4.10.

Tabela 4.3. Lista dos nós habilitados para serem o primeiro nó sorteado para o caso 1.

Nó do primeiro o sorteio	nós possíveis para o segundo sorteio	Nº de arcos disjuntivos remanesc.	Nº de arcos disjuntivos planejam.	Nó do primeiro sorteio	nós possíveis para o segundo sorteio	nº de arcos disjuntivos remanesc.	nº de arcos disjuntivos planejam.
TA1/6	TC1/1	1	1	TC2/1	TB2/7	1	1
TA1/7	TC1/1 – 2	2	2	TB3/15	-	-	1
TA1/8	-	-	2	TB3/16	-	-	1
TA1/9	-	-	1	TB3/17	-	-	2
TC1/1	TA1/6	1	3	TB3/18	TC3/1 – 2	2	2
TC1/2	TA1/7 – 9	3	3	TC3/1	TB3/15 – 18	4	4
TA2/5	TA4/2	1	1	TC3/2	-	-	2
TA2/6	TA4/3	1	1	TA3/3	TC5/1 – 4	3	4
TA2/9	TA4/6	1	2	TA3/4	TC5/3 – 5	3	3
TA2/10	TA4/6 – 7	2	2	TC5/1	TA3/3	1	1
TA4/2	TA2/5	1	1	TC5/2	TA3/3	1	1
TA4/3	TA2/6	1	1	TC5/3	TA3/4	1	2
TA4/6	TA2/9 – 10	2	2	TC5/4	TA3/4	1	2
TA4/7	TA2/10	1	2	TC5/5	-	-	1
TB2/7	TC2/1	1	1				

Dos 29 nós inicialmente definidos (na tabela 4.2) como habilitados para o primeiro sorteio, 7 nós não estão mais habilitados quando se consideram os arcos definidos pela solução vigente, porque não possuem nenhuma troca factível. Por exemplo, no final do planejamento foi definido que haviam dois arcos disjuntivos ligados ao nó *TA1/8*, como pode ser visto na figura 4.4. A inclusão dos arcos dirigidos da solução vigente fez estes dois arcos disjuntivos tornarem-se arcos dirigidos, onde os nós *TC1/1* e *TC1/2* precedem o nó *TA1/8*. Com isso, a consideração dos arcos da solução vigente reduziu a apenas 22 nós, dos 85 nós do problema, o número de nós que estão habilitados para o primeiro sorteio, o que significa uma grande redução no espaço de busca da solução ótima. Além disso, a redução do espaço de busca também se manifesta no número de nós habilitados para serem o segundo nó sorteado. Isto porque alguns dos nós que no final do planejamento estavam ligados por arcos disjuntivos ao primeiro nó sorteado não estarão mais habilitados para serem o segundo nó sorteado.

Tem-se então na tabela 4.3, para cada nó habilitado para o primeiro sorteio, os nós habilitados para o segundo sorteio. A partir desta relação sorteia-se o segundo nó para a permutação. No exemplo tratado em que o primeiro nó sorteado é o *TC1/1*, há apenas o nó *TA1/6* para ser selecionado. A figura 4.11 apresenta a construção do novo candidato.

Antigo ordenamento em **P1**: TA1/1, ..., TA1/5, **TA1/6**, TC1/1, TC1/2, TA1/7, ..., TA1/10.
 Novo ordenamento em **P1**: TA1/1, ..., TA1/5, **TC1/1**, **TA1/6**, TC1/2, TA1/7, ..., TA1/10.

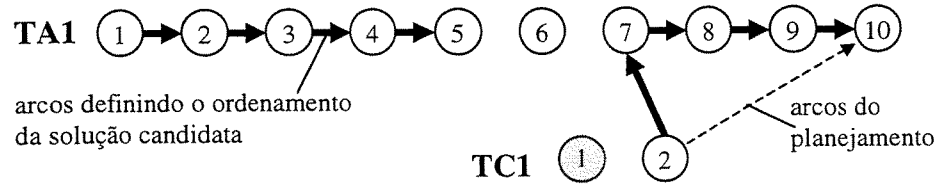


Figura 4.11a. Primeiro nó liberado.

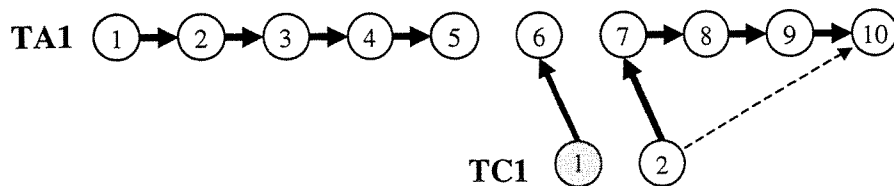


Figura 4.11b. Permutação.

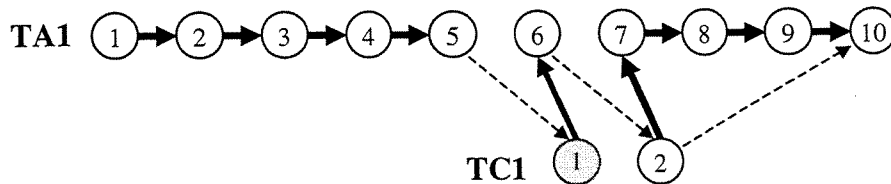


Figura 4.11c. Arcos fixos do planejamento.

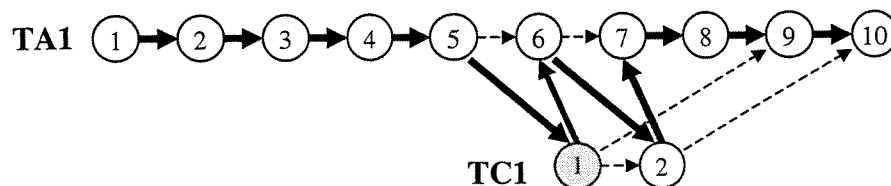


Figura 4.11d. Novo candidato.

4.4.1.1. Resultados da comparação do SA para as duas abordagens

O SA proposto é comparado com um SA sem filtragem dos candidatos. A tabela 4.4 apresenta, em função do nó do primeiro sorteio, o número de permutações possíveis. Pode-se notar que no caso do SA com filtragem o conjunto de nós para o primeiro sorteio é reduzido, como já foi discutido anteriormente.

Tabela 4.4. Possibilidades de permutação em função do primeiro nó sorteado.

nó	Arcos disjuntivos planejam.	Nº de trocas no SA com filtragem	Nº de trocas no SA sem filtragem	nó	Arcos disjuntivos planejam.	Nº de trocas no SA com filtragem	Nº de trocas no SA sem filtragem
TA1/1	-	-	2	TC4/5	-	-	7
TA1/2	-	-	2	TC2/1	TB2/7	1	7
TA1/3	-	-	2	TC2/2	-	-	7
TA1/4	-	-	2	TB2/1	-	-	2
TA1/5	-	-	2	TB2/2	-	-	2
TA1/6	TC1/1	1	2	TB2/3	-	-	2
TA1/7	TC1/1 – 2	2	2	TB2/4	-	-	2
TA1/8	TC1/1 – 2	2	2	TB2/5	-	-	2
TA1/9	TC1/ 2	1	2	TB2/6	-	-	2
TA1/10	-	-	2	TB2/7	TC2/1	1	2
TC1/1	TA1/6 – 8	3	10	TB3/1	-	-	5
TC1/2	TA1/7 – 9	3	10	TB3/2	-	-	5
TA2/1	-	-	10	TB3/3	-	-	5
TA2/2	-	-	10	TB3/4	-	-	5
TA2/3	-	-	10	TB3/5	-	-	5
TA2/4	-	-	10	TB3/6	-	-	5
TA2/5	TA4/2	1	10	TB3/7	-	-	5
TA2/6	TA4/3	1	10	TB3/8	-	-	5
TA2/7	-	-	10	TB3/9	-	-	5
TA2/8	-	-	10	TB3/10	-	-	5
TA2/9	TA4/6 – 7	2	10	TB3/11	-	-	5
TA2/10	TA4/6 – 7	2	10	TB3/12	-	-	5
TA4/1	-	-	10	TB3/13	-	-	5
TA4/2	TA2/5	1	10	TB3/14	-	-	5
TA4/3	TA2/6	1	10	TB3/15	TC3/1	1	5
TA4/4	-	-	10	TB3/16	TC3/1	1	5
TA4/5	-	--	10	TB3/17	TC3/1 – 2	2	5
TA4/6	TA2/9 – 10	2	10	TB3/18	TC3/1 – 2	2	5
TA4/7	TA2/9 – 10	2	10	TC3/1	TB3/15 – 18	4	18
TA4/8	-	-	10	TC3/2	TB3/17 – 18	2	18
TA4/9	-	-	10	TC3/3	-	-	18
TA4/10	-	-	10	TC3/4	-	-	18
TB1/1	-	-	5	TC3/5	-	-	18
TB1/2	-	-	5	TA3/1	-	-	5
TB1/3	-	-	5	TA3/2	-	-	5
TB1/4	-	-	5	TA3/3	TC5/1 – 4	4	5
TB1/5	-	-	5	TA3/4	TC5/3 – 5	3	5
TB1/6	-	-	5	TC5/1	TA3/3	1	4
TB1/7	-	-	5	TC5/2	TA3/3	1	4
TC4/1	-	-	7	TC5/3	TA3/3 – 4	2	4
TC4/2	-	-	7	TC5/4	TA3/3 – 4	2	4
TC4/3	-	-	7	TC5/5	TA3/4	1	4
TC4/4	-	-	7				

A soma do número de trocas possíveis (na tabela 4.4) mostra que, devido aos arcos do planejamento e da solução vigente, o SA com filtragem possui 52 permutações possíveis. Por sua vez o SA sem filtragem possui 558 permutações possíveis.

Foram usados dois valores de temperatura inicial (1 e 5) para que se pudesse testar a convergência das duas abordagens. A comparação das duas abordagens do SA é feita a partir da evolução da melhor solução encontrada (S^*). Para que esta comparação fosse feita, o caso 1 foi resolvido 100 vezes para cada uma das duas abordagens, tendo sempre a

mesma solução inicial. As figuras 4.12 e 4.13, apresentam a evolução dos valores médios nas duas abordagens do SA. Nas duas figuras observa-se que o SA com filtragem atinge a solução ótima após a avaliação de um número menor de candidatos que o SA sem filtragem.

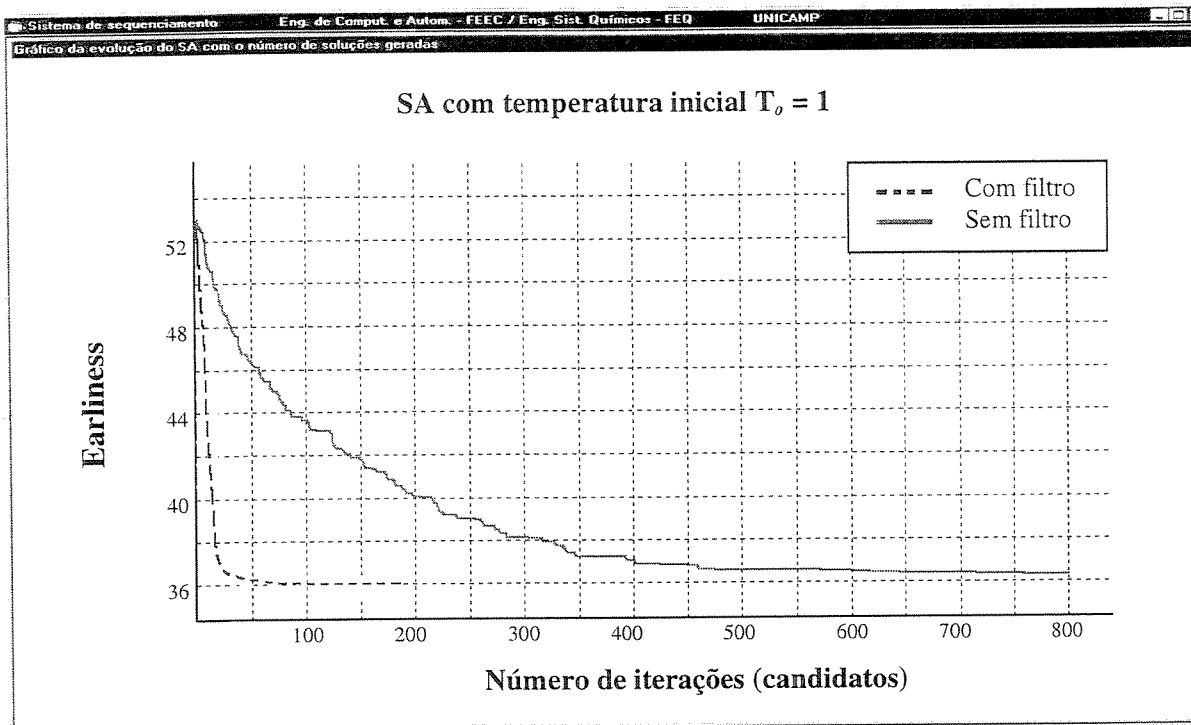


Figura 4.12. Comparação entre as duas abordagens do SA para temperatura inicial igual a 1.

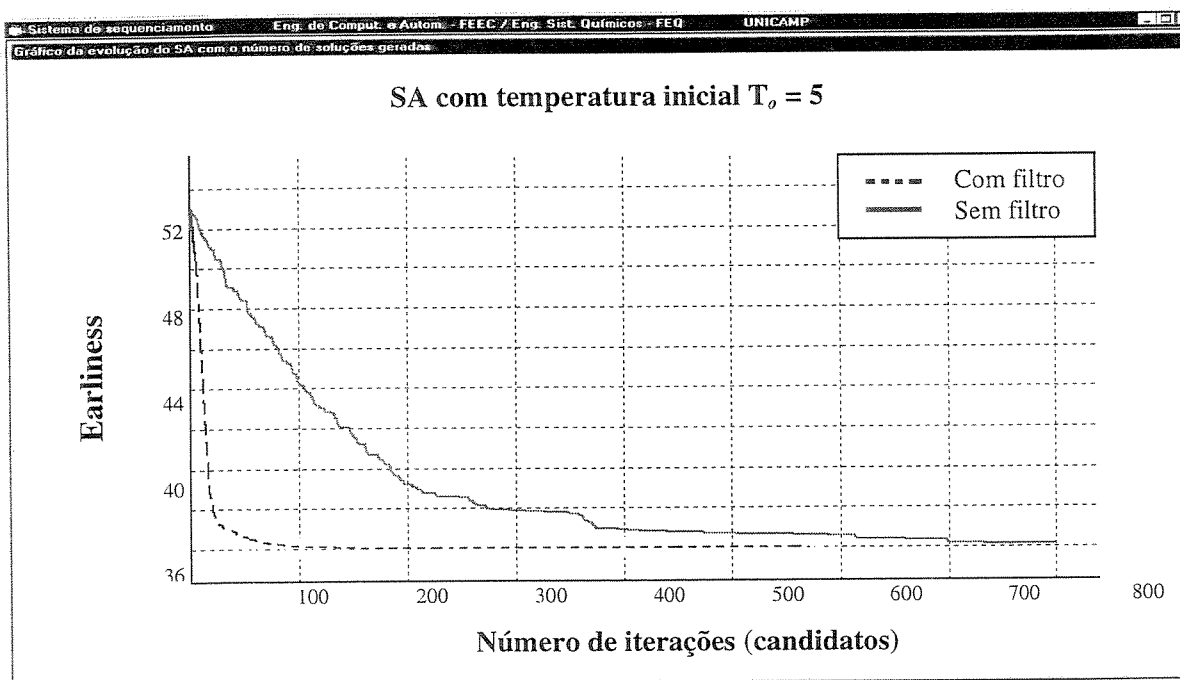


Figura 4.13. Comparação entre as duas abordagens do SA para temperatura inicial igual a 5.

