



1150064336



BAE

T/UNICAMP AL86e

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA QUÍMICA

ÁREA DE CONCENTRAÇÃO
SISTEMAS DE PROCESSOS QUÍMICOS E INFORMÁTICA

**Estudo da estratégia de busca orientada por restrições
em plantas multipropósito operando em batelada**

Autor: Wesley de Lima Alvarenga
Orientadora: Prof^a. Dr.^a Maria Teresa Moreira Rodrigues
Co-Orientador: Prof. Dr. Luis Gimeno Latre

Dissertação de Mestrado apresentada à Faculdade de Engenharia Química como parte dos requisitos exigidos para a obtenção do título de Mestre em Engenharia Química.

Campinas - São Paulo
Março de 2001

200511488

UNIDADE	FEV-2006
Nº CHAMADA	AL86e
EX	
TOMBO BC/	64336
PROC.	16-000000-00
C	<input type="checkbox"/>
D	<input checked="" type="checkbox"/>
PREÇO	11,00
DATA	13/06/05
Nº OPD	

Bibid: 353165

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

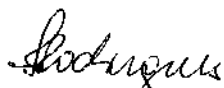
AL86e Alvarenga, Wesley de Lima
Estudo da estratégia de busca orientada por restrições em plantas multipropósito operando em batelada / Wesley de Lima Alvarenga. --Campinas, SP: [s.n.], 2001.

Orientadores: Maria Teresa Moreira Rodrigues, Luis Gimeno Latre.

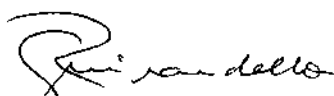
Dissertação (mestrado) - Universidade Estadual de Campinas, Faculdade de Engenharia Química.

1. Processos químicos. 2. Agenda de execução (Administração). 3. Engenharia de produção. 4. Just-in-time. I. Rodrigues, Maria Teresa Moreira. II. Latre, Luis Gimeno. III. Universidade Estadual de Campinas. Faculdade de Engenharia Química. IV. Título.

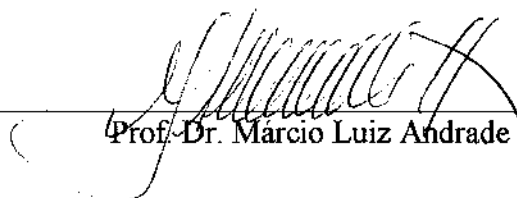
Dissertação de Mestrado defendida por Wesley de Lima Alvarenga e aprovada em 06 de abril de 2.001 pela banca examinadora constituída pelos doutores:



Prof. Dr. - Maria Teresa Moreira Rodrigues



Prof. Dr. Reginaldo Guirardello



Prof. Dr. Márcio Luiz Andrade Netto

O presente exemplar corresponde à versão final aprovada.

Rodrigues
mat 05535-2

Agradecimentos

À professora Maria Teresa Moreira Rodrigues pela oportunidade de realizar este trabalho e pela orientação concedida.

Ao professor Luis Gimeno Latre por toda a orientação e apoio durante o desenvolvimento do trabalho.

Aos meus pais, Lucacir e Laura de Lima Alvarenga por terem me apoiado e incentivado desde o começo. Obrigado por tudo.

Aos amigos, Fábio Conselheiro, Marcos Fraga, Gilsinei Campese, Luis Rodrigues e Adriana e tantos outros que por aqui passaram, obrigado pelos conselhos.

Ao CNPq pela bolsa e por ter garantido a realização deste trabalho.

E a Claudia, minha esposa, por todo apoio, incentivo e compreensão.

Resumo

O problema abordado é o de programação de produção (*scheduling*) em plantas operando em batelada. Considera-se problemas de curto prazo (*short-term*) com restrições fortes sobre a capacidade de produção da planta e de armazenagem de produtos intermediários, situação que ocorre na indústria de processos com frequência.

A abordagem proposta visa a utilização da informação sobre gargalos de capacidade dos equipamentos como forma de orientar o procedimento de busca pela solução, através da sucessiva redução do espaço de busca.

Para esse fim, foram implementadas as principais técnicas de propagação de restrições sobre recursos unitários existentes na literatura, e é proposta uma estratégia de propagação de restrições baseada nas restrições de armazenagem de produtos intermediários. Também foram propostas estratégias de identificação e avaliação das restrições do problema, de modo melhor a orientar o procedimento de busca pela solução.

A abordagem proposta identifica as restrições e avalia quais são mais duras, ou seja, mais propensas a serem violadas, e toma decisões sobre as variáveis envolvidas nessas restrições que permitam resolver o problema de programação de produção mantendo a maior liberdade possível para as decisões restantes, de forma que as restrições não sejam violadas.

Os resultados mostram um ganho de eficiência na busca por uma solução no sentido de um menor número de nós visitados na árvore de busca para atingir uma solução final, assim como menor número de nós infactíveis visitados (soluções parciais que não podem ser completadas). Esse ganho resulta em menor gasto de tempo e recursos computacionais empregados em elementos de pouca influência sobre a resolução do problema.

Abstract

The problem considered is short term scheduling of batch chemical plants. Constraints over plant capacity are taken as well as intermediate storage constraints since they are frequent in the process industries area.

The proposed approach utilizes a constraint based search technique based on bottlenecks identification to guide the search procedure and reduce the search space.

To reach this goal, the usual constraint propagation techniques over unitary resources are implemented, and it is proposed a new intermediate storage constraints propagation technique. New bottlenecks identification and measurement strategies are proposed to have a more efficient search procedure.

The search procedure identifies the hardest constraints, and takes decisions that keep the greater flexibility for the remaining variables.

The results show a big efficiency gain in the search procedure by reaching a complete solution with less visited nodes in the search tree, as well as visiting fewer infeasible nodes. The result is a reduction of computational time and resources.

Sumário

<u>CAPÍTULO 1 - INTRODUÇÃO</u>	1
1.1 - O PROBLEMA DE <i>SCHEDULING</i>	1
<u>CAPÍTULO 2 - ABORDAGEM AO PROBLEMA DE <i>SCHEDULING</i></u>	3
2.1 - INTRODUÇÃO	3
2.2 - REPRESENTAÇÃO STN	3
2.3 - TÉCNICAS DE SOLUÇÃO PARA O PROBLEMA DE <i>SCHEDULING</i> DE CURTO PRAZO	6
2.3.1 - BUSCA EM ÁRVORE	6
2.3.2 - PROGRAMAÇÃO MATEMÁTICA	8
2.3.3 - BUSCA ORIENTADA POR RESTRIÇÕES	9
2.3.4 - TÉCNICAS EVOLUTIVAS	10
2.4 - ABORDAGEM PROPOSTA	10
2.4.1 - MÓDULO DE PLANEJAMENTO	11
2.4.2 - MÓDULO DE <i>SCHEDULING</i>	13
<u>CAPÍTULO 3 - BUSCA ORIENTADA POR RESTRIÇÕES</u>	17
3.1 - INTRODUÇÃO	17
3.2 - TIPOS DE RESTRIÇÕES	20
3.3 - MEDIDAS DE GARGALOS	22
3.3.1 - CRITICALIDADE E DEMANDA INDIVIDUAL DE UMA OPERAÇÃO	22
3.3.2 - CRUCIALIDADE E DEMANDA AGREGADA DE UM EQUIPAMENTO	24
3.3.3 - TEMPO DE FOLGA	29
3.3.4 - FLEXIBILIDADE ASSOCIADA A UM PAR DE OPERAÇÕES	30
3.4 - HEURÍSTICAS PARA DECISÕES DE <i>SCHEDULING</i>	32
3.4.1 - HEURÍSTICA DE MENOR IMPACTO	32
3.4.2 - HEURÍSTICA DE SOBREVIVÊNCIA DA RESERVA	34
3.4.3 - HEURÍSTICA DE ARBITRAGEM DE PRECEDÊNCIA MAXIMIZANDO FOLGA	35
3.4.4 - HEURÍSTICA DE TENDÊNCIA DO MENOR CAMINHO	36

3.5 - MECANISMOS DE PROPAGAÇÃO DE RESTRIÇÕES	38
3.5.1 - PROPAGAÇÃO DE RESTRIÇÕES DE COMPARTILHAMENTO DE EQUIPAMENTOS	39
3.5.2 - PROPAGAÇÃO DE RESTRIÇÕES DE BALANÇO DE MASSA	47
3.5.3 - ALOCAÇÃO DE UMA BATELADA	48
3.5.4 - RESTRIÇÕES DE ARMAZENAGEM	49
3.5.5 - FUNCIONAMENTO GLOBAL DOS MECANISMOS DE PROPAGAÇÃO DE RESTRIÇÕES	52
3.6 - ABORDAGEM PROPOSTA	55
3.6.1 - CONSOLIDAÇÃO DO ESPAÇO DE BUSCA PELA IDENTIFICAÇÃO DE CICLOS	59
3.6.2 - LIMITAÇÕES DE ARMAZENAGEM	63
<u>CAPÍTULO 4 - RESULTADOS</u>	<u>64</u>
4.1 - EXEMPLOS CONSIDERADOS	64
4.2 - USO DA HEURÍSTICA PROPOSTA PARA BUSCA EM PROFUNDIDADE	66
4.2.1 - USO DA DEMANDA AGREGADA NA HEVAR	66
4.2.2 - USO DA DEMANDA AGREGADA E DA PROPAGAÇÃO DE RESTRIÇÕES DE ARMAZENAGEM	68
4.2.3 - HEVAL DE MÁXIMA REDUÇÃO DE JANELAS	71
4.2.4 - POSSIBILIDADE E INTERESSE DE OUTRAS HEVAR / HEVAL	72
4.2.5 - USO DAS LIMITAÇÕES DE ARMAZENAGEM NA HEVAR	75
4.3 - USO DAS HEURÍSTICAS PROPOSTAS PARA OBTENÇÃO DA SOLUÇÃO ÓTIMA	80
<u>CAPÍTULO 5 - CONCLUSÕES E SUGESTÕES</u>	<u>91</u>
<u>APÊNDICE A - ASPECTOS DE IMPLEMENTAÇÃO</u>	<u>94</u>
<u>REFERÊNCIAS BIBLIOGRÁFICAS</u>	<u>97</u>

Terminologia

Janela de tempo - intervalo de tempo onde uma operação pode ser alocada

JT_i - janela de tempo da operação i

JT_{ij} - janela de tempo da operação i no equipamento j

TP_i - tempo de processamento da operação i

Cr_i - criticalidade da operação i

EST_i - (*earliest starting time*) instante de início mais cedo da batelada i

EFT_i - (*earliest finishing time*) instante de término mais cedo da batelada i

LST_i - (*latest starting time*) instante de início mais tarde da batelada i

LFT_i - (*latest finishing time*) instante de término mais tarde da batelada i

CBS - (*constraint based search*) busca orientada por restrições

HEVar - heurística de escolha de variável

HEVal - heurística de escolha de valor

ZW - (*zero wait*) armazenagem do tipo sem espera

NIS - (*no intermediate storage*) armazenagem do tipo inexistente

FIS - (*finite intermediate storage*) armazenagem do tipo limitada

UIS - (*unlimited intermediate storage*) armazenagem do tipo ilimitada

MRP - Material Requirements Planning

Capítulo 1 - Introdução

1.1 - O Problema de *scheduling*

Alguns setores da indústria, como por exemplo a farmacêutica, de alimentos e de siderurgia são caracterizados pela produção de um grande número de produtos diferentes utilizando a mesma planta de produção, ou produtos de alto valor, sujeitos a problemas operacionais, como limpeza constante dos equipamentos. Produtos desse gênero possuem processamento muito complexo, necessitam de um controle de qualidade rigoroso, ou ainda possuem demandas sazonais e influenciadas pela situação de mercado. Para os setores que com essas características de produção, o processamento descontínuo em batelada é preferível ou é a única opção.

Um dos principais atrativos do processo descontínuo é sua grande flexibilidade em utilizar e compartilhar os recursos disponíveis para a produção de volumes variáveis de um grande número de diferentes produtos utilizando as mesmas instalações. Entretanto, para que se obtenha a melhor utilização de todos os recursos, de modo a tomar completa vantagem da flexibilidade oferecida pelas plantas descontínuas e para acomodar alterações no perfil de demanda, são necessárias ferramentas de planejamento sofisticadas.

A maneira com que as plantas descontínuas operam é fortemente influenciada pela natureza da demanda imposta a ela. Se existem previsões de demanda bem conhecidas, a um prazo relativamente longo, a produção é orientada para recompor os estoques no longo prazo, e pode-se estabelecer um modelo de operação cíclica. Esta forma de operação da planta é conhecida como modo “campanha”, que é uma forma de operação regular da planta com a repetição de um perfil de produção (ciclo) durante um intervalo de tempo definido (duração da campanha). Por outro lado, quando a demanda é variável, orientada por pedidos de clientes em curto prazo, dispõe-se de um menor horizonte de tempo e não é possível estabelecer um modelo de operação cíclico. Este modo de operação é conhecido como de curto prazo. Devido às rápidas mudanças de mercado, a segunda situação tende a se tornar cada vez mais freqüente.

Definido o modo de operação da planta, existem duas abordagens para o problema de produção: abordagem em nível hierárquico e único. Na abordagem de nível hierárquico o problema de produção se divide em duas etapas. A primeira é o planejamento de

produção, onde são definidas as quantidades a serem produzidas, as datas de compra de matérias-primas e entrega de produtos. A segunda é a programação ou *scheduling*, onde é definido o programa de produção, ou seja, onde e quando cada operação será realizada. Na abordagem de nível único os aspectos envolvidos no planejamento e na programação de produção são considerados simultaneamente, pois em alguns problemas estas duas etapas não são facilmente separáveis. Obviamente a primeira abordagem é subótima, mas diminui a complexidade do problema.

Neste trabalho é suposto que as quantidades a serem produzidas, as datas de entrega de produtos finais e a disponibilidade de matérias-primas são conhecidas, tratando-se de um problema de curto prazo. Assim, nesta tese é abordado o problema de programação de produção (*scheduling*) de curto prazo, que consiste em determinar o instante de início de cada operação necessária para cumprir a demanda de produtos finais, satisfazendo as restrições do problema. As restrições a serem satisfeitas são normalmente:

- receita de produção
- limitação na quantidade de equipamentos
- limitação na oferta de recursos compartilhados
- limitação na disponibilidade de armazenagem intermediária

Neste trabalho propõe-se utilizar uma medida da competição pelos equipamentos compartilhados entre as operações em cada instante de tempo, chamada crucialidade, que permite identificar os intervalos de tempo com demanda por equipamentos mais elevada. Esta informação é utilizada para identificar regiões de elevada competição, que orientam a busca pela solução.

Os exemplos abordados procuram mostrar que as técnicas de solução de problemas de *scheduling* orientadas por restrições apresentadas na literatura não consideram completamente todas as restrições do problema, e é possível diminuir o número de decisões de *scheduling* necessárias para obter uma solução ao considerar simultaneamente essas restrições, assim como inferir novos meios de eliminar gargalos na planta e dirigir as decisões de *scheduling* além dos apresentados na literatura.

Capítulo 2 - Abordagem ao problema de *scheduling*

2.1 - Introdução

Cada produto final é produzido através de um conjunto de operações como por exemplo aquecimento, reação e separação. O problema de *scheduling* consiste basicamente em definir em que instante de tempo será iniciada cada operação da receita de produção, denominado “alocação” da operação. A alocação da operação determina em que intervalo de tempo os recursos por ela utilizados, tais como equipamentos e energia, estarão comprometidos. A decisão de quando e onde será realizada uma operação certamente possui algum efeito ou consequência sobre uma ou mais das demais operações, em virtude do balanço de massa ou compartilhamento dos recursos.

É necessário então que o problema de programação de produção seja modelado de forma a considerar estas relações de balanço de massa e compartilhamento, existentes entre as diversas operações que compõem a receita de produção. Uma consideração primária a ser feita na modelagem do problema é justamente a representação da receita de produção ou estrutura de processamento. Uma vez representada a estrutura de processamento e de posse de dados relevantes ao problema, como quantidade a ser produzida, prazos de entrega e critérios de otimização, o problema pode ser resolvido através de diversas abordagens, como programação matemática, busca em árvore ou busca orientada por restrições.

A representação usada para a estrutura de processamento é mostrada na seção a seguir. As principais abordagens ao problema de *scheduling* são brevemente apresentadas nas seções 2.3.1 a 2.3.4. O trabalho desenvolvido nesta tese está inserido em um projeto mais amplo que envolve planejamento e *scheduling*, como descrito na seção 2.4. Esse projeto vem sendo desenvolvido pelo grupo de pesquisa do qual o autor faz parte, e engloba outras técnicas de para resolução do problema de *scheduling*.

2.2 - Representação STN

A representação da estrutura de processamento é um elemento essencial do modelamento do problema. Desenvolver um modelo do problema de programação de produção para cada estrutura de processamento seria muito trabalhoso e, além disso, modificações na estrutura exigiriam alterações no modelo do problema.

As estruturas presentes na indústria de processos são complexas, com produtos intermediários consumidos e gerados por várias operações. O tamanho de batelada das diferentes operações na receita de um produto pode ser variável, exigindo que o número de bateladas de cada operação a ser realizada seja variável ao longo da estrutura de processamento. De forma a permitir o modelamento de uma variedade ampla de estruturas de processamento, será usada a representação Rede Estado-Tarefa (*State Task Network*, STN) introduzida por Kondili (Kondili E., Pantelides C. C., Sargent R. W. H. 1993).

Os processos em batelada são usualmente representados em termos da rota de produção. A rota de produção de cada produto é semelhante ao de plantas de processamento contínuo, do ponto de vista da representação do processo. Assim, os processos em batelada são também representados por diagramas que descrevem a rota de produção. Embora a representação por diagramas seja adequada para estruturas de processamento contínuo, o uso em estruturas mais complexas pode provocar ambigüidades na interpretação.

A figura 2.1 mostra uma estrutura de processamento com cinco operações e o fluxo de matéria entre as operações é indicado por setas. Essa representação não deixa claro se a operação 1 produz dois produtos diferentes que constituem a alimentação das operações 2 e 3 ou se há apenas um produto que é dividido entre as duas operações. Da mesma forma, não é claro se a operação 4 requer dois tipos diferentes de alimentação provenientes de 2 e 5 ou se as correntes são do mesmo tipo com um reciclo de 5.

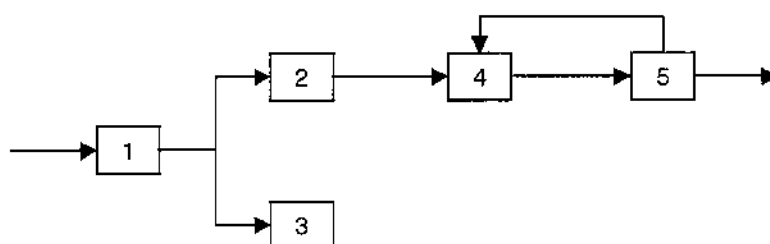


Figura 2.1 - Estrutura de processamento

A representação rede estado-tarefa foi proposta para eliminar interpretações ambíguas. Essa representação possui dois tipos de elementos; os estados, ou seja, matérias-primas, produtos intermediários e produtos finais, representados por círculos; e as tarefas, ou seja, as operações realizadas ao longo do processo, representadas por retângulos.

A figura 2.2 mostra duas representações STN diferentes para o processo da figura 2.1, onde cada um constitui um processo distinto do outro. No primeiro processo a operação 1 gera um único estado intermediário que é dividido entre as operações 2 e 3, e a operação 4 consome um único estado gerado pelas operações 2 e 5. No segundo processo a operação 1 gera dois estados intermediários diferentes para as operações 2 e 3, e a operação 4 consome dois estados diferentes gerados pelas operações 2 e 5, respectivamente.

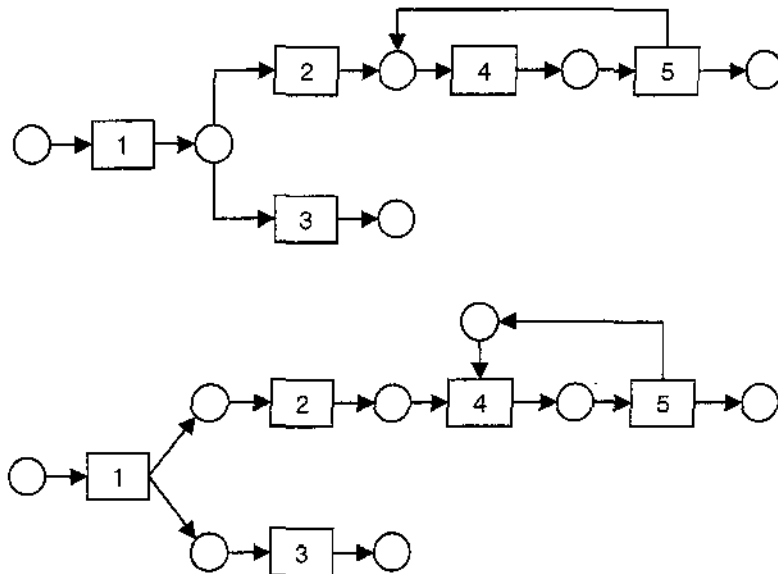


Figura 2.2 - Representações STN do processo da figura 2.1

As regras para construção da representação STN de um processo são:

a) uma operação tem tantos estados de entrada (saída) quantos forem os diferentes materiais consumidos (produzidos).

b) duas ou mais correntes entrando no mesmo estado são necessariamente do mesmo material. Se a mistura de diferentes correntes é envolvida no processo, então esse processo deve constituir uma operação separada.

2.3 - Técnicas de solução para o problema de *scheduling* de curto prazo

A solução do problema de *scheduling* requer a definição do instante de tempo em que uma determinada operação é iniciada e que recursos ela utiliza (equipamento, mão-de-obra, utilidades, etc.). Assim, o problema de *scheduling* pode ser definido como a alocação de N operações a M equipamentos. Isso impõe uma natureza combinatorial ao problema, pois dependendo do problema, podem existir $(N!)^M$ soluções possíveis. Esse tipo de problema é NP-completo (*non-polynomial complete*). Em problemas NP-completos, o tempo de solução cresce exponencialmente com a dimensão do problema. Como consequência, para problemas de dimensão industrial, é necessário recorrer a estratégias que permitam obter uma solução razoável em um prazo de tempo aceitável. Existem diferentes abordagens para resolução do problema de *scheduling* apresentadas na literatura. A seguir são descritas as principais abordagens.

2.3.1 - Busca em árvore

Devido à natureza combinatorial do problema de *scheduling* é inviável realizar uma busca indiscriminada através de todas as soluções existentes. A alternativa é recorrer a métodos de busca controlada que possam encontrar uma solução de maneira mais eficiente. As técnicas de busca em árvore (*Branch and Bound*, BAB) realizam uma busca parcial e controlada das soluções possíveis do problema. As atividades de busca e controle podem ser realizadas por meio de diferentes tipos de critérios e pela incorporação de informações específicas de cada problema, além da avaliação do critério de otimização empregado. Dentro dessa abordagem, deve-se definir qual o mecanismo de busca e quais são as variáveis de decisão.

A solução do problema de *scheduling* evolui conforme vão sendo alocadas as operações. Até que todas as operações estejam alocadas, o que se possui são soluções parciais do problema. Definido o critério de otimização, essas soluções parciais podem ter calculado um custo a elas relacionado. Para efeito de ilustração, é considerado que o critério de otimização visa a minimização da função de custo.

O BAB é uma estrutura em forma de árvore, constituída de nós, que representam soluções parciais do problema, e de ramos, que unem os nós de níveis sucessivos. Durante o processo de busca obtêm-se uma série de nós abertos (as soluções parciais) e um custo

associado a cada nó. O custo associado a cada nó aberto é chamado de limitante inferior (*lower bound*), e é composto por duas parcelas:

- a) o custo da solução parcial obtida até esse nó
- b) a estimativa do custo mínimo para se completar a solução

O menor custo associado a um nó indica o melhor caminho a ser seguido na árvore, auxiliando a escolha do melhor caminho para a solução ótima e descartando, ao menos temporariamente, soluções menos promissoras. Assim, a alocação das operações restantes segue até que seja obtida uma solução completa. O custo associado a essa solução completa constitui um limitante superior (*upper bound*), ou seja, após obtida essa solução, somente interessa seguir outras soluções parciais que possuam custo inferior.

Uma vez encontrada a solução completa, o método retrocede a nós anteriores, (operação conhecida como *backtracking*), e procura por soluções parciais de custo inferior ao limitante superior, repetindo o processo de alocação de operações até que seja encontrada uma solução completa. A figura abaixo ilustra o uso do método para o seqüenciamento de quatro operações.

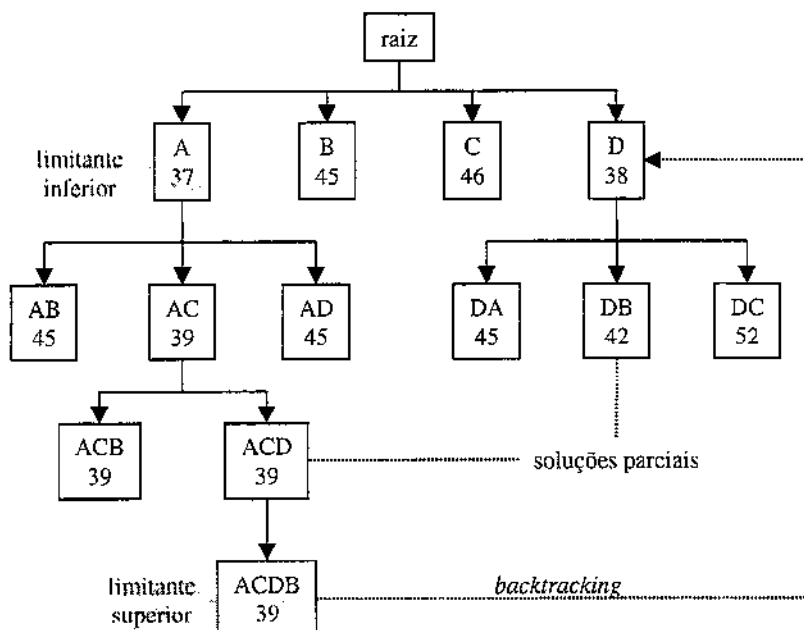


Figura 2.3 - Árvore de busca

2.3.2 - Programação Matemática

Um modo de representar o problema de *scheduling* é através de uma formulação matemática. Neste tipo de representação todas as restrições do processo, como as restrições de ocupação dos equipamentos, de armazenagem e de balanço de massa originada pela receita são representadas por equações. Pode-se então utilizar ferramentas como a programação inteira mista (*mixed integer linear programming*, MILP), amplamente difundida na literatura, para resolver o problema de *scheduling* (Kondili E., Pantelides C. C., Sargent R. W. H. 1993). A formulação MILP utiliza variáveis binárias para determinar o início do processamento e o equipamento utilizado para cada operação, juntamente com variáveis contínuas que representam outras grandezas, como massa e tamanho de batelada.

Formulações MILP são resolvidas através da utilização de um algoritmo de programação linear combinado com uma técnica de busca em árvore. As restrições do problema são relaxadas de modo a permitir que as variáveis binárias assumam valores reais entre 0 e 1. O problema relaxado é resolvido pelo algoritmo de programação linear (PL), e o valor da função de custo obtido é utilizado como limitante inferior, assim como descrito na técnica BAB. São gerados dois subproblemas, no qual o problema original agora possui o valor de uma das variáveis binárias fixada em um valor inteiro igual a 0 ou 1, e cada subproblema é resolvido pelo algoritmo PL. O procedimento segue pelo nó com menor custo (no caso de problemas de minimização) até que todas as variáveis binárias tenham seu valor fixado em 0 ou 1.

Um aspecto da eficiência das formulações MILP diz respeito à diferença entre o custo obtido com a solução relaxada e o custo da solução inteira ótima, denominado *gap* de integralidade. Quando esse *gap* possui valor elevado diversas soluções parciais serão perseguidas resultando em mais nós abertos na árvore de busca e conseqüentemente maior tempo de cálculo. Obviamente esse *gap* de integralidade não é conhecido *a priori*.

Outro aspecto importante na formulação do problema de *scheduling* é a representação do tempo. Algumas formulações utilizam uma representação discreta do tempo definida *a priori*. O horizonte de tempo disponível 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 tempos de processamento das operações. A vantagem dessa

representação é que facilita a formulação por criar uma grade de referência na qual as operações que competem pelo mesmo recurso são posicionadas. Eventos de qualquer tipo (início ou fim de processamento, ocupação de um equipamento, etc.) são permitidos apenas nos extremos desses *slots*. A desvantagem é que a discretização deve ser suficientemente fina para representar todos os eventos de interesse, o que pode resultar em um número extremamente grande de variáveis binárias. Como a dimensão da árvore de busca é função do número de variáveis binárias, para problemas de proporções médias ou grandes pode ser inviável obter uma solução final em tempo aceitável.

Outras formulações usam uma representação contínua do tempo, de forma a reduzir a dimensão do problema. O horizonte de tempo é dividido em intervalos de duração variável, chamados eventos. As durações desses eventos são variáveis do problema. As operações são associadas a um evento, ao invés de definidas em relação ao tempo. O início do processamento de uma operação só pode ocorrer no início do evento a ela associado. O número de eventos é em geral muito menor que o número de intervalos decorrente da discretização do tempo, diminuindo a dimensão do problema pela redução de variáveis binárias, porém a formulação do problema torna-se não-linear. É possível linearizar a formulação, mas isso envolve o uso de dicotomias com constantes grandes, o que aumenta o *gap* de integralidade do problema.

2.3.3 - Busca orientada por restrições

Essa abordagem é adequada para problemas com restrições fortes, como por exemplo janelas de processamento restritas e equipamentos muito utilizados. Sua principal característica é que o processo de busca é orientado pelos gargalos do processo, e não pela função de custo. Em situações muito restritas parece lógico que o algoritmo se concentre primeiro nas “regiões” onde as restrições são mais duras, com a hipótese de que solucionados esses gargalos o restante do problema não apresentará dificuldades.

A ferramenta fundamental dessa técnica é o procedimento de propagação de restrições, que a cada nó da árvore de busca realiza a propagação dos efeitos da decisão de *scheduling* que gerou o nó. Em problemas muito restritos esta propagação pode reduzir consideravelmente o número de decisões remanescentes reduzindo portanto a árvore de busca. Considerações a respeito da função de custo são feitas após obtida uma solução

completa. O valor limitante superior é adicionado como uma restrição ao procedimento de busca, da mesma forma que nas técnicas descritas anteriormente.

2.3.4 - Técnicas evolutivas

Técnicas como *simulated annealing* estão baseadas na obtenção de uma solução inicial e na melhora sucessiva dessa solução. Essas técnicas podem sair de mínimos locais permitindo que sejam admitidas soluções piores do que a atual, dentro de um nível pré-definido, com o objetivo de obter uma posterior melhora da solução. A vantagem dessas técnicas é a disponibilidade contínua de uma solução.

No processo de montagem da solução são escolhidos dois candidatos (operações) aleatoriamente, e definido um ordenamento entre eles, para avaliar a função objetivo. Caso haja uma melhora da função, a solução parcial (ordenamento) é aceita, e o procedimento segue até que estejam definidos todos os ordenamentos. Deste modo, os candidatos estão definidos em termos de ordem de precedência entre eles, não em termos de tempo. A principal razão para isso é ter uma representação de dimensão razoável. Assim, o processo de construção das sucessivas soluções é rápido, porém a avaliação do custo associado a cada solução somente é rápido para algumas funções objetivo, como a minimização de *makespan* (tempo total de processamento). Caso o problema seja muito restrito, por exemplo em termos de janelas de processamento, condições de armazenagem ou equipamentos muito utilizados, a quantidade de candidatos ineficazes gerados pode ser extremamente grande, diminuindo a eficiência da técnica.

2.4 - Abordagem proposta

A abordagem proposta neste trabalho é uma técnica de busca orientada por restrições. Essas técnicas vem sendo utilizadas desde 1.976 (Erschler J. 1976, Fox M. S. 1983, Keng N. P., Yun D. Y. Y., Rossi M. 1988, Sadeh N. 1991, ILOG 1997, Claire *Schedule* 1.0 1999) e tem mostrado bom desempenho em problemas de *scheduling* com restrições duras.

Todos os trabalhos nessa área apresentados na literatura partem da definição do problema de *scheduling* como sendo o de alocação de um número pré-determinado de bateladas de operações dentro das respectivas janelas de tempo para processamento.

Como geralmente os equipamentos de plantas descontínuas possuem pequena capacidade, as operações devem ser repetidas para produzir a quantidade de massa necessária, ou seja, deve ser feita mais de uma batelada para satisfazer completamente a demanda. Portanto, cada operação é constituída por um conjunto de bateladas e o conjunto de todas as operações e suas bateladas formam por assim dizer o “cenário” de produção.

As janelas de tempo representam o intervalo de tempo disponível para execução de uma batelada. Essas janelas estão limitadas pelo menor instante de início (*earliest starting time*, EST) e o maior instante de término (*latest finishing time*, LFT) que uma operação pode assumir quando alocada.

Nos sistemas de *scheduling* comerciais, as janelas de processamento das bateladas são dados de entrada, ou então obtidas numa fase inicial, a partir das janelas de processamento das bateladas de produtos finais. Essas janelas de produtos finais são também dados de entrada, na forma de datas de entrega e data de início mais cedo possível para cada batelada de produto. Assim, para utilização destes sistemas, as janelas de tempo devem ser definidas por um sistema de planejamento à parte, e este aspecto tem sido uma crítica recorrente à utilização destes sistemas (Shah N., Chung P. W. H., Das B. P. 1998).

A abordagem proposta utiliza o sistema de planejamento desenvolvido pelo grupo de pesquisa para fornecer as janelas de processamento das bateladas. Esse sistema está incorporado ao sistema de *scheduling*, de modo que não é necessário o uso de outro software. O sistema é descrito sucintamente a seguir.

2.4.1 - Módulo de planejamento

As janelas de tempo são determinadas na etapa de planejamento, a partir dos seguintes dados:

- Demanda dos produtos
- Datas de entrega dos produtos
- Em que equipamento será realizada cada operação (*assignment*)
- Plano de compras

Fornecido o *assignment*, ficam fixados o tamanho de batelada e o tempo de processamento da operação, e assim é possível calcular o número de bateladas de cada operação necessário para satisfazer os requisitos de demanda. Através de um processo de

explosão semelhante ao utilizado no MRP (Material Requirements Planning), são definidos os fins das janelas das operações. A partir dessa informação, o sistema calcula as datas-limite para compra de matérias-primas. A partir do plano de compras de matérias-primas definido pelo usuário, são calculados os inícios das janelas, através de balanço de massa. A estrutura do sistema de planejamento é ilustrada a seguir.

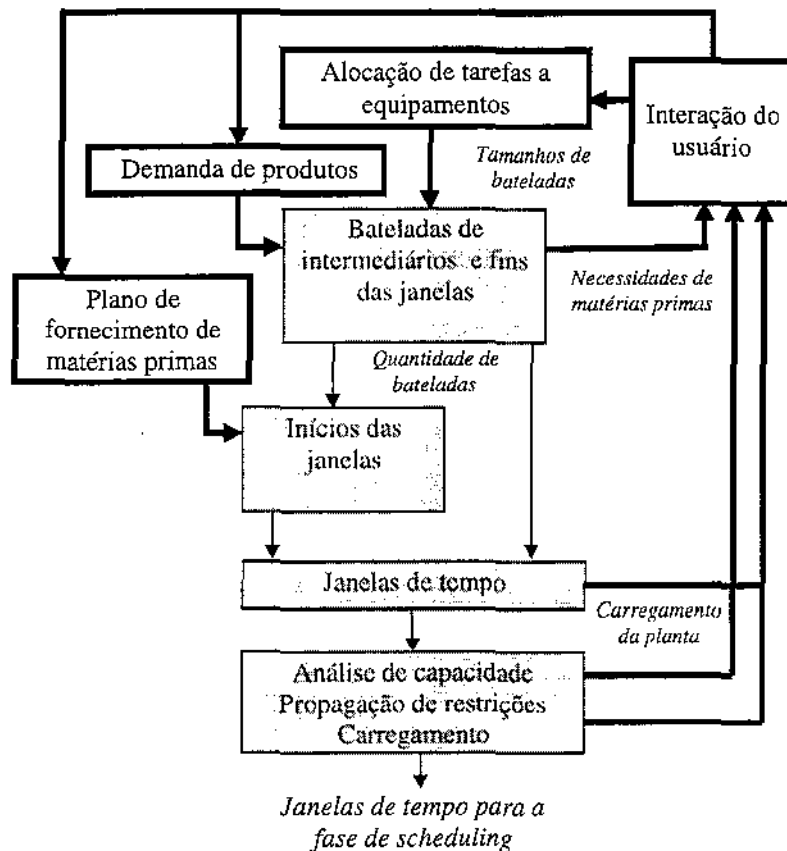


Figura 2.4 – Estrutura do sistema de planejamento

A figura 2.5 mostra o resultado fornecido pelo módulo de planejamento. No painel esquerdo são mostradas as necessidades de matérias-primas, representadas por barras brancas, sendo as regiões em cinza o plano de compras fornecido pelo usuário. O painel da direita exibe as janelas de estados resultantes do planejamento. Os estados são produtos finais ou intermediários produzidos por uma operação e consumidos por outras. No exemplo mostrado, a matéria-prima C foi lançada no instante 1, a B no instante 3 e a A dividida em dois lançamentos, nos instantes 1 e 10, respectivamente.

O sistema de planejamento transforma as janelas dos estados em janelas de tempo das operações, e realiza a análise do compartilhamento dos equipamentos, que verifica se as bateiadas a serem executadas em cada equipamento não ultrapassam a capacidade do

mesmo, permitindo que a etapa de *scheduling* não seja executada caso o problema seja infactível. Se a capacidade da planta estiver subutilizada, ou se as janelas resultantes não forem satisfatórias, o usuário pode redefinir o plano de compras ou alterar a data de entrega de algum produto. É possível também definir um novo *assignment*, o que pode mudar completamente o problema se o tamanho de batelada for modificado. Essa interação com o usuário permite que sejam criados cenários de busca diferentes para uma avaliação prévia. Após o cenário ser aprovado, está finalizada a etapa de planejamento, que passa as janelas resultantes ao módulo de *scheduling*.

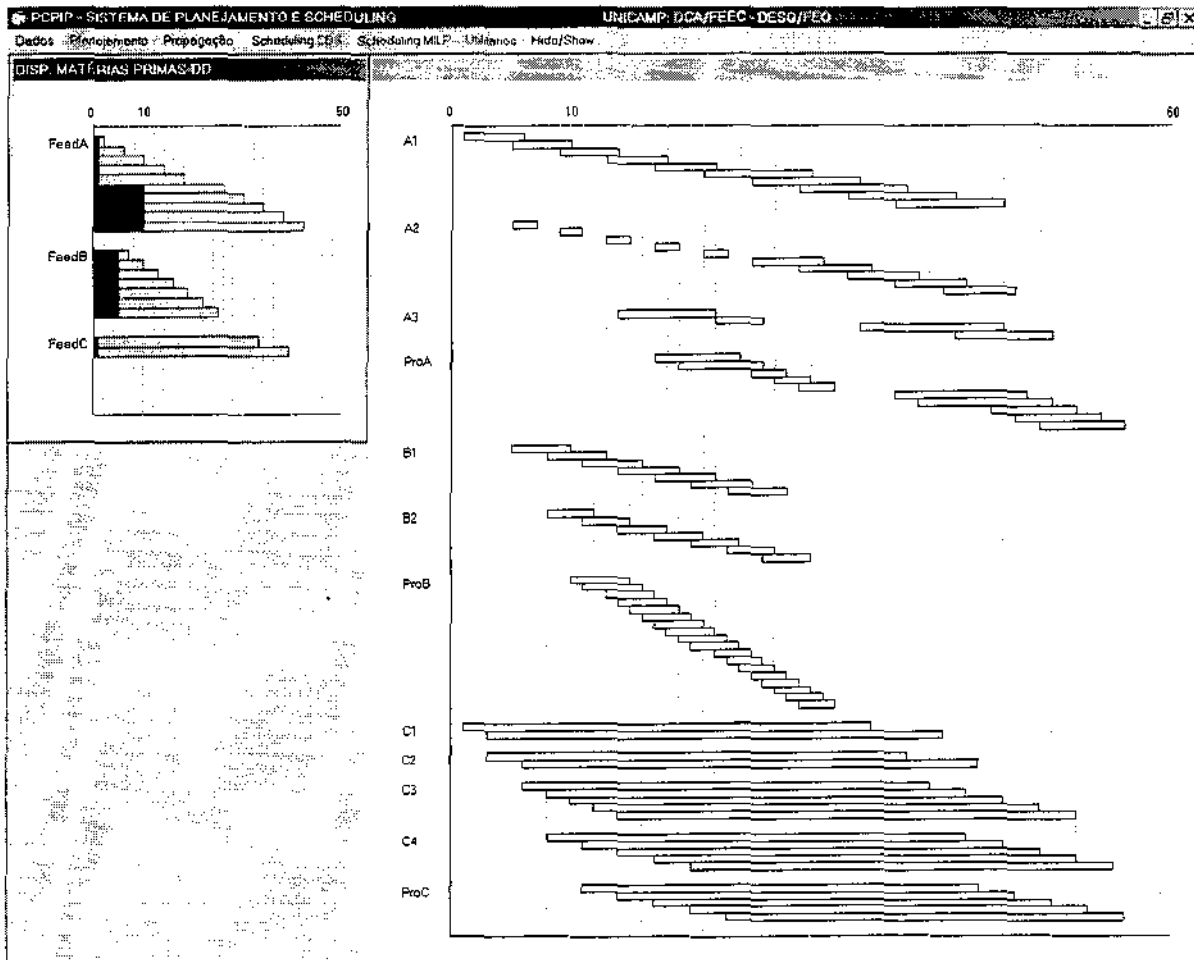


Figura 2.5 – Módulo de planejamento

2.4.2 - Módulo de *scheduling*

No módulo de *scheduling* são usadas as janelas fornecidas pelo módulo de planejamento, ilustradas na figura 2.6. Janelas da mesma cor mostram operações que utilizam o mesmo equipamento. Os espaços em preto representam intervalos que serão ocupados obrigatoriamente (também denominados intervalos de *total reliance*), e os

espaços em branco nas janelas, a indisponibilidade dos equipamentos ou proibição de ocupação em virtude destes intervalos. Os intervalos de ocupação obrigatória e os mecanismos de propagação de restrições que conduziram a este cenário serão detalhados no capítulo seguinte.

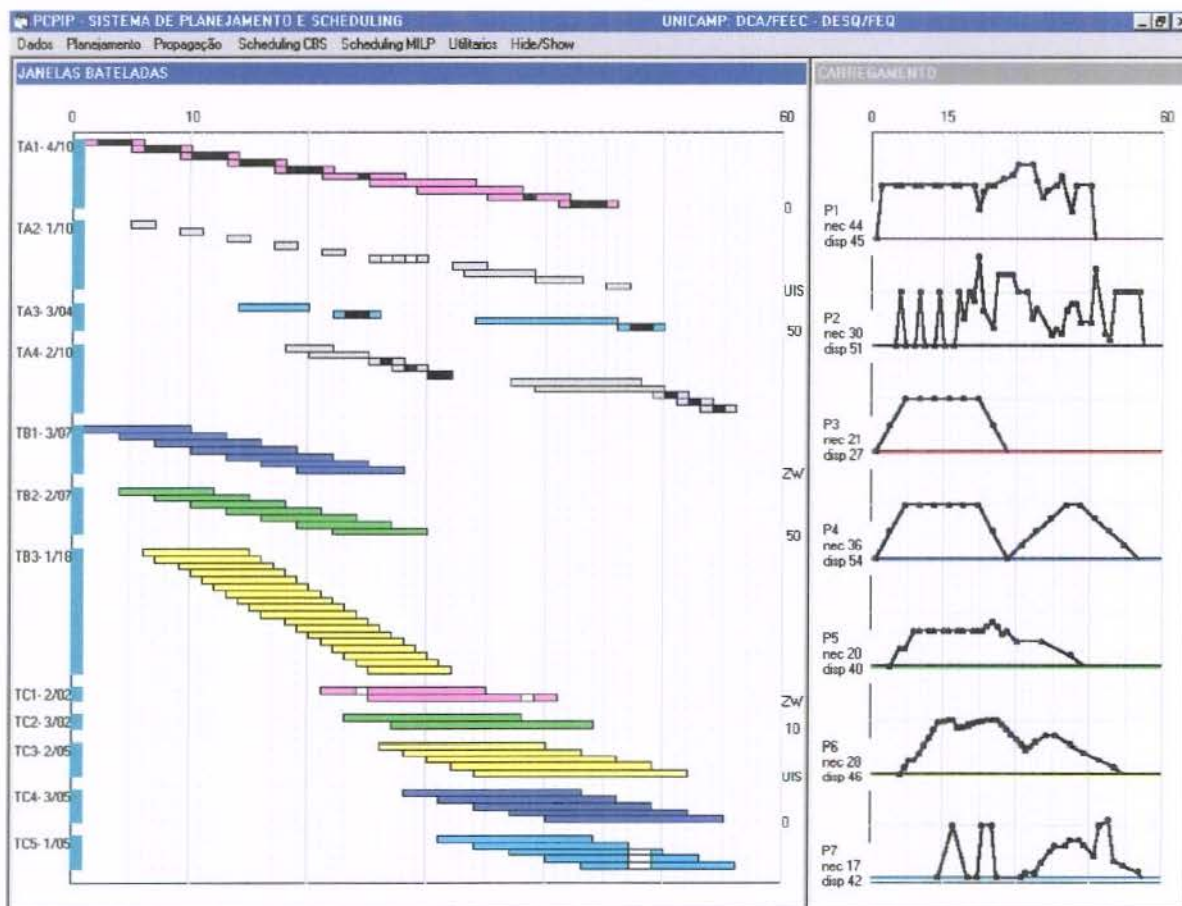


Figura 2.6 – Módulo de *scheduling*

Como resultado do *scheduling* obtêm-se a Carta de Gantt, um gráfico contendo a alocação das bateladas de cada operação aos equipamentos utilizados, bem como os respectivos instantes de início de cada batelada da operação, mostrando quando e onde cada operação deve ser realizada. Na figura 2.7 é exibida uma Carta de Gantt para o resultante da alocação das operações mostradas na figura 2.6.

A abordagem de janelas de tempo traz algumas vantagens, como permitir a redução do espaço de busca através da propagação das restrições do problema. Outra vantagem da utilização das janelas é que como elas mostram o intervalo de tempo disponível para realização das operações, o cenário formado pelas janelas representa também o espaço de solução, o que permite uma avaliação do problema antes de ser

iniciado qualquer procedimento de solução. Através das janelas pode-se calcular o carregamento nos equipamentos em virtude da concorrência entre as operações por esse equipamento. Por outro lado, a abordagem de janelas traz algumas limitações, pois para obtê-las é necessário conhecer previamente o tempo de processamento e o tamanho de batelada das operações.



Figura 2.7 – Carta de Gantt

O capítulo 3 cobre os aspectos da busca orientada por restrições e os principais trabalhos desenvolvidos até o momento. Na seção 3.2 são mostradas as restrições mais comuns presentes na indústria. A seção 3.3 descreve as principais formas de qualificar e mensurar gargalos apresentadas na literatura. As heurísticas de decisões de scheduling mais importantes são detalhadas na seção 3.4. A seção 3.5 descreve os meios de propagação de restrições empregados neste trabalho. Finalmente na seção 3.6 é introduzida a abordagem proposta deste trabalho.

Capítulo 3 - Busca orientada a por restrições

3.1 - Introdução

As técnicas de busca orientada por restrições (*Constraint Based Search*, CBS) são técnicas de busca em árvore e portanto são uma abordagem construtiva do problema de *scheduling*. Como em qualquer técnica de busca em árvore, a árvore é constituída por nós representando soluções parciais do problema de *scheduling*, sendo que uma decisão de *scheduling* leva de um nó pai a um nó filho. Os nós finais da árvore representam soluções completas do problema. Um ramo desde o nó raiz até o nó final contém todas as decisões de *scheduling* que levam à solução representada por este nó final. A principal característica dessas técnicas é que as decisões de *scheduling* nos nós são definidas através de análise das restrições do problema, concentrando os esforços de solução em partes específicas do problema de cada vez.

As técnicas de busca em árvore do tipo *branch & bound* aplicam-se a um problema de *scheduling* onde uma função objetivo deve ser otimizada (para fins dessa discussão considera-se um problema de minimização). A busca em árvore realizada por um algoritmo *branch & bound* é feita de uma forma inteligente utilizando dois procedimentos: o corte de ramos (*pruning*) através da utilização do limitante superior (*upper bound*) disponível, e o prosseguimento do ramo mais promissor através da utilização do limitante inferior (*lower bound*) do último nó de cada ramo.

As técnicas de busca orientada por restrições foram inicialmente utilizadas para problemas de *scheduling* altamente restritos. Nestas situações é difícil obter uma solução em tempo hábil, de modo que é mais importante obter uma solução factível, ou seja, que satisfaça todas as restrições, do que obter a solução ótima, ainda que obviamente atingido o primeiro objetivo o usuário desejará abordar o segundo.

Dentro do contexto da obtenção de uma solução factível em problemas com restrições fortes, as técnicas de busca orientadas por restrições utilizam:

- a) procedimentos específicos para determinar a decisão de *scheduling* em cada nó, e
- b) técnicas de propagação de restrições após cada decisão de *scheduling* de forma a reduzir a árvore de busca.

As decisões de *scheduling* são tomadas levando em consideração as restrições, normalmente detectando onde essas são mais duras. É interessante se concentrar primeiro nas restrições mais fortes, pois elas determinam a qualidade da solução final que será obtida. A decisão escolhida satisfazendo essas restrições freqüentemente é aquela que minimiza o seu impacto sobre o problema, de forma a manter a maior flexibilidade possível para a alocação das operações ainda não alocadas.

As decisões de *scheduling* em um ramo podem conduzir a uma situação inactível, ou seja, em que não sejam satisfeitas todas as restrições do problema. Neste caso é necessário retroceder a um nó anterior na árvore para continuar a busca em outro ramo (*backtracking*).

Quando as técnicas de busca orientada pelas restrições são utilizadas em um contexto de otimização, os sistemas atuais (e.g. ILOG, ILOG 1997) mantêm o mesmo esquema de busca, apenas avaliando o valor da função objetivo em cada nó, obtendo assim um limitante inferior. Após a obtenção de uma solução completa, o valor da função objetivo correspondente constitui um limitante superior que permite cortar ramos. É importante notar que as decisões de *scheduling* ainda são tomadas através da análise de gargalos e não dirigida através dos valores dos limitantes inferiores, como em estratégias clássicas de busca em árvore.

Os algoritmos apresentados na literatura e os *softwares* comerciais existentes utilizam fundamentalmente as restrições originadas pela capacidade dos equipamentos. Quando se consideram problemas de *scheduling* os equipamentos são caracterizados como recursos unitários, ou seja, em cada instante de tempo um equipamento só pode ser utilizado por uma operação.

A busca orientada pelas restrições de capacidade define as decisões de *scheduling* através da detecção dos equipamentos e intervalos de tempo que constituem gargalos de produção. De fato os gargalos podem ter diversas origens (energia elétrica limitada, produtos intermediários requisitados por mais de uma operação, etc.), no entanto a maior parte das técnicas desenvolvidas até recentemente se restringe a gargalos originados pela competição entre operações por um mesmo equipamento.

Isto traz a questão de identificação de gargalos, detalhadas na seção 3.3. Para realizar essa identificação estas técnicas utilizam como dados de entrada as janelas de tempo para execução das operações, e supõe-se que essas janelas já foram definidas em uma fase anterior do problema.

Após a escolha de um gargalo, procede-se então à escolha da operação ou operações sobre as quais serão tomadas as decisões de *scheduling*. As regras utilizadas para esta escolha são heurísticas e na literatura de CBS são conhecidas por heurísticas de escolha de variável (*variable ordering heuristics*, HEVar), e as decisões tomadas sobre as operações escolhidas, heurísticas de escolha de valor (*value ordering heuristics*, HEVal). Tomada uma decisão sobre uma ou mais operações, o número de decisões possíveis sobre as demais operações é reduzido através da propagação das restrições. A existência de uma fase de propagação dos efeitos ou conseqüências de uma decisão tomada é uma característica fundamental das técnicas CBS. A resolução do gargalo inicial gera um novo cenário para o problema que pode conduzir a novos gargalos em outras regiões, e então o procedimento é repetido. Assim é possível obter uma solução pela redução seguida ou refinamento do espaço de busca, que segue então as seguintes etapas:

- Detecção do conflito que será resolvido
- Identificação das operações envolvidas no conflito
- Escolha das operações sobre as quais será tomada uma decisão
- Fixar uma decisão sobre a operação ou operações escolhidas
- Propagação e revisão das restrições do problema

3.2 - Tipos de restrições

Os recursos compartilhados que as operações requisitam podem ser classificados em dois tipos: recursos unitários e cumulativos. Os recursos compartilhados unitários são aqueles que não podem ser utilizados por mais de uma operação ao mesmo tempo, como por exemplo equipamentos, pois supõe-se operações não preemptivas, ou seja, uma vez iniciadas, não podem ser interrompidas até sua conclusão. Os recursos compartilhados cumulativos podem ser utilizados por várias operações simultaneamente até um determinado limite, como por exemplo eletricidade, vapor e mão-de-obra.

Os sistemas de *scheduling* orientados por restrições existentes consideram apenas as restrições por recursos compartilhados unitários (ILOG 1997). Recentemente vem sendo desenvolvidas extensões considerando os recursos compartilhados cumulativos (Claire *Schedule* 1.0 1999). No trabalho desenvolvido as restrições consideradas foram as originadas por recursos compartilhados unitários, no caso, equipamentos, e restrições provenientes de limitações de armazenagem.

Como a atribuição das operações aos equipamentos é fixada na fase de planejamento, são conhecidas previamente as operações que concorrerão pelos equipamentos, e assim a chave da questão não é que recurso as operações requisitam, mas sim os intervalos de tempo em que as operações requisitam o equipamento.

O intervalo de tempo em que uma operação requisita um equipamento é denominado janela de tempo para processamento da operação. O sistema de planejamento utilizado (RODRIGUES L. C. A. 2000d) fornece estas janelas de tempo para a fase de *scheduling*.

Por exemplo, se duas operações compartilham o mesmo equipamento mas suas janelas não se sobrepõem no tempo, como ilustrado na figura 3.1, não existe concorrência pelo equipamento, pois este será de uso exclusivo de cada operação dentro de seu intervalo de tempo, e a decisão de *scheduling* sobre uma operação não terá impacto sobre a outra. Caso as janelas de tempo estejam sobrepostas, como na figura 3.2, existe concorrência pelo mesmo espaço de tempo no equipamento, e assim a decisão tomada sobre uma operação reduzirá as possibilidades de alocação da outra. Deste modo se torna necessário qualificar e

quantificar a exigência de uma determinada operação sobre um equipamento originada pela sua janela de tempo.

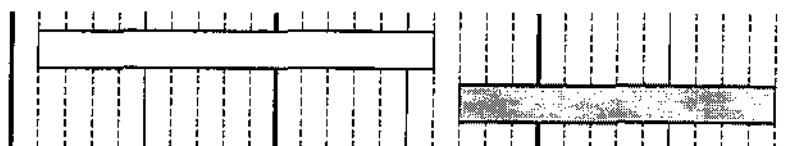


Figura 3.1 – Janelas sem sobreposição

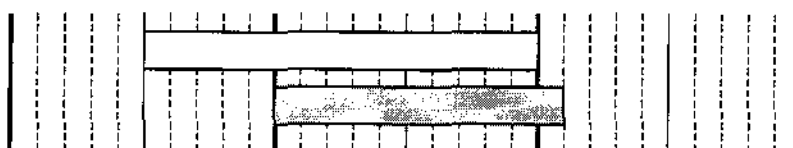


Figura 3.2 – Janelas sobrepostas

A janela de tempo para o processamento de cada operação é definida pelo instante de início mais cedo (*earliest starting time* ou EST) e o instante de término mais tarde (*latest finishing time* ou LFT). O primeiro é determinado, na fase de planejamento, a partir da disponibilidade das matérias-primas, e o último a partir das datas de entrega dos produtos finais que utilizam os estados gerados pela operação. Como ilustrado na figura 3.3, existem dois casos extremos, em que a operação pode ser alocada mais ao início ou mais ao fim da sua janela.

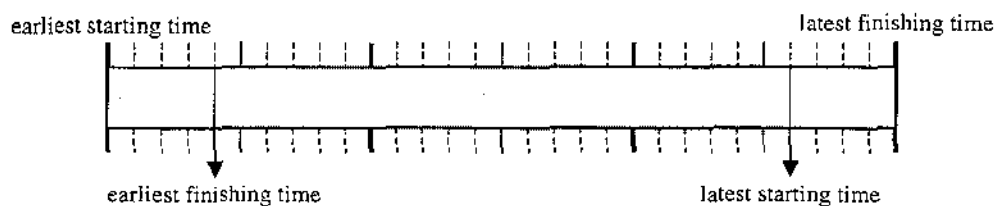


Figura 3.3 – Earliest starting time e latest finishing time

Se a operação for alocada no início da sua janela, será finalizada o mais cedo possível. Neste caso, o instante em que a operação termina é chamado de instante de término mais cedo (*earliest finishing time*, EFT), definido pela equação 3.1. Por outro lado, se a operação for alocada no fim da sua janela, será iniciada o mais tarde possível. Neste caso, o instante em que a operação é iniciada é chamado de instante de início mais tarde (*latest starting time*, LST), definido pela equação 3.2.

$$EFT = EST + \text{Tempo de Processamento} \quad [3.1]$$

$$LST = LFT - \text{Tempo de Processamento} \quad [3.2]$$

3.3 - Medidas de gargalos

Algumas ferramentas podem orientar o procedimento de busca e identificação das regiões onde ocorrem gargalos e as operações envolvidas nestes. As técnicas apresentadas a seguir são baseadas na competição de operações por recursos compartilhados unitários. Para tanto, exigem que sejam conhecidos o tempo de processamento das operações e as janelas de tempo onde é permitida sua alocação.

3.3.1 - Criticalidade e demanda individual de uma operação

A. Criticalidade (Keng N. P., Yun D. Y. Y., Rossi M. 1988) - A criticalidade mede de maneira quantitativa a flexibilidade de alocação de uma operação dada a sua janela de tempo, ou em outras palavras, o quão “apertado” é o intervalo de tempo disponível para realizar a operação. A criticalidade de uma operação i está definida como a razão entre os valores do seu tempo de processamento e da janela de tempo disponível para sua execução.

$$Cr_i = TP_i / JT_i \quad [3.3]$$

Onde: TP_i = tempo de processamento da operação i
 JT_i = duração da janela de tempo da operação i
 Cr_i = criticalidade da operação i

Valores de criticalidade altos indicam operações com pouca flexibilidade temporal, ou seja, com poucas possibilidades de alocação. A operação é dita mais crítica quanto maior a sua criticalidade, que no caso de possuir valor unitário, indica que a alocação da operação é única, ou seja, não existem outras alternativas de alocação. Considerando o tempo discretizado com intervalo de discretização unitário, o número de possíveis alocações é dado por $JT_i - TP_i + 1$.