4.4.2. Caso 2

O caso 2 corresponde à demanda e prazos de conclusão dos produtos finais dados na tabela 4.5. A tabela 4.5 (caso 2) difere da tabela 4.1 (caso 1) apenas nos prazos de conclusão dos produtos finais, fazendo o caso 2 ficar menos restrito que o caso 1.

Tabela 4.5. Demanda dos produtos finais para o caso 2.

Produto final	Demanda	Prazo (due date)
ProA	100	35
	90	61
ProB	270	35
ProC	75	61

A partir das janelas de tempo é possível identificar, no final da fase de planejamento, os ordenamentos existentes entre as bateladas. A figura 4.14 mostra as janelas de tempo obtidas no final da fase de planejamento para todas as bateladas do caso 2.

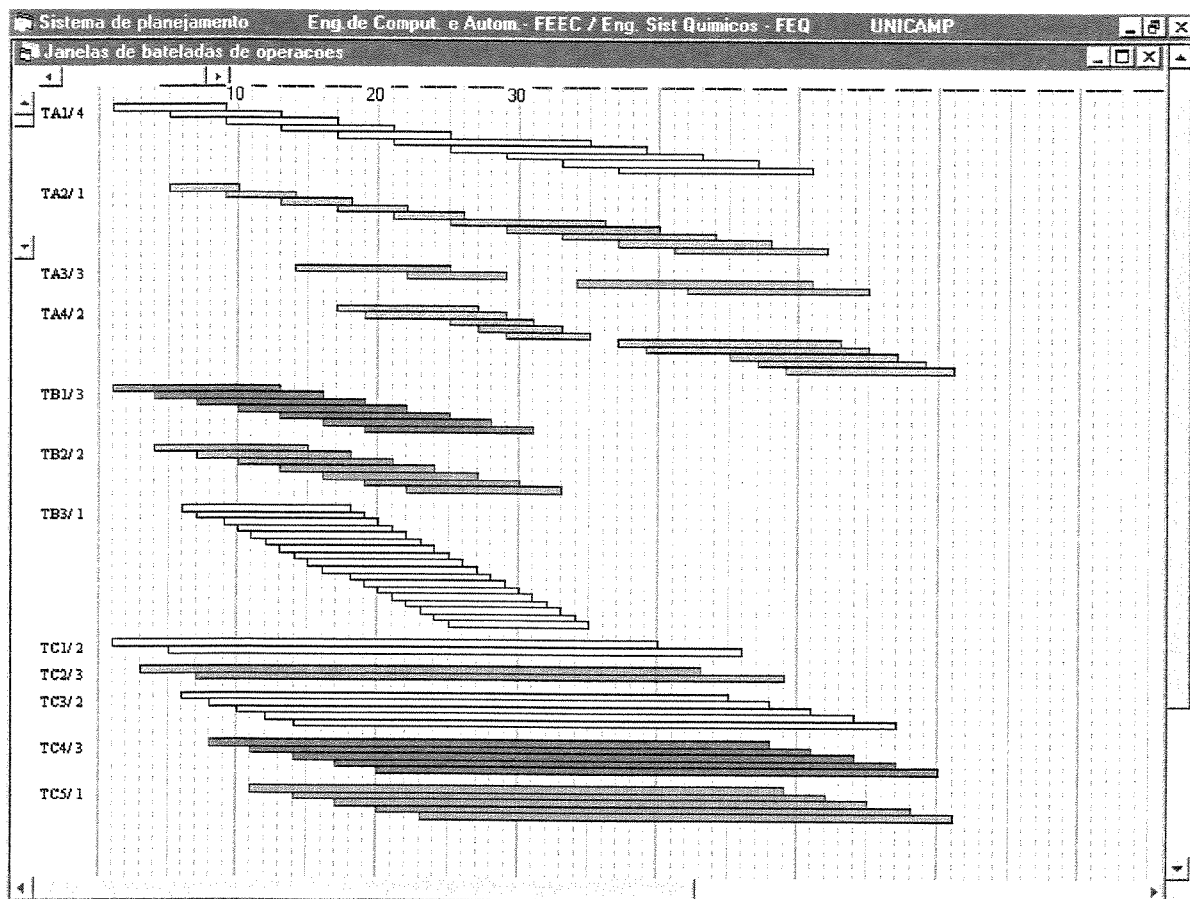


Figura 4.14. Janelas de tempo das bateladas.

A figura 4.15 apresenta os arcos dirigidos identificados no final da fase de planejamento. Os arcos dirigidos obtidos a partir das janelas de tempo são indicados em azul. Os arcos dirigidos entre bateladas de uma mesma operação são indicados em preto. Os arcos dirigidos obtidos a partir das relações de precedência decorrentes do balanço de massa e das relações de armazenamento são indicados em vermelho. Os nós habilitados para serem o primeiro nó sorteado no SA são apresentados pintados de preto.

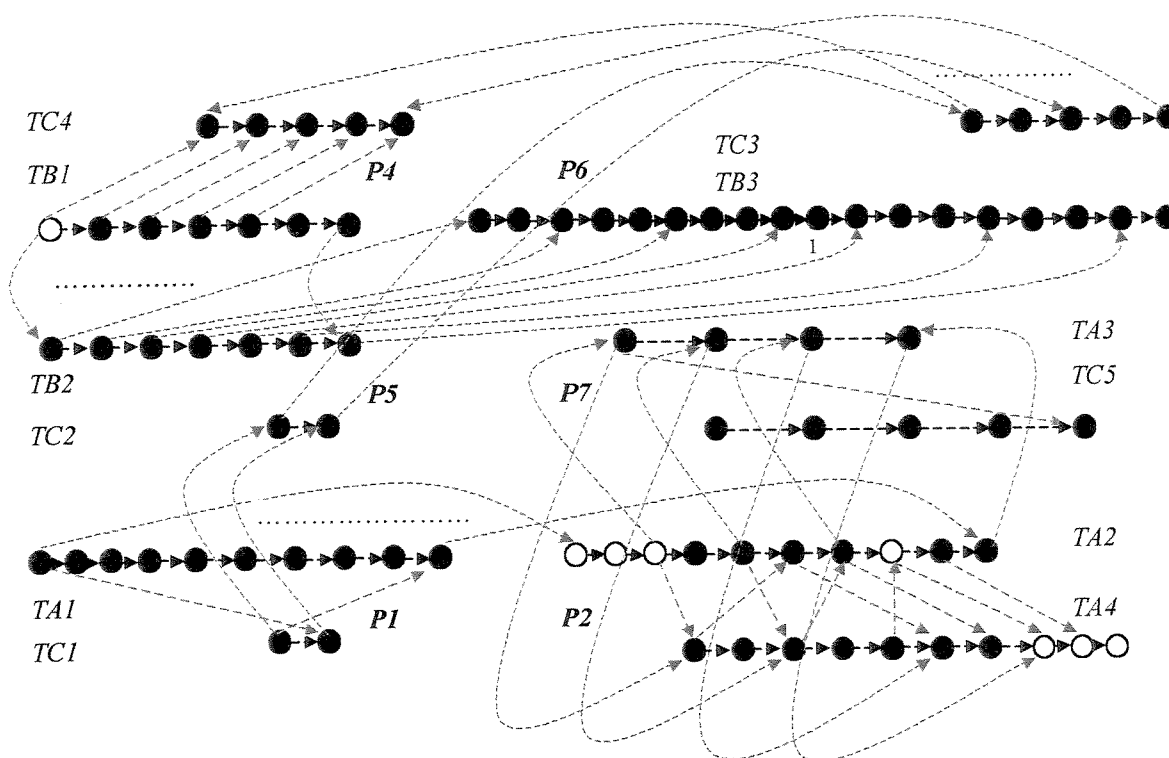


Figura 4.15. Arcos dirigidos identificados na fase de planejamento.

A tabela 4.6 apresenta a lista dos nós habilitados para serem o primeiro nó sorteado para o caso 2. O procedimento de filtragem no final da fase de planejamento permitiu reduzir o número de bateladas habilitadas para o primeiro sorteio de 85 para 77 bateladas.

Tabela 4.6. Lista dos nós habilitados para serem o primeiro nó sorteado.

processadores	nós habilitados para serem o primeiro nó sorteado
P1	TA1/1 a 10, TC1/1 e 2
P2	TA2/4 a 7, TA2/9 e 10, TA4/1 a 7
P3	-
P4	TB1/2 a 7, TC4/1 a 5
P5	TB2/1 a 7, TC2/1 e 2
P6	TB3/1 a 18, TC3/1 a 5
P7	TA3/1 a 4, TC5/1 a 5

Para o caso 2, a figura 4.16 apresenta os ordenamentos da solução inicial adotada. A figura 4.17 apresenta a carta de Gantt para a solução inicial (earliness = 60).

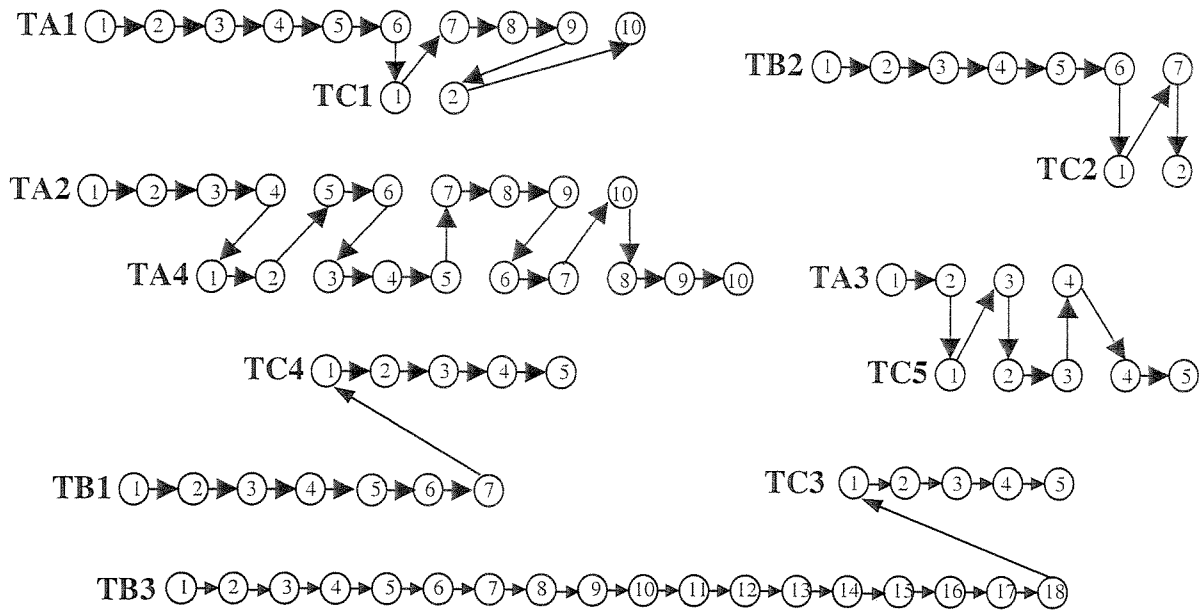


Figura 4.16. Ordenamentos da solução inicial no caso 2.

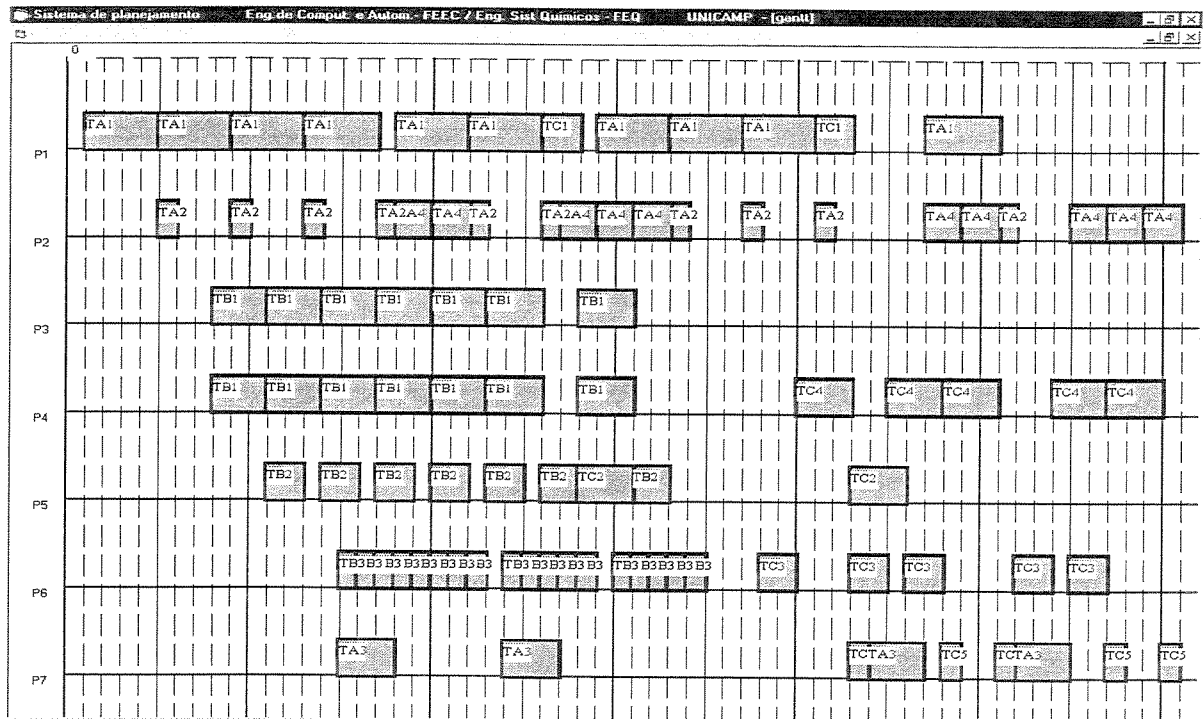


Figura 4.17. Solução inicial do caso 2 de valor 60.

A figura 4.18 apresenta o grafo com os arcos dirigidos do planejamento, os arcos gerados pelas relações de precedência e os ordenamentos da solução inicial adotada, sendo que os nós habilitados para serem sorteados são destacados.

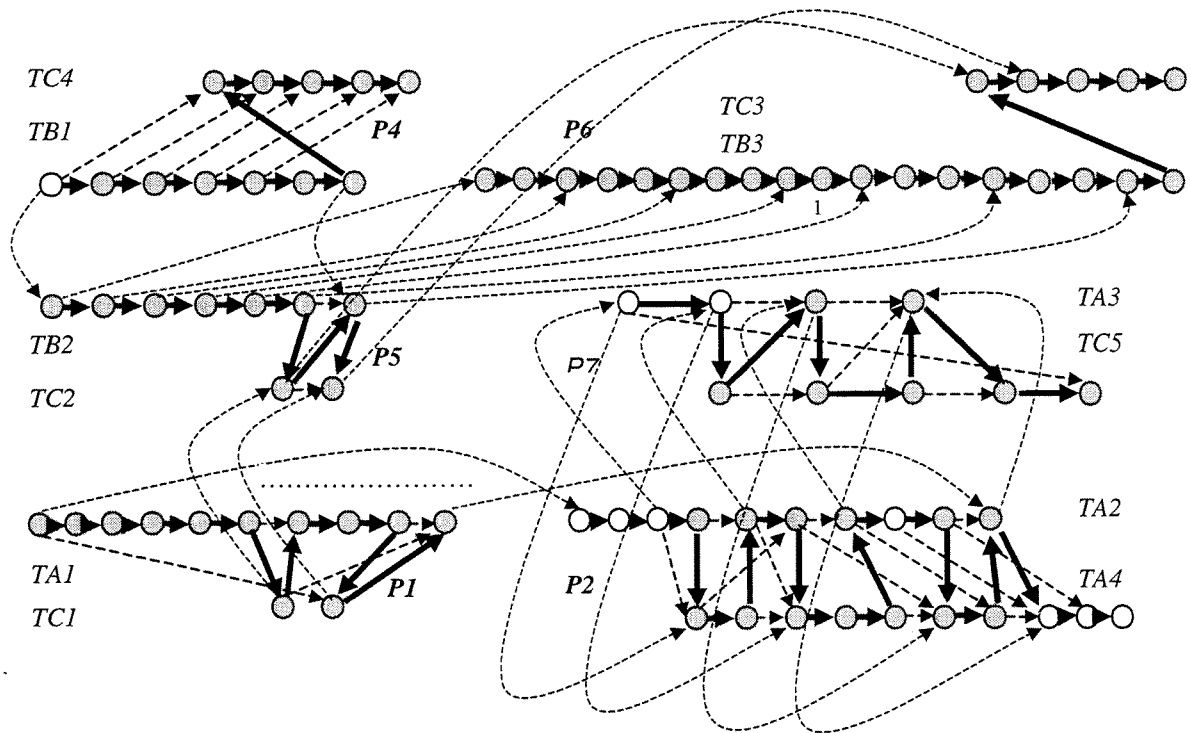


Figura 4.18. Ordenamentos da solução inicial para o caso 3.

A tabela 4.7 mostra para cada possível primeiro sorteio a quantidade de arcos disjuntivos remanescentes (considerando-se os arcos dirigidos do planeamento e da solução vigente) e a quantidade inicial de arcos disjuntivos (considerando-se só os arcos dirigidos do planeamento). A tabela 4.7 mostra que há uma redução no número de arcos disjuntivos remanescentes quando são considerados os arcos do planeamento e os arcos definidos pela solução vigente. A tabela 4.7 identifica quais os nós habilitados para serem o primeiro nó sorteado do SA com filtragem quando se consideram os arcos disjuntivos do planeamento e quando se consideram os arcos disjuntivos remanescentes. Somente podem ser sorteadas as bateladas com uma ou mais trocas (arcos disjuntivos) permitidas.

Dos 77 nós inicialmente definidos (na tabela 4.6) como habilitados para o primeiro sorteio, 39 nós não estão mais habilitados quando se consideram os arcos definidos pela solução vigente, porque não possuem nenhuma troca factível, como mostra a tabela 4.7. Com isso, a consideração dos arcos da solução vigente reduziu a apenas 38 nós, dos 85 nós do problema, o número de nós que estão habilitados para o primeiro sorteio. Além disso, a redução do espaço de busca também se manifesta no número de permutações possíveis. Isto porque alguns dos nós que no final do planeamento estavam ligados por arcos disjuntivos ao primeiro nó sorteado não estarão mais habilitados para serem o segundo nó sorteado. Após o planeamento são identificadas 348 permutações possíveis. A consideração também dos arcos dirigidos da solução vigente reduziu o total de permutações a 99.

Tabela 4.7. Lista dos nós habilitados para serem o primeiro nó sorteado para o caso 2.

Nó do primeiro sorteio	Nós possíveis para o segundo sorteio	Nº de arcos disjuntivos remanesc.	Nº de arcos disjuntivos planejam.	Nó do primeiro sorteio	Nós possíveis para o segundo sorteio	Nº de arcos disjuntivos remanesc.	Nº de arcos disjuntivos planejam.
TA1/1	TC1/1	1	1	TB2/2	-	-	2
TA1/2	TC1/1	1	2	TB2/3	-	-	2
TA1/3	TC1/1	1	2	TB2/4	-	-	2
TA1/4	TC1/1	1	2	TB2/5	-	-	2
TA1/5	TC1/1	1	2	TB2/6	TC2/1	1	2
TA1/6	TC1/1	1	2	TB2/7	TC2/1 – 2	2	2
TA1/7	TC1/1	1	2	TB3/1	-	-	5
TA1/8	-	-	2	TB3/2	-	-	5
TA1/9	TC1/ 2	1	2	TB3/3	-	-	5
TA1/10	TC1/2	1	1	TB3/4	-	-	5
TC1/1	TA1/1 – 9	9	9	TB3/5	-	-	5
TC1/2	TA1/7 – 10	4	9	TB3/6	-	-	5
TA2/4	TA4/1 – 2	2	2	TB3/7	-	-	5
TA2/5	TA4/1 – 2	2	2	TB3/8	-	-	5
TA2/6	TA4/3 – 5	3	4	TB3/9	-	-	5
TA2/7	TA4/4 – 5	2	2	TB3/10	-	-	5
TA2/9	TA4/6 – 7	2	2	TB3/11	-	-	5
TA2/10	TA4/6 – 7	2	2	TB3/12	-	-	5
TA4/1	TA2/4	1	2	TB3/13	-	-	5
TA4/2	TA2/5 – 6	2	3	TB3/14	-	-	5
TA4/3	TA2/6	1	1	TB3/15	-	-	5
TA4/4	-	-	2	TB3/16	-	-	5
TA4/5	TA2/7	1	2	TB3/17	-	-	5
TA4/6	TA2/9	1	2	TB3/18	TC3/1 – 5	5	5
TA4/7	TA2/10	1	2	TC3/1	TB3/1 – 18	18	18
TB1/2	-	-	1	TC3/2	-	-	18
TB1/3	-	-	2	TC3/3	-	-	18
TB1/4	-	-	3	TC3/4	-	-	18
TB1/5	-	-	4	TC3/5	-	-	18
TB1/6	-	-	5	TA3/1	-	-	4
TB1/7	TC4/1 – 5	5	5	TA3/2	TC5/1	1	5
TC4/1	TB1/2 – 7	6	6	TA3/3	TC5/1 – 3	3	5
TC4/2	-	-	5	TA3/4	TC5/2 – 3	2	5
TC4/3	-	-	4	TC5/1	TA3/1 – 3	3	4
TC4/4	-	-	3	TC5/2	TA3/3	1	4
TC4/5	-	-	2	TC5/3	TA3/ 4	1	4
TC2/1	TB2/1 – 7	7	7	TC5/4	TA3/4	1	4
TC2/2	TB2/7	1	6	TC5/5	-	-	3
TB2/1	-	-	1				

4.4.2.1. Resultados da comparação do SA para as duas abordagens

O SA proposto é comparado com um SA sem filtragem dos candidatos. A tabela 4.8 apresenta, em função do nó do primeiro sorteio, o número de permutações possíveis. Pode-se notar que no caso do SA com filtragem o conjunto de nós para o primeiro sorteio é reduzido, como já foi discutido anteriormente.

Tabela 4.8. Lista dos nós habilitados para serem o primeiro nó sorteado para o caso 2.

nó	Arcos disjuntivos planejam.	Nº de trocas no SA com filtragem	Nº de trocas no SA sem filtragem	nó	Arcos disjuntivos planejam.	Nº de trocas no SA com filtragem	Nº de trocas no SA sem filtragem
TA1/1	TC1/1	1	2	TC4/5	TB1/6 – 7	2	7
TA1/2	TC1/1 – 2	2	2	TC2/1	TB2/1 – 7	7	7
TA1/3	TC1/1 – 2	2	2	TC2/2	TB2/2 – 7	6	7
TA1/4	TC1/1 – 2	2	2	TB2/1	TC2/1	1	2
TA1/5	TC1/1 – 2	2	2	TB2/2	TC2/1 – 2	2	2
TA1/6	TC1/1 – 2	2	2	TB2/3	TC2/1 – 2	2	2
TA1/7	TC1/1 – 2	2	2	TB2/4	TC2/1 – 2	2	2
TA1/8	TC1/1 – 2	2	2	TB2/5	TC2/1 – 2	2	2
TA1/9	TC1/1 – 2	2	2	TB2/6	TC2/1 – 2	2	2
TA1/10	TC1/2	1	2	TB2/7	TC2/1 – 2	2	2
TC1/1	TA1/1 – 9	9	10	TB3/1	TC3/1 – 5	5	5
TC1/2	TA1/2 – 10	9	10	TB3/2	TC3/1 – 5	5	5
TA2/1	-	-	10	TB3/3	TC3/1 – 5	5	5
TA2/2	-	-	10	TB3/4	TC3/1 – 5	5	5
TA2/3	-	-	10	TB3/5	TC3/1 – 5	5	5
TA2/4	TA4/1 – 2	2	10	TB3/6	TC3/1 – 5	5	5
TA2/5	TA4/1 – 2	2	10	TB3/7	TC3/1 – 5	5	5
TA2/6	TA4/2 – 5	4	10	TB3/8	TC3/1 – 5	5	5
TA2/7	TA4/4 – 5	2	10	TB3/9	TC3/1 – 5	5	5
TA2/8	-	-	10	TB3/10	TC3/1 – 5	5	5
TA2/9	TA4/6 – 7	2	10	TB3/11	TC3/1 – 5	5	5
TA2/10	TA4/6 – 7	2	10	TB3/12	TC3/1 – 5	5	5
TA4/1	TA2/4 – 5	2	10	TB3/13	TC3/1 – 5	5	5
TA4/2	TA2/4 – 6	3	10	TB3/14	TC3/1 – 5	5	5
TA4/3	TA2/6	1	10	TB3/15	TC3/1 – 5	5	5
TA4/4	TA2/6 – 7	2	10	TB3/16	TC3/1 – 5	5	5
TA4/5	TA2/6 – 7	2	10	TB3/17	TC3/1 – 5	5	5
TA4/6	TA2/9 – 10	2	10	TB3/18	TC3/1 – 5	5	5
TA4/7	TA2/9 – 10	2	10	TC3/1	TB3/1 – 18	18	18
TA4/8	-	-	10	TC3/2	TB3/1 – 18	18	18
TA4/9	-	-	10	TC3/3	TB3/1 – 18	18	18
TA4/10	-	-	10	TC3/4	TB3/1 – 18	18	18
TB1/1	-	-	5	TC3/5	TB3/1 – 18	18	18
TB1/2	TC4/1	1	5	TA3/1	TC5/1 – 4	4	5
TB1/3	TC4/1 – 2	2	5	TA3/2	TC5/2 – 5	4	5
TB1/4	TC4/1 – 3	3	5	TA3/3	TC5/3 – 5	3	5
TB1/5	TC4/1 – 4	4	5	TA3/4	TC5/4 – 5	2	5
TB1/6	TC4/1 – 5	5	5	TC5/1	TA3/1 – 4	4	4
TB1/7	TC4/1 – 5	5	5	TC5/2	TA3/1 – 4	4	4
TC4/1	TB1/2 – 7	6	7	TC5/3	TA3/1 – 4	4	4
TC4/2	TB1/3 – 7	5	7	TC5/4	TA3/1 – 4	4	4
TC4/3	TB1/4 – 7	4	7	TC5/5	TA3/2 – 4	4	4
TC4/4	TB1/5 – 7	3	7				

A soma do número de trocas possíveis (na tabela 4.8) mostra que, devido aos arcos do planejamento e da solução vigente, o SA com filtragem possui 99 permutações possíveis. Por sua vez o SA sem filtragem possui 558 permutações possíveis.

As figuras 4.19 e 4.20, apresentam a evolução dos valores médios nas duas abordagens do SA. Nas duas figuras observa-se que o SA com filtragem atinge a solução ótima após a avaliação de um número menor de candidatos que o SA sem filtragem.

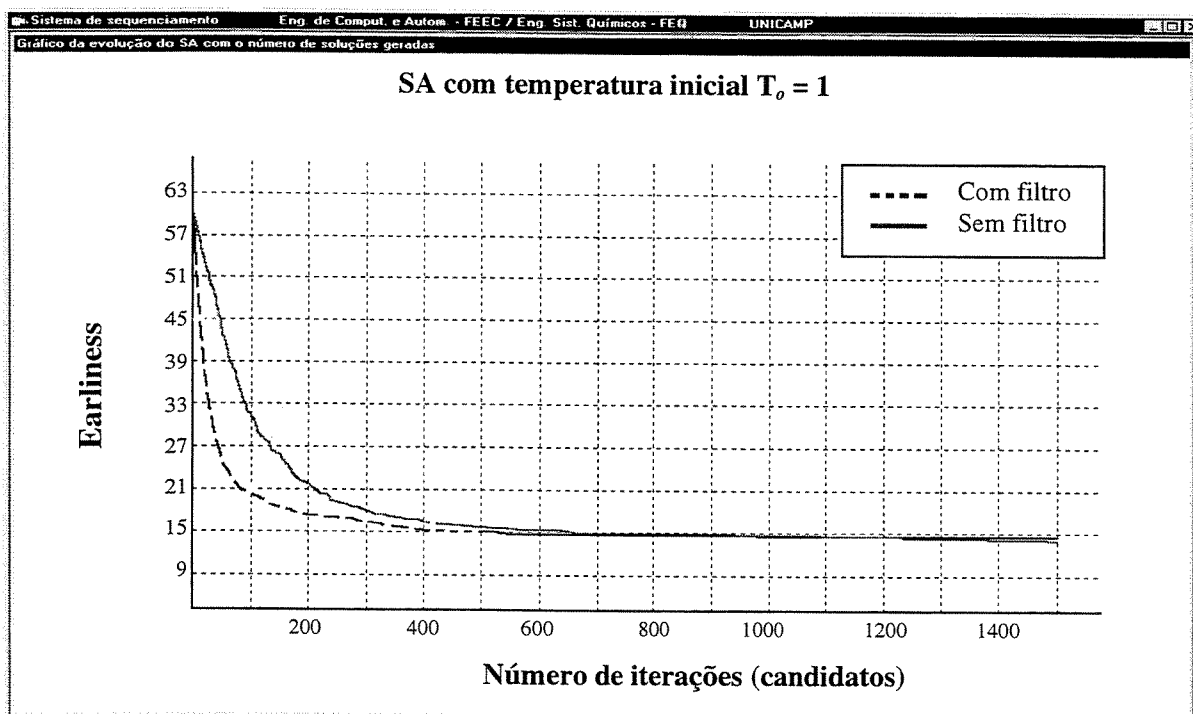


Figura 4.19. Evolução do valor de earliness para o SA com temperatura inicial 1.