Uma operação pode ser executada em mais de um equipamento ao mesmo tempo, como ocorre com o processamento em paralelo operando em fase. Neste caso a operação possui duas ou mais janelas de tempo, que podem ter diferentes durações em função da ocupação dos equipamentos. Para efeito do cálculo de criticalidade, é considerado que existe apenas uma janela, de tal forma que o número de possíveis alocações nessa nova janela seja igual à soma das possíveis alocações em cada uma das janelas da operação. Assim, para uma operação i sendo executada em um conjunto J de equipamentos a criticalidade é calculada como:

$$Cr_i = \frac{TP_i}{\sum_{j \in J} [(JT_{ij} - TP_i + 1) + TP_i - 1]} \quad [3.4]$$

Onde: JT_{ij} = janela de tempo da operação i no equipamento j

A parcela $(JT_{ij} - TP_i + 1)$ é o número de possíveis alocações de i em j , considerando o tempo discretizado com intervalo de discretização unitário.

B. Demanda Individual (Sadeh N. 1991) - A demanda individual possui um enfoque um pouco diferente da criticalidade. Trata-se de uma medida da possibilidade de que um intervalo de tempo seja utilizado na alocação da operação. Para tanto, é avaliado o número de possíveis alocações que a operação pode ter dentro de sua janela (considerando o tempo discretizado), e é considerado que cada possibilidade de alocação tem a mesma probabilidade de ocorrer. Assim, cada intervalo de tempo de cada janela possui também uma probabilidade de ser ocupado, determinado como a razão entre o número de alocações ocupando este intervalo de tempo e o número total de alocações possíveis. Este método tem como vantagem mostrar mais claramente do que o anterior qual é o intervalo de tempo com maior possibilidade de ser utilizado pela operação.

A figura 3.4 ilustra o método. A operação possui tempo de processamento de 15 unidades de tempo e uma janela de 20 unidades para ser executada. Utiliza-se um intervalo de discretização unitário. Existem 6 possíveis alocações, indicadas por setas, cada qual com igual chance de ocorrer. Contudo, as janelas das possíveis alocações (setas) se sobrepõem, indicando que alguns intervalos de tempo têm maior possibilidade de serem ocupados. Assim, o intervalo de 5 a 6, onde somente uma das possíveis alocações está presente, tem possibilidade de 1/6. Já no intervalo de 6 a 7 existem duas possíveis alocações que podem ocupar este intervalo, e a possibilidade é de 2/6. Os intervalos com maior chance de serem utilizados pela operação estão entre 10 e 20, com possibilidade de 6/6 para todos eles, pois existem 6 possíveis alocações, como indicado pelas setas. De fato, o intervalo de tempo [10, 20] tem possibilidade de 100% de ser utilizado (6/6), e será ocupado qualquer que seja a alocação escolhida.

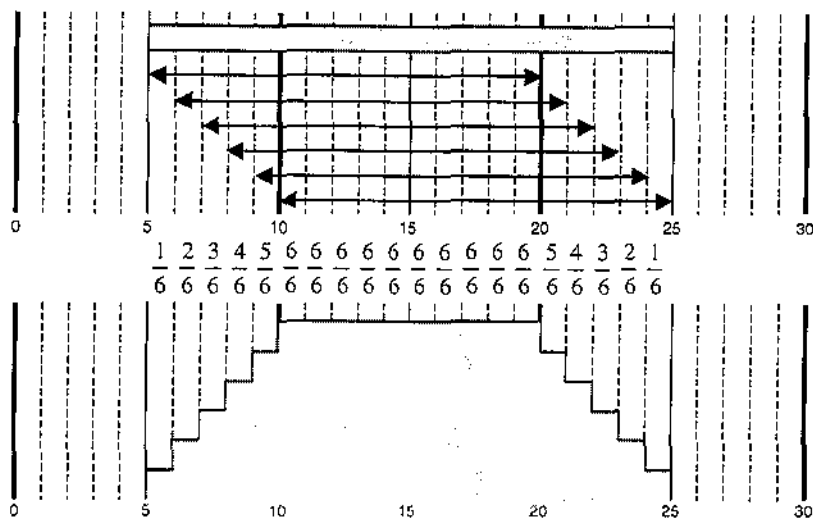


Figura 3.4 – Demanda individual

Outra vantagem da demanda individual é a capacidade de lidar com janelas de tempo disjuntas, que possuem um ou mais intervalos de tempo indisponíveis para alocação entre elas. O conceito da crucialidade não é capaz de quantificar a flexibilidade de alocação nesses casos.

3.3.2 - Crucialidade e demanda agregada de um equipamento

Mesmo que uma operação possua pouca flexibilidade de alocação em um equipamento, este só poderá vir a ser um gargalo se houver outras operações competindo pelo mesmo recurso no mesmo intervalo de tempo. Os conceitos descritos a seguir permitem a identificação dos intervalos de tempo onde ocorre grande competição por equipamentos devido às operações que os requisitam.

A. Crucialidade (Keng N. P., Yun D. Y. Y., Rossi M. 1988) - A crucialidade avalia de maneira qualitativa a concorrência entre operações por um determinado equipamento no tempo. Para representar essa concorrência é construída a curva de crucialidade, que é obtida a partir da soma dos valores de crucialidade das operações que competem pelo mesmo equipamento, dentro do intervalo de suas respectivas janelas de tempo. Regiões onde a crucialidade assume valores altos representam grande disputa entre as operações e portanto constituem gargalos na produção.

A seguir é ilustrado o conceito da crucialidade. Na figura 3.5 são exibidas as janelas de tempo de quatro operações, A, B, C e D, com tempo de processamento 3, 2, 2 e 3 unidades de tempo, respectivamente, onde cada operação é constituída por duas ou três bateladas. Há dois equipamentos; P₁, compartilhado por A e B e P₂, compartilhado por C e

D. Na figura 3.6 está ilustrada a curva de crucialidade do equipamento P1 e na figura 3.7 a do P2. Este último mostra um pico de grande concorrência entre as operações no intervalo de 13 a 19.

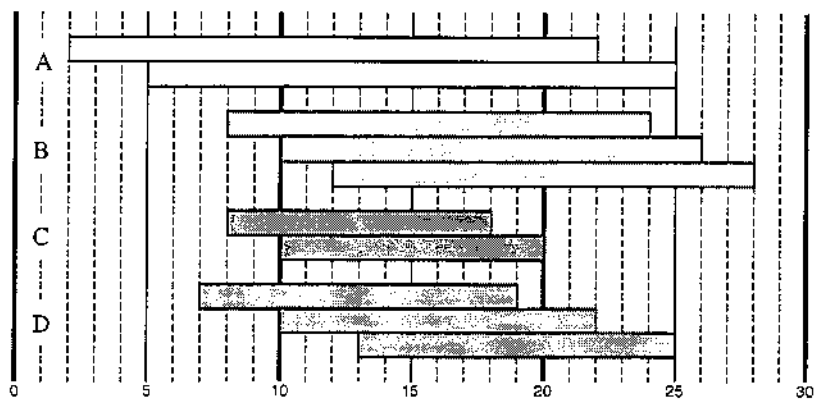


Figura 3.5 – Janelas das operações

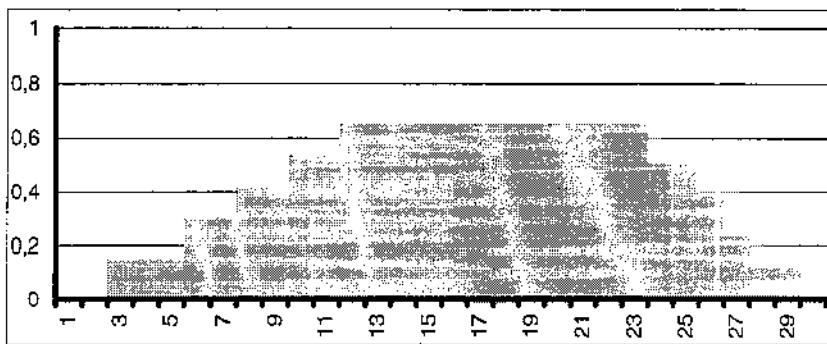


Figura 3.6 – Curva de crucialidade do equipamento P1

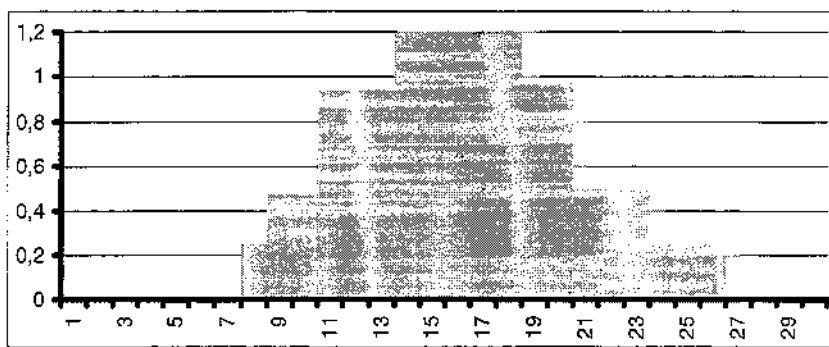


Figura 3.7 - Curva de crucialidade do equipamento P2

Para ilustrar a obtenção da curva de crucialidade, considere apenas as operações do equipamento P1. Para cada batelada que compõe a operação é calculado o valor da crucialidade, de modo que este valor existe somente no intervalo de tempo pertencente a janela da batelada. A crucialidade é obtida somando os valores da crucialidade, de modo que quando as janelas de tempo se sobrepõem, o valor das suas crucialidades é acumulado.

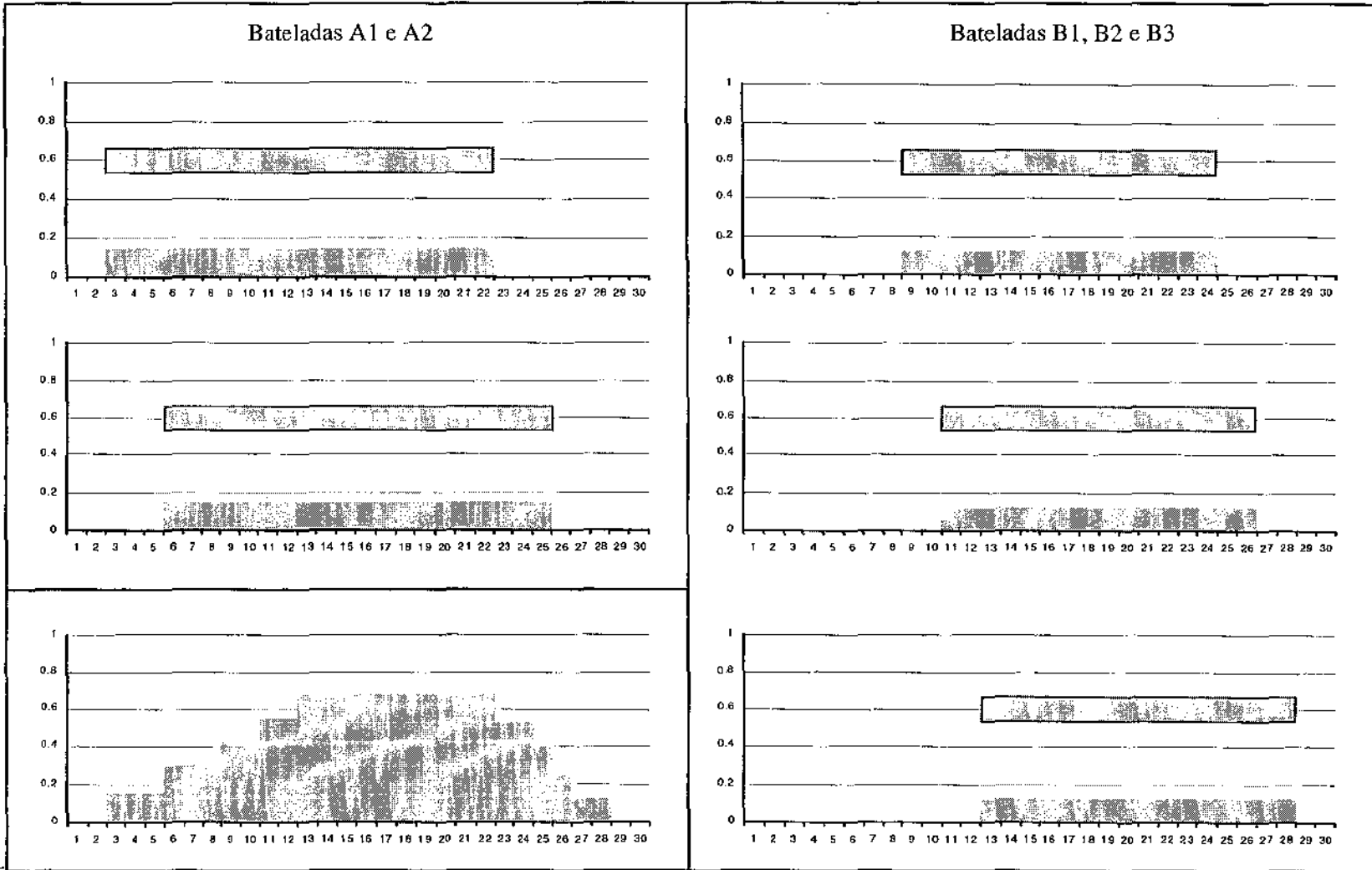


Figura 3.8

B. Demanda Agregada (Sadeh N. 1991) - O método da demanda agregada é similar ao da crucialidade. São somados no tempo os perfis de demanda individual (Sadeh N. 1991) das operações para construir a curva de demanda agregada. Usando as mesmas janelas da figura 3.5, a curva de demanda agregada para o primeiro equipamento, resultante da soma das demandas individuais, é mostrada na figura 3.9. De modo análogo é obtida a curva de demanda agregada para o segundo equipamento, exibida na figura 3.10.

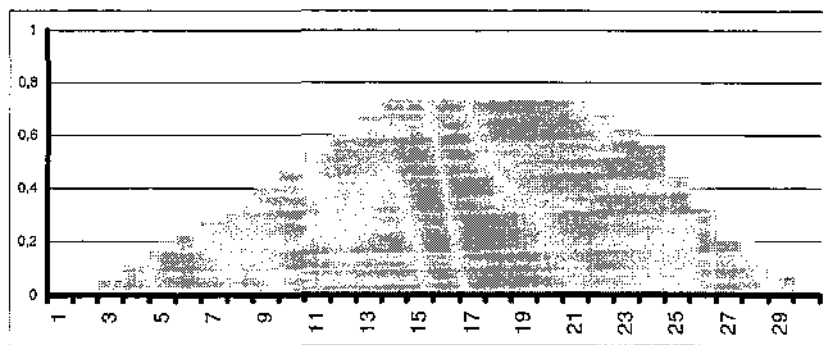


Figura 3.9 – Demanda agregada do equipamento P1

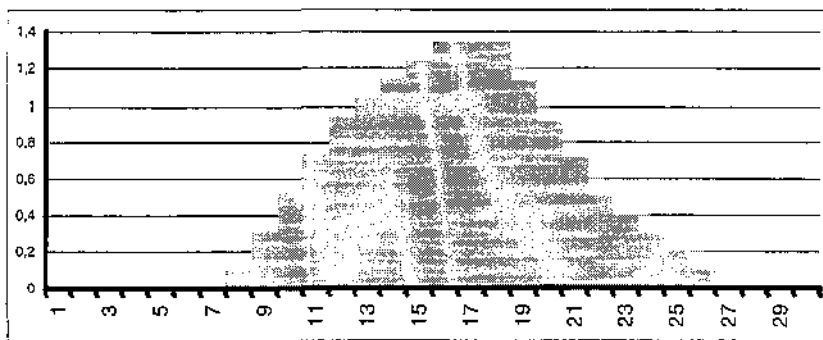


Figura 3.10 – Demanda agregada do equipamento P2

As curvas de crucialidade e de demanda agregada são um pouco diferentes, sendo que esta última claramente fornece um nível de detalhe maior. Como é possível notar, o maior pico está agora no intervalo [15, 18].

Novamente, para ilustrar a obtenção da curva de demanda agregada, considere apenas as operações do equipamento P1. Para cada batelada que compõe a operação é calculado o valor da demanda individual, de modo que este valor existe somente no intervalo de tempo pertencente à janela da batelada. A demanda agregada é obtida somando os valores da demanda individual, de modo que quando as janelas de tempo se sobrepõem, o valor das suas demandas individuais é acumulado.

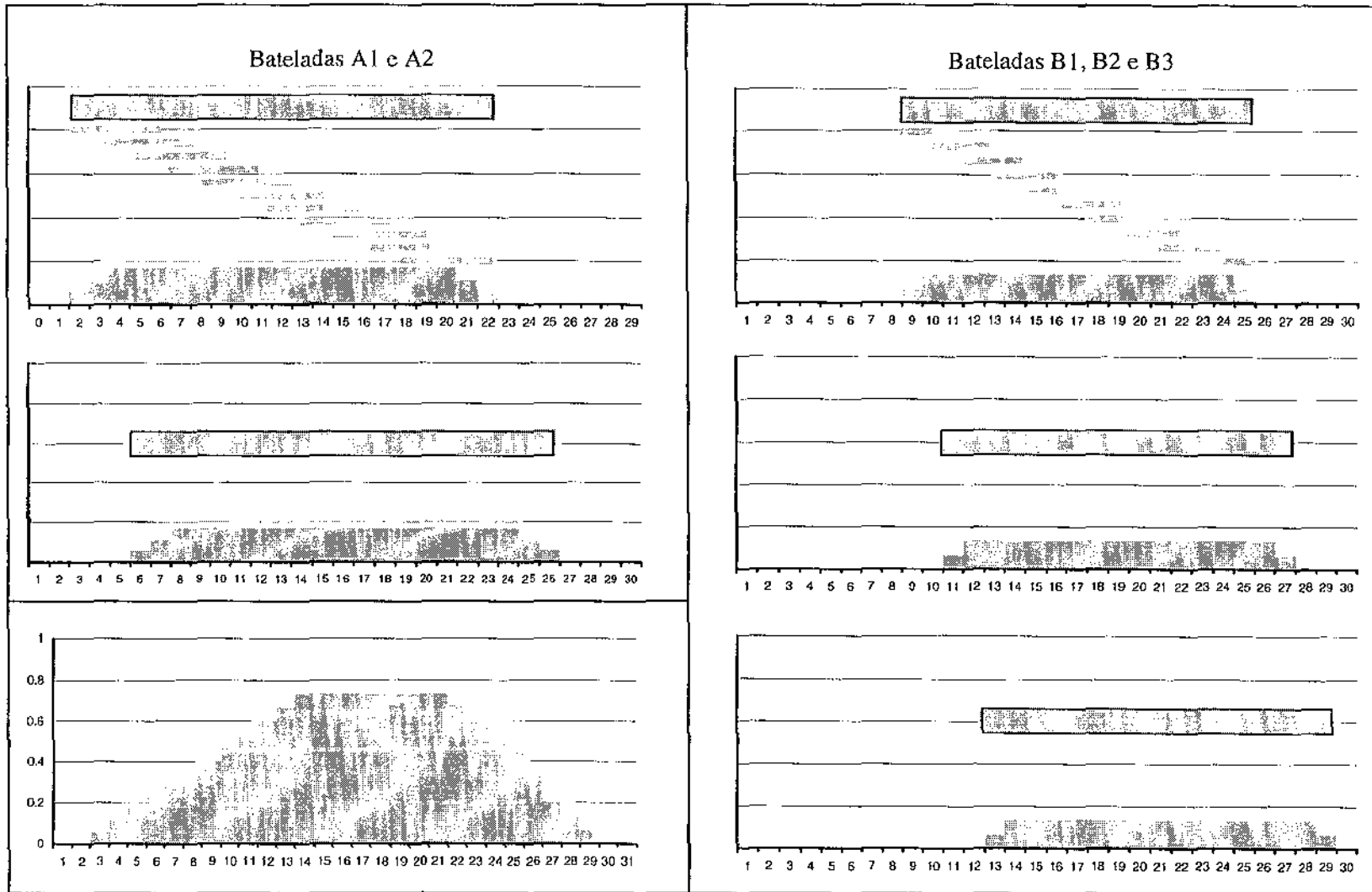


Figura 3.11

3.3.3 - Tempo de Folga

A folga (*slack time*, Smith S. F., Cheng C. C. 1993) representa o tempo de folga existente para a realização de uma operação, e é definida como a diferença entre a janela de tempo disponível e o tempo de processamento. Na referência acima citada o conceito de folga é estendido para um par de operações competindo pelo mesmo equipamento, ou seja, com janelas sobrepostas. A diferença $\max(LFT_i, LFT_j) - \min(EST_i, EST_j)$ mostra o tempo total disponível para execução de ambas, como uma única janela resultante da união das janelas das duas operações. A diferença entre este tempo total disponível e a soma dos tempos de processamento fornece uma medida da folga existente para realização das duas operações. Este conceito é diferente dos conceitos de criticalidade e demanda individual, pois fornece uma medida da flexibilidade de alocação apenas para um par de operações que competem pelo mesmo equipamento.

A figura abaixo ilustra o método. O tempo de processamento das operações A (branco) e B (cinza) são 10 e 5, respectivamente. A medida de folga é feita analisando o tempo total disponível e o tempo total necessário para execução do par de operações.

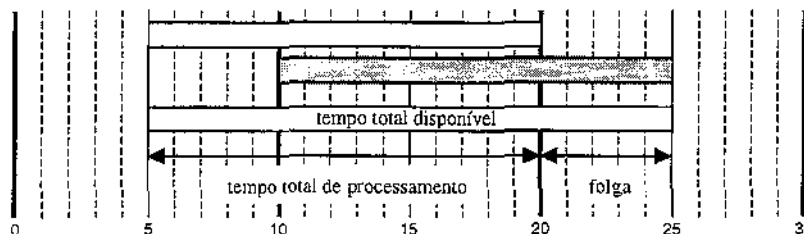


Figura 3.12 – Folga de um par de operações

Considerando agora as janelas do primeiro exemplo, mostrado na figura 3.5, pode ser realizado o cálculo da folga para todos os pares de operações que competem pelo mesmo equipamento. O resultado é mostrado na tabela a seguir. Nesta tabela é mostrada também a folga relativa, definida como a razão entre o tempo de folga e o tempo total disponível. A folga relativa tem o objetivo de distinguir entre casos onde o tempo de folga absoluto é igual, mas as janelas disponíveis para o processamento são diferentes, de modo que algumas situações podem ser mais apertadas do que outras.

Tabela 3.1 – Folga dos pares de operações

Pares de Operações	Tempo Total		Tempo de Folga		Folga Relativa	
	max(LFT) - min(EST)		(TTotal) - ΣTP		(TFolga/TTotal) * 100	
A1	B1	22	17	77%		
	B2	24	19	79%		
	B3	26	21	81%		
A2	B1	20	15	75%		
	B2	21	16	76%		
	B3	23	18	78%		
C1	D1	12	7	58%		
	D2	14	9	64%		
	D3	17	12	71%		
C2	D1	13	8	62%		
	D2	12	7	58%		
	D3	15	10	67%		

3.3.4 - Flexibilidade associada a um par de operações

Na referência acima, propõe-se também caracterizar o gargalo constituído por um par de operações através da flexibilidade de ordenamento associada a esse par. Uma forma simples de estimar a flexibilidade de um ordenamento é o *menor caminho*, proposto por Smith e Cheng (Cheng C. C., Smith S. F. 1995). Para um par de operações *i* e *j* competindo pelo mesmo recurso, é definido o menor caminho (*shortest path*, *sp*) como sendo a diferença entre o fim da janela de uma operação e o início da janela da outra, como ilustrado na figura 3.13. Assim, obtêm-se dois valores para o menor caminho; sp_{ij} e sp_{ji} .

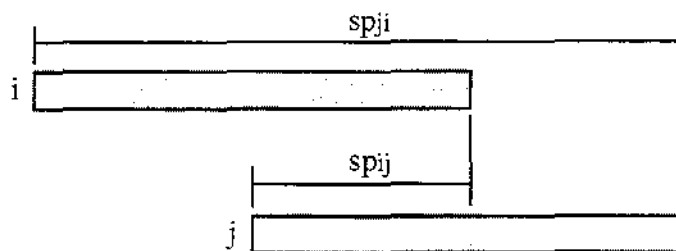


Figura 3.13 – Menor caminho para duas operações *i* e *j*

$$sp_{ij} = LFT_i - EST_j \quad [3.5]$$

$$sp_{ji} = LFT_j - EST_i \quad [3.6]$$

Se o valor de sp_{ij} for menor do que a soma dos tempos de processamento, o ordenamento *i* precede *j* é obrigatório, dado que o ordenamento *j* precede *i* é infactível. O termo menor caminho refere-se à menor distância (em unidades de tempo) entre o fim e o início das janelas de um par de operações. Quanto menor essa distância, mais próximo o par está de assumir um ordenamento obrigatório. A partir desse resultado os autores propõem uma medida da flexibilidade associada ao par de operações (*i*, *j*), denominado comprimento mínimo do menor caminho (*minimum shortest path length*), w_{ij} , definido na equação a seguir.

$$w_{ij} = \min(sp_{ij}, sp_{ji}) \quad [3.7]$$

Quanto menor o valor de w_{ij} , menor flexibilidade possui o par (i, j) de poder assumir qualquer uma das duas ordens de precedência, pois está mais próximo de assumir um ordenamento obrigatório. Quando se consideram várias operações, o par com menor w_{ij} constitui o maior gargalo. Uma situação importante a ser considerada é aquela em que o valor de w é igual para dois ou mais pares de operações. Considere as operações mostradas na figura 3.14. Ambos os pares de janelas (A_1, A_2) e (B_1, B_2) possuem um valor para w de 4 unidades de tempo.

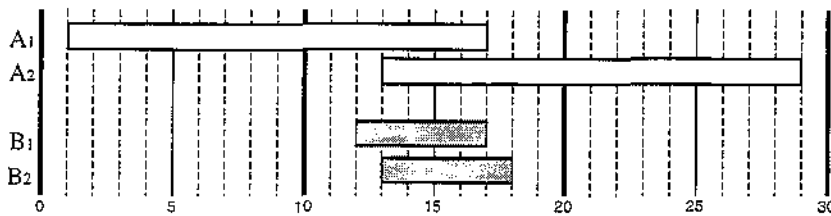


Figura 3.14 – Janelas das operações

Para distinguir entre esses casos os autores propõem um novo conceito, a *tendência do menor caminho (biased shortest path, bsp)*. É introduzida a variável S , que estima o grau de similaridade entre os valores de sp_{ij} , de forma a distinguir entre casos como o ilustrado acima.

$$S = \min(sp_{ij}, sp_{ji}) / \max(sp_{ij}, sp_{ji}) \quad [3.8]$$

Quanto maior a diferença entre sp_{ij} e sp_{ji} , maior é a tendência de uma das duas ordens do par de ser obrigatória, e menor o valor de S . O valor do sp modificado, bsp , é definido então como:

$$bsp_{ij} = sp_{ij} / \sqrt{S} \quad e \quad [3.9]$$

$$bsp_{ji} = sp_{ji} / \sqrt{S} \quad [3.10]$$

O valor de bsp é então maior quanto maior a tendência de uma das ordens do par de ser obrigatória. O comprimento mínimo do menor caminho w_{ij} é determinado como anteriormente:

$$w_{ij} = \min(bsp_{ij}, bsp_{ji}) \quad [3.11]$$

É escolhido o par cuja escolha de um ordenamento é menos obrigatória, de forma a preservar maior flexibilidade de tempo para as operações ainda não alocadas. No exemplo da figura 3.14 valores de w com base na tendência do menor caminho são mostrados na tabela a seguir:

Tabela 3.2 – Valores da tendência do menor caminho para as operações da figura 3.14

Operações	sp	s	bsp	w
(A1, A2)	4		10,6	
(A2, A1)	28	1 / 7	74,1	10,6
(B1, B2)	4		4,9	
(B2, B1)	6	2 / 3	7,3	4,9

Apesar de ambos os pares possuírem valores iguais para o menor sp, o bsp distingue entre casos onde as janelas são diferentes e o menor w mostra o par mais crítico, e que neste contexto constitui um gargalo mais grave.

3.4 - Heurísticas para decisões de *scheduling*

Os conceitos descritos até agora servem de base para heurísticas de identificação de gargalos. Identificada a região que constitui o gargalo mais grave, procede-se à escolha da operação ou operações que contribuem mais para formação deste gargalo, sobre as quais serão tomadas as decisões de *scheduling*. Essas decisões podem ser decisões de alocação ou de ordenamento.

A alocação define o instante em que uma operação é iniciada, enquanto que o ordenamento define uma relação de precedência entre duas operações. No primeiro caso o problema de *scheduling* está resolvido quando são alocadas todas as operações. Neste caso, o instante específico em que as operações iniciam e terminam é conhecido e a partir destes obtêm-se a Carta de Gantt diretamente. Nas técnicas que determinam ordenamentos o problema de *scheduling* está resolvido quando todas as ordens estão fixadas. Uma vantagem desta última abordagem é que muitas operações ainda podem apresentar janelas de tempo maiores do que o tempo de processamento, ao invés de estarem alocadas. Isso permite uma maior flexibilidade da solução, pois o instante específico de início das operações pode então ser decidido com base em critérios não previstos anteriormente, o que fornece maior grau de liberdade ao usuário. A alocação pode até mesmo ser alterada após ser iniciada a produção, caso ocorra algum imprevisto, como atraso na chegada de uma matéria-prima ou então manutenção de um equipamento.

3.4.1 - Heurística de menor impacto

Os autores Keng, Yun e Rossi (Keng N. P., Yun D. Y. Y., Rossi M. 1988) usam o conceito de crucialidade e criticalidade para identificar o equipamento com intervalo de tempo mais crítico. É considerado gargalo o intervalo de tempo em que a curva de crucialidade assume valores mais elevados. Valores altos de crucialidade mostram que

existe maior concorrência pelo equipamento no intervalo em questão. Entre as operações envolvidas neste intervalo, ou seja, cujas janelas pertencem ao intervalo, é escolhida a que mais contribui para o gargalo, que é a operação de maior criticalidade (HEVar). Isso se deve ao fato de uma operação com criticalidade alta requisitar fortemente o equipamento no intervalo de tempo da sua janela. Em virtude da baixa flexibilidade de alocação, o equipamento tem maior chance de ser ocupado nesse intervalo de tempo por esta operação com alta criticalidade. Em seguida é tomada uma decisão de alocação da operação escolhida, que é alocada no intervalo de sua janela onde a curva de crucialidade possui menor valor, ou seja, onde há menor concorrência pelo recurso, de forma a minimizar o impacto da alocação sobre as demais operações (HEVal).