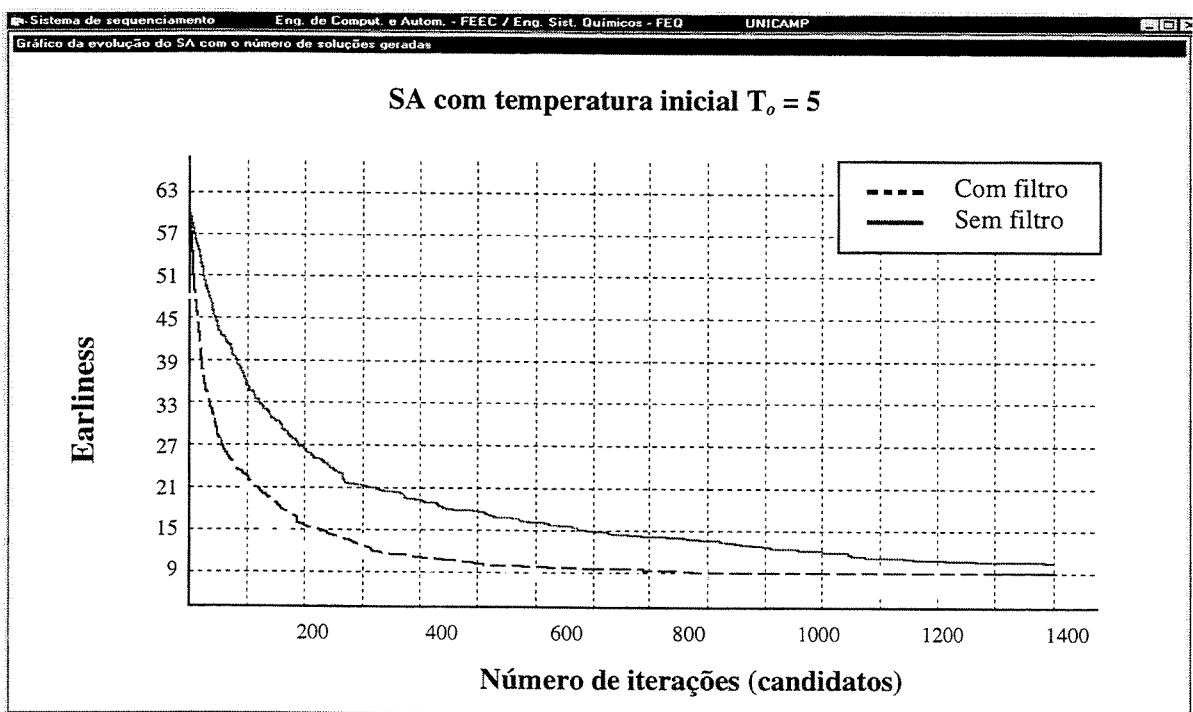


Figura 4.20. Evolução do valor de earliness para o SA com temperatura inicial 5.

O caso 2 é menos restritivo que o caso 1 e neste caso a vantagem no uso do SA com filtragem não foi tão evidente quanto no primeiro caso. Ainda assim, o SA com filtragem apresenta um número menor de trocas permitidas que o SA sem filtragem. O

resultado desta diferença menor entre as duas abordagens também foi manifestado no número de candidatos gerados necessários para a convergência do SA para próximo da solução ótima.

4.5. Conclusão

Neste capítulo foi proposta uma estratégia de solução do problema de *scheduling* usando uma abordagem SA com janelas de tempo para a realização das operações. As janelas de tempo associadas à competição pelos equipamentos podem levar a um conjunto de restrições ativas para as quais algoritmos do tipo SA clássico (sem filtragem) não são eficientes e/ou adequados, já que o número de candidatos inactivos gerados pode ser elevado.

Na abordagem SA com filtragem é proposto um procedimento em que estas restrições são previamente analisadas. O impacto das restrições é estabelecido através da definição de ordenamentos obrigatórios, que limitam o número de candidatos possíveis de serem gerados. Os efeitos são a redução do espaço de busca, através da redução no número de candidatos inactivos gerados, e do esforço computacional na avaliação da função objetivo.

CAPÍTULO 5. CONCLUSÕES

O trabalho desenvolvido nesta tese tem como objetivo principal propor abordagens para a solução de problemas de planejamento e programação da produção de curto prazo na indústria de processos operando em batelada. Este tipo de problema vem se tornando cada vez mais importante no contexto industrial não só pelas condições de competitividade que vem crescendo na última década, assim como a alta taxa de substituição/atualização de produtos (moléculas) que se tem observado nos últimos vinte anos.

A principal característica das plantas utilizadas na produção de tais produtos é a flexibilidade que permite a inserção e/ou retirada de produtos do mercado, mantendo-se, em geral com pequenas alterações, os equipamentos e o *lay-out* da planta. Se, por um lado, a flexibilidade é benéfica, ela traz também um problema complexo de programação da produção, provocado pelas demandas que variam ao longo do tempo e pela competição das operações da receita de produção pelos equipamentos compartilhados.

O sistema de planejamento implementado permite que o usuário tenha informação sobre a factibilidade do cenário de produção gerado e sobre o carregamento dos processadores decorrente deste cenário. A possibilidade de interação do usuário com o sistema permite que novos cenários de produção sejam gerados sempre que for detectada alguma infactibilidade ou quando o usuário achar conveniente.

Os pontos essenciais no sistema de planejamento proposto são:

- Utilização de técnicas de explosão para frente e para trás para a obtenção das janelas de processamento;
- Utilização de técnicas de propagação de restrições para obter as janelas de tempo reais decorrentes das restrições do problema;
- Utilização dos conceitos de crucialidade/demanda agregada para visualizar o carregamento dos processadores.

As técnicas de propagação de restrições utilizadas em sistemas de busca orientada por restrições foram ampliadas para incluir a propagação de restrições de armazenagem. Utilizam-se em conjunto dois mecanismos originados na área de busca orientada por restrições: a propagação de restrições e as funções de crucialidade. Esta abordagem nova levou a uma proposta para automatizar a ativação dos mecanismos de propagação de restrições apenas nas situações mais promissoras.

A utilização de janelas de tempo das bateladas, na formulação MILP do problema de *scheduling*, permite reduzir sensivelmente a dimensão do problema. Esta redução é maior para situações com carregamento alto porque este carregamento implica em janelas de tempo menores. A redução na dimensão do problema é conseguida basicamente através da redução do número de variáveis binárias.

A formulação MILP reduzida foi desenvolvida com o intuito de eliminar do problema de *scheduling* as bateladas que não envolvem problemas de alocação difíceis, por exemplo, por estarem atribuídas a processadores dedicados ou com carregamento baixo.

Estas situações levam geralmente a degenerescência na solução do modelo MILP com os softwares comerciais hoje existentes.

A inclusão de um procedimento de filtragem na abordagem SA, limitando o número de candidatos inactivos gerados, tem se mostrado eficiente para reduzir o espaço de busca e o esforço computacional. O esforço adicional para a eliminação de candidatos inactivos é extremamente baixo, dado que utiliza resultados disponíveis da fase de planeamento e um algoritmo de detecção de ciclos em grafos. O procedimento de filtragem utilizado parece poder ser estendido para outras técnicas evolutivas de *scheduling*.

O sistema de planeamento atual deve ser ampliado no futuro. Especificamente através de:

- Introdução de mecanismos de propagação de restrições para recursos compartilhados cumulativos;
- Desenvolvimento de ferramentas para ajuda ao usuário na atribuição de operações a processadores;
- Introdução de ferramentas para o diagnóstico e alternativas para situações inactivos.

Uma abordagem de busca orientada por restrições para o problema de *scheduling* está em desenvolvimento. Com isto, poderão ser iniciados trabalhos com abordagens híbridas.

REFERÊNCIAS BIBLIOGRÁFICAS

- Baptiste P. e Le Pape C. (1995).** A Theoretical and Experimental Comparison of Constraint Propagation Techniques for Disjunctive Scheduling. *Proceedings of 14th International Joint Conference on Artificial Intelligence*. Montreal, Canadá.
- Basset M.H., Dave P., Doyle III F.J., Kudva G.K., Pekny J.F., Reklaitis G.V., Subrahmanyam S., Miller D.L. e Zentner M.G. (1996).** Perspectives on model based integration of process operations. *Computers and Chemical Engineering*, 20 (6/7), 821-844.
- Birewar D.B. e Grossmann I.E. (1990).** Simultaneous production planning and scheduling in multiproduct batch plants. *Industrial Engineering and Chemical Research*, 29, 570-580.
- Battiti R e Tecchiolli G. (1994).** Simulated annealing and tabu search in the long run: a comparison on QAP tasks. *Computer and Mathematics with Applications*, 28 (6), 1-8.
- Beck J.C., Davenport A.J., Sitarski E.M. e Fox M.S. (1997).** Texture-based heuristics for scheduling revisited.
- Cantón J. (2000).** Tese de doutorado na Universidade Politécnica da Catalunha (em preparação).
- Caseau Y. e Laburthe F. (1995).** Improving branch and bound for job shop scheduling with constraint propagation. *Proceedings of the 8th Franco-Japanese 4th Franco-Chinese Conference*.
- Cerny V. (1985).** A thermodynamic approach to the traveling salesman problem: an efficient simulated annealing algorithm. *Journal of Optimization Theory and Applications*, 45, 41-51.
- Claire schedule 1.0 (1999).** Reference manual. <http://www.ens.fr/~laburthe>
- Das H., Cummings P.T. e Le Van M.D. (1990).** Scheduling of serial multiproduct batch processes with simulated annealing. *Computers and Chemical Engineering*, 14, 1351-1357.
- Desai R. e Patil R. (1996).** SALO: Combining simulated annealing and local optimization for efficient global optimization. *9th Florida AI Research Symposium (FLAIRS)*.
- Drexel A. (1988).** A simulated annealing approach to the multiconstraint zero-one knapsack problem. *Computing*, 40, 1-8.
- Erschler J. (1976).** Analyse sous contraintes et aide à la décision pour certains problèmes d'ordonnancement. Tese de doutorado, Universidade Paul Sabatier, Toulouse, França.
- Erschler J. e Esquirol P. (1986).** Decision-aid in job shop scheduling: A knowledge based approach. *IEEE International Conference on Robotics and application*, San Francisco, EUA.
- Fox M. S. (1983).** Constraint-Directed Search: A Case Study of Job Shop Scheduling. Tese de doutorado, School of Computer Science, Carnegie Mellon University, EUA.
- Fox M. S. e Sycara K. (1990).** Overview of CORTES: A Constraint Based Approach to Production Planning, Scheduling and Control. *Proceedings of the Fourth International Conference on Expert Systems in Production and Operations Management*.

- GAMS (1998).** A user's guide. Gams Development Corporation. Washington, EUA.
- Gimeno L., Rodrigues M.T.M., Rodrigues L.A. (2000).** Constraint propagation tools in multipurpose batch plants short term planning. *Proceedings of 2nd Conference on Management and Control and Production and Logistics* (em CD-Rom). Grenoble, França.
- Graells M., Espuña A. e Puigjaner L. (1996).** Evolutionary identification of best schedules for optimum production planning. *Second International Conference on Computer Integrated Manufacturing in the Process Industries*. Eindhoven, Holland.
- Henning G.P. e Cerdá J. (1996).** A knowledge-based approach to production scheduling for batch processes. *Computers and Chemical Engineering*, 20 (supplement), S1295-S1300.
- ILOG (1997).** *Scheduler 4.0*. User's Manual, ILOG. Mountain View, EUA.
- i2 Technologies, Inc. (1996).** Rhythm tutorial manual. Irving, EUA.
- Johnson D.S., Aragon C.R., McGeoch L.A. e Schevon C. (1991).** Optimization by simulated annealing: an experimental evaluation, part II, graph coloring and number partitioning. *Operations Research*, 39 (3), 378-406.
- Keng N.P., Yun D.Y.Y. e Rossi M. (1988).** Interaction sensitive planning system for job shop scheduling. *Expert systems and intelligent manufacturing*, 57-69. Elsevier. Amsterdam, Holanda.
- Kirkpatrick S., Gelatt C.D. e Vecchi P.M. (1983).** Optimization by simulated annealing. *Science*, 220, 671-680.
- Kondili E., Pantelides C.C., Sargent e R.W.H. (1993).** A general algorithm for short term scheduling of batch operations – I. MILP formulation. *Computers and Chemical Engineering*, 17 (2), 211-227.
- Ku H. e Karimi I.A. (1991).** An evaluation of simulated annealing for batch process scheduling. *Industrial Engineering and Chemical Research*, 30, 163-169.
- Löhl T., Schulz C. e Engell S. (1998).** Sequencing of batch operations for a highly coupled production process. *Computers and Chemical Engineering*, 22 (supplement), 579-585.
- Lundy S. e Mees A. (1986).** Convergence of an annealing algorithm. *Math. Prog.*, 21, 498-516.
- Mamalis A.G. e Malagardis I. (1996).** Determination of due dates in job shop scheduling by simulated annealing. *Computer Integrated Manufacturing Systems*, 9 (2), 65-72.
- Martin O.C. e Otto S.W. (1994).** Combining simulated annealing with local search heuristics.
- Metropolis N., Rosenbluth A.N., Rosenbluth M.N., Teller A.H. e Teller H. (1953).** Equation of state calculation by fast computing machines. *J. Chem. Phys.*, 26 (6), 1087-1092.
- Mockus L. e Reklaitis G.V. (1997).** Mathematical programming formulation for scheduling of batch operations based on nonuniform time discretization. *Computers and Chemical Engineering*, 21 (10), 1147-1156.

- Mockus L. e Reklaitis G.V. (1999a).** Continuous time representation approach to batch and continuous process scheduling. 1. MINLP formulation. *Industrial Engineering and Chemical Research*, 38, 197-203.
- Mockus L. e Reklaitis G.V. (1999b).** Continuous time representation approach to batch and continuous process scheduling. 2. Computational issues. *Industrial Engineering and Chemical Research*, 38, 204-210.
- Morton T.E. e Pentico D.W. (1993).** Heuristic scheduling systems: With applications to production systems and project management. John Wiley & Sons.
- Orlicky, J. (1975).** *Material Requirements Planning*. McGraw-Hill. Nova York, EUA.
- Papageorgiou L.G. e Pantelides C.C. (1993).** A hierarchical approach for campaign planning of multipurpose batch plants. *Computers and Chemical Engineering*, 17 (supplement), 27-32.
- Papageorgiou L.G. e Pantelides C.C. (1996).** Optimal campaign planning/scheduling of multipurpose batch/semicontinuous plants. 2. A mathematical decomposition approach. *Industrial Engineering and Chemical Research*, 35, 510-529.
- Pinto J.M. e Grossmann I.E. (1995).** A continuous time mixed-integer linear programming model for short term scheduling of multistage batch plants. *Industrial Engineering and Chemical Research*, 34, 3037-3051.
- Preactor (2000).** Preactor News, vol. 2. Chippenham, Inglaterra.
- Rich S.H. e Prokopakis G.J. (1986).** Scheduling and sequencing of batch operations in a multipurpose plant. *Ind. Eng. Chem. Process Des. Dev.*, 25, 979-988.
- Rodrigues M.T.M., Gimeno L., Passos C.A.S. e Campos M.D. (1996).** Reactive scheduling approach for multipurpose batch plants. *Computers and Chemical Engineering*, 20 (supplement), S1215-S1220.
- Rodrigues M.T.M., Latre L.G. e Rodrigues L.C.A. (2000a).** Short-term planning and scheduling in multipurpose batch chemical plants: A multi-level approach. *Computers and Chemical Engineering*, 24, 2247-2258.
- Rodrigues M.T.M., Latre L.G. e Rodrigues L.C.A. (2000b).** Production planning using time windows for short term multipurpose batch plants scheduling problems. *Industrial Engineering and Chemical Research*, 39, 3823-3834.
- Rodrigues L.C.A., Graells M., Cantón J., Gimeno L., Rodrigues M.T.M., Espuña A. e Puigjaner L. (2000c).** Utilization of processing time windows to enhance planning and scheduling in short term multipurpose batch plants. *Computers and Chemical Engineering*, 24, 353-359.
- Romero R., Gallego R.A. e Monticelli A. (1996).** Transmission system expansion planning by simulated annealing. *IEEE Transactions on Power Systems*, 11 (1), 364-369.
- Sadeh N. (1991).** Look-ahead techniques for micro-opportunistic job shop scheduling, Tese de doutorado, School of Computer Science, Carnegie Mellon University.
- SAP (1999).** SAP Advanced Planner & Optimizer. Production Planning and Detailed Scheduling. <http://www.sap.com>.