Considerando o exemplo da figura 3.5, existe um pico acentuado no equipamento P_2 , no intervalo de 13 a 19. As operações envolvidas nesse intervalo são C_1 , C_2 e D_1 a D_3 . De acordo com a equação 3.3, o valor da criticalidade das operações C_1 e C_2 é 0,2 e das operações D_1 a D_3 é 0,25. Como as operações D_1 a D_3 possuem criticalidade maior do que C_1 e C_2 , uma delas deve ser escolhida. Seja tomada D_2 , por exemplo. Dentro do espaço da sua janela a curva de crucialidade é menor no intervalo 19 a 22, e portanto a operação será alocada nesse intervalo. Uma vez que a operação é alocada, uma nova restrição é gerada (o intervalo de tempo ocupado pela alocação de D_2 é agora impossível de ser ocupado por qualquer outra operação), e esta restrição é propagada, levando a uma nova situação das janelas, ilustradas na figura 3.15. Estes mecanismos de propagação de restrições serão discutidos na seção 3.5. Uma nova curva de crucialidade é construída com base nas novas janelas. O resultado é mostrado na figura 3.16. O cume de valor unitário no intervalo de 19 a 15 é resultado das janelas alocadas, e por isso não representa mais um gargalo. O gargalo inicial foi eliminado, porém há outro no intervalo de tempo 10 a 16. Este procedimento continua até que todas as operações estejam alocadas.

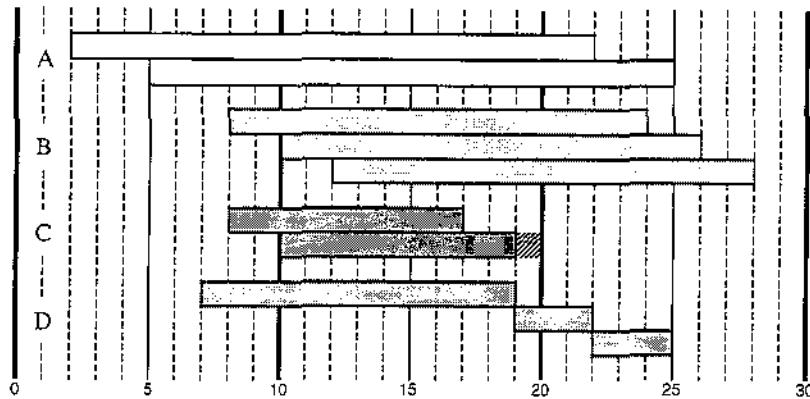


Figura 3.15 – Janelas de tempo com a operação D2 alocada

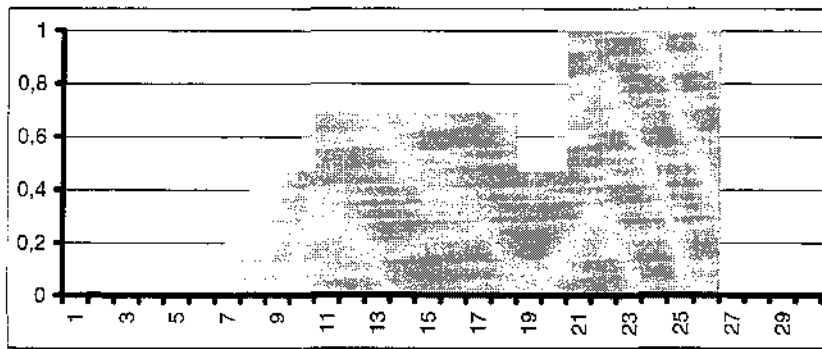


Figura 3.16 – Nova curva de crucialidade do equipamento P2

3.4.2 - Heurística de sobrevivência da reserva

Na heurística de sobrevivência da reserva (Sadeh N. 1991), é escolhido o intervalo com maior demanda agregada e a operação que mais contribui para este, ou seja, a que possui maior demanda individual (HEV_{ar}). Em seguida a operação escolhida é alocada usando uma medida de sobrevivência (HEV_{al}), que estima a possibilidade de uma determinada alocação não causar um resultado infactível em um ponto mais avançado da árvore de busca, ou seja, estima a possibilidade de sobrevivência de uma determinada alocação. Intervalos de tempo de pequena concorrência fornecem maiores valores de sobrevivência, e portanto são escolhidos para a alocação. A sobrevivência é uma função da demanda agregada e do número de possíveis alocações da operação ainda não alocada que disputa um intervalo de tempo. Para uma operação, quanto menores esses dois valores, maior a sobrevivência da alocação, ou seja, menor a possibilidade de que haja um conflito posterior no intervalo reservado para a operação alocada. De forma análoga à apresentada na seção anterior, o procedimento é repetido até que todas as operações sejam alocadas.

Ambas as heurísticas possuem dimensão pequena, porém possuem espaço de busca grande em relação a outros métodos. A dimensão se refere à quantidade de variáveis

do problema, ao passo que o espaço de busca se refere ao domínio dessas variáveis. A dimensão é pequena no caso das heurísticas acima citadas porque o número de elementos (ou variáveis) que podem ser escolhidos é igual ao número de operações. O espaço de busca é grande pois a quantidade de valores (ou domínio) que uma variável pode assumir é grande, visto que o número de possíveis alocações de uma operação geralmente é alto.

3.4.3 - Heurística de arbitragem de precedência maximizando folga

Na heurística de arbitragem de precedência (Smith S. F., Cheng C. C. 1993) são escolhidas duas operações e fixada uma ordem de precedência entre elas. É escolhido o par de operações que possuir menor tempo de folga (HEVar), ou seja, o par com menor flexibilidade temporal. Para decidir qual a ordem de precedência entre duas operações A e B, é feito o cálculo do tempo de folga para ambos os casos, A precede B e B precede A. É então escolhida a precedência que preservar o maior tempo de folga, ou seja, a que deixará maior grau de liberdade para a alocação das operações, depois de fixada (HEVal). No exemplo da figura 3.5, existem dois pares que possuem pequeno valor de folga: C₁/D₁ e C₂/D₂, conforme a tabela 3.1. Tomando por exemplo o par C₂ e D₂, fixando ambas as ordens de precedência, obtêm-se as janelas ilustradas na figura 3.17. Quando uma operação precede outra, o fim de sua janela deve ser menor ou igual ao fim da segunda menos o tempo de processamento, e esta última deve ter início maior do que o início da primeira mais o tempo de processamento.

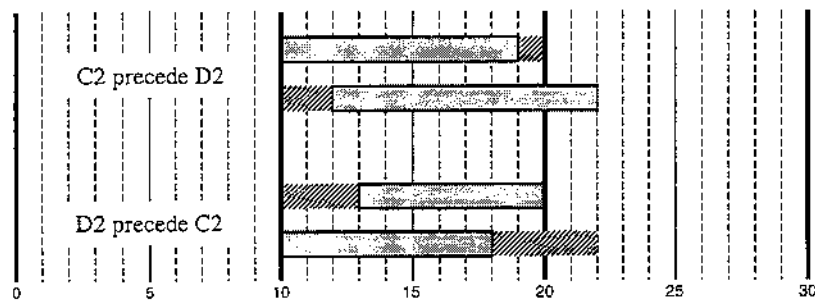


Figura 3.17 – Ordenamentos entre A2 e B1

O primeiro ordenamento produz uma redução pequena nas janelas, que continuam com um tempo total para execução de 12 unidades de tempo. Já o segundo ordenamento diminui mais as janelas e reduz o tempo total de execução para 10 unidades de tempo, reduzindo a flexibilidade de alocação das operações. Assim, o primeiro ordenamento é escolhido.

Essa heurística possui uma dimensão grande em comparação às duas anteriores, ou seja, o número de variáveis é maior pois o número de elementos que podem ser escolhidos é igual a combinação de todos os pares de operações que competem pelo mesmo equipamento. Porém, o espaço de busca é bem menor pois o domínio das variáveis é menor, já que o número de decisões que se pode tomar sobre cada par são apenas duas.

3.4.4 - Heurística de tendência do menor caminho

Na heurística da tendência do menor caminho (Cheng C. C., Smith S. F. 1995) é escolhido o par de operações que possui menor flexibilidade de ordenamento, descrito na seção 3.3.4 (HEVar). A heurística de escolha de valor de tendência do menor caminho escolhe o par cujo ordenamento está mais próximo de se tornar obrigatório, ou seja, com menor flexibilidade de ordenamento. Esse par é identificado através da equação 3.7. Caso dois ou mais pares de operações possuam valores iguais para o menor caminho (como mostrado na **figura 3.17**), é escolhido o par cujo ordenamento está menos próximo de se tornar obrigatório, identificados pela equação 3.11. Dado que os valores do menor caminho são iguais entre dois pares de bateladas, a variável S (equação 3.8), ao medir o grau de similaridade das janelas, mostra qual dos pares possui janelas menores, nos quais a inserção de um dos dois ordenamentos pode produzir maior redução das janelas e conduzir a um resultado infactível. Essa situação reduz a flexibilidade de estabelecer qualquer um dos dois ordenamentos, e constitui o maior gargalo.

É então fixada a ordem de precedência de forma a minimizar a redução das janelas em consequência do ordenamento (HEVal), mantendo a maior flexibilidade para as alocações posteriores. Essa decisão corresponde a escolher o ordenamento com menor bsp, ou seja:

$$Se \ bsp_{ij} \leq bsp_{ji} \Rightarrow i \text{ precede } j \quad [3.12]$$

$$Se \ bsp_{ji} < bsp_{ij} \Rightarrow j \text{ precede } i \quad [3.13]$$

Essa decisão faz sentido pois a cada passo, estabelecendo a ordem que está mais próxima de realizar obrigatoriamente, preserva-se a precedência que as janelas têm maior tendência a assumir, de forma a obter a menor redução das janelas após o par ser ordenado.

Como as janelas sofrem a menor redução possível, também é mantida a maior flexibilidade na alocação do par de operações envolvidas. A grande vantagem da tendência do menor caminho é a pequena possibilidade de obter uma solução ineficaz, já que esta busca é orientada para uma solução na qual as janelas das operações se acomodam mais facilmente. Entretanto, visto que o ordenamento sugerido pode ter pouco impacto na redução das janelas, um número maior de decisões pode ser necessário para reduzir as janelas de forma a eliminar a concorrência entre as operações pelo equipamento.

3.5 - Mecanismos de propagação de restrições

As técnicas apresentadas a seguir constituem uma parte importante dos algoritmos de busca orientada por restrições. A propagação de restrições tem como objetivo garantir que o espaço de busca após cada decisão de *scheduling* seja atualizado face às decisões anteriormente tomadas no ramo, motivo pelo qual na literatura é muitas vezes referida como imposição de consistência (*consistency enforcing*). Uma decisão de alocação faz com que a janela da operação seja igual ao tempo de processamento e que todo o intervalo da janela obviamente seja ocupado. Uma decisão de ordenamento entre um par de operações pode dar lugar a reduções nas janelas de uma ou ambas as operações. Estas reduções podem conduzir a reduções nas janelas das operações ligadas àquelas por compartilhamento de equipamentos e balanço de massa (receita de produção). Deste modo, a cada passo, a propagação das restrições detecta reduções obrigatórias nas janelas das operações em virtude de uma alocação ou ordenamento, reduzindo também o espaço de busca.

Quando a estratégia de busca prevê *backtracking*, pode haver também uma redução na quantidade de *backtracking* necessário, pois a possibilidade de um ramo atingir um nó infactível na árvore diminui, já que a propagação de restrições elimina as decisões de *scheduling* que desrespeitam as restrições do problema, impedindo que tais decisões sejam tomadas em nós mais avançados no mesmo ramo. Essa estratégia pode diminuir o tempo necessário para obter uma solução final, entretanto a propagação de restrições tem também um custo computacional, e deve ser estabelecido um compromisso entre a economia de tempo pela redução do *backtracking* e o gasto devido à propagação.

Os mecanismos de propagação de restrições considerados em sistemas comerciais (ILOG 1997) envolvem:

- Restrições de balanço de massa
- Restrições de capacidade

Neste trabalho estes mecanismos foram estendidos de forma a considerar também as restrições de armazenagem, frequentemente presentes na indústria de processos.

3.5.1 - Propagação de restrições de compartilhamento de equipamentos

3.5.1.1 - Precedência entre bateladas de uma mesma operação

Existem situações em que uma operação precisa ser executada diversas vezes para atender à demanda, introduzindo o conceito de bateladas de uma mesma operação. As bateladas de uma operação compartilham o mesmo equipamento, de modo que uma condição especial de monotonicidade pode ser imposta de forma que:

$$EST_i \geq EST_{i-1} + TP_i \text{ e} \quad [3.14]$$

$$LFT_i \leq LFT_{i+1} - TP_i \quad [3.15]$$

Essa condição restringe os instantes de início e fim das bateladas de forma a estabelecer uma ordem entre elas. A segunda batelada de uma operação obviamente só poderá ser iniciada após finalizada a primeira, e a terceira após finalizada a segunda. De forma análoga, a segunda batelada deve ser finalizada antes de iniciar a terceira, e a primeira antes de iniciar a segunda. Assim, uma batelada pode ser iniciada, no mínimo, no menor instante em que sua antecessora termina (eq. 3.14). Um raciocínio semelhante pode ser feito com respeito aos instantes de término. Uma batelada pode ser finalizada, no máximo, no maior instante em que sua sucessora inicia (eq. 3.15).

Existe portanto uma precedência fixa entre as bateladas da mesma operação, e suas janelas devem obedecer essa precedência. Para ilustrar a propagação resultante dessa precedência, suponha uma operação constituída por três bateladas cujo tempo de processamento é de quatro unidades de tempo. De acordo com as equações 3.14 e 3.15, as janelas ficariam como ilustrado a seguir:

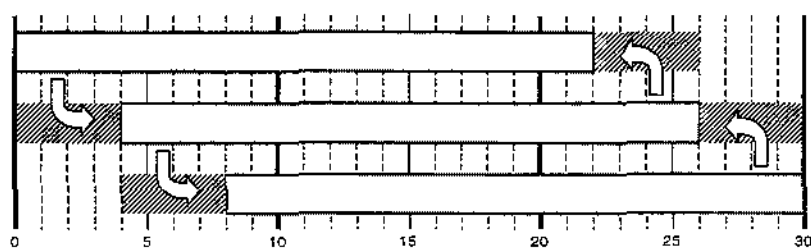


Figura 3.18 – Reduções em virtude da precedência entre bateladas de uma operação

3.5.1.2 - Intervalos de ocupação obrigatória

Um intervalo de ocupação obrigatória (*interval of total reliance*, Sadeh N. 1991) é um intervalo de tempo que será utilizado obrigatoriamente por uma determinada operação. Isto quer dizer que o equipamento onde a operação é realizada inevitavelmente será

ocupado por essa operação neste intervalo de tempo, independente do instante em que a operação seja alocada. Assim, o equipamento se torna indisponível para outras operações que o requisitam neste intervalo de tempo.

Se para uma operação i , $EFT_i > LST_i$, como ilustrado na figura abaixo, então mesmo que a operação seja alocada no início ou no fim da janela, o intervalo de tempo entre EFT_i e LST_i será ocupado obrigatoriamente.

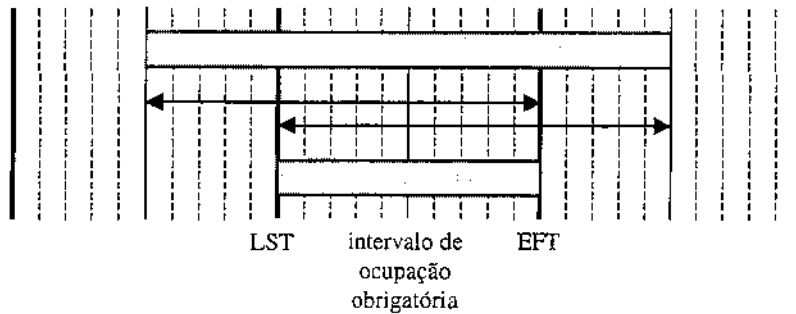


Figura 3.19 – Intervalo de ocupação obrigatória

Ocupações obrigatórias são facilmente identificados analisando as janelas de tempo das operações. Se a janela disponível para execução de uma operação for menor que o dobro do seu tempo de processamento, a janela possui um intervalo de ocupação obrigatória $[LST_i, EFT_i]$.

Para ilustrar a propagação em virtude da restrição imposta pelos intervalos de ocupação obrigatória, considere agora as operações cujas janelas são mostradas na figura 3.20. O tempo de processamento para ambas as operações A e B, em branco e cinza, respectivamente, é de 5 unidades de tempo. As operações utilizam o mesmo equipamento. As áreas hachuradas representam intervalos de ocupação obrigatória dentro das janelas das operações, e que devem ser eliminadas das janelas das operações que competem pelo mesmo equipamento neste intervalo de tempo.

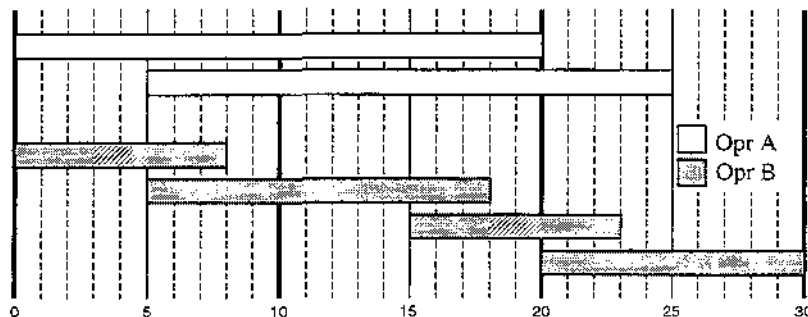


Figura 3.20 – Janelas das operações

A figura 3.21 mostra a propagação destas restrições para as janelas destas operações. Visto que a operação A tem duração de 5 unidades, o intervalo de 0 a 3 antes do primeiro intervalo de ocupação obrigatória é insuficiente para acomodar a batelada A₁, e como não pode ocupar o intervalo de 3 a 5, comprometido com a batelada B₁, esta só poderá ser iniciada no instante 5. O mesmo ocorre com o intervalo 18 a 20 comprometido com a batelada B₃, que impede que as bateladas A₁ e A₂ ocupem este intervalo. As janelas ficariam então como ilustrado abaixo:

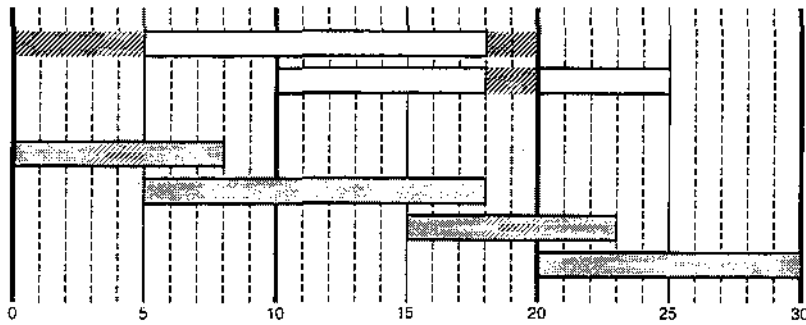


Figura 3.21 – Reduções causadas por intervalos de ocupação obrigatória

3.5.1.3 - Precedência obrigatória entre pares de operações

A análise de precedência descrita a seguir verifica se existe uma ordem de precedência obrigatória entre duas operações que competem pelo mesmo recurso, forçada pela disposição de suas janelas. Dependendo de fatores como o tempo de processamento, o tamanho e a posição relativa das janelas de um par de operações que competem pelo mesmo equipamento, é possível que determinadas alocações de uma das operações violem restrições de capacidade, ou seja, não deixem espaço para a execução da outra operação.

Situações como essa podem ser identificadas a partir do algoritmo conhecido como arbitragem de restrições de precedência (*Precedence Constraint Post*, PCP), apresentado por Cheng e Smith (Cheng C. C., Smith S. F. 1995), que procura identificar casos em que somente um de dois ordenamentos é possível, e então fixá-lo. O algoritmo pode ser resumido pelas quatro condições a seguir :

$$\text{Se } LFT_A - EST_B \geq 0 \text{ e } LFT_B - EST_A < 0 \Rightarrow B \text{ precede } A \quad [3.16]$$

$$\text{Se } LFT_B - EST_A \geq 0 \text{ e } LFT_A - EST_B < 0 \Rightarrow A \text{ precede } B \quad [3.17]$$

$$\text{Se } LFT_A - EST_B < 0 \text{ e } LFT_B - EST_A < 0 \Rightarrow \text{infectível} \quad [3.18]$$

$$\text{Se } LFT_A - EST_B \geq 0 \text{ e } LFT_B - EST_A \geq 0 \Rightarrow \text{ambas as ordens são possíveis} \quad [3.19]$$

Para exemplificar, sejam duas operações A e B, em branco e cinza, respectivamente, ambas atribuídas ao mesmo equipamento, competindo pelo mesmo. A figura abaixo mostra as janelas das operações, cujo tempo de processamento é de 10 unidades para ambas.

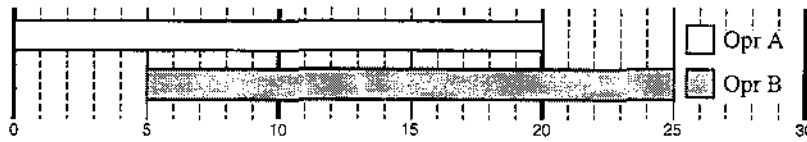


Figura 3.22 – Janelas das operações

É perfeitamente possível alocar a operação A no início e a B no fim das suas respectivas janelas, entretanto não é possível o contrário. Caso se deseje alocar a operação B no início de sua janela, não haverá espaço para a execução de A. O mesmo vale para a operação A, que se alocada no fim de sua janela não deixará espaço para a execução de B, como mostrado nas figuras abaixo. Deste modo, a operação A pode preceder B, mas B não pode preceder A.

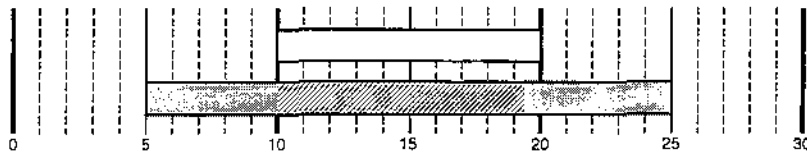


Figura 3.23 – Operação B alocada no início da janela

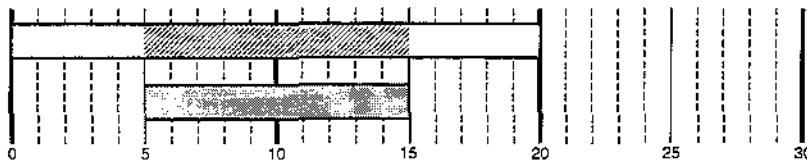


Figura 3.24 – Operação A alocada no fim da janela

Esta precedência forçada é facilmente identificada se constatado que entre o fim da janela de uma operação e o início da outra não existe espaço para a execução de ambas as operações, o que é equivalente a verificar se o LST de uma operação é menor que o EFT da outra. De fato, o LST_A (10) é menor do que o EFT_B (15), e portanto A pode preceder B. Por outro lado, o LST_B (15) é maior do que EFT_A (10), e logo, B não pode preceder A.

Como somente a opção A precede B é válida, a precedência entre as duas operações é obrigatória e os efeitos dessa nova restrição propagados. No caso, como ambas estão competindo pelo mesmo equipamento e não podem utilizá-lo ao mesmo tempo, B deve ser iniciada de modo a permitir o processamento anterior de A, e A deve ser finalizada

de modo a permitir o processamento posterior de B. Essa nova restrição pode ser traduzida nas seguintes equações:

$$\text{Se A precede B então: } LFT_A \leq LFT_B - TP_B \quad [3.20]$$

$$EST_B \geq EST_A + TP_A \quad [3.21]$$

Na figura abaixo é mostrada a propagação em virtude da restrição de precedência obrigatória, aplicando as equações acima.

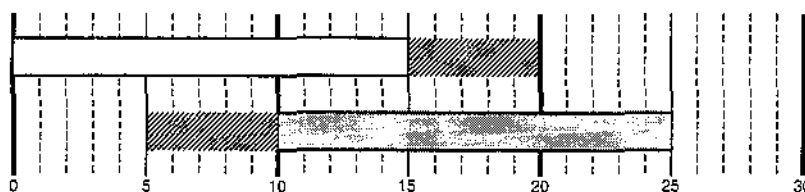


Figura 3.25 – Reduções induzidas pelo ordenamento A precede B

3.5.1.4 - Precedência entre uma operação e um conjunto de operações

Foi visto que pode ser feita uma análise para identificar precedências obrigatórias entre pares de operações. A análise de precedência descrita a seguir verifica se uma determinada operação deve ou não preceder não uma, mas um conjunto de operações. O objetivo dessa análise é a redução da dimensão do problema.

Existem ao todo $(2^n - 1)$ pares para n operações. Uma forma de reduzir a dimensão do problema foi apresentada por Caseau e Laburthe (Caseau Y., Laburthe F. 1995) através da utilização do conceito de intervalo de operações (*task interval*). Um intervalo de operações é definido como o conjunto de operações I construído a partir de duas operações A e B (que podem ser a mesma operação) escolhendo as operações tais que $EST_{i \in I} \geq EST_A$ e $LFT_{i \in I} \leq LFT_B$. Os autores provam que em lugar de considerar todos os pares apenas é necessário considerar estes conjuntos, a vantagem sendo que o número de conjuntos é no máximo n^2 . O ordenamento entre uma operação e um intervalo de operações é deduzido, na referência acima citada, nas seguintes situações:

A) Exclusão: Ordenamento entre uma operação A e um intervalo de operações Ω ($A \notin \Omega$)

Para um conjunto Ω o LFT (ou EST) é definido como o maior LFT (ou menor EST) das operações pertencentes a Ω e o tempo de processamento como a soma dos tempos de processamento das operações. Definindo $\Phi = \Omega \cup A$ a situação será infactível se $LFT_\Phi - EST_\Phi < TP_\Omega + TP_A$. Em uma situação factível o ordenamento pode ser deduzido em dois casos:

(i) se $LFT_{\Omega} - EST_A < TP_{\Omega} + TP_A$ então A não precede o conjunto Ω , e portanto a seguinte restrição deve ser satisfeita :

$$EBT_A \geq \min_{i \in \Omega} (EBT_i + TP_i) \quad [3.22]$$

(ii) se $LFT_A - EST_{\Omega} < TP_{\Omega} + TP_A$ então A não segue o conjunto Ω , e portanto a seguinte restrição deve ser satisfeita.

$$LFT_A \leq \max_{i \in \Omega} (LFT_i - TP_i) \quad [3.23]$$

Nos dois casos podem ser obtidas condições de ordenamento mais fortes. No caso (i) se, adicionalmente $LFT_{\Omega} - EST_{\Omega} < TP_{\Omega} + TP_A$ significa que A não pode ser processada entre as operações i pertencentes a Ω . Dado que já é conhecido que A não precede o conjunto Ω tem-se que Ω precede A. Neste caso EST_A deve ser maior que o EFT do conjunto Ω . Um limitante inferior para este é $EST_{\Omega} + TP_{\Omega}$, portanto:

$$EBT_A \geq EBT_{\Omega} + TP_{\Omega} \quad [3.24]$$

Neste caso também os LFTs das operações i pertencentes ao conjunto Ω devem satisfazer:

$$LFT_i \leq LFT_A - TP_A \quad \forall i \in \Omega \quad [3.25]$$

No mesmo caso (i) a conclusão Ω precede A também é obtida se a operação A deve seguir qualquer operação $i \in \Omega$ porque $EST_A + TP_A > LFT_i - TP_i \quad \forall i \in \Omega$, e neste caso as equações 3.24 e 3.25 devem também ser satisfeitas.

No caso (ii), se adicionalmente $LFT_{\Omega} - EST_{\Omega} < TP_{\Omega} + TP_A$, isto significa que A não pode ser processada entre as operações i pertencentes a Ω . Dado que já é conhecido que A não segue o conjunto Ω , conclui-se que A precede Ω . Neste caso LFT_A deve ser menor que o LBT do conjunto Ω . Um limitante superior para este é $LFT_{\Omega} - TP_{\Omega}$ e portanto:

$$LFT_A \leq LFT_{\Omega} - TP_{\Omega} \quad [3.26]$$

Também neste caso os EST para as operações i pertencentes a Ω devem satisfazer:

$$EBT_i \geq EBT_A + TP_A \quad \forall i \in \Omega \quad [3.27]$$

No mesmo caso (b) a conclusão A precede Ω também é obtida se a operação A deve preceder qualquer operação $i \in \Omega$ porque $LFT_A - TP_A < EST_i + TP_i \forall i \in \Omega$, e então as equações 3.26 e 3.27 devem ser satisfeitas.

B) Edge Finding: Ordenamento entre uma operação A e um intervalo de operações $\Omega (A \in \Omega)$

A situação é infactível se $LFT_\Omega - EST_\Omega < TP_\Omega$. Em uma situação factível podem ser deduzidos ordenamentos em dois casos:

$$LFT_\Omega - EBT_A < TP_\Omega \quad [3.28]$$

então A não precede o conjunto remanescente $\Omega - \{A\}$

$$LFT_A - EBT_\Omega < TP_\Omega \quad [3.29]$$

então A não segue o conjunto remanescente $\Omega - \{A\}$

Nestes casos são obtidas condições idênticas às equações 3.22 e 3.23. No primeiro caso EST_A é modificado para satisfazer:

$$EBT_A \geq \min_{i \in \Omega, i \neq A} (EBT_i + TP_i) \quad [3.30]$$

No segundo LFT_A deve satisfazer:

$$LFT_A \leq \max_{i \in \Omega, i \neq A} (LFT_i - TP_i) \quad [3.31]$$

A determinação de ordenamentos induzidos pela janelas de tempo das operações competindo pelo mesmo equipamento pode originar desta forma reduções nas janelas de tempo através de aumentos no EFT e/ou reduções no LFT.