- Shah N., Pantelides C.C. e Sargent R.W.H. (1993).** A general algorithm for short term scheduling of batch operations – II. Computational issues. *Computers and Chemical Engineering*, 17 (2), 229-244.
- Shapiro J.F. (1993).** Mathematical programming models and methods for production planning and scheduling (Graves S.C. *et al.*, editores). Em Handbooks in OR & MS, Cap. 8, vol. 4. Elsevier.
- Smith, S.F. (1994).** Opis: A methodology and architecture for reactive scheduling, Intelligent Scheduling, (eds.: Fox, M. e Zweben, M.), Morgan Kaufmann Publishers.
- Subrahmanyam S., Pekny J.F. e Reklaitis G.V. (1996).** Decomposition approaches to batch plant design and scheduling. *Industrial Engineering and Chemical Research*, 35 (6), 1866-1876.
- van Laarhoven P.J.M., Aarts E.H.L. (1987).** Simulated annealing: Theory and applications. Kluwer Academic Publishers. Holanda.
- van Laarhoven P.J.M., Aarts E.H.L. e Lenstra J.K. (1992).** Job shop scheduling by simulated annealing. *Operations Research*, 40 (1), 113-125.

ANEXO 1. DEFINIÇÃO DOS EXEMPLOS ESTUDADOS

A.1. Introdução

Foram adotados dois problemas da literatura para serem aplicados ao longo do trabalho, sendo que algumas alterações foram feitas nestes problemas. O primeiro problema foi proposto por (Papageorgiou e Pantelides, 1996) e o segundo problema foi proposto por (Papageorgiou e Pantelides, 1993). Estes problemas foram adotados por apresentarem estados com restrições de armazenagem variadas, gargalos de produção, processadores compartilhados e a receita do primeiro problema possuir estados compartilhados. São criados vários casos destes problemas com o propósito de comparar diferentes situações de carregamento da planta, gerando-se casos com demanda de produtos finais e disponibilidade de matérias primas diferentes.

A.2. Exemplo 1

A figura A.1 indica a representação STN da receita dos produtos finais do problema. A planta possui oito processadores que são usados na produção de quatro produtos finais.

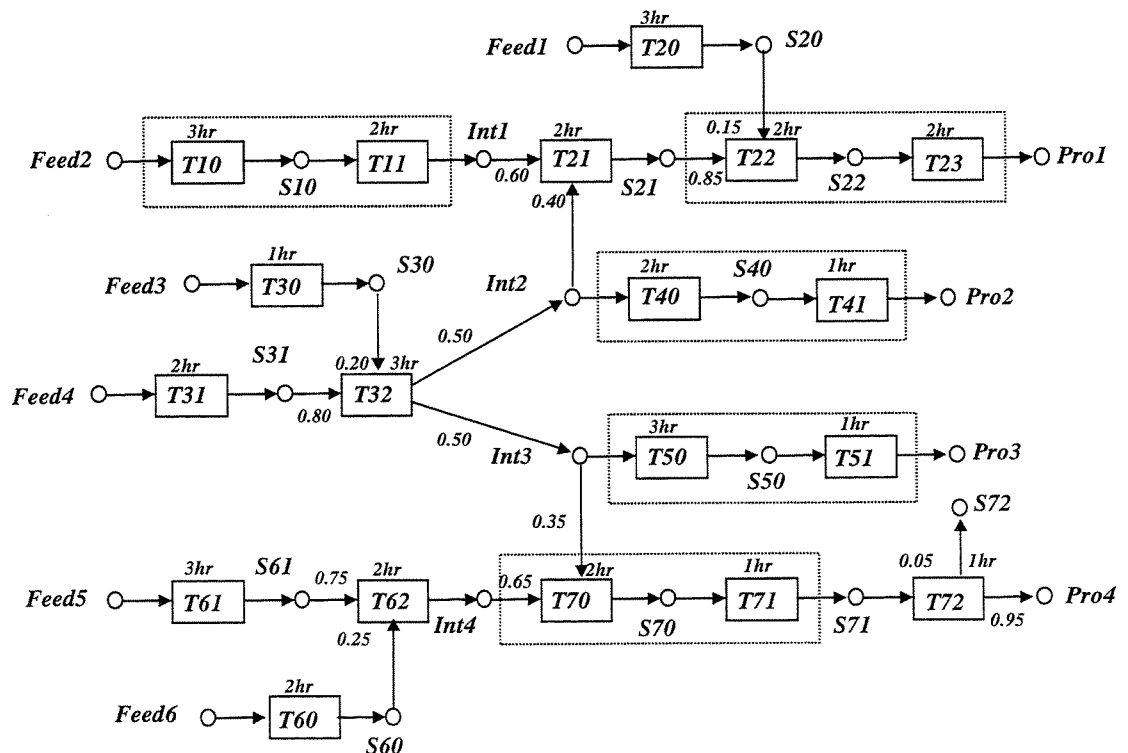


Figura A.1. Representação STN do caso 1.

A tabela A.1 contém os dados de entrada deste problema. Nesta tabela são apresentados na mesma linha uma operação e o estado gerado por esta operação.

Tabela A.1 - Dados de entrada do exemplo 1.

processador	operação	tamanho da batelada	estados gerados pela operação	restrições de armazenagem
P1	<i>T10</i>	6	<i>S10</i>	ZW
P1	<i>T21</i>	6	<i>S21</i>	UIS
P2	<i>T32</i>	8	<i>Int2 e Int3</i>	UIS
P3	<i>T31</i>	7	<i>S31</i>	UIS
P3	<i>T72</i>	7	<i>Pro4 e S72</i>	UIS
P4	<i>T23</i>	7	<i>Pro1</i>	UIS
P4	<i>T30</i>	7	<i>S30</i>	UIS
P4	<i>T60</i>	7	<i>S60</i>	UIS
P5	<i>T20</i>	8	<i>S20</i>	UIS
P5	<i>T40</i>	7	<i>S40</i>	ZW
P5	<i>T50</i>	8	<i>S50</i>	ZW
P6	<i>T61</i>	6	<i>S61</i>	UIS
P6	<i>T70</i>	6	<i>S70</i>	ZW
P7	<i>T11</i>	6	<i>Int1</i>	UIS
P7	<i>T22</i>	7	<i>S22</i>	ZW
P7	<i>T41</i>	7	<i>Pro2</i>	UIS
P8	<i>T51</i>	8	<i>Pro3</i>	UIS
P8	<i>T62</i>	8	<i>Int4</i>	UIS
P8	<i>T71</i>	6	<i>S71</i>	UIS

A3. Exemplo 2

O problema possui sete processadores que são usados na produção de três produtos finais. A figura A.2 apresenta a receita de produção de cada um dos produtos finais usando a representação STN.

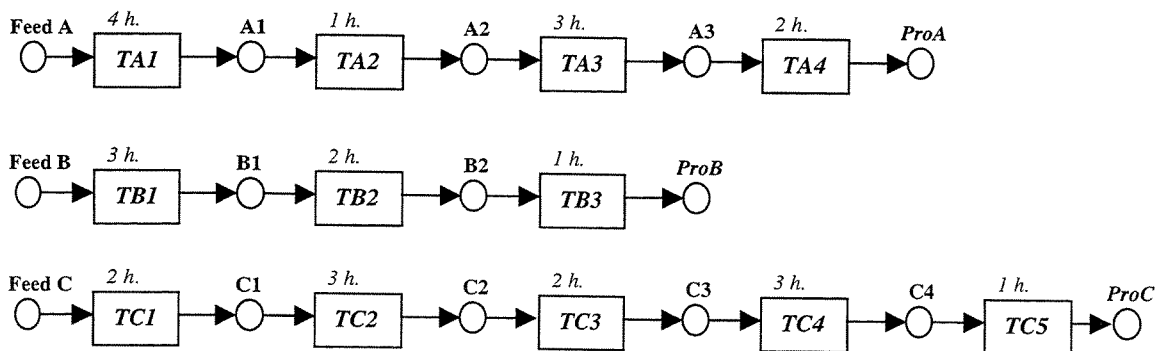


Figura A2. Representação STN do exemplo 2.

A tabela A.2 apresenta a atribuição de operações a processadores e as restrições de estocagem em cada um dos estados.

Tabela A.2 - Dados de entrada do exemplo 2.

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

ANEXO 2. INSTÂNCIAS DO EXEMPLO 2 APLICADAS À FORMULAÇÃO MILP

Foram criados vários casos do exemplo 2 para analisar o desempenho da formulação MILP reduzida (seção 3.4). Em todos os casos gerados, o problema MILP reduzido exclui as operações **TB3** e **TC3**. A tabela 3.9 é reapresentada, descrevendo as modificações nas instâncias do exemplo 2.

Tabela 3.9. Instâncias do exemplo 2.

caso	particularidade do caso
2.1	a situação anterior sem modificações
2.2	caso 2.1 com disponibilidade de matérias primas em 0
2.3	caso 2.1 com estado <i>A1</i> com restrição <i>ZW</i>
2.4	caso 2.2 com estado <i>A1</i> com restrição <i>ZW</i>
2.5	caso 2.1 com prazos de entrega de <i>ProB</i> em 52 (em lugar de 32). Isto leva a janelas de <i>TB3</i> com maior sobreposição com as janelas de <i>TC3</i>
2.6	caso 2.5 com disponibilidade de <i>FeedB</i> em 10 (em lugar de 1). O carregamento em <i>P6</i> induzido pela sobreposição aumenta porque as janelas de <i>TB3</i> são menores
2.7	caso 2.5 com disponibilidade de <i>FeedB</i> em 15 (em lugar de 1). O carregamento em <i>P6</i> aumenta

Nas páginas seguintes são apresentadas, para todas os casos gerados, as informações do planejamento (janelas de tempo e carregamento dos processadores) e as cartas de Gantt para a formulação MILP global, para o primeiro passo e o segundo passo da formulação MILP reduzida.

a) Caso 2.1

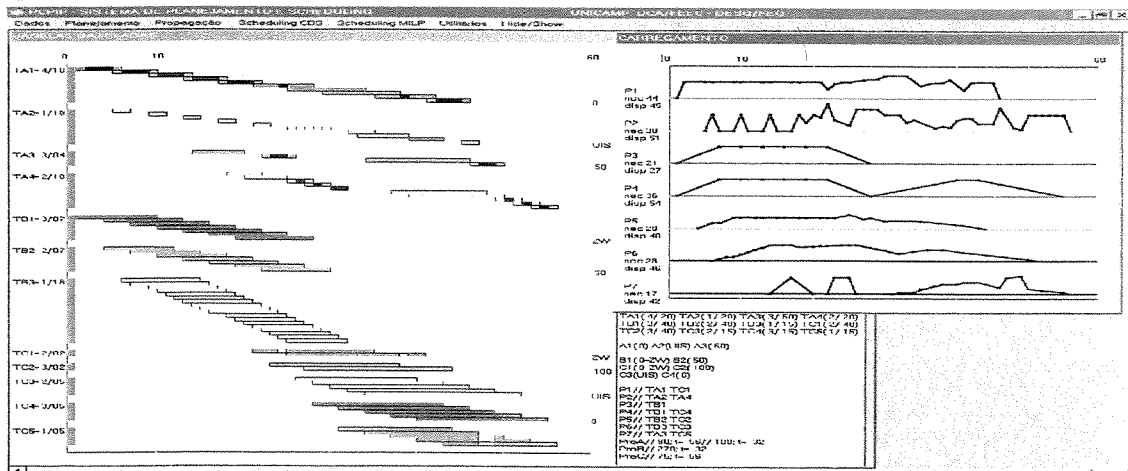


Figura A.3. Janelas de tempo e carregamento do caso 2.1.

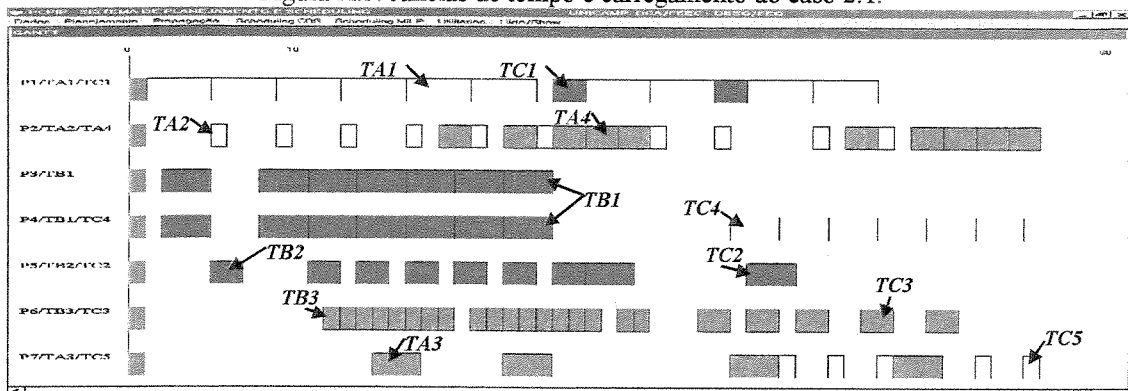


Figura A.4. Solução do MILP global para o caso 2.1.