Para ilustrar, sejam duas operações A e B, em branco e cinza, respectivamente, atribuídas ao mesmo equipamento. A figura abaixo mostra as janelas das operações. O tempo de processamento de A é de 5 unidades de tempo e o de B é de 3.

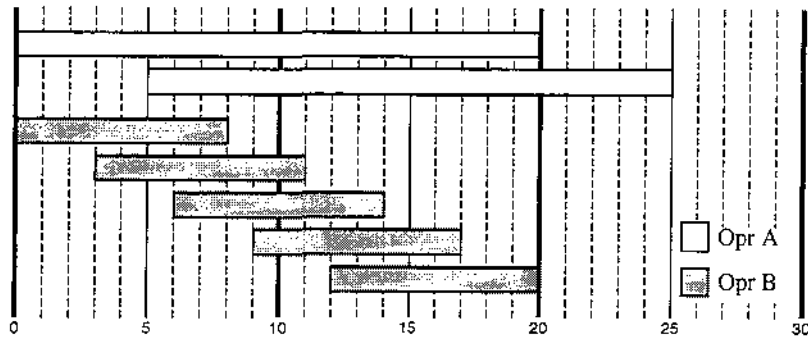


Figura 3.26 – Janelas das operações

Usando as equações para exclusão ($A \notin \Omega$), seja considerado o conjunto Ω como constituído pelas operações A_1 e B_1 a B_5 , e a operação A pela operação A_2 . Desta forma, temos:

$$\begin{array}{ll} EST_A = 5 & EST_\Omega = 0 \\ LFT_A = 25 & LFT_\Omega = 20 \\ TP_A = 5 & TP_\Omega = 5 + 3(5) = 20 \end{array}$$

Substituindo estes valores nas condições (i) e (ii) descritas na exclusão, verifica-se que a primeira é verdadeira e segunda não. Portanto, é possível aplicar a equação 3.22. Além disso, a condição para ordenamentos mais fortes também se verifica. Assim, aplica-se também a equação 3.24. Portanto, o conjunto Ω deve preceder A . As janelas ficariam dispostas da forma indicada na figura 3.27:

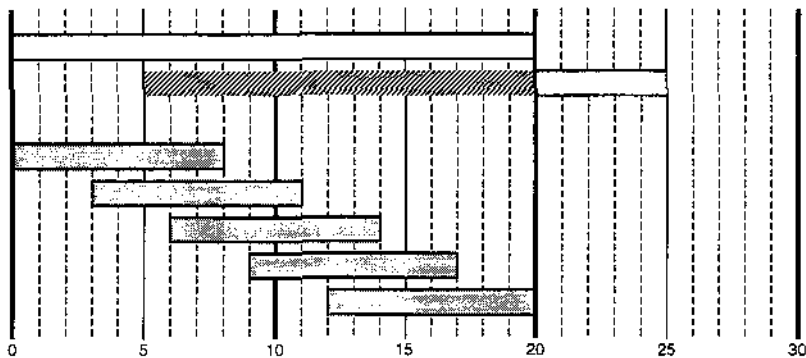


Figura 3.27 – Reduções induzidas pelo ordenamento

Apesar do uso do conceito de *task interval* para diminuir o espaço de busca do problema, as técnicas de exclusão e *edge finding* ainda podem ser onerosas do ponto de vista do esforço computacional. Por esse motivo sistemas comerciais como ILOG *scheduler* e Claire *schedule* permitem ao usuário desabilitar a sua execução.

Para contornar o problema de custo computacional, foi implementado um filtro que condiciona o uso da análise apresentada nesta seção somente quando satisfeitas condições pré-estabelecidas de demanda individual e de folga. O objetivo desse filtro é evitar que sejam realizados os cálculos de exclusão e *edge finding* quando as condições existentes de demanda individual e folga não são críticas, e portanto existem poucas chances de resultarem em redução das janelas. Se não há redução obrigatória das janelas, as restrições do problema na verdade já estão satisfeitas, e esses cálculos são desnecessários. Assim, na análise dos conjuntos de bateladas é verificada se os valores dos parâmetros obedecem às seguintes condições, para realização da análise apresentada nesta seção:

- folga do conjunto \geq folga mínima
- demanda individual média das operações do conjunto \geq demanda individual média mínima
- demanda agregada no intervalo de tempo do conjunto \geq demanda agregada mínima

Esses valores mínimos são definidos pelo usuário. Valores nulos para esses parâmetros levam a uma análise completa.

3.5.2 - Propagação de restrições de balanço de massa

Outras restrições que as janelas das operações devem obedecer dizem respeito à disponibilidade de massa. A fase de planejamento, ao calcular as janelas das operações, estabelece ordens de precedência entre as operações que produzem e que consomem um estado intermediário de forma a satisfazer o balanço de massa. Isso quer dizer que uma operação que consome um estado só poderá ser iniciada quando a operação que produz esse estado tiver produzido massa suficiente para atender as exigências de massa da primeira. A exigência de massa é traduzida por um número inteiro de bateladas da operação que produz o estado a ser consumido posteriormente, seguindo a receita de produção. Assim, as janelas das bateladas relacionadas pelo balanço de massa devem obedecer a ordens de precedência. No trabalho desenvolvido, as precedências por balanço de massa são estabelecidas na fase de planejamento e usadas pelo módulo de *scheduling* como novas restrições a serem satisfeitas. O seguinte exemplo ilustra a propagação por balanço de massa. Suponha que uma operação A produza a cada batelada 50 Kg de um estado intermediário que é consumido pela operação B, cujo tamanho de batelada é de 100 Kg, como indicado na figura 3.28. Para que uma batelada da operação B seja iniciada, é necessário que a operação A tenha produzido 100 Kg do estado intermediário, correspondentes a realização de duas

bateladas. Assim, uma batelada da operação B só poderá ser iniciada após serem finalizadas duas bateladas de A, constituindo uma precedência entre as bateladas que produzem um estado e a que consome, como mostrado na figura 3.29.

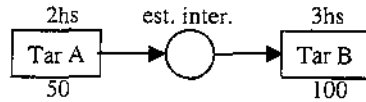


Figura 3.28 – Rota de processamento

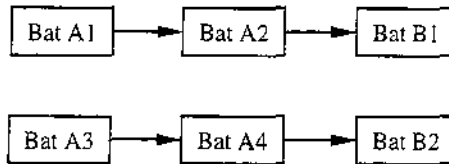


Figura 3.29 – Precedência entre as bateladas

A figura 3.30 mostra as janelas das operações A (branco) e B (cinza), de tempos de processamento de 2 e 3 unidades, respectivamente. As operações estão atribuídas a equipamentos diferentes, e suas janelas não consideraram as restrições de disponibilidade de massa. A figura 3.31 mostra as reduções produzidas nas janelas resultantes da aplicação das restrições de precedência de balanço de massa mostradas na figura 3.29.

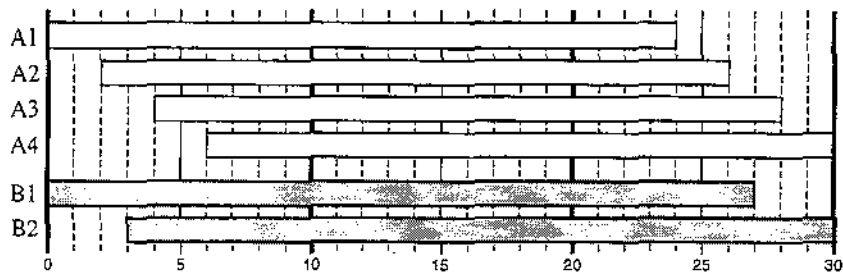


Figura 3.30 – Janelas das operações

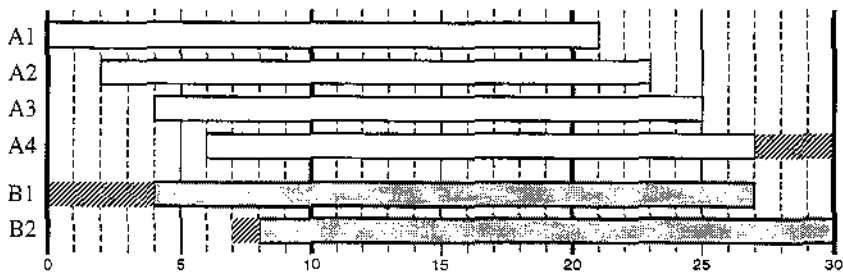


Figura 3.31 – Reduções induzidas por balanço de massa

3.5.3 - Alocação de uma batelada

Quando uma batelada é alocada, se tem a certeza de que o equipamento que ela requer para si estará indisponível para as demais operações que utilizam o mesmo equipamento, no intervalo de tempo de duração da batelada. Na verdade, uma batelada

alocada pode ser considerada como uma janela de tempo com duração igual ao tempo de processamento da operação, o que pode ser interpretado como um intervalo de ocupação obrigatória de duração desta janela. Portanto, as operações que concorrem pelo equipamento com a batelada alocada ficam proibidas de utilizar o intervalo dessa. Assim sendo tanto a propagação das restrições referentes a bateladas de uma operação quanto a de intervalo de ocupação obrigatória descritas anteriormente podem ser imediatamente aplicadas. Desta forma, após a execução da propagação das restrições, não existe qualquer outra operação competindo com a batelada alocada pelo equipamento.

3.5.4 - Restrições de armazenagem

Neste trabalho foi utilizada a propagação das restrições impostas pelas limitações de armazenagem de estados intermediários, sempre presentes em problemas reais da indústria de processos. São quatro os tipos de armazenagem existentes;

- Sem espera (*zero wait* ou ZW)
- Inexistente (*no intermediate storage* ou NIS)
- Limitada (*finite intermediate storage* ou FIS)
- Ilimitada (*unlimited intermediate storage* ou UIS)

A política de processamento ZW é usada quando os estados intermediários não podem ser armazenados para aguardar a próxima etapa de seu processamento, o que obriga a que as bateladas que produzem e consomem um estado estejam sincronizadas, ou seja, a operação seguinte deve ser executada imediatamente após terminada a anterior. Na armazenagem do tipo NIS não existem reservatórios específicos para armazenagem, entretanto o estado intermediário pode ficar estocado no próprio equipamento em que foi processado, até que outra operação consuma esse estado ou o equipamento seja requisitado pela próxima batelada ou outra operação, o que obriga a execução da próxima operação da receita de produção. Na armazenagem do tipo FIS existem reservatórios de capacidade limitada para armazenagem, o que impõe ao problema a restrição de que os estoques de estados intermediários não podem superar a capacidade dos reservatórios. A armazenagem do tipo UIS supõe capacidade de armazenagem ilimitada, e não impõe qualquer tipo de restrição ao problema. Seguem exemplos que ilustram a propagação das restrições em função da política de processamento ZW e restrições de armazenagem NIS e FIS.

a) **Sem espera (ZW)** - Considere duas operações A e B, atribuídas a equipamentos diferentes, como mostrado na figura 3.32. A operação A produz e a B consome um estado intermediário instável. Caso a operação B sofra um aumento de seu EST por algum motivo, a operação A não sofreria nenhuma modificação pelos mecanismos de propagação de restrições descritos nas seções 3.5.1, 3.5.2 e 3.5.3. Entretanto, como a operação A deve se manter em sincronia com a B, esta também sofre um aumento de seu EST de modo que o estado de saída da primeira seja imediatamente transferido para a segunda e essa seja executada. Um raciocínio análogo pode ser feito caso a operação A sofra uma redução no seu LFT.

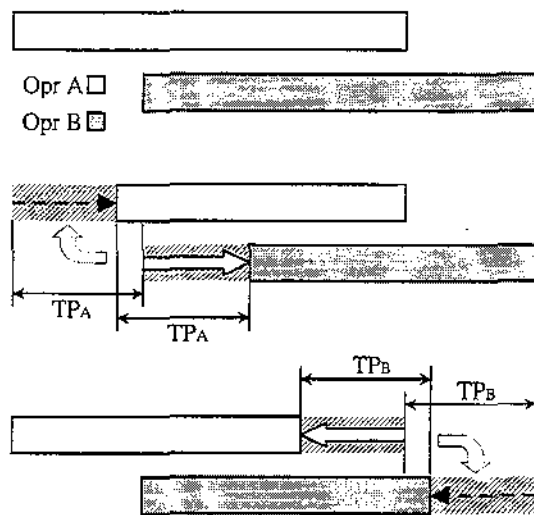


Figura 3.32 – Reduções induzidas por armazenagem ZW

b) **Inexistente (NIS)** - No caso NIS o estado que é gerado pela operação pode permanecer armazenado no equipamento em que foi processado até que seja consumido. Esse procedimento é conhecido na literatura como *hold*, e pode continuar até que o equipamento seja requisitado por outra operação. A partir do instante em que não for mais possível a permanência do estado no equipamento, este deve ser transferido à operação que o consome, que portanto deve ser iniciada nesse instante.

Para ilustrar, considere duas operações A e B, atribuídas a equipamentos diferentes, como mostrado na figura a seguir. A massa produzida pela batelada A_1 é consumida por B_1 , e a produzida por A_2 é consumida por B_2 . A seta cinza indica o intervalo de tempo em que o estado produzido por A_1 pode ficar armazenado no próprio equipamento, no caso em que a alocação seja feita o mais tarde possível.

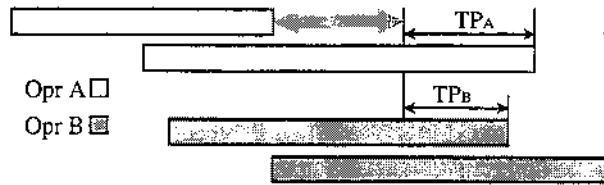


Figura 3.33 – Janelas das operações

Supondo que A_2 sofra uma redução no seu LFT, o intervalo disponível para o *hold* diminui. Como a batelada A_2 vai utilizar o equipamento, o estado nele contido deve ser consumido pela batelada B_1 . Essa batelada portanto, deve ser iniciada assim que o *hold* termina, obrigando a uma redução do seu LFT, como mostra a figura 3.34.

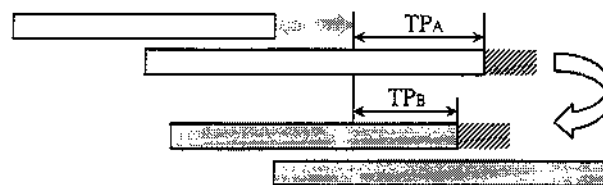


Figura 3.34 – Reduções induzidas por armazenagem NIS

c) **Limitada (FIS)** - No caso FIS, as bateladas da operação que consome o estado estão interligadas com as bateladas que produzem o estado devido ao fato de que o estoque resultante não pode superar a capacidade de armazenagem. A propagação de restrições FIS pode ser determinada considerando a situação em que as bateladas produtoras são processadas no início e as consumidoras no fim das suas respectivas janelas.

A figura abaixo mostra as janelas de duas operações: A produz o estado que é consumido pela operação B, e o perfil de armazenagem do estado intermediário. As operações A e B tem tamanho de batelada de 40 e 15 unidades de massa, e os tempos de processamento são de 3 e 1 unidades de tempo, respectivamente. A capacidade de armazenagem do estado intermediário é de 50 unidades de massa.

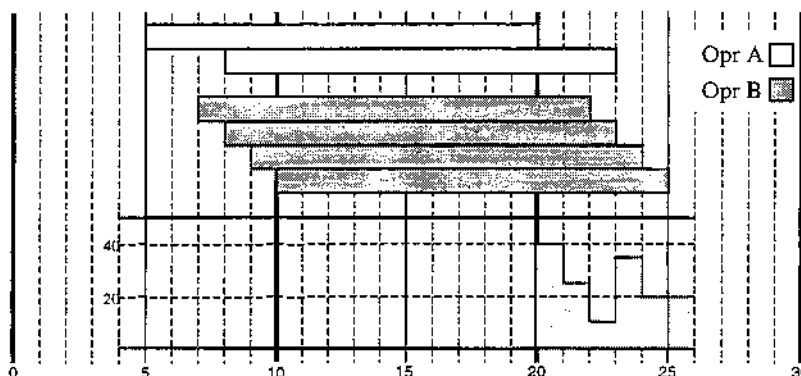


Figura 3.35 – Janelas das operações e perfil de armazenagem

Caso a operação A sofra uma redução no seu LFT (figura 3.36), o estado intermediário será produzido mais cedo, entretanto continuará a ser consumido no mesmo instante. O perfil de armazenagem mostra que há uma produção do estado maior do que a capacidade de armazenagem, e essa quantidade extra deve ser consumida pela operação B antes que isso ocorra. Assim, a operação B também deve ter seu LFT reduzido de forma a restabelecer o equilíbrio entre produção e consumo e respeitar o limite de armazenagem (figura 3.37).

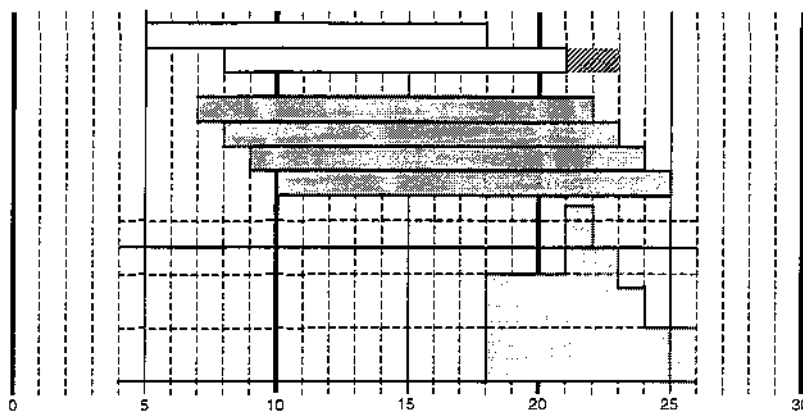


Figura 3.36 – Novo perfil de armazenagem induzido pela redução em A2

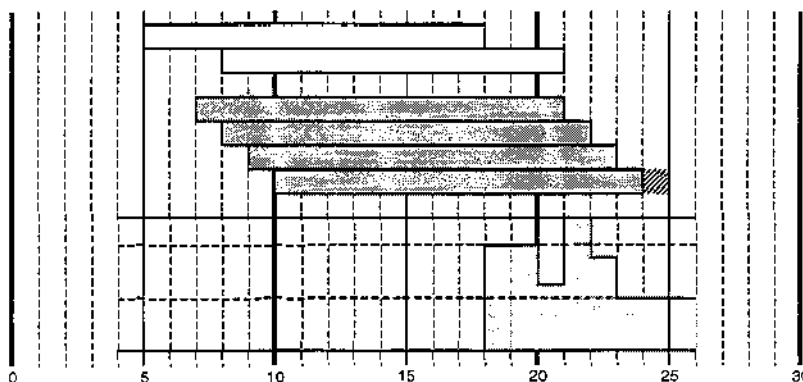


Figura 3.37 – Reduções induzidas por armazenagem FIS

3.5.5 - Funcionamento global dos mecanismos de propagação de restrições

A figura 3.38 mostra o funcionamento conjunto dos mecanismos de propagação de restrições. Após cada passo da propagação, é realizada a análise de factibilidade das janelas. Caso seja detectada infactibilidade em alguma janela a execução dos próximos passos é abortada e o usuário deve modificar a sua decisão.

A tabela 3.3 relaciona as causas e conseqüências das diferentes técnicas de propagação de restrições. O terceiro passo provoca a execução da propagação dos aumentos de todas as janelas que tiveram seu início alterado seguindo a receita de produção. O quinto passo faz o mesmo para as janelas cujo final foi reduzido. O sétimo passo faz uma busca por novos intervalos de ocupação obrigatória caso os passos 4 e/ou 6 tenham definido alguma modificação nas janelas de tempo. Os procedimentos de propagação são repetidos até que nenhuma nova propagação seja detectada (oitavo passo).

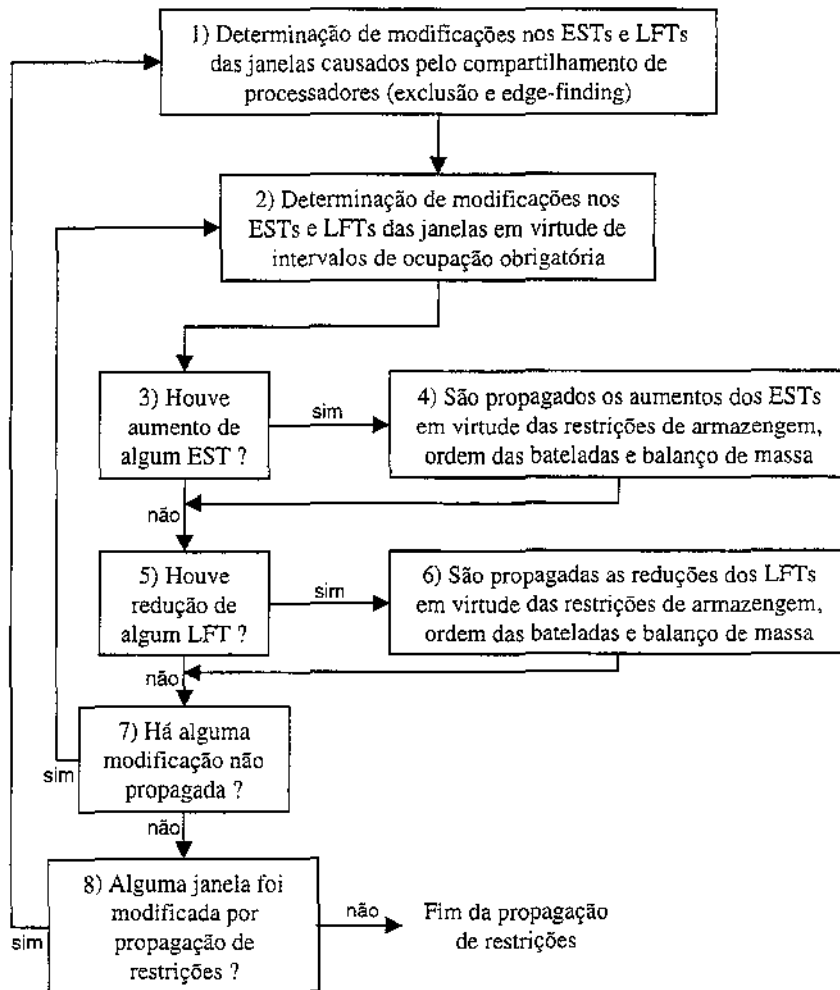


Figura 3.38 – Algoritmo de funcionamento das propagações de restrições

Tabela 3.3 – Causas e conseqüências das técnicas de propagação de restrições

Chamado por	Tipo de propagação	Ativado por	Modificação realizada
passo 4	restrição de armazenagem ZW	aumento do EST da operação consumidora	aumento do EST da operação produtora
passo 6	restrição de armazenagem ZW	redução do LFT da operação produtora	redução do LFT da operação consumidora
passo 6	restrição de armazenagem NIS	redução do LFT da operação produtora	redução do LFT da operação consumidora
passo 4	restrição de armazenagem NIS	aumento do EST da operação consumidora	aumento do EST da operação produtora
passo 4 e 6	restrição de armazenagem NIS	armazenagem obrigatória no equipamento	aumento do LFT ou redução do EST das operações no intervalo de armazenagem
passo 6	restrição de armazenagem FIS	redução do LFT da operação produtora	redução do LFT da operação consumidora
passo 4	restrição de armazenagem FIS	aumento do EST da operação consumidora	aumento do EST da operação produtora
passo 4	ordem fixa entre as bateladas de uma operação	aumento do LST da batelada b	aumento do LST das bateladas $b' > b$
passo 6	ordem fixa entre as bateladas de uma operação	redução do LFT da batelada b	redução do LFT das bateladas $b' < b$
passo 4	balanço de massa	aumento do LST da batelada b	aumento do EST da batelada b' que consome b
passo 6	Balanço de massa	redução do LFT da batelada b	redução do LFT da batelada b' que produz b

3.6 - Abordagem proposta

O sistema proposto é uma evolução em relação às abordagens descritas nas seções 3.3 e 3.4, na medida em que combina e estende suas características. Os sistemas de busca orientada por restrições propostos na literatura se utilizam basicamente de três ferramentas:

- Técnicas de identificação de gargalos (seção 3.3)
- Heurísticas para as decisões de *scheduling* (seção 3.4)
- Mecanismos de propagação de restrições (seção 3.5)

O sistema proposto tem como principais características:

i) A extensão dos mecanismos de propagação de restrições se refere à propagação por restrições de armazenagem, apresentada na seção 3.5.4, inexistente nos métodos apresentados, seja nas heurísticas ou na fase de propagação de restrições.

ii) A extensão da heurística de escolha de valor se refere à utilização da heurística de tendência do menor caminho combinada a um algoritmo de detecção de ciclos para eliminar os ordenamentos que não podem ser arbitrados por já estarem fixos devido a relações de receita de produção.

iii) A extensão das técnicas de identificação de gargalos se refere ao uso combinado da demanda agregada e da medida de folga. Isso se deve ao fato de que as técnicas apresentadas falham em considerar simultaneamente os aspectos da concorrência entre as operações e da capacidade dos equipamentos, como é discutido a seguir.

A idéia é que se um equipamento possui restrições de capacidade fortes, ou seja, pouco tempo disponível para realização de todas as bateladas das operações, uma decisão sobre uma dessas operações terá maior impacto sobre as demais, reduzindo a flexibilidade de alocação dessas. Existe assim uma propagação mais forte dos efeitos de uma decisão sobre as demais operações na receita do produto e que compartilham o mesmo equipamento, comparativamente a uma situação com maior “folga” de tempo.

Para exemplificar a importância dessas restrições em um gargalo, considere as operações A, B e C, em cinza escuro, cinza claro e branco, respectivamente, ilustradas na figura 3.39. Na mesma figura é mostrada a curva de demanda agregada para essas operações. Existe um intervalo com um alto pico de demanda agregada constituído por operações de demanda individual de baixo valor (A e B) e apenas uma de alto valor (C).

Apesar de aparentemente existir uma concorrência forte entre as operações envolvidas, o gargalo não é tão grave quanto se pode pensar, pois apenas operação C possui pequena flexibilidade de alocação, e uma decisão sobre ela não trará grandes perturbações às janelas das demais, que possuem grande flexibilidade.

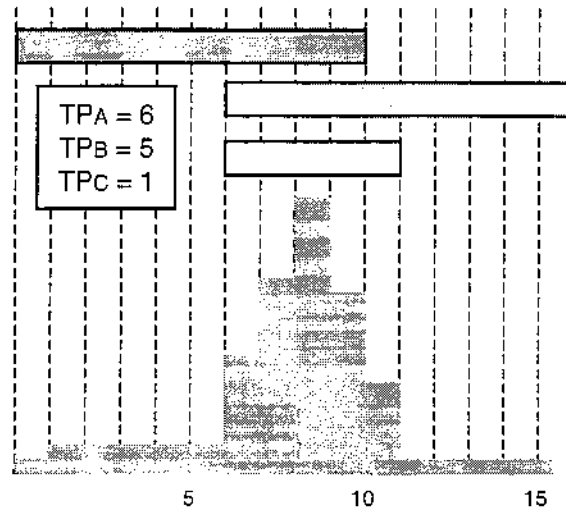


Figura 3.39 – Janelas das operações e curva de demanda agregada

Considere agora as mesmas janelas, porém com tempos de processamento diferentes, como mostrado na figura 3.40. Nessa nova situação, existe um pico de demanda agregada de mesmo tamanho, porém menos acentuado. A alocação da operação A ou B reduzirá a flexibilidade de alocação das demais muito mais do que no primeiro caso. Isso se deve ao fato de no primeiro caso a folga ser maior do que no segundo, já que os tempos de processamento de A e B são maiores apesar das janelas serem iguais. Essa folga pode ser estimada através do tempo total livre, por exemplo.

no primeiro caso: $16 - 2 - 1 - 4 = 9$ unidades de tempo

no segundo caso: $16 - 6 - 5 - 1 = 4$ unidades de tempo

Essa flexibilidade das operações concorrentes pode ser estimada através do conceito de folga, descrito na seção 3.3.3, estendendo esse conceito para um conjunto de operações.

Propõe-se então unir as técnicas de demanda agregada e de folga de modo a abordar os gargalos resultantes da concorrência que possuam também menor folga, pois espera-se que decisões precoces sobre gargalos tenham um efeito de redução da árvore de busca.

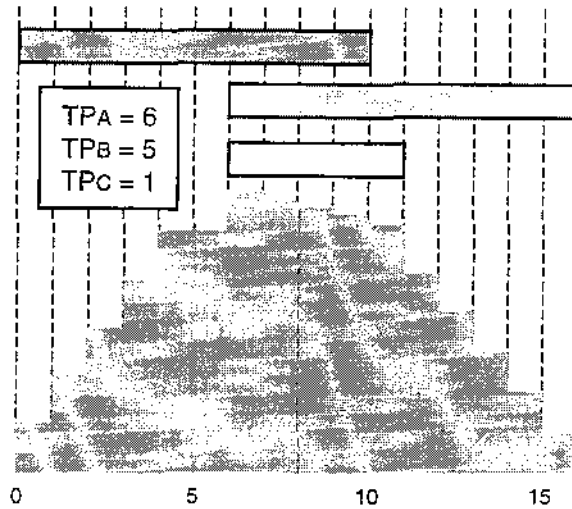


Figura 3.40 – Janelas das operações

Na estratégia proposta são inicialmente selecionados intervalos onde a curva de demanda agregada é superior a um valor pré-estabelecido pelo usuário (neste trabalho foi utilizado um valor unitário para este parâmetro). Para cada intervalo selecionado, são identificadas as operações que possuem parte da janela dentro deste intervalo, ou seja, cuja janela contribua para o gargalo nesse intervalo. Assim, cada intervalo j selecionado tem associado a si um conjunto I de operações. Para cada um desses conjuntos de operações, é feita uma medida de folga, descrita pela equação abaixo:

$$Folga_j = \frac{\left[\max_{i \in I} (LFT_i) - \min_{i \in I} (EST_i) \right] - \sum_{i \in I} TP_i}{\max_{i \in I} (LFT_i) - \min_{i \in I} (EST_i)} \quad [3.32]$$

Quanto menor a folga, menor a flexibilidade de alocação das operações que pertencem a esse conjunto, e portanto o intervalo ao qual essas operações estão associadas é definido como o gargalo mais crítico. Para exemplificar, considere o conjunto de janelas mostrado na figura 3.41. As operações A (branco) e B (cinza) possuem tempo de processamento de 4 e 2 unidades de tempo, respectivamente. Nos intervalos de tempo [10, 14] e [19, 24] a curva de demanda agregada assume valores superiores à unidade, constituindo um gargalo.

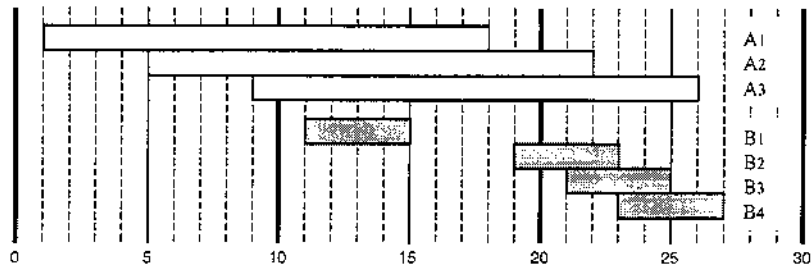


Figura 3.41 – Janelas das operações

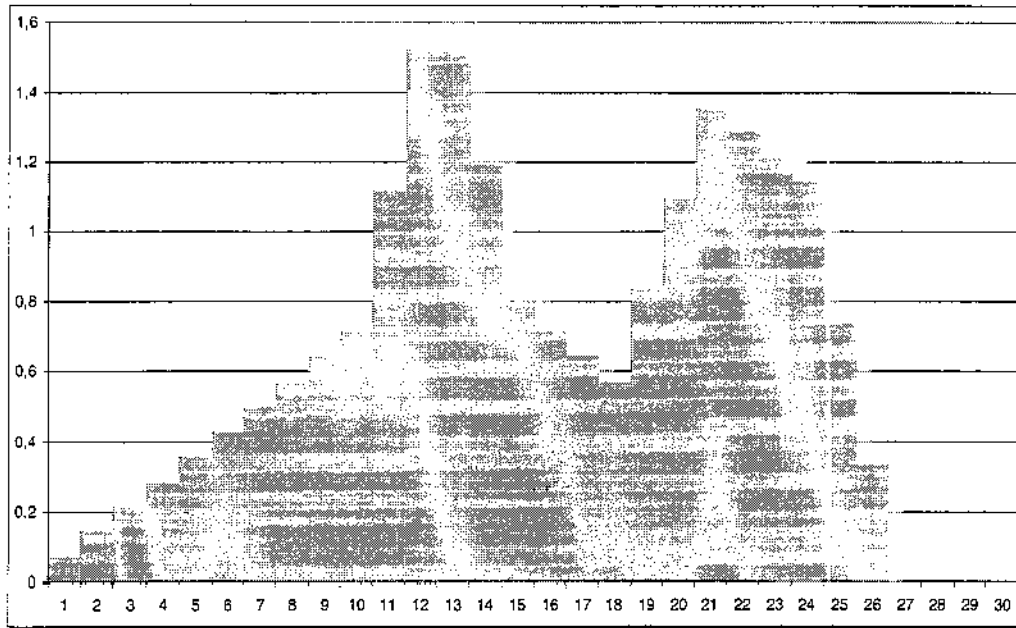


Figura 3.42 – Curva de demanda agregada para as janelas da figura 3.41

Estão envolvidas no primeiro intervalo as operações A_1 a A_3 e B_1 , pois suas janelas contribuem para a curva de demanda agregada neste intervalo. Do mesmo modo, estão envolvidas no segundo intervalo as operações A_2 e A_3 e B_2 a B_4 . Assim, é calculada a folga para este conjunto de operações, de acordo com a equação 3.32, que fornece os resultados abaixo. Como a folga do segundo intervalo é menor, esse é considerado o gargalo mais crítico.

$$\text{Folga}_{\text{intervalo 1}} = 0,44$$

$$\text{Folga}_{\text{intervalo 2}} = 0,32$$

As decisões de *scheduling* utilizadas são decisões de ordenamento entre pares de operações. A heurística de escolha de variável que seleciona o par de operações utiliza os conceitos da tendência do menor caminho (seção 3.4.4), apoiada por um algoritmo de detecção de ciclos em grafos. Este algoritmo é discutido na próxima seção. Seu objetivo é limitar a escolha do par de operações aos pares que ainda não possuem um ordenamento

entre si, visto que devido à disposição das janelas e conexões da receita de produção, não é óbvio quais ordenamentos já estão realmente fixados.

A abordagem apresentada acima caracteriza-se então por:

- a) Decisões sobre ordens de precedência
- b) Uso da demanda agregada para identificar gargalos de concorrência
- c) Uso do tempo de folga para avaliar restrições de capacidade do equipamento entre os gargalos
- d) Uso de restrições de compartilhamento de equipamentos, balanço de massa e de armazenagem na propagação de restrições
- e) Uso de técnica de detecção de ciclos em grafos para eliminar ramos inactíveis da árvore de busca

A proposta **a** garante um menor espaço de busca, em virtude do menor número de decisões que existe para um par de operações, como descrito na técnica de arbitragem de precedência (seção 3.4.3). Visto que as decisões de *scheduling* são sobre ordens de precedência, a dimensão do problema é grande, como explicado anteriormente. Para reduzir essa dimensão, a proposta **b** faz uso de informações fornecidas pela curva de demanda agregada para atacar primeiro regiões de maior concorrência e selecionar o conjunto de operações sobre os quais será feita a análise para decidir a precedência, de modo que esta seja realizada somente entre os pares de operações envolvidos em regiões de alta concorrência. A proposta **c** pretende distinguir entre as regiões de conflito aquela que possui maior restrição de tempo de folga, de forma a atacar primeiro a região mais crítica do que a que seria detectada somente com o uso da demanda agregada.

A proposta **d** tem o objetivo de reduzir o espaço de busca, pela redução das janelas das operações de forma a adequá-las às decisões já tomadas. A proposta **e** reduz o espaço de busca eliminando decisões que não podem fazer parte da solução final do problema, através da detecção de ordenamentos inactíveis. Essas duas últimas propostas reduzem, conseqüentemente, a quantidade de resultados inactíveis e a de *backtracking*. A proposta **d** incorpora a propagação das restrições de armazenagem na solução do problema, visto que condições duras de armazenagem dos estados intermediários tornam o problema mais restrito.

3.6.1 - Consolidação do espaço de busca pela identificação de ciclos

Em receitas complexas, geralmente presentes na indústria de processos, as possíveis decisões de ordenamento entre as operações em um equipamento estão restritas

não apenas pelas suas janelas mas também pelas relações de precedência provenientes da receita e do balanço de massa. Como essas relações podem ser muito complexas, se torna necessário analisá-las em conjunto para determinar quais decisões de ordenamento são possíveis. Neste trabalho utiliza-se uma representação de grafos para determinação dos ordenamentos possíveis através de um algoritmo de detecção de ciclos.

Grafos são uma representação matemática de objetos interligados em um sistema, onde as relações entre estes objetos representam um papel dominante. Os grafos podem ser usados para modelar redes, estruturas de dados e *scheduling* de processos, entre outras aplicações. Um grafo consiste de um conjunto de elementos denominados vértices conectados entre si por arestas (ou arcos).

A figura abaixo mostra a representação geométrica de um grafo. Os vértices são mostrados como círculos e os arcos como linhas conectando pares de círculos.

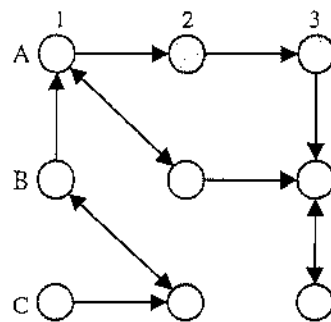


Figura 3.43 – Representação de um grafo

No grafo mostrado acima, os arcos que unem os vértices indicam uma relação entre os dois. Alguns desses arcos não têm uma direção específica, indicando que não existe uma ordem de precedência definida, e outros possuem uma orientação, de forma que está fixada uma relação de precedência entre os dois vértices.

No trabalho desenvolvido, as operações que constituem o processo são representadas por um grafo, onde os vértices representam as operações e os arcos as relações de precedência entre elas. Como neste trabalho a natureza das decisões tomadas no *scheduling* são de ordenamento entre as operações, se torna necessário identificar quais precedências podem ser arbitradas e quais estão fixas, que são representadas por arcos disjuntivos (não orientados) e fixos (orientados), respectivamente. Algumas ordens de precedência estão fixadas de início, devido aos ordenamentos obrigatórios impostos pelas janelas obtidas na etapa de planejamento. As precedências ou arcos fixos são identificados

pelo método de precedência entre as bateladas de uma operação e entre pares de operações (seções 3.5.1.1 e 3.5.1.3).

O grafo é de grande utilidade na visualização de quais ordenamentos já estão realmente fixados. Os arcos que não foram fixados pelos métodos acima citados não são necessariamente disjuntivos. Em virtude da relação com outras operações alguns arcos que estão definidos como disjuntivos são de fato arcos fixos. Esta situação pode ser identificada com a ajuda da ferramenta de detecção de ciclos em grafos.

O ciclo é um circuito fechado de arcos fixos que é formado quando se toma uma decisão de ordenamento que, em princípio, não viola as restrições impostas pelos ordenamentos já existentes. A figura 3.44 mostra um exemplo. Sejam duas operações A e B de tempo de processamento de 2 e 3 unidades, respectivamente. Ambas as operações estão atribuídas ao mesmo equipamento e competem pelo mesmo. O grafo correspondente mostrando os possíveis ordenamentos é mostrado em seguida.

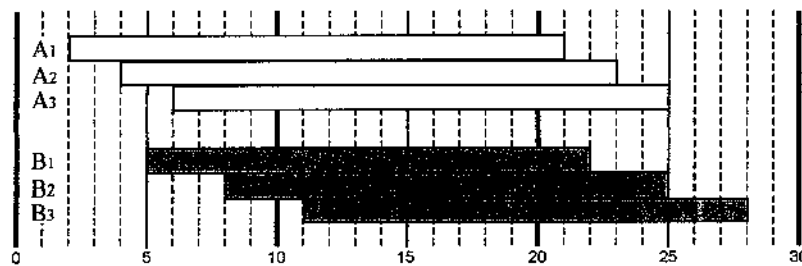


Figura 3.44 – Janelas das operações

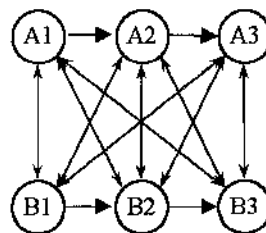


Figura 3.45 – Representação das operações em forma de grafo

As setas azuis representam arcos fixos por precedências entre bateladas da mesma operação. As setas verdes representam os arcos disjuntivos. Todos os ordenamentos entre as operações concorrentes não estão definidos e qualquer um pode ser escolhido para ser fixado. Suponha que seja fixado um ordenamento tal que A_2 precede B_1 . Com a propagação das restrições de precedência entre uma par de operações (equações 3.20 e 3.21) as novas janelas seriam como as mostradas abaixo. O grafo para essa nova situação é mostrado na figura 3.47.

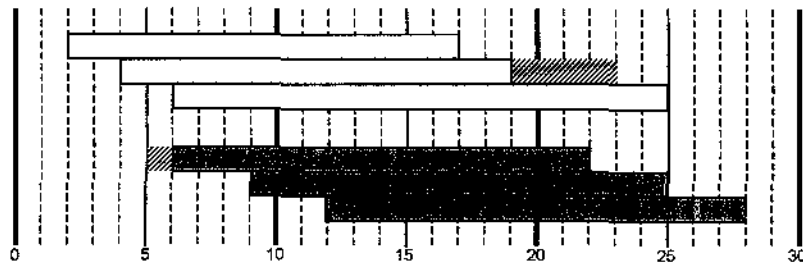


Figura 3.46 – Novas janelas após ordem A2 precede B1

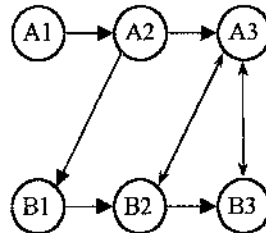


Figura 3.47 – Novo grafo após ordenamento

Após fixado o arco, somente alguns ordenamentos são possíveis. Entretanto, como as janelas das operações ainda são relativamente grandes, aparentemente seria possível estabelecer outros ordenamentos além dos mostrados no grafo. Por exemplo, considerando apenas as janelas de tempo, de acordo com os conceitos de precedência entre pares de bateladas (seção 3.5.1.3), as operações A_1 e B_2 não estariam obrigatoriamente ordenadas, pois não é verificado nenhum ordenamento obrigatório (equações 3.16 e 3.17). Portanto, poderia ser fixada a ordem B_2 precede A_1 , por exemplo. Porém, devido à ordem fixada anteriormente, essa nova precedência é inconsistente, pois conduz ao ciclo de operações A_1 , A_2 , B_1 , B_2 e novamente A_1 . Está então formado um ciclo, que uma vez identificado deve eliminar das possibilidades de decisão o arco que o gerou. Assim, fixando apenas uma precedência, o número de arcos disjuntivos foi reduzido de nove para apenas dois, pois os demais arcos disjuntivos são redundantes ou podem conduzir a um ciclo caso fixados.

O algoritmo de detecção de ciclos (Canton, 2000) procura e marca o vértice inicial, aquele que não possui precedente, que no exemplo citado, pode ser a operação A_1 ou B_1 . Este vértice é marcado como um vértice já visitado, e a partir daí é feita uma varredura em todo o grafo, marcando os demais vértices cujos precedentes já foram marcados. Os arcos disjuntivos são temporariamente fixados um de cada vez para verificar se podem conduzir a um ciclo, caso fixados. Caso exista um ciclo, o algoritmo só conseguirá marcar os vértices que pertencem a este ciclo, e não será possível marcar os demais. Fica assim caracterizado o ciclo no arco disjuntivo temporariamente fixado, que é

então removido da lista de possíveis decisões. O algoritmo é mostrado no fluxograma a seguir:

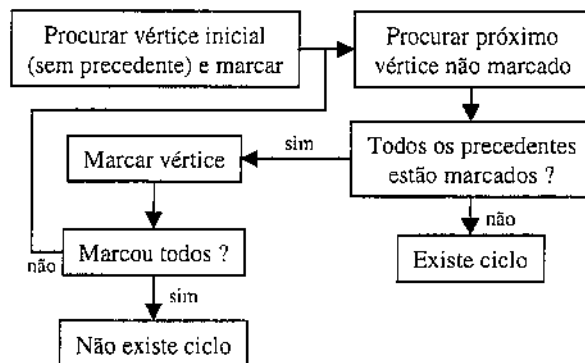


Figura 3.48 – Algoritmo de detecção de ciclos

3.6.2 - Limitações de armazenagem

Limitações de armazenagem de estados intermediários estão sempre presentes na indústria de processos. Os estados sempre relacionam pelo menos duas operações, a que produz e a que consome o estado. Quando existem condições duras de armazenagem desses estados, a propagação da decisão em uma dessas operações para a outra é mais forte.

É interessante investigar a interferência das restrições de armazenagem no processo de busca por uma solução. Se as restrições de armazenagem são fortes, desempenhando um papel importante no processo de busca, essas podem ser consideradas também como um gargalo, e pode ser estabelecido um modelo para acrescentar o uso de informações sobre as condições de armazenagem como critério para orientar a busca.

Capítulo 4 - Resultados

Neste capítulo são mostrados os resultados obtidos comparando a abordagem proposta com as existentes. A seção 4.1 descreve a natureza e as características do exemplo considerado. O desempenho da heurística proposta é comparado com as técnicas existentes nas seções 4.2.1, 4.2.2 e 4.2.3. Na seção 4.2.4 é discutida uma nova heurística de ordenamento. Na seção 4.2.5 é feita uma análise do uso das restrições de armazenagem para orientar as decisões de ordenamento. Finalmente, na seção 4.3 é realizado um estudo do uso da heurística proposta em problemas de otimização.

4.1 - Exemplos considerados

O exemplo considerado neste trabalho é o apresentado por Papageorgiou e Pantelides (Papageorgiou L. G., Pantelides C. C. 1996). A representação STN do exemplo considerado é mostrada na figura 4.1, onde constam as operações, tamanhos de batelada, tempos de processamento, estados e condições de armazenagem. A atribuição das operações aos equipamentos e condições de armazenagem estão detalhados na tabela 4.1. A demanda de produtos finais é mostrada na tabela 4.2.

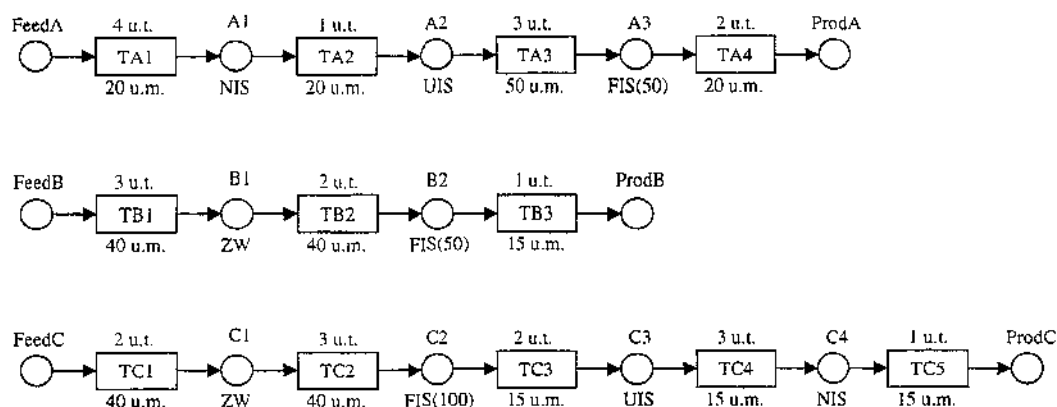


Figura 4.1 – STN do processo

Tabela 4.1 – Habilitação das operações e condições de armazenagem

Operações	Equipamento	Estados	Armazenagem
TA1, TC1	P1	B1, C1	ZW
TA2, TA4	P2	A1, C4	NIS
TB1	P3	A3, B2	FIS (50)
TB1, TC3	P4	C2	FIS (100)
TB2, TC2	P5	A2, C3	UIS
TB3, TC3	P6		
TA3, TC5	P7		

Tabela 4.2 – Demanda e data de entrega dos produtos

Produto	Massa	Data de Entrega	Massa	Data de entrega
Prod A	100	32	90	56
Prod B	270	32		
Prod C	75	56		

As datas de disponibilidade de todas as matérias-primas são fixadas em $t = 1$, obtendo-se como resultado final da fase de planejamento as janelas de processamento das operações e a demanda agregada nos equipamentos apresentadas na figura 4.2.

A função objetivo considerada é a minimização do adiantamento dos LFTs (*earliness*). O *earliness* é definido pela seguinte equação:

$$earliness = \sum_{i \in I} (DD_i - TF_i) \quad [4.1]$$

onde: I = conjunto de bateladas que produzem um produto final
 DD_i = instante de entrega da batelada i
 TF_i = instante de término da batelada i

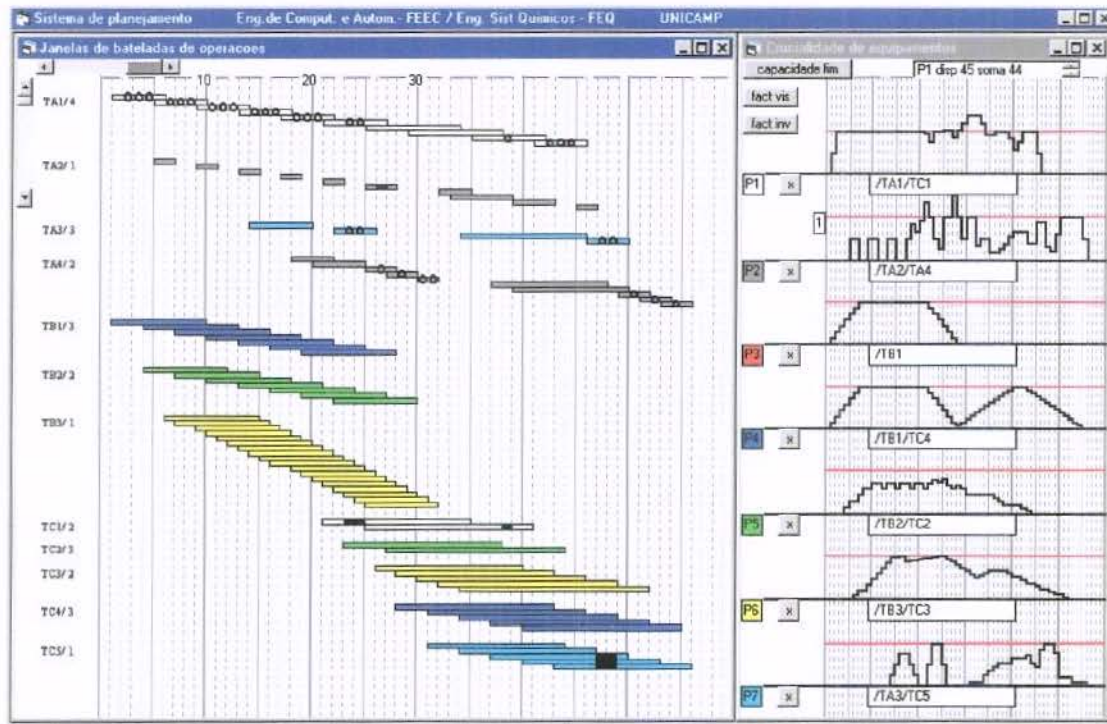


Figura 4.2 – Janelas de tempo das operações

Para definir quais ordenamentos estão fixos são realizadas a análise de precedência obrigatória entre pares de operações e a análise de ciclos em grafos. Para as janelas da figura acima obtêm-se os arcos ilustrados na figura 4.3. As setas vermelhas são arcos fixos (ordenamentos fixos) e as verdes arcos disjuntivos (ordenamentos que podem ser fixados). As setas azuis são arcos fixos provenientes do ordenamento fixo entre bateladas da mesma operação.

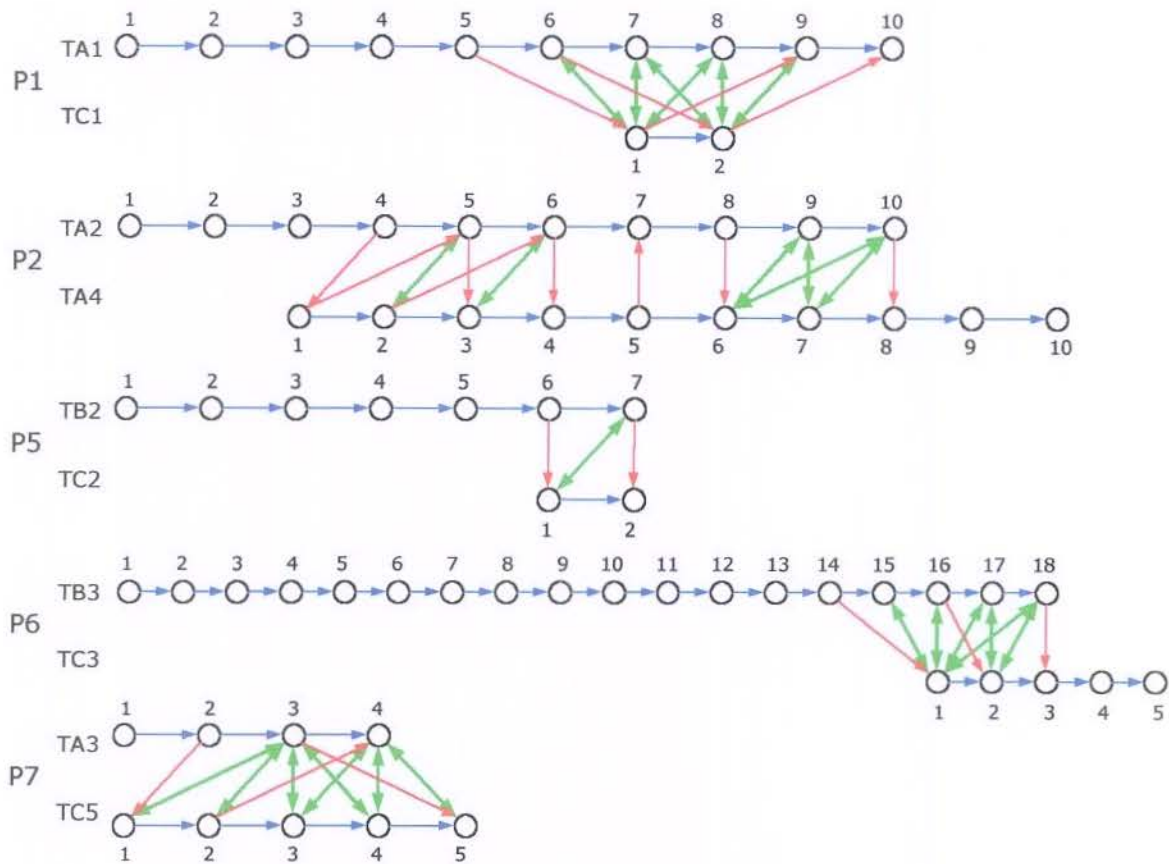


Figura 4.3 – Arcos fixos e disjuntivos para as janelas da figura 2.1

4.2 - Uso da heurística proposta para busca em profundidade

A abordagem proposta na seção 3.6 será ilustrada na determinação de um limitante superior para redução do espaço de busca. Nesse contexto é realizada uma busca em profundidade para obtenção de uma solução factível o mais rápido possível, que servirá então de limitante superior para dirigir a busca por uma solução ótima. O objetivo é mostrar que o uso da demanda agregada e da propagação das restrições de armazenagem permitem atingir uma solução final com menos nós visitados.

4.2.1 - Uso da demanda agregada na HEVar

No método apresentado por Smith para a resolução do problema (seção 3.4.3), as decisões de *scheduling* são tomadas com respeito à ordem de precedência do par de operações escolhido. De acordo com a HEVar deste método, o par de operações é escolhido através de uma análise exaustiva da folga considerando a combinação de todos os pares de operações possíveis que concorrem pelo mesmo equipamento. Na figura 4.2, ainda que

considerando somente os pares de operações concorrentes cujas janelas se sobrepõem, são ao todo 80 pares de bateladas para a análise de folga.

Para diminuir a dimensão do conjunto a ser analisado, utiliza-se as informações fornecidas pela curva de demanda agregada, onde o valor mínimo de demanda agregada usado para indicar a existência de um gargalo é igual à unidade, como descrito na seção 3.6. Assim, obtêm-se três intervalos de tempo considerados gargalos, listados abaixo juntamente com as operações envolvidas nestes intervalos.

Tabela 4.3 – Intervalos de gargalos

Equipamento	Intervalo	Operações Envolvidas
P1	29 - 34	TA1/7 e 8; TC1/1 e 2
P2	21 - 22	TA2/5; TA4/1 e 2
P2	27 - 28	TA2/6; TA4/3 e 4

Com o auxílio da demanda agregada existem então apenas 8 pares de operações para analisar, uma quantidade menor do que a exigida anteriormente. Certamente uma decisão sobre um desses oito pares terá impacto muito maior sobre a solução, por estarem envolvidas em gargalos. As decisões de ordenamento entre um par de operações foram tomadas usando o algoritmo da *tendência do menor caminho*, de modo a realizar a menor redução possível das janelas e preservar maior flexibilidade para alocação.

Foi implementada inicialmente uma busca em profundidade automática até todos os ordenamentos estarem estabelecidos. Neste problema foram necessárias 12 decisões. A árvore de busca contendo as decisões tomadas e as janelas finais resultantes dessas decisões são exibidas na figura 4.4. O número entre parênteses ao lado de cada decisão é uma estimativa mínima do custo associado a essa decisão, calculada através da equação 4.1. Trata-se de uma estimativa pois como as bateladas estão na forma de janelas, não é conhecido ao certo qual o seu instante de término. Assim, o cálculo é realizado assumindo que a batelada poderá ser alocada o mais tarde possível, ou seja, a parcela TF_i da equação 4.1 é substituída pelo LFT da janela, porém esse valor pode ser alterado quando a batelada for alocada de fato.

As decisões se concentram principalmente nos equipamentos P1 e P2, pois estes apresentam a maior carga imposta pelas janelas. Em virtude disso, as decisões tomadas nas operações habilitadas a estes equipamentos, TA1/TC1 e TA2/TA4, têm grande impacto na redução das suas janelas.