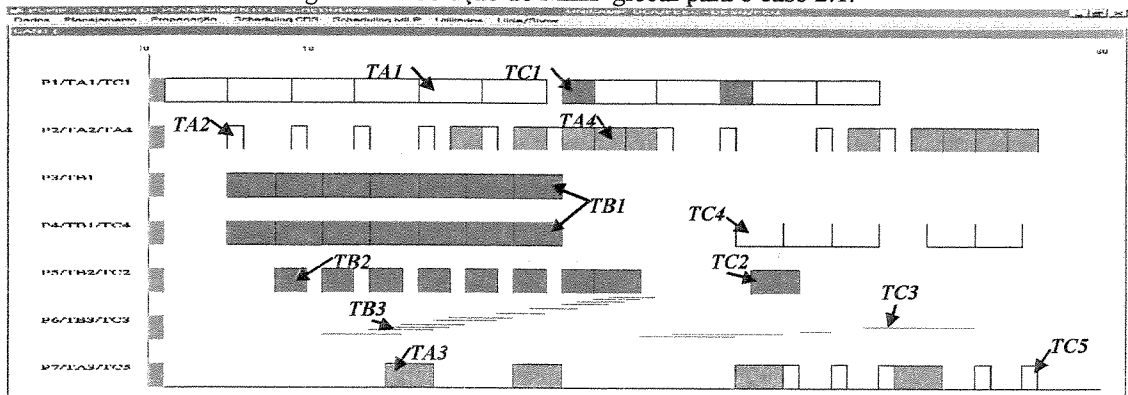


Figura A.5. Solução do primeiro passo do MILP reduzido para o caso 2.1.

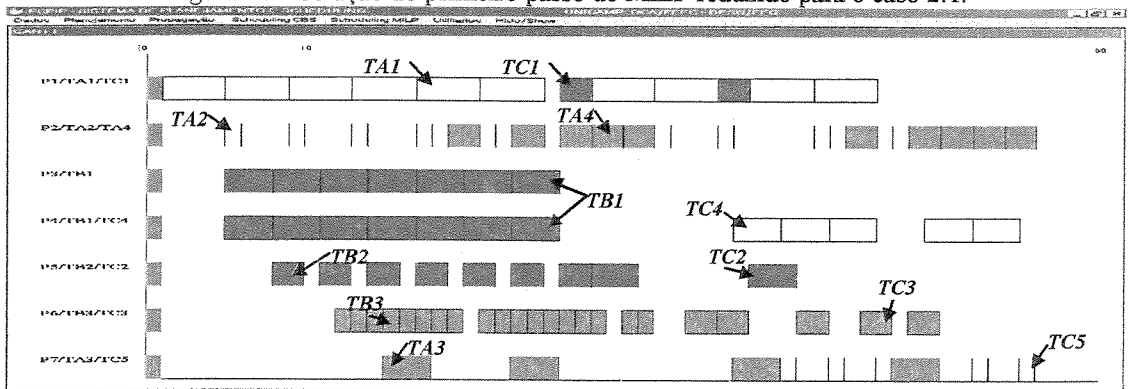


Figura A.6. Solução do segundo passo do MILP reduzido para o caso 2.1.

b) Caso 2.2

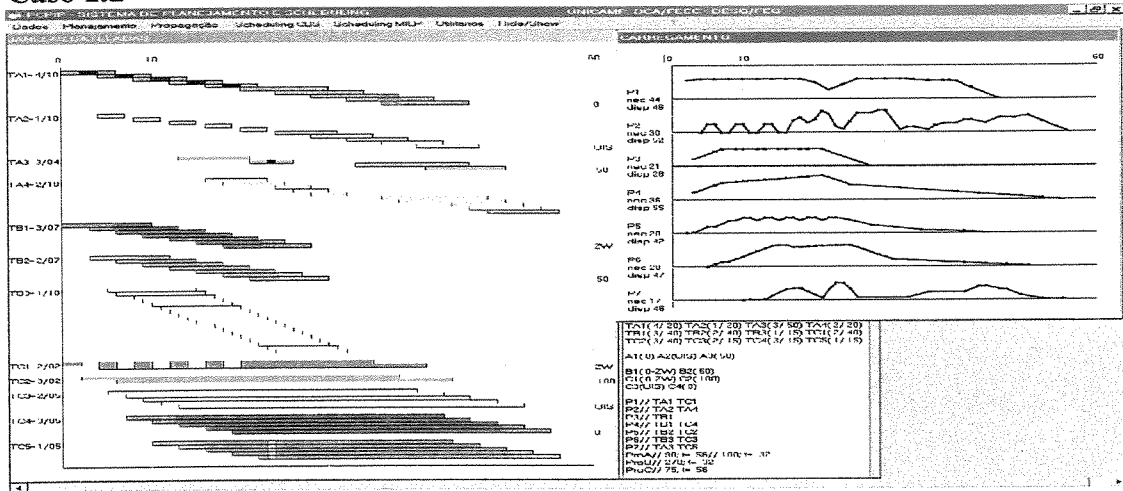


Figura A.7. Janelas de tempo e carregamento do caso 2.2.

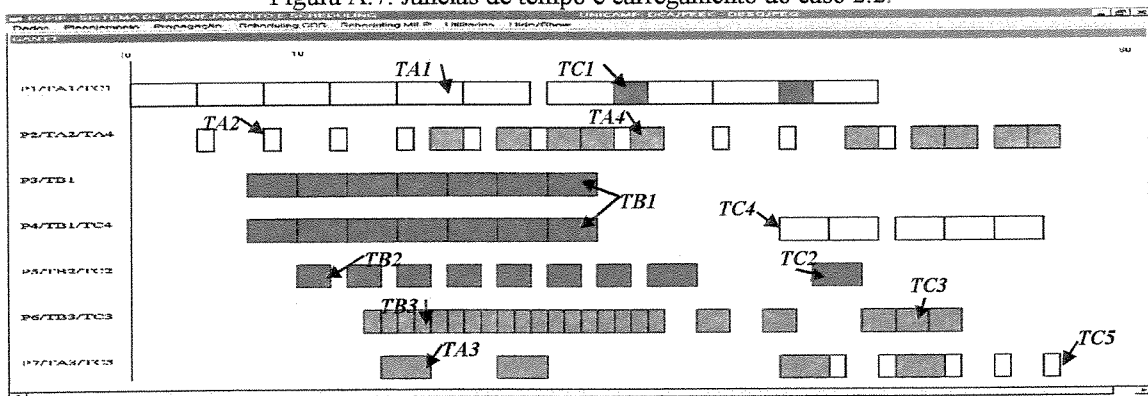


Figura A.8. Solução do MILP global para o caso 2.2.

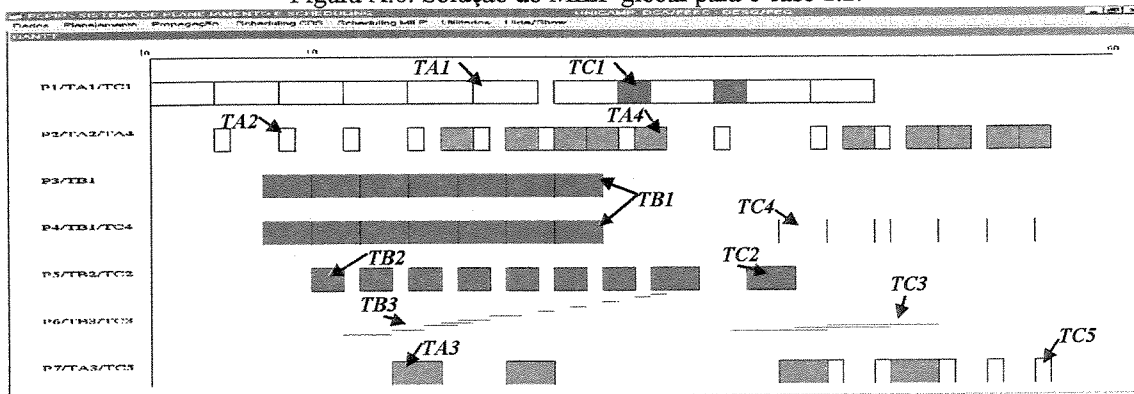


Figura A.9. Solução do primeiro passo do MILP reduzido para o caso 2.2.

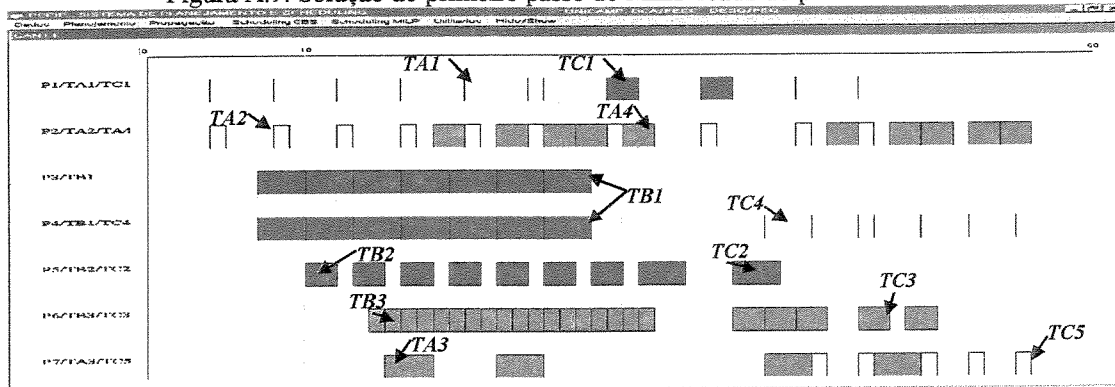


Figura A.10. Solução do segundo passo do MILP reduzido para o caso 2.2.

c) Caso 2.3

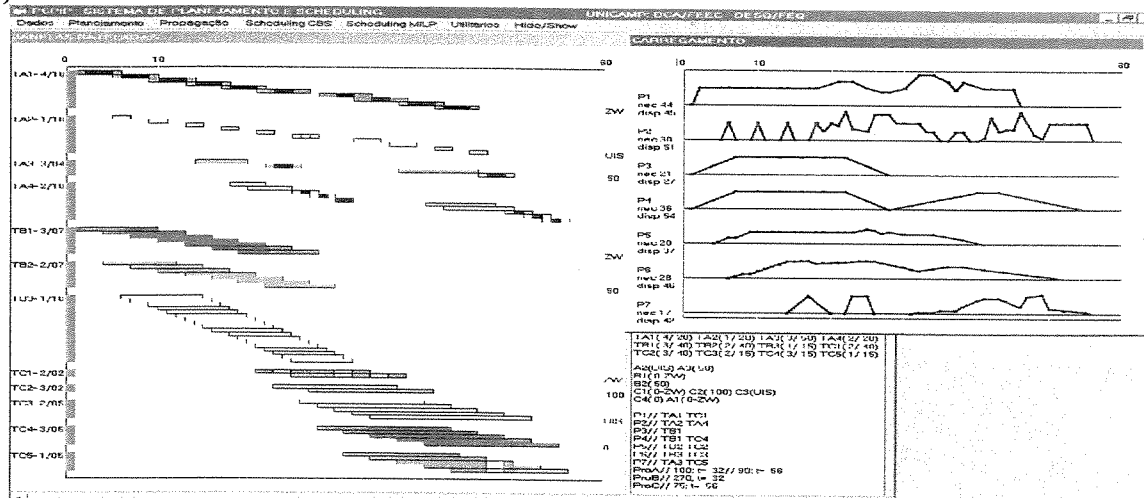


Figura A.11. Janelas de tempo e carregamento do caso 2.3.

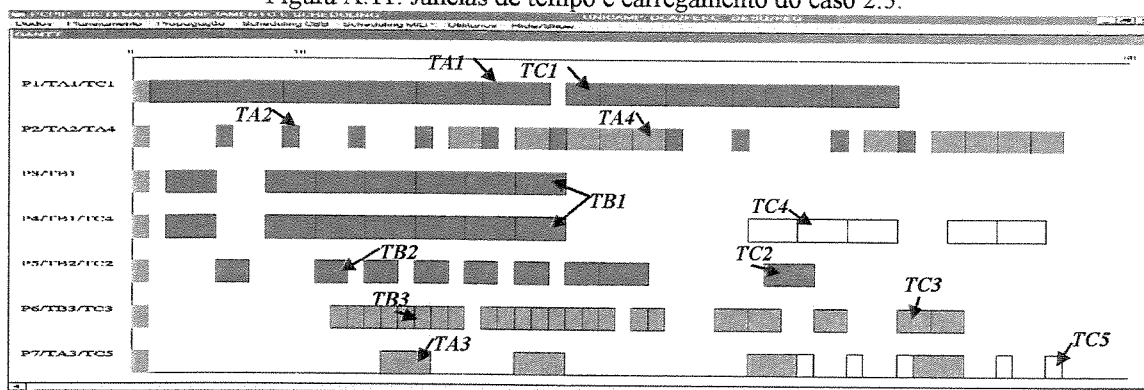


Figura A.12. Solução do MILP global para o caso 2.3.

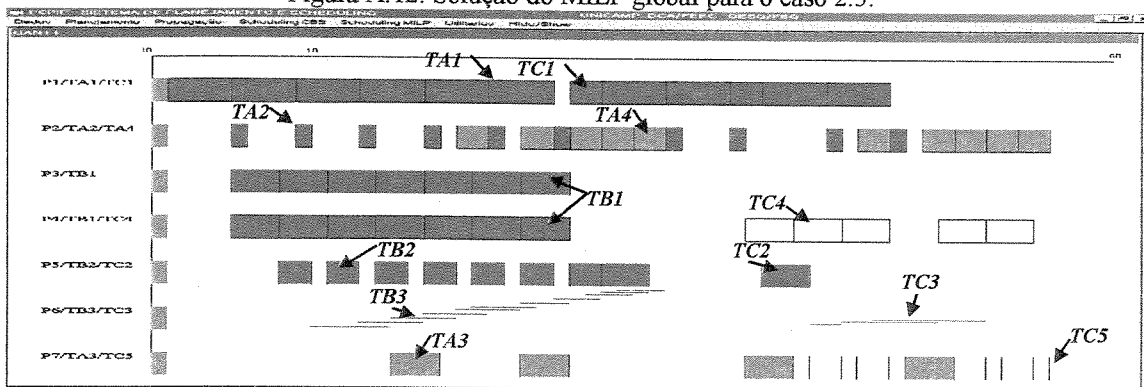


Figura A.13. Solução do primeiro passo do MILP reduzido para o caso 2.3.

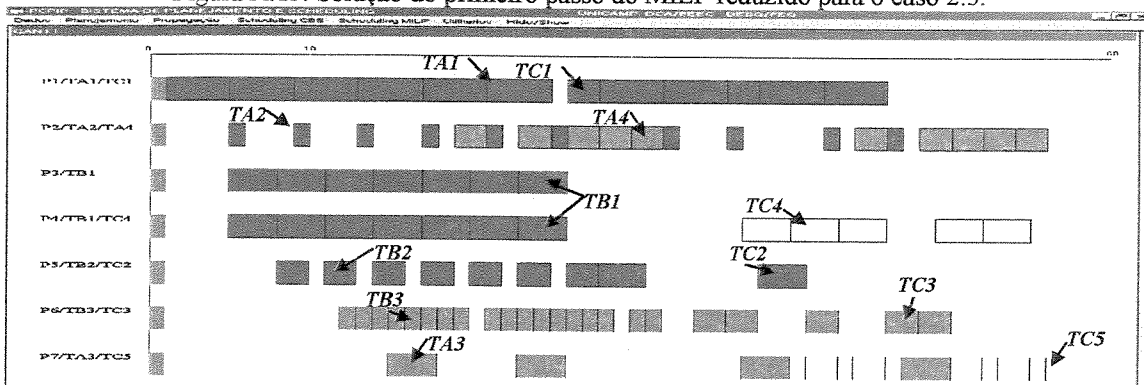


Figura A.14. Solução do segundo passo do MILP reduzido para o caso 2.3.

d) Caso 2.4

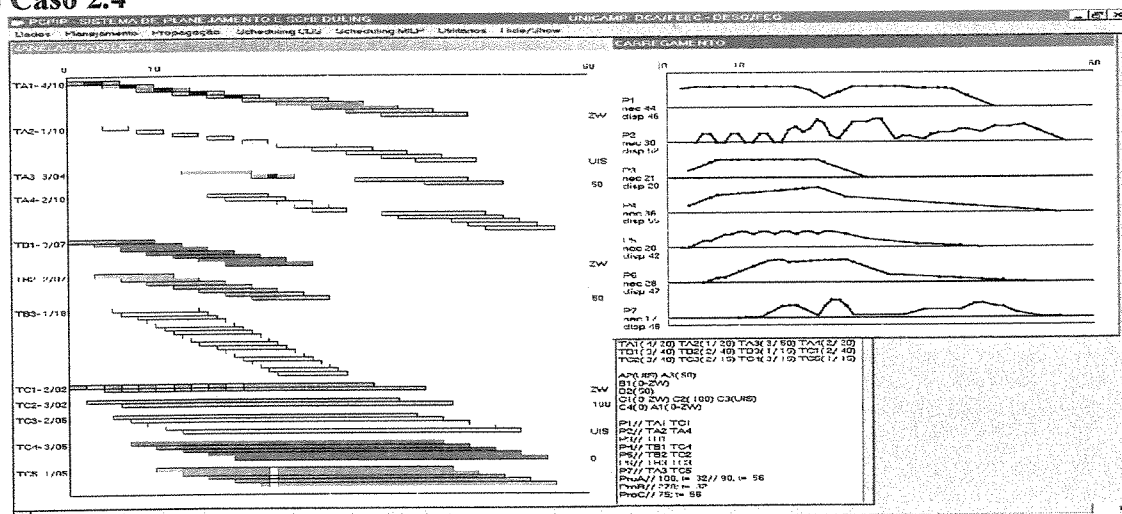


Figura A.15. Janelas de tempo e carregamento do caso 2.4.

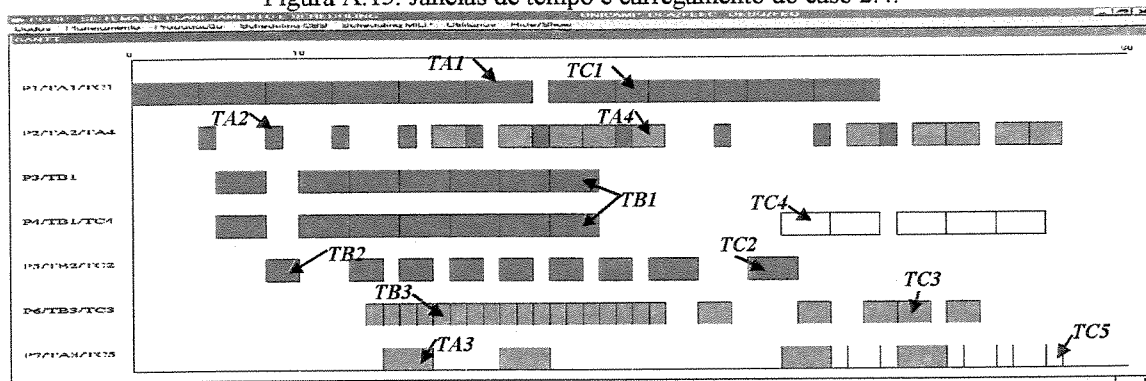


Figura A.16. Solução do MILP global para o caso 2.4.

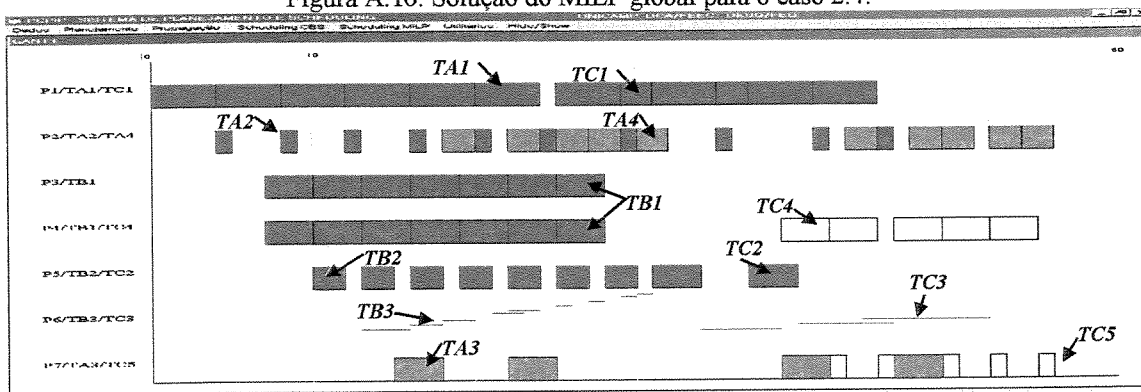


Figura A.17. Solução do primeiro passo do MILP reduzido para o caso 2.4.

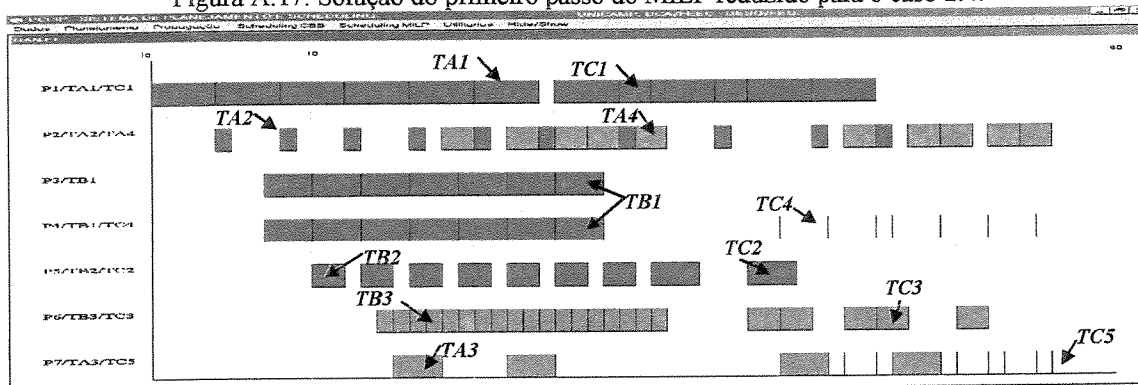


Figura A.18. Solução do segundo passo do MILP reduzido para o caso 2.4.

e) Caso 2.5

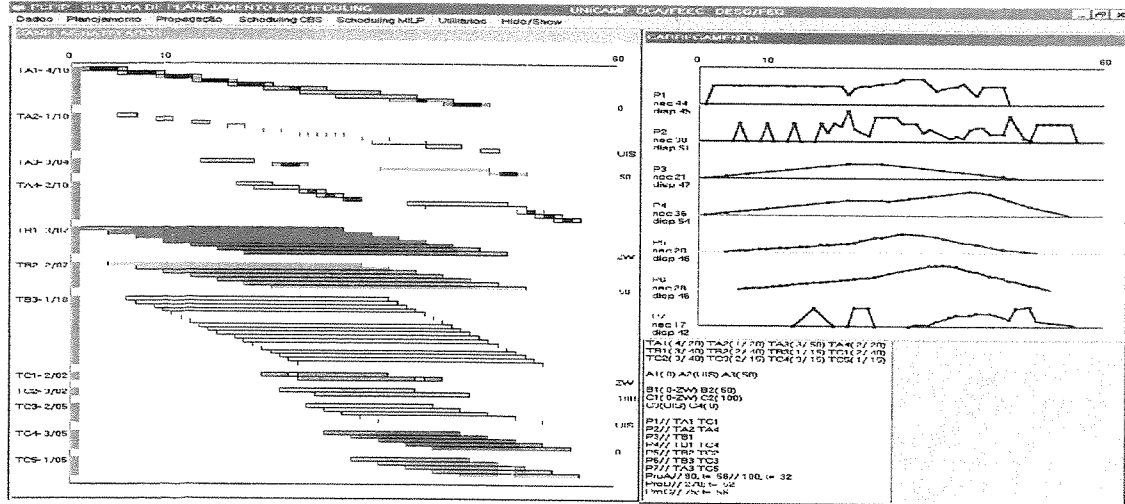


Figura A.19. Janelas de tempo e carregamento do caso 2.5.

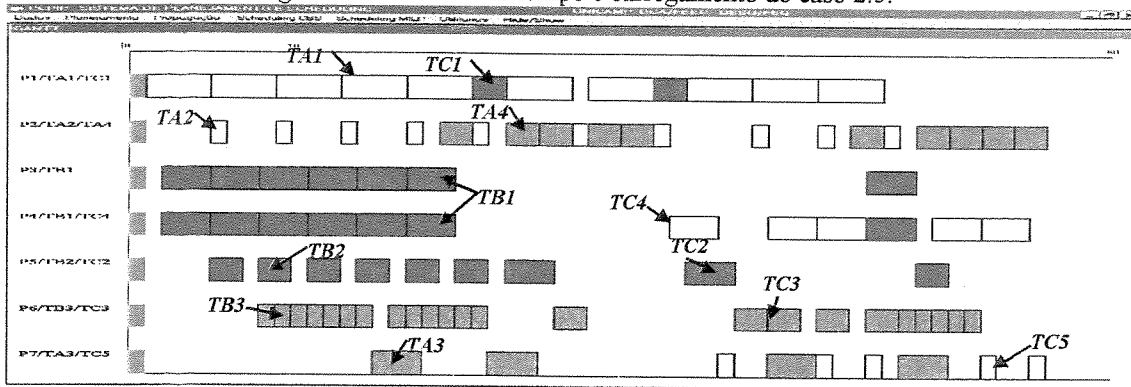


Figura A.20. Solução do MILP global para o caso 2.5.

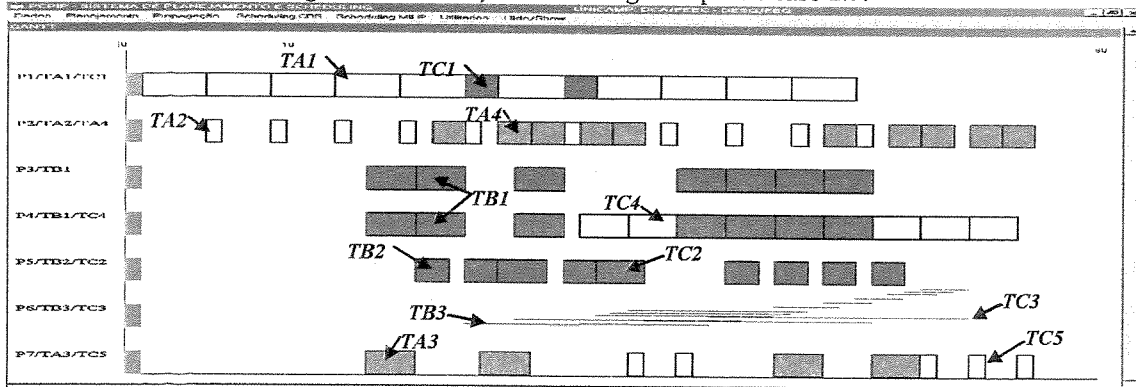


Figura A.21. Solução do primeiro passo do MILP reduzido para o caso 2.5.

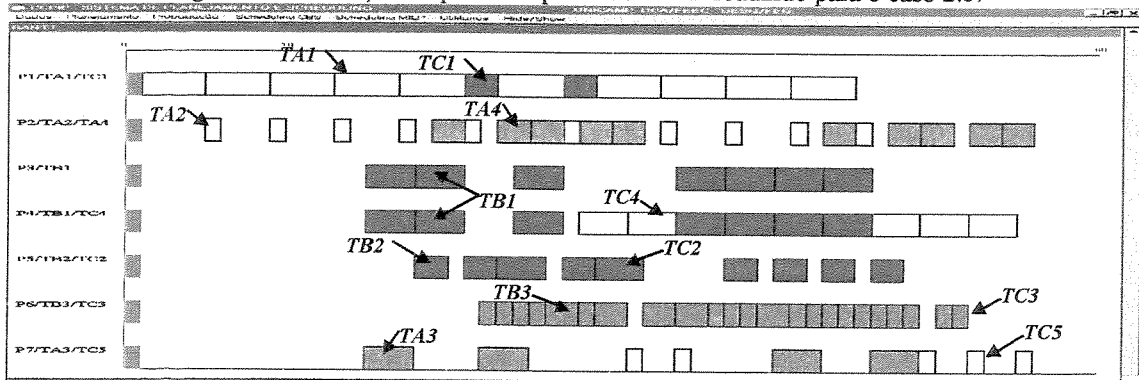


Figura A.22. Solução do segundo passo do MILP reduzido para o caso 2.5.

f) Caso 2.6

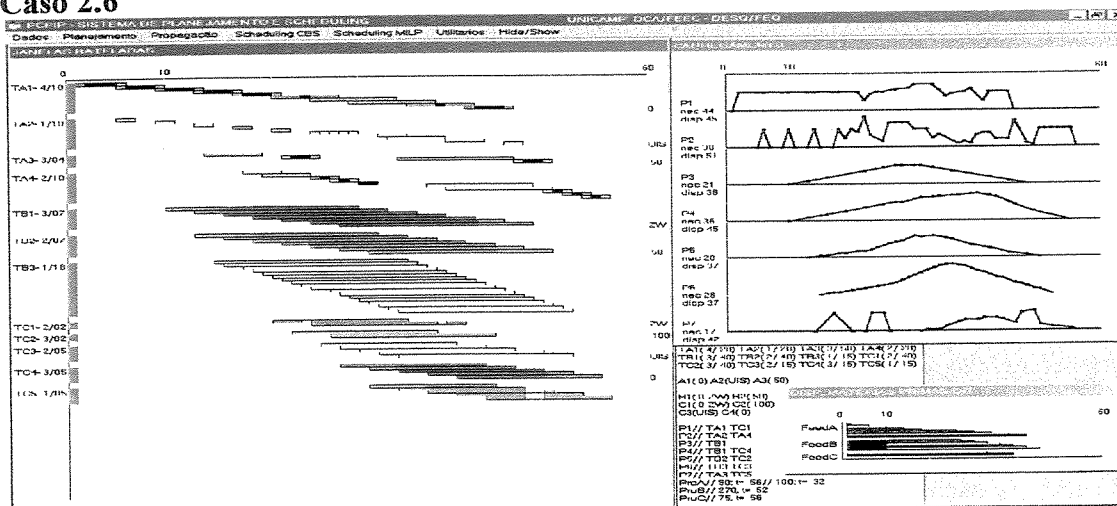


Figura A.23. Janelas de tempo e carregamento do caso 2.6.