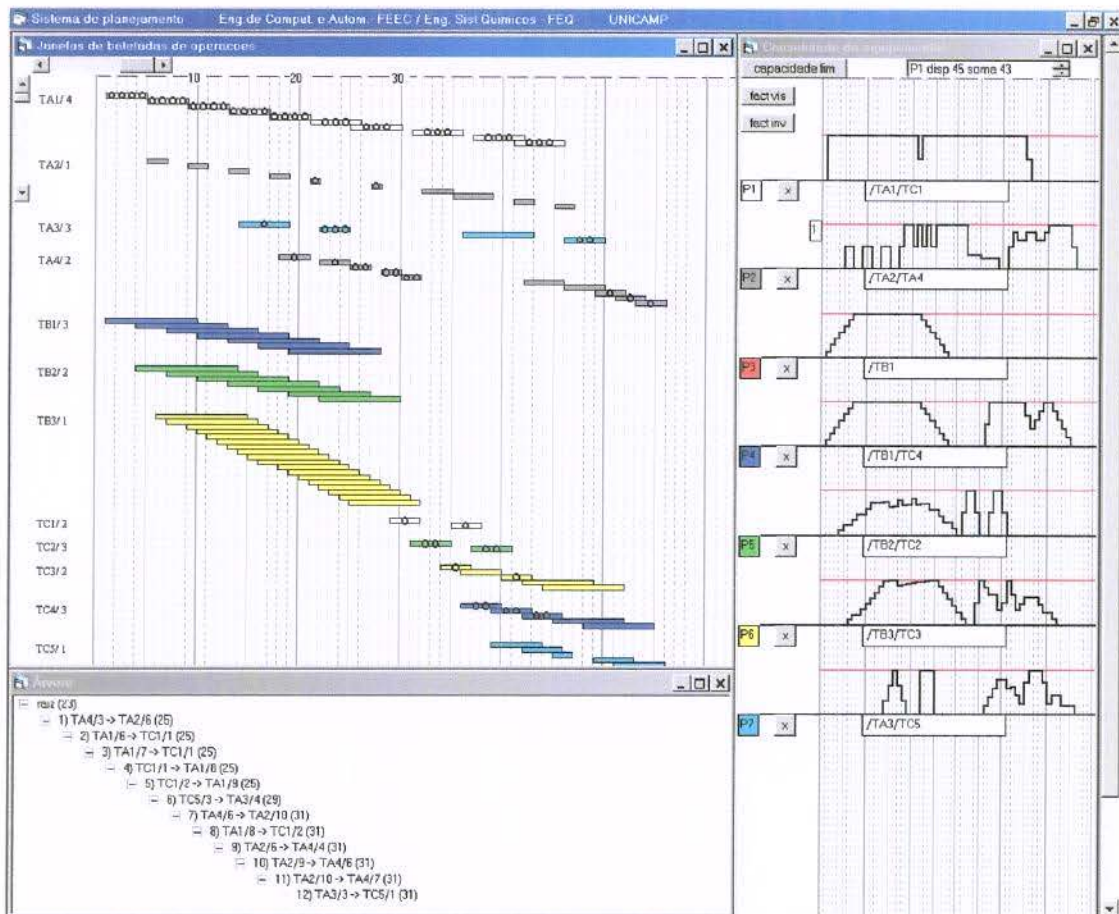


Figura 4.4 – Janelas finais da proposta 4.2.1

A influência das bateladas envolvidas em gargalos sobre as demais é visível quando se consideram as reduções nas janelas das operações que não participaram das decisões de ordenamento. Isso é verificado nas operações TC2 a TC5, que através do balanço de massa, foram bastante reduzidas pelas decisões sobre TC1 e também TC5. O uso da demanda agregada portanto direcionou a heurística de escolha de variável proposta por Smith, de forma que todas as ordens de precedência fossem estabelecidas tomando decisões sobre um pequeno conjunto de operações, evitando gasto de tempo na análise de todos os 80 pares de bateladas.

4.2.2 - Uso da demanda agregada e da propagação de restrições de armazenagem

No exemplo anterior as janelas fornecidas pela fase de planejamento obedecem as restrições impostas pelas limitações de armazenagem. Essas restrições no entanto não foram consideradas no processo de busca, ou seja, a propagação de restrições após cada decisão na árvore não incluiu a propagação de restrições de armazenagem. Por esse motivo qualquer resultado obtido desta maneira corre o risco de não respeitar essas restrições. De

fato, uma rápida inspeção na solução obtida mostra que entre as bateladas TA1/1 e TA2/1 não existe armazenagem disponível (NIS), e portanto essa última não pode ser iniciada no instante 6, como permite a sua janela. Esta batelada deve obrigatoriamente ser iniciada no instante 5.

A adição de restrições induzidas pelas limitações de armazenagem na forma de propagação de restrições a cada decisão tomada na árvore garante que as janelas das operações se adaptem às exigências de armazenagem, evitando assim que assumam valores incorretos de EST e LFT.

Resolvendo o exemplo anterior novamente, com as técnicas já descritas e em adição considerando a propagação das restrições de armazenagem a cada decisão de ordenamento fixada, obtêm-se a solução da figura 4.5.

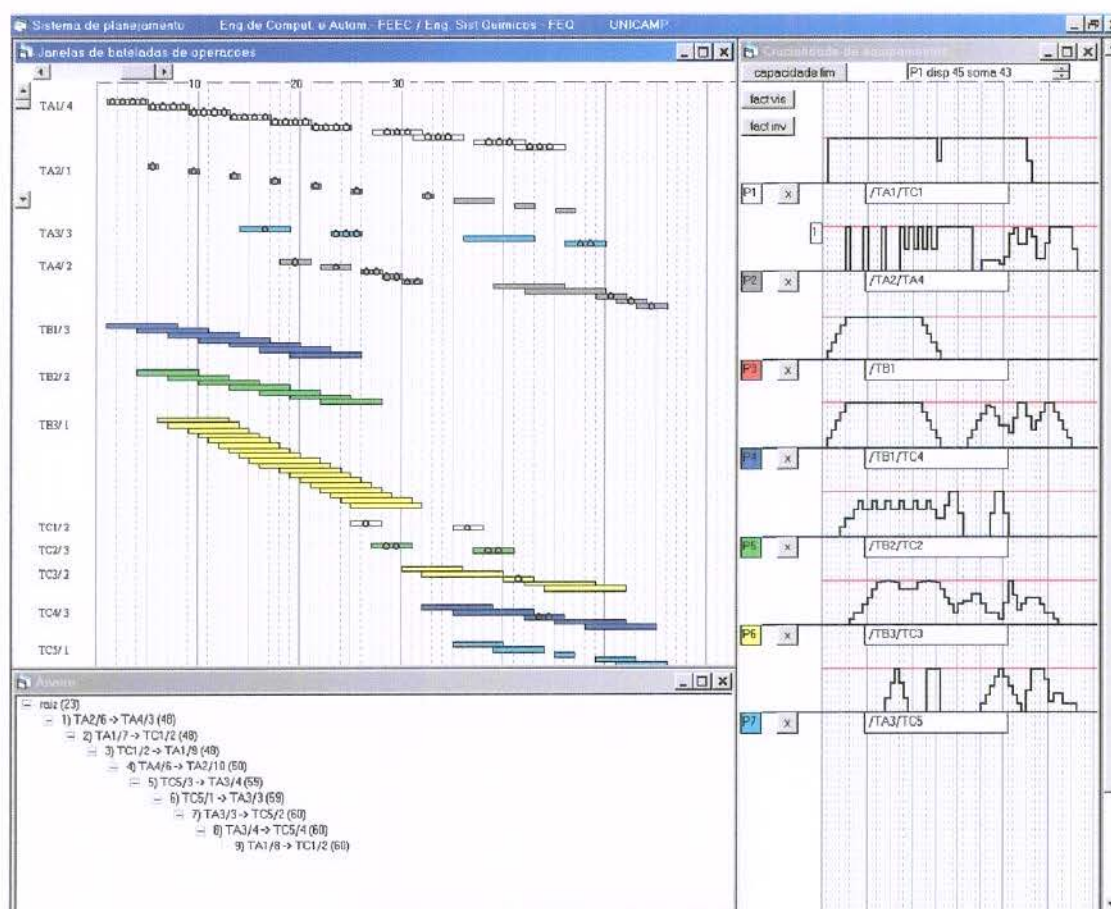


Figura 4.5 – Janelas finais da proposta 4.2.2

Pode-se notar uma redução maior nas janelas das operações em função das condições de processamento ZW e condições de armazenagem NIS e FIS. Nas operações TA1 e TA2, em particular nas bateladas 1 a 6, a redução ocorre devido à condição de

armazenagem NIS. Há também uma grande redução nas janelas das operações TB1 a TB3, porém nenhuma decisão foi tomada envolvendo essas operações. Essas reduções se devem exclusivamente à propagação de restrições de armazenagem, e são verificadas já na primeira decisão (TA2/6 precede TA4/3), como mostrado nas figuras 4.6 e 4.7.

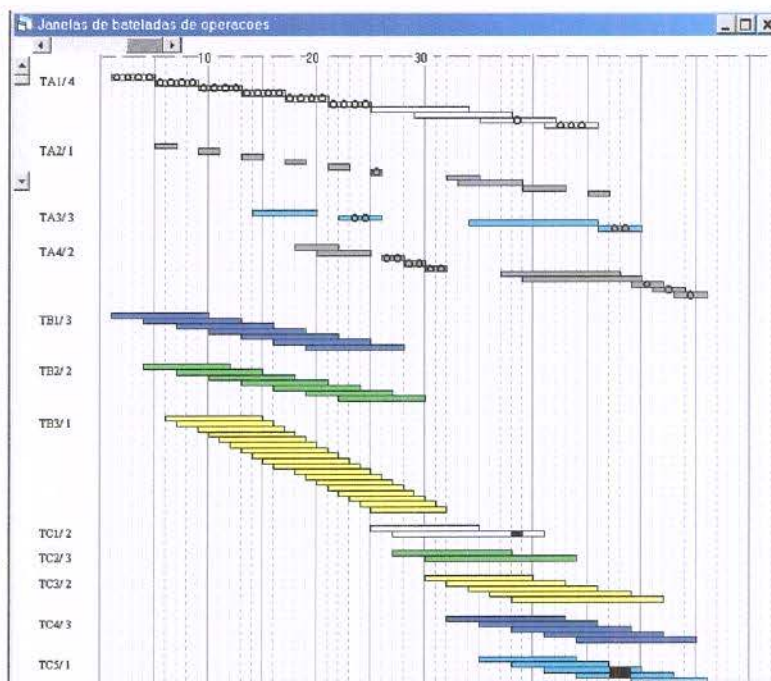


Figura 4.6 – Janelas após a primeira decisão (TB2/6 precede TB4/3) sem propagações por armazenagem

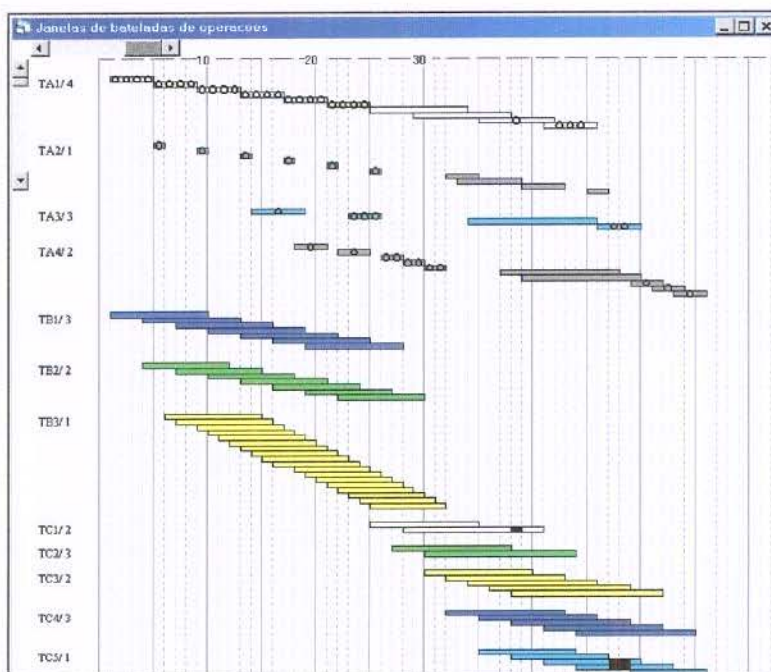


Figura 4.7 - Janelas após a primeira decisão (TB2/6 precede TB4/3) com propagações por armazenagem

Essa nova solução agora obedece às limitações de armazenagem do problema, e a solução final foi obtida com um menor número de nós do que no caso analisado na seção anterior. Isso se deve ao fato de que a propagação das restrições de armazenagem reduziu mais as janelas, o que obrigou alguns ordenamentos, que antes foram inseridos manualmente, a serem definidos automaticamente. Fica evidente a importância da propagação das restrições de armazenagem quando se consideram problemas bastante restritos.

4.2.3 – HEVal de máxima redução de janelas

Até agora se usou como heurística de escolha de variável a decisão de ordenamento que preservasse o maior grau de flexibilidade para as decisões remanescentes, ou seja, que reduzisse as janelas o menos possível. Em termos de busca em árvore esta heurística tende a gerar poucos nós inactíveis. Porém, no problema as restrições são fortes e as janelas de qualquer solução final serão pequenas. Como discutido por Sadeh (Sadeh N. 1991), a solução ótima pode conter ordenamentos que não seriam obtidos através de uma heurística que preserve a maior flexibilidade. Visto que as janelas obtidas na solução final tendem a ser pequenas para problemas restritos, isso levanta a questão de que uma heurística oposta, que produzisse a maior redução das janelas, pode ser interessante, especialmente em uma busca em profundidade, para a obtenção de um limitante superior para a busca.

O ordenamento sugerido pode ter grande impacto na redução das janelas. Como consequência, um número menor de decisões é necessário para reduzir as janelas de forma a eliminar a concorrência entre as operações pelo equipamento, e um limitante superior pode ser obtido mais rapidamente. Porém, em vista dessa maior redução, o risco das janelas não absorverem essas reduções e gerar nós inactíveis é maior, o que exigiria mais *backtracking*. No caso de serem gerados apenas nós inactíveis, deve ser adotada a heurística apresentada na seção anterior.

Resolvendo novamente o problema, através das técnicas do algoritmo proposto na seção 4.2.2, mas desta vez tomando a decisão que produz a maior redução, obtêm-se as janelas abaixo:

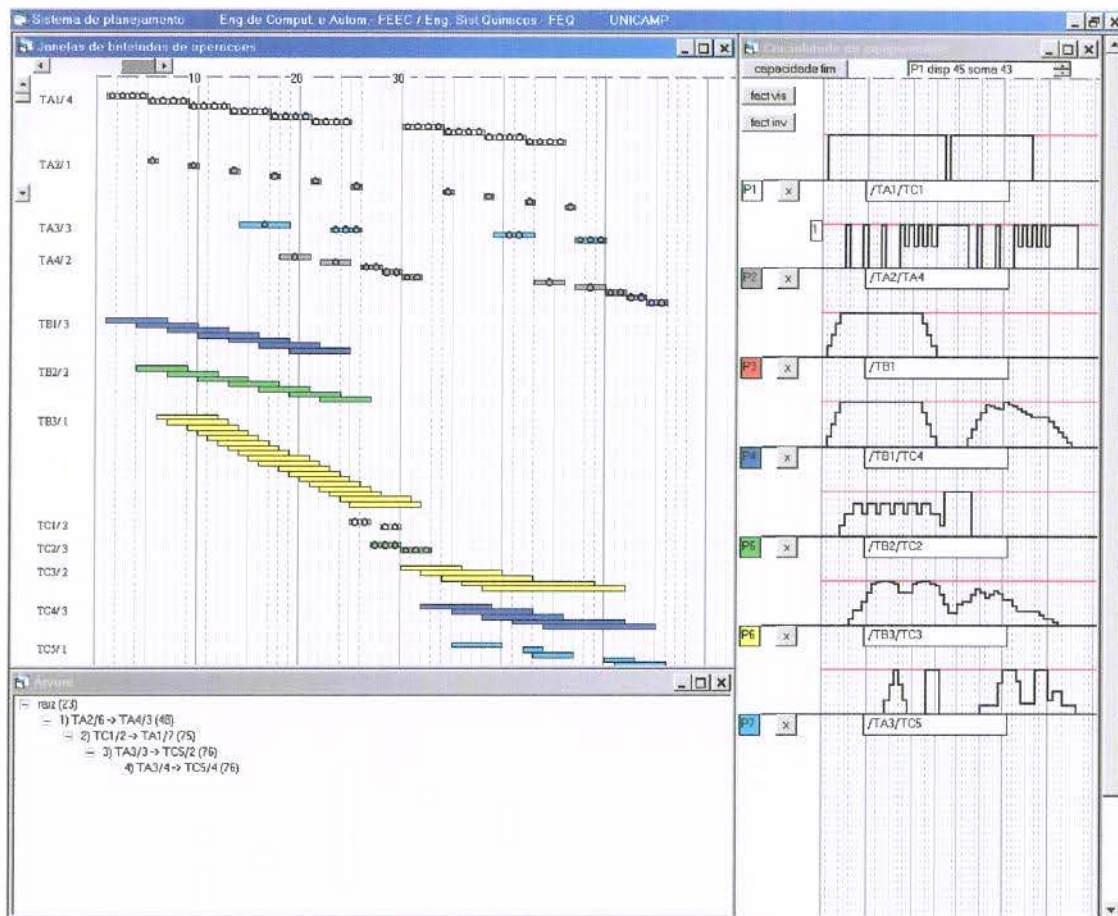


Figura 4.8 – Janelas finais da proposta 4.2.3

As janelas das operações ficaram ainda mais reduzidas que no exemplo anterior, com exceção das janelas de TC3 e TC4. Isso se deve à particularidade dos ordenamentos tomados sobre TA1/7 e TC1/2. Como esperado, a solução foi obtida brevemente, com apenas 4 decisões.

4.2.4 - Possibilidade e interesse de outras HEVar / HEVal

O exemplo a seguir tem como objetivo mostrar que as janelas de processamento permitem desenvolver outras heurísticas de ordenamento além da utilizada até agora, que se limita a escolher um par de operações e então escolher um ordenamento entre eles.

Considera-se o mesmo problema da figura 4.2 com uma nova demanda dada na tabela 4.4 e com a condição de armazenagem do estado C4 alterada de UIS para FIS com capacidade de 50 toneladas. Da fase de planejamento obtêm-se as janelas da figura 4.9.

Tabela 4.4 – Nova demanda e datas de entrega

Produto	Massa	Data de Entrega	Massa	Data de entrega
Prod A	150	50	180	100
Prod B	300	100		
Prod C	100	100		

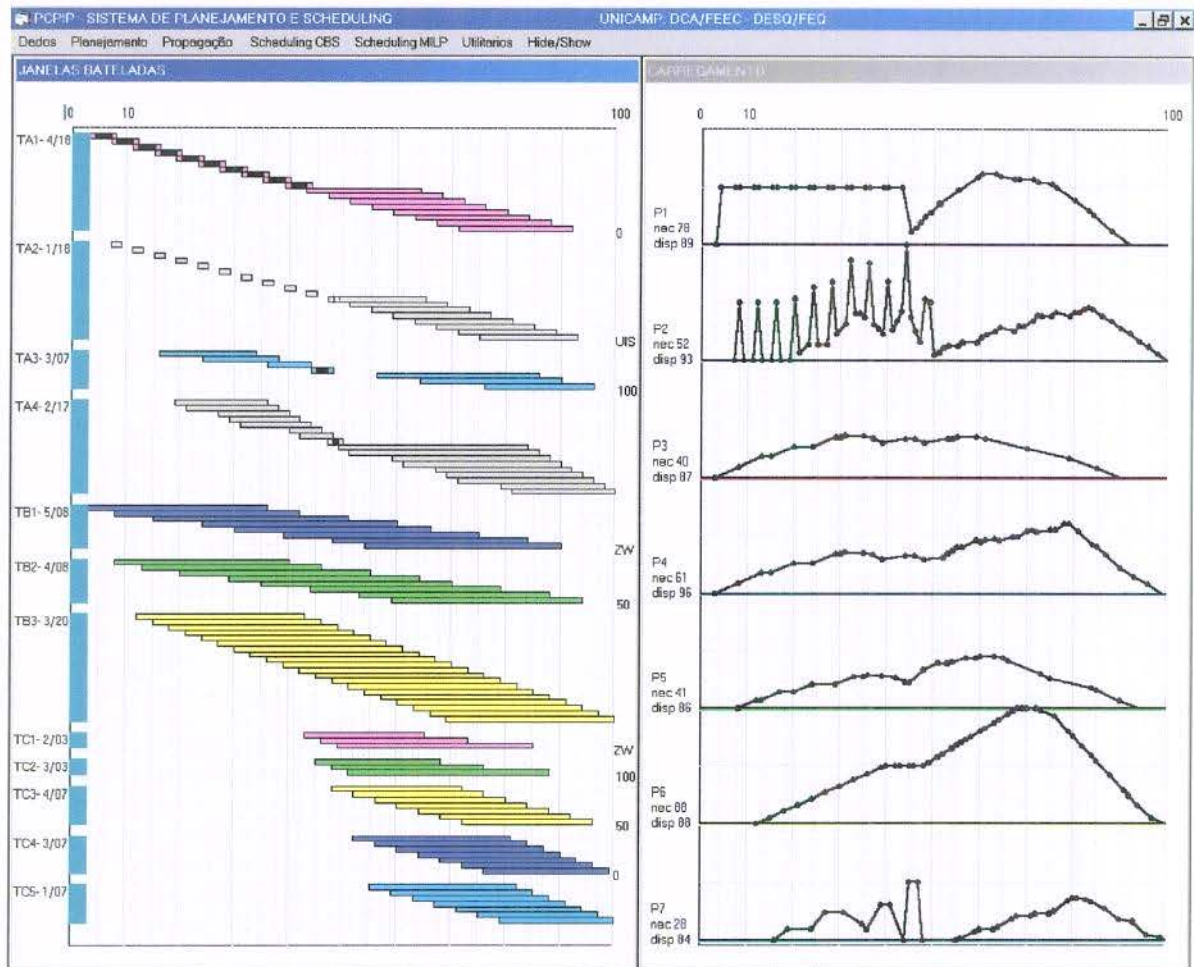


Figura 4.9 – Novas janelas de tempo

Uma breve análise na curva de demanda agregada revela a existência de um forte gargalo no equipamento P6. Há uma região de grande concorrência no intervalo de tempo de 48 a 86, como mostrado no detalhe da figura 4.10, onde os esforços de solução serão concentrados inicialmente. Existem ao todo 20 bateladas da operação TB3 e mais 7 da operação TC3, porém as duas primeiras bateladas da operação TB3 não estão envolvidas no gargalo, pois suas janelas não englobam o intervalo [48, 86]. De acordo com a heurística utilizada até agora, considerando somente as bateladas cujas janelas estão envolvidas no gargalo, há ainda um total de 25 bateladas das duas operações. Excluídos os ordenamentos obrigatórios, existem inicialmente 160 possíveis ordens de precedência que podem ser estabelecidas nesse conjunto de bateladas. Dado o grande número de decisões que podem ser tomadas e de bateladas envolvidas, é conveniente uma análise do gargalo sob outros aspectos.

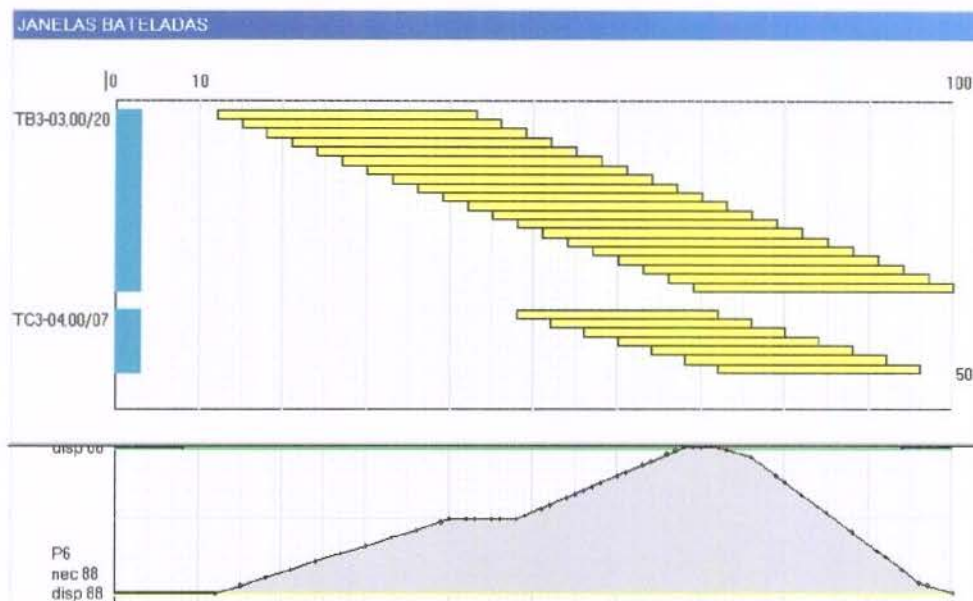


Figura 4.10 – Operações TB3, TC3 e demanda agregada do equipamento P6 em detalhe

Um ponto relevante é que a folga disponível para execução de todas as bateladas das operações TB3 e TC3 é zero. Ou seja, o intervalo compreendido pelo conjunto de janelas (de 0 a 100) será completamente ocupado. Pela função de custo adotada, seria melhor alocar as bateladas de TB3 no fim das suas janelas, pois essas produzem produto final. No entanto isso impediria a execução de bateladas de TC3 dentro de suas janelas. As bateladas de TC3 requerem ao todo 28 unidades de tempo (7 bateladas de 4 unidades de tempo cada). Como suas janelas permitem a alocação a partir do instante 48, todas as bateladas podem ser finalizadas até o instante 76, o que limita o tempo para as bateladas de TB3 a 24 unidades. Essas 24 unidades são suficientes apenas para oito bateladas de TB3. As 12 primeiras bateladas de TB3 portanto não poderão ocupar um intervalo de tempo que ultrapasse o instante 48, ou seja, esse conjunto de bateladas deve ser finalizado antes deste instante. Isto pode ser traduzido em uma forte redução obrigatória do LFT das 12 primeiras bateladas de TB3, que portanto não competem com as bateladas de TC3. Impondo que o LFT de TB3/12 seja 48, obtêm-se como resultado a figura 4.11. Após essa redução, o número de possíveis ordenamentos para as duas operações obtidos pelo algoritmo de detecção de ciclos é de 108, uma quantidade razoavelmente menor do que os 160 anteriores.

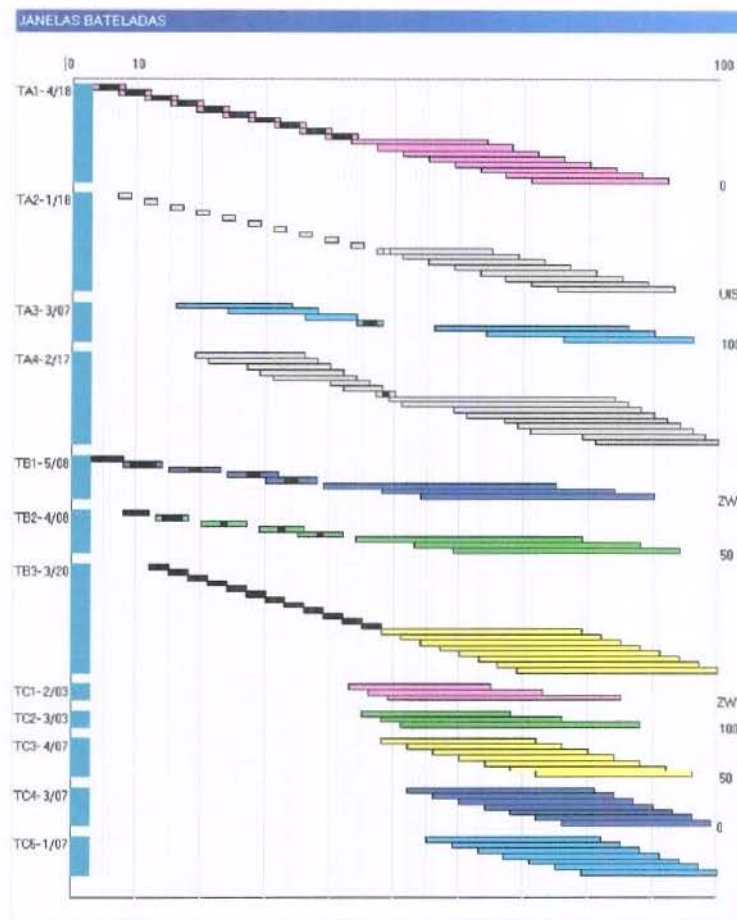


Figura 4.11 – Janelas após redução de TB3/1 a TB3/12

Com essa simples análise do tempo de folga, evita-se tomar decisões sobre as operações TB3/1 a 12 que seriam ineficazes, caso fosse fixada uma ordem em que alguma batelada de TC3 as preceda (TC3/X precede TB3/[1-12]). O uso do tempo de folga e da demanda agregada numa análise menos local das janelas envolvidas no gargalo, além de diminuir sensivelmente a dimensão do conjunto a ser trabalhado no problema, pode conseqüentemente diminuir o espaço de busca e o número de ramos a serem explorados.

4.2.5 - Uso das limitações de armazenagem na HEVar

Continuando a resolução do problema a partir das janelas exibidas na figura 4.11, pode-se utilizar uma visão menos local do problema de forma a estimar o impacto de uma decisão sobre a função de custo. Nesta seção discute-se o interesse em usar informações a respeito da capacidade de armazenagem para prever as conseqüências da propagação de restrições sobre o custo, de forma a tomar uma decisão que estabeleça um compromisso não sobre a otimização do gargalo, mas do problema global.

Como a operação TB3 produz um produto final, é interessante que as bateladas sejam concluídas o mais tarde possível, de modo a não aumentar a função de custo. Isso exigiria que as bateladas de TC3, que utilizam o mesmo equipamento, precedessem as de TB3. Entretanto, existe um outro ponto a considerar. A operação TC3 está ligada pela receita à operação TC4, que por sua vez está ligada a operação TC5, que também produz um produto final. Portanto, reduções no fim das janelas de TC3 podem provocar reduções no fim das janelas de TC5, e também aumentar a função de custo. O impacto das reduções de LFTs de TC3 sobre TC5 depende da dureza das restrições de armazenagem. Seria interessante então, verificar o quanto essas restrições influenciam as janelas de TC5, de modo a poder estabelecer um compromisso entre o aumento de custo originado por reduções nos LFTs das bateladas de TB3 e o originado por reduções nos LFTs das bateladas de TC5. Deste modo a heurística de escolha do valor seria direcionada pela função de custo, considerando-se o impacto das restrições de armazenagem.