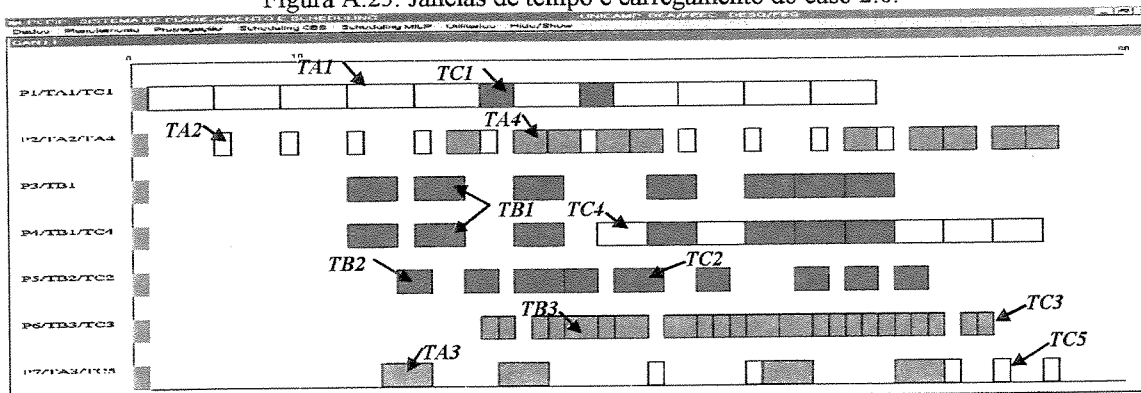


Figura A.24. Solução do MILP global para o caso 2.6.

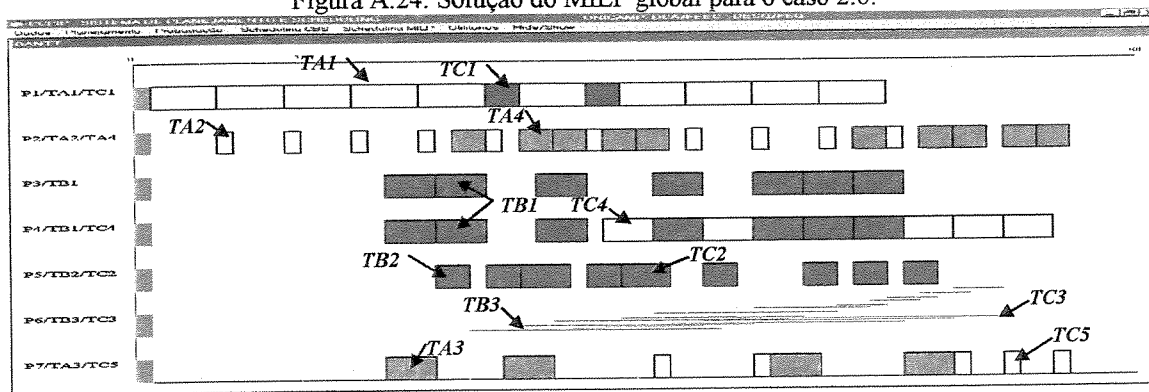


Figura A.25. Solução do primeiro passo do MILP reduzido para o caso 2.6.

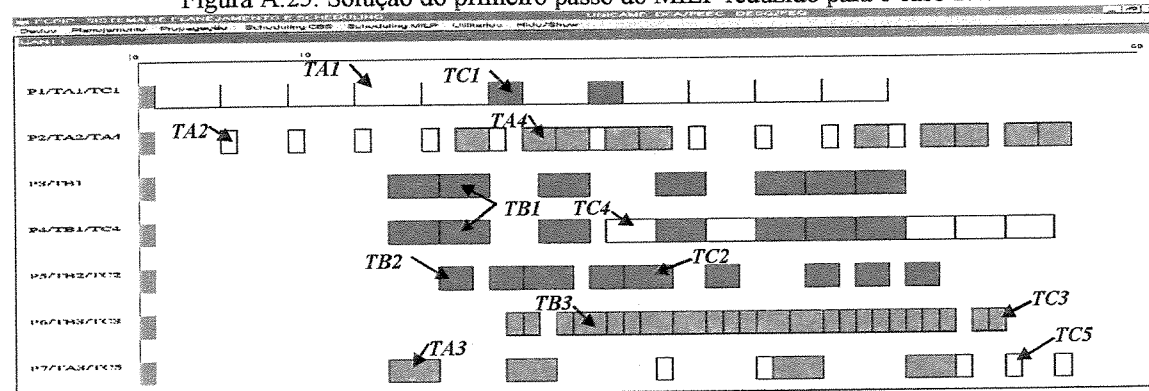


Figura A.26. Solução do segundo passo do MILP reduzido para o caso 2.6.

g) Caso 2.7

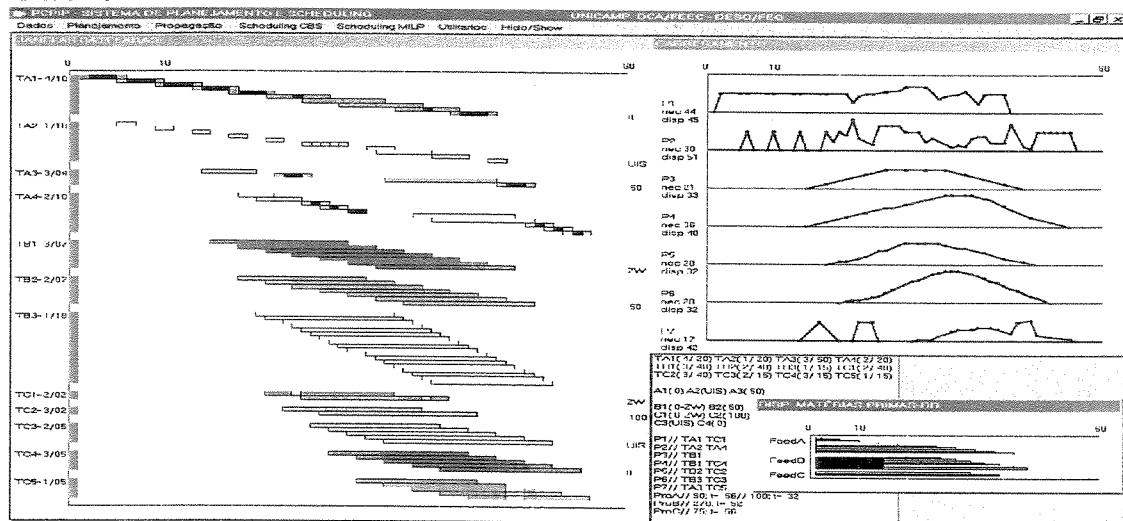


Figura A.27. Janelas de tempo e carregamento do caso 2.7.

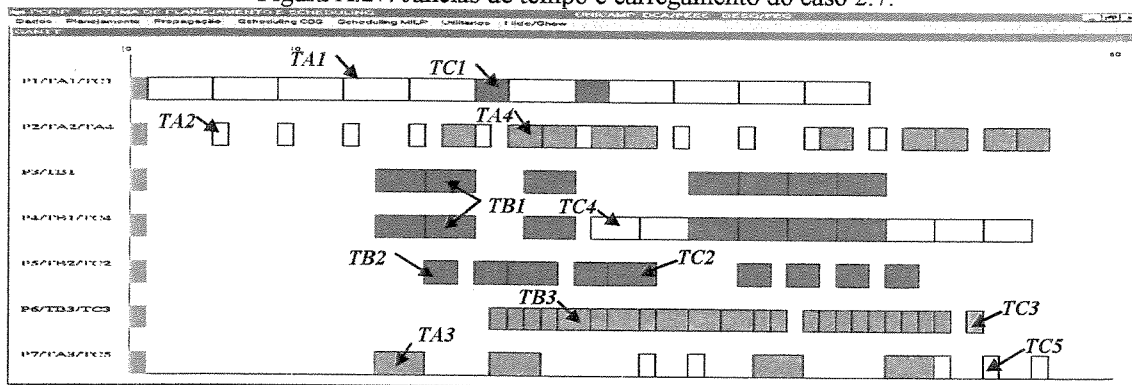


Figura A.28. Solução do MILP global para o caso 2.7.

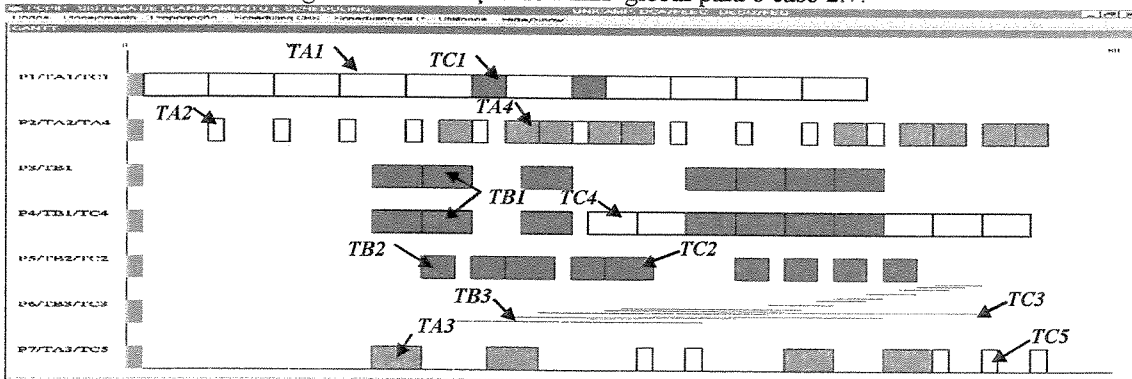


Figura A.29. Solução do primeiro passo do MILP reduzido para o caso 2.7.

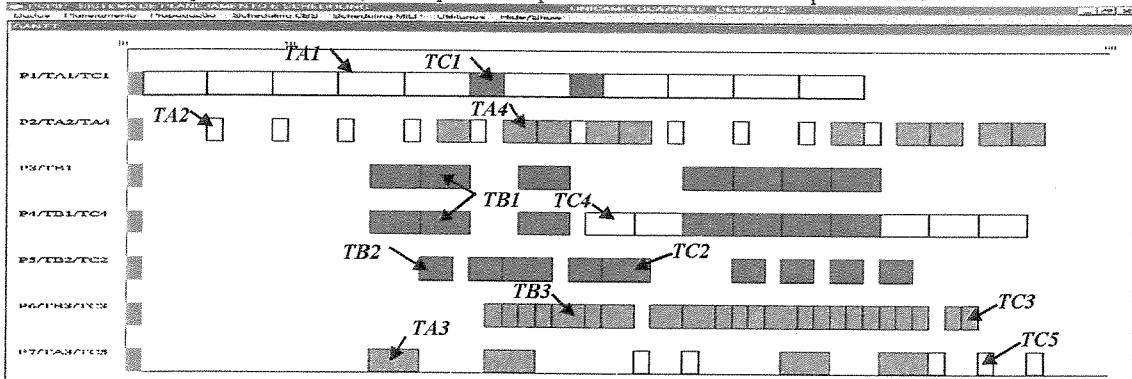


Figura A.30. Solução do segundo passo do MILP reduzido para o caso 2.7.

ANEXO 3. PARÂMETROS USADOS NO SA

Há vários procedimentos aplicados ao SA para que este possa fugir dos pontos de ótimo local. Nesta seção são apresentados alguns dos procedimentos encontrados na literatura para a definição dos parâmetros usados no SA.

Das diversas propostas para o cálculo da temperatura, a mais popular é a expressão A.1, proposta por Kirkpatrick *et al.* (1982), onde cada nova temperatura é identificada pelo índice $k+1$. Johnson *et al.* (1991), Drexl (1988), Romero *et al.* (1996) e Mamalis e Malagardis (1996) utilizam a expressão A.1 e propõem que o fator de redução da temperatura α tenha o seu valor fixado dependendo do problema entre 0,5 e 0,99.

$$T_{k+1} = \alpha * T_k \quad [A.1]$$

No cálculo do valor da temperatura para a próxima iteração (T_{k+1}), vários autores adotaram a expressão A.2 proposta por Lundy e Mees (1986). A constante β , usada na expressão A.2, é definida de tal forma que em M iterações a temperatura desça do valor da temperatura inicial (T_o) até o valor da temperatura final (T_f), como mostra a expressão A.3.

$$T_{k+1} = \frac{T_k}{(1 + \beta * T_k)} \quad [A.2]$$

$$\beta = \frac{T_o - T_f}{M * T_o * T_f} \quad [A.3]$$

Van Laarhoven *et al.* (1992) utilizam a expressão A.4 para o cálculo do valor da temperatura para a próxima iteração (T_{k+1}). Nesta expressão o fator de redução da temperatura δ é utilizado para controlar o decréscimo da temperatura a cada iteração. O parâmetro σ_k é o desvio padrão dos valores da função objetivo até a iteração k .

$$T_{k+1} = \frac{T_k}{1 + [T_k * \ln(1 + \delta) / 3\sigma_k]} \quad [A.4]$$

Battiti e Tecchiolli (1994) apresenta um outro procedimento para o cálculo do valor da temperatura da próxima iteração (T_{k+1}), que é dado pela expressão A.5. Nesta expressão, M é o número de iterações dos SA e ϕ é o parâmetro de decréscimo da temperatura a cada iteração. θ_{min} é a temperatura final após M iterações do SA. θ_{max} provém do valor da temperatura inicial que é: $T_o = \theta_{min} + \phi (\theta_{max} - \theta_{min})$. No problema tratado pelos autores, ϕ é igual a 0,1. No entanto, se ϕ for igual a 1, a temperatura inicial (T_o) passará a ser igual a θ_{max} e a expressão A.5 será um procedimento de decréscimo linear da temperatura, baixando de θ_{max} (temperatura inicial) a θ_{min} (temperatura final) em M iterações.

$$T_{k+1} = T_k - \phi (\theta_{max} - \theta_{min}) / M \quad [A.5]$$

O procedimento de gerar um número de candidatos (ξ) a uma dada temperatura (T_k) é utilizado por vários autores. Johnson *et al.* (1991) propõe que o parâmetro ξ seja definido a partir da multiplicação do número de vizinhos existentes (N) de uma solução por um fator constante (*SizeFactor*), tal que: $\xi = N * SizeFactor$.

Seja k o índice que identifica uma temperatura T_k , Drexl (1988) e Romero *et al.* (1996) definem o número de candidatos gerados (ξ_k) a uma temperatura T_k , onde o número de repetições (ξ_k) é definido por: $\xi_k = \xi_{k-1} * SizeFactor$. Por esta proposta, a medida que a temperatura (T_k) é reduzida, aumenta-se o número de repetições (ξ_k). Segundo Romero *et al.*, o número de repetições (ξ_k) deve permitir que a cada temperatura se atinja um estado de quase equilíbrio térmico, segundo a analogia com o processo de recozimento. E este quase equilíbrio térmico não depende somente de ξ_k , mas também da diferença entre duas temperaturas consecutivas ($T_{k-1} - T_k$) porque quanto maior for esta diferença entre temperaturas, maior será o número o número de repetições necessárias para se atingir o “equilíbrio térmico”.

Martin e Otto (1994) e Desai e Patil (1996) não propõem o uso da repetição na geração de candidatos (ξ_k) a cada temperatura (T_k). Em lugar disto, eles propõem que a cada temperatura (T_k) se utilize uma heurística de busca local e que se use o critério da exponencial de $(-\Delta/T_k)$ para sair dos ótimos locais, após a sua localização.