A armazenagem entre TC3 e TC4 é do tipo limitada (FIS), com capacidade para 50 unidades de massa, e TC3 e TC4 possuem tamanho de batelada de 15 unidades. Isto indica que se três bateladas de TC3 (correspondentes a 45 unidades de massa) fossem alocadas no início de suas janelas, não haveria redução das janelas de TC4, pois a massa necessária para a sua execução poderia ficar armazenada até que TC4 seja iniciada, como exemplificado na figura 4.12.

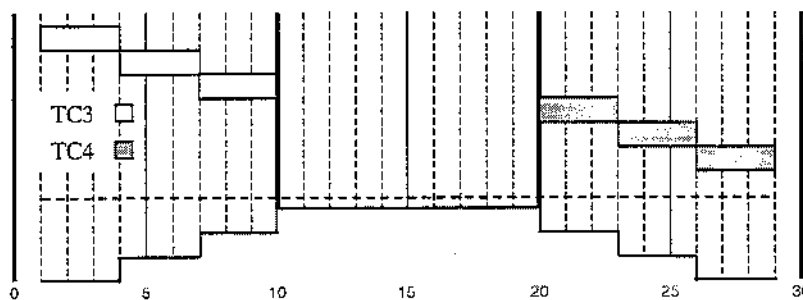


Figura 4.12 - Perfil de armazenagem com três bateladas de TC3 no início das janelas

Quando for produzida uma quarta batelada de TC3, a quantidade de estado intermediário será de 60 unidades de massa, e a capacidade de armazenagem será superada (figura 4.13), exigindo que uma batelada de TC4 seja iniciada imediatamente após concluída a quarta batelada de TC3, de modo a consumir a massa extra, como indicado na figura 4.14.

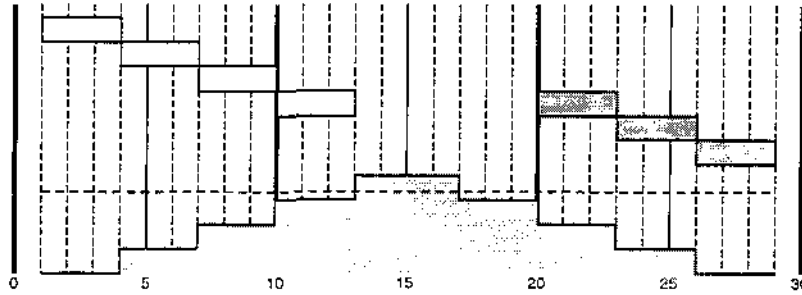


Figura 4.13 - Perfil de armazenagem com quatro bateladas de TC3 no início das janelas

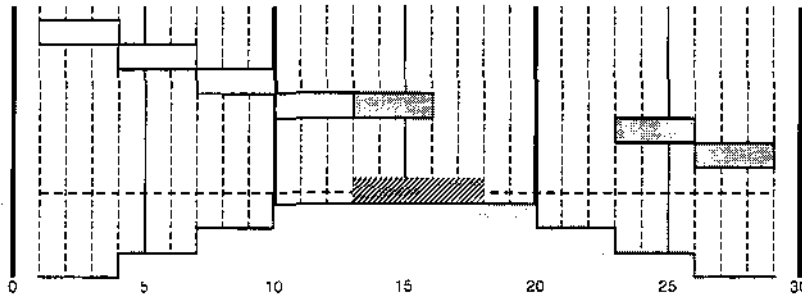


Figura 4.14 - Adiantamento obrigatório de TC4 devido a limitações de armazenagem

Pode-se então concluir que as três primeiras bateladas de TC3 podem ter o fim das suas janelas reduzido, sem impacto na função de custo, deixando assim espaço para que TB3 seja realizado o mais tarde possível. Até este ponto, ou seja, reduzido o LFT de apenas três bateladas de TC3, não existe propagação dessas reduções nas janelas de TC4, e conseqüentemente de TC5, portanto sem aumento do custo.

A armazenagem entre TC4 e TC5 é do tipo inexistente (NIS), o que indica que a redução do fim de uma janela de TC4 não implicará necessariamente em uma redução em TC5, pois a massa produzida poderá ficar armazenada no próprio equipamento até que este seja requisitado por outra batelada de TC4.

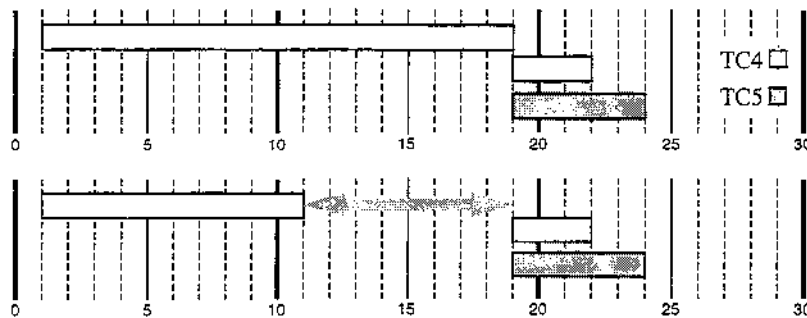


Figura 4.15 - Operação de *hold* após a redução de uma janela

Considerando a batelada i de TC4, se a próxima batelada ($i + 1$) tiver o LFT da sua janela reduzida, ela ocupará o equipamento mais cedo, diminuindo assim o tempo disponível para operação de *hold* da batelada i . Deste modo, a batelada de TC5 que consome a massa da batelada i deve também ter o fim da sua janela reduzido, como mostrado na figura abaixo.

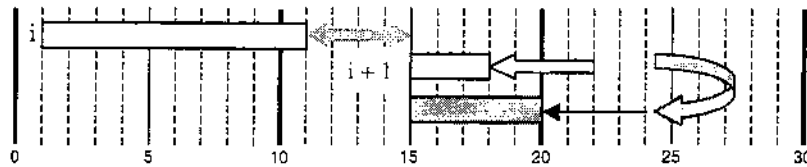


Figura 4.16 - Redução da janela em virtude da redução do tempo de *hold*

Assim, pode-se concluir que as janelas de TC5 só sofrerão redução no LFT se duas janelas de TC4 tiverem seus LFTs reduzidos, pois a segunda batelada exigirá o equipamento, obrigando uma batelada TC5 a consumir a massa produzida pela primeira batelada de TC4, e portanto levando a uma redução do seu LFT.

A partir destas considerações, é aceitável que se coloque mais bateladas de TC3 precedendo TB3, pois em caso contrário a redução do fim das janelas em TB3 terá conseqüências diretas sobre a função de custo, enquanto que a redução do fim das janelas de TC3 pode ser absorvida pela armazenagem. É óbvio que nem todas as bateladas de TC3 podem preceder as de TB3, pois as condições de armazenagem são duras e certamente implicarão em uma forte redução das janelas de TC5 após ultrapassado o ponto em que a capacidade de armazenagem pode absorver essas mudanças.

Esta análise mostra o tipo de considerações que podem ser feitas para diminuir o espaço de busca. Obviamente deverá ser avaliado o esforço de cálculo adicional para chegar a um compromisso adequado entre o tempo gasto em uma análise mais refinada e o gasto em *backtracking*.

Seguindo o exemplo anterior, a alocação de três bateladas de TC3 no início de suas janelas não implicará em reduções em TC4, pois a massa produzida por essas pode ficar armazenada, sem implicar na redução das janelas da última.

A alocação de quatro bateladas de TC3 irá provocar a redução de uma batelada de TC4. A redução dessa batelada de TC4, entretanto, não trará conseqüências sobre TC5, pois o estado que esta consome poderá ficar armazenado no próprio equipamento, sem exigir uma redução da sua janela.

A alocação de cinco bateladas de TC3 no início das janelas implicará na redução de duas bateladas de TC4. Essas duas janelas de TC4 por sua vez induzirão a redução de uma batelada em TC5, pois desta vez a segunda batelada reduzida de TC4 exigirá o equipamento mais cedo, obrigando que o fim de uma batelada de TC5 também seja reduzido de modo a receber a massa que não pode mais ficar armazenada no equipamento.

Definindo ordens de precedência de modo a refletir as conclusões descritas acima, obtêm-se as janelas mostradas na figura 4.17. Dessa forma, o aumento da função de custo foi menor do que se obteria no caso contrário, reduzindo o fim de TB3 para preservar as janelas de TC5.

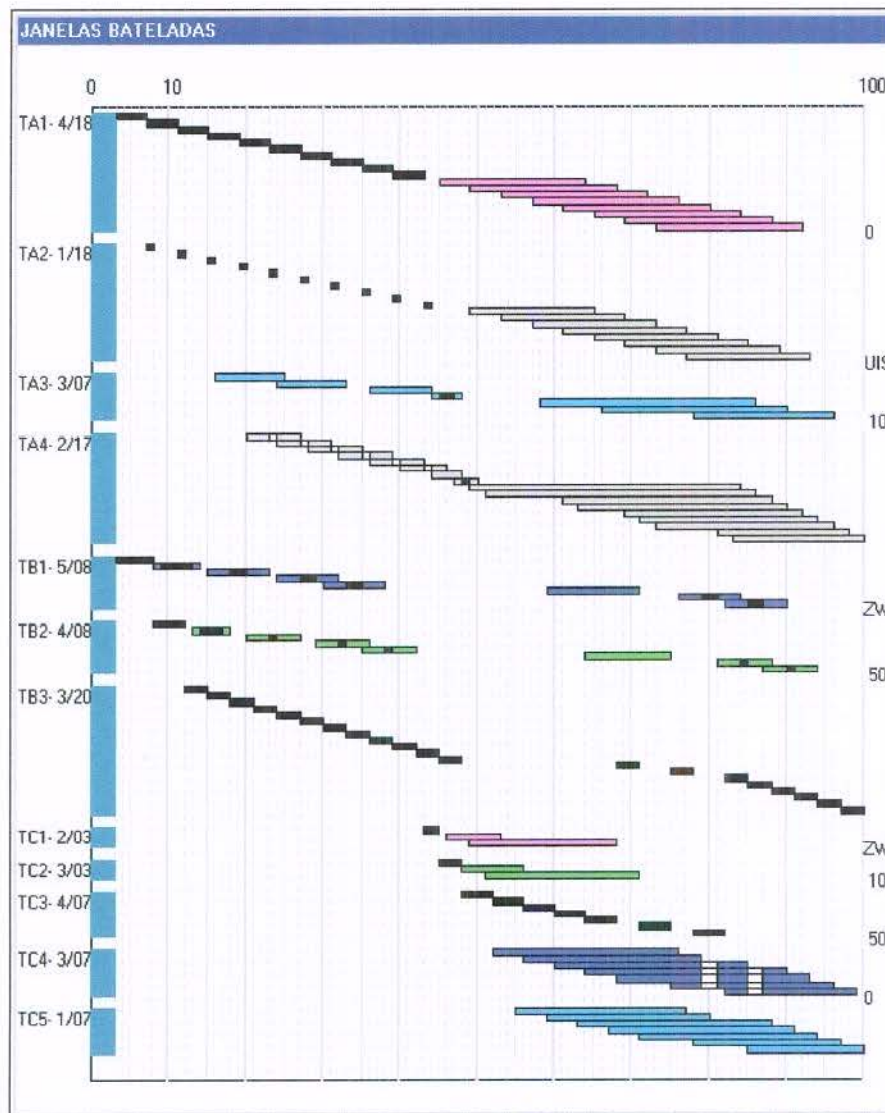


Figura 4.17 – Janelas após a otimização de TB3/TC5

Houveram outras reduções nas janelas de TC5, porém, não foram devidas a questões de compartilhamento do equipamento com TC3. Essas reduções foram induzidas por intervalos de ocupação obrigatória gerados por TB1, alocado ao mesmo equipamento que TC4, que reduziu as janelas das bateladas dessa operação, e cujas reduções foram propagadas para TC5.

Desta maneira, foi possível estabelecer um compromisso entre diminuir o custo proveniente das decisões em TC3 e as consequências no aumento de custo em TC5 que estas decisões provocavam. Quaisquer outras decisões de ordenamento no equipamento P6 diferentes das que foram tomadas elevam o valor da função de custo.

4.3 - Uso das heurísticas propostas para obtenção da solução ótima

Na seção 4.2 o objetivo foi determinar um limitante superior do problema, onde o importante é obter uma solução factível o mais rápido possível. Nesse contexto, a resolução de gargalos sucessivos onde só se expande um nó na árvore de busca para encontrar uma solução mais rápida é razoável. Quando se resolve um problema de otimização porém, a estratégia de abertura de nós deve ser diferente. Isto porque a heurística da tendência do menor caminho, que leva à menor redução das janelas, não pode garantir que o valor da função de custo seja menor do que tomando a decisão oposta. Deste modo propõe-se abrir os dois nós correspondentes aos dois ordenamentos possíveis, proceder à propagação das restrições nos dois casos e estimar o valor da função de custo para os dois nós. O processo de busca seguirá perseguindo o nó com estimativa custo mais promissora (conhecido na literatura como *best first search*).

Como agora o custo resultante da abertura de um nó é relevante, deve-se orientar a busca pelo nó onde o custo é mais promissor. O ideal seria possuir uma estimativa do custo final resultante de um determinado ordenamento para então fixar aquele que possuir a menor estimativa. Quando a função de custo não é o *makespan*, não é possível fazer uma estimativa da função de custo e orientar a busca por esse critério é impossível. Deste modo, uma maneira de abordar o problema de otimização é simplesmente testar ambos os ordenamentos possíveis do par escolhido pela heurística de escolha de variável, calcular o custo até os dois nós assim gerados e, então, seguir o ramo que possuir menor custo. Fixar o ordenamento entre um par de operações é uma atividade rápida, de modo que arbitrar ambas as ordens é barato do ponto de vista de tempo de cálculo. A ressalva que pode ser

feita com respeito ao teste de ambos os ordenamentos é que a propagação de restrições resultante de cada ordenamento pode consumir muito tempo, tornando esta abordagem custosa. Essa é uma grande vantagem da heurística proposta, que usa ordens de precedência entre pares de operações ao invés de alocação de uma operação. Neste último, o número de possíveis decisões de alocação tornaria o teste de todas as possibilidades oneroso, ao passo que na ordem de precedência existem apenas duas possibilidades.

Quando se chega a uma solução final, onde não existem mais ordenamentos livres, o custo obtido com essa solução serve como limitante superior dos demais ramos. Caso o nó em que se esteja investigando ultrapassar este custo, não é necessário aprofundar mais a busca neste nó, pois a solução que será obtida seguindo essa direção com certeza será pior do que a já obtida com o limitante superior.

O exemplo a seguir ilustra o uso da heurística proposta em um problema de otimização. Considerando o STN mostrado na figura 4.1, habilitação das operações e condições de armazenagem descritos na tabela 4.1 e demanda e datas de entrega de produtos dados na tabela 4.2, obtêm-se as janelas da figura 4.18. A heurística proposta revela os intervalos de gargalos listados na tabela 4.5, onde o valor da demanda agregada ultrapassa o valor unitário, estabelecido como critério para definir gargalos.

Tabela 4.5 – Intervalos, operações envolvidas e folga dos gargalos da figura 4.18

Equipamento	Intervalo	Operações envolvidas	Folga do conjunto de operações
P1	25 - 34	TA1/6 a 8 e TC1/1 e 2	20,0 %
P1	38 - 40	TA1/9 e TC1/2	64,7 %
P2	21 - 23	TA2/5 e TA4/1 e 2	28,6 %
P2	27 - 29	TA2/6 e TA4/3 e 4	0,0 %
P2	45 - 47	TA2/10 e TA4/6 e 7	64,5 %
P7	47 - 49	TA3/4 e TC5/3 a 5	68,4 %

O gargalo cujas operações envolvidas possuem menor folga é o quarto. Os cálculos da tendência do menor caminho para escolha do par de operações a ordenar é então realizado somente sobre esse pequeno conjunto de operações. Os cálculos para as operações que competem pelo equipamento P2 são mostrados na tabela 4.6.

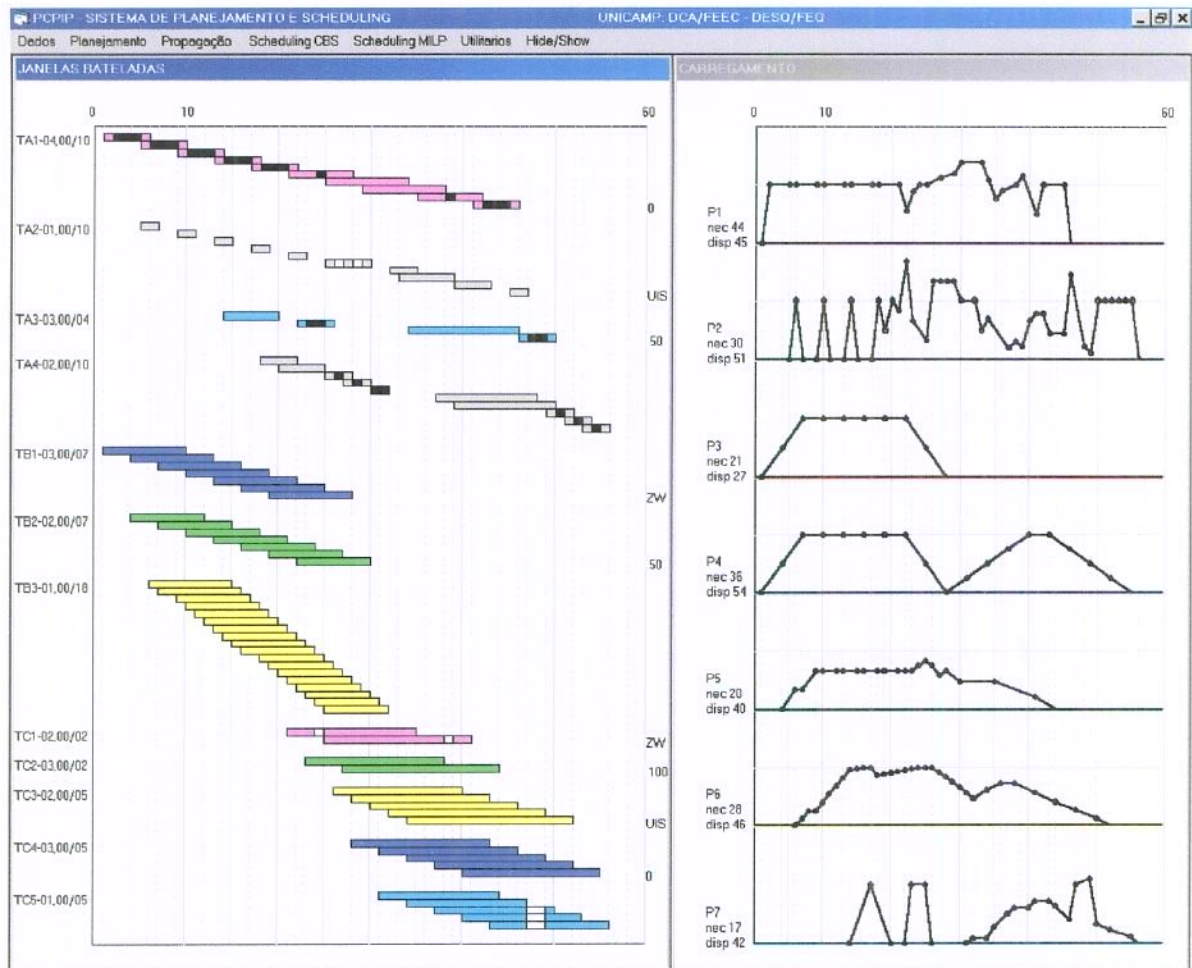


Figura 4.18 - Janelas iniciais do problema de otimização

Tabela 4.6 – Menor caminho e tendência do menor caminho dos pares de operações em P2

Opr A	Opr B	spAB	spBA	S	bspAB	bspBA
TA2/6	TA4/3	5	3	0,60	6,45	3,87
TA2/6	TA4/4	3	5	0,60	3,87	6,45

Pela heurística da tendência do menor caminho, tanto o ordenamento TA4/3 precede TA2/6 quanto TA2/6 precede TA4/4 são igualmente promissores, ou seja, implicarão em menor redução das janelas após fixados. Caso escolhida a primeira ordem de precedência, por exemplo, se obtêm as janelas mostradas a seguir.

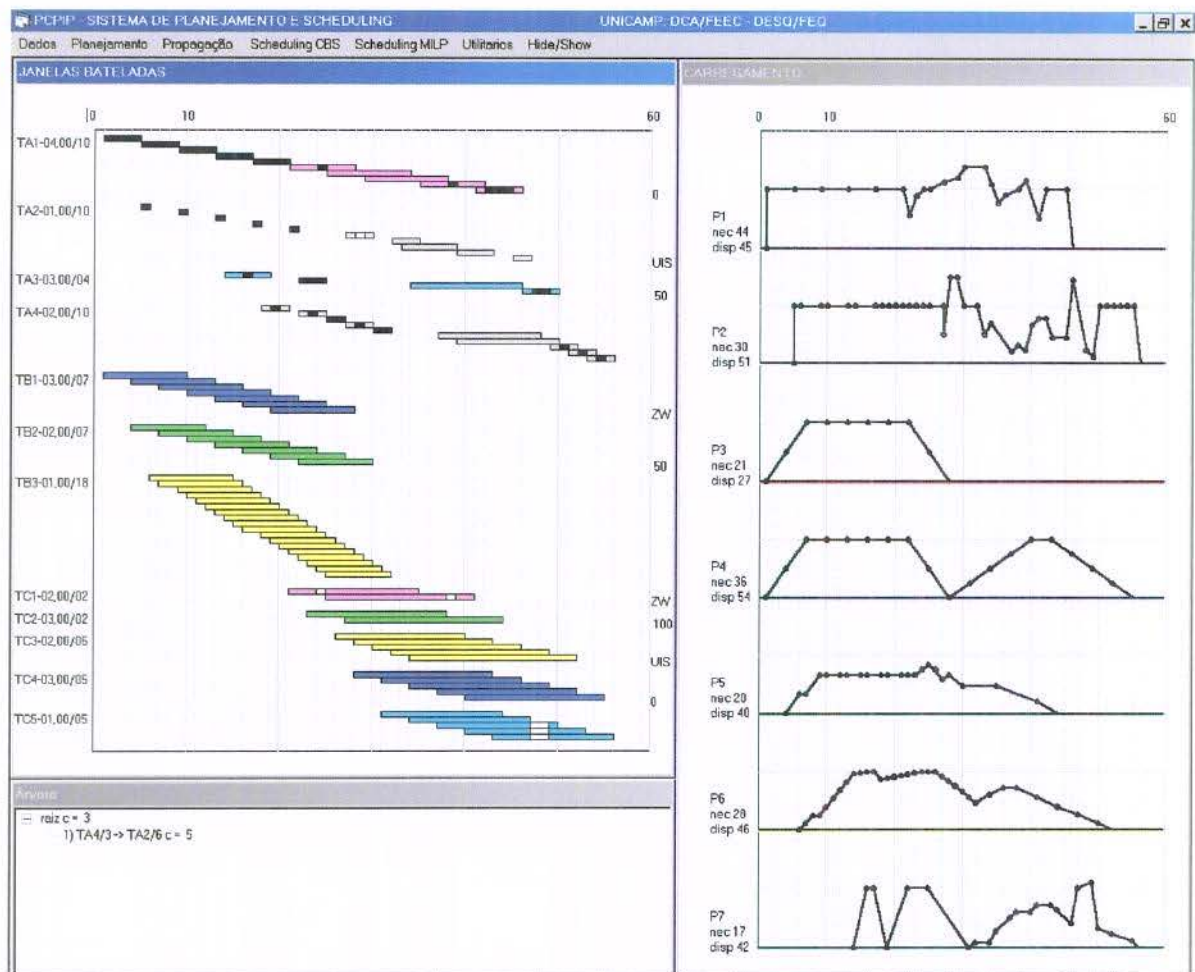


Figura 4.19 – Janelas após a ordem TA4/3 precede TA2/6

Como a operação TA4 produz um produto final, o adiantamento no seu LFT aumenta a função de custo. Como se pretende minimizar a função de custo, o outro nó, correspondente à ordem de precedência inversa, também será investigado. Fixando a ordem oposta, obtêm-se as janelas ilustradas na figura 4.20. Como é possível notar, a função de custo para esse segundo caso é menor, apesar de a precedência sugerida pela tendência do menor caminho não ser essa. Isso se deve ao fato que essa heurística propõe a menor redução das janelas, que apesar de geralmente impor mudanças suaves na função de custo, não implica obrigatoriamente em obter o menor custo.

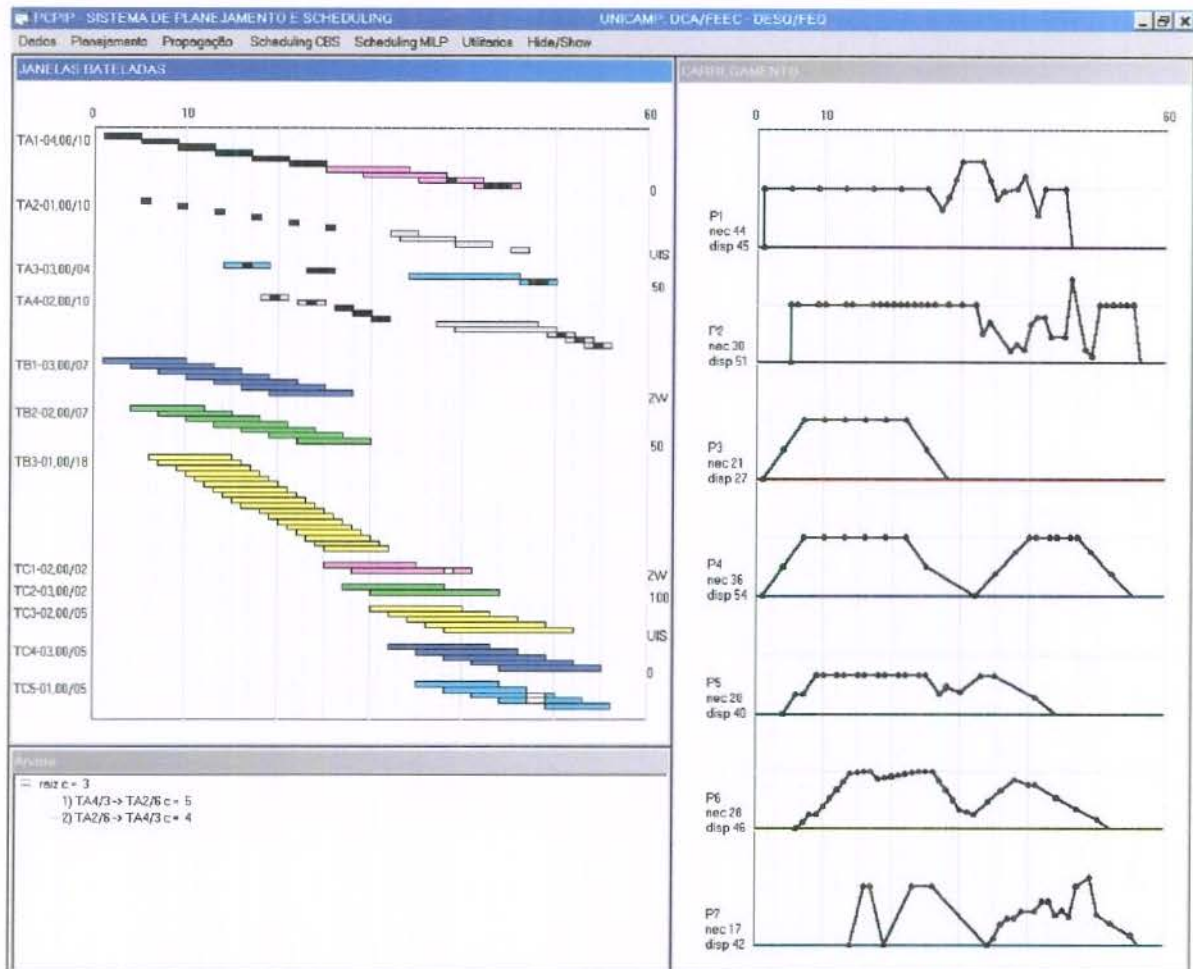


Figura 4.20 – Janelas após inverter o ordenamento

Visto que esse último ordenamento tem menor custo, a busca seguirá deste ponto. As novas janelas mostram agora os gargalos listados na tabela 4.7. O gargalo com menor folga é o primeiro. A análise da tendência do menor caminho para as operações envolvidas é mostrada na tabela 4.8.

Tabela 4.7 - Intervalos, operações envolvidas e folga dos gargalos da figura 4.20

Equipamento	Intervalo	Operações envolvidas	Folga do conjunto de operações
P1	29 - 34	TA1/7 e 8 e TC1/1 e 2	25,0 %
P1	38 - 40	TA1/9 e TC1/2	57,1 %
P2	45 - 47	TA2/10 e TA4/6 e 7	61,5 %
P7	47 - 49	TA3/4 e TC5/3 a 5	58,3 %

Tabela 4.8 - Tendência do menor caminho dos pares de operações

Opr A	Opr B	spAB	spBA	S	bspAB	bspBA
TA1/7	TC1/1	9	10	0,90	9,49	10,54
TA1/7	TC1/2	6	16	0,38	9,80	26,13
TA1/8	TC1/1	13	6	0,46	19,14	8,83
TA1/8	TC1/2	10	12	0,83	10,95	13,15

Apesar de dois ordenamentos possuírem o mesmo valor para o menor caminho (TA1/7 precede TC1/2 e TC1/1 precede TA1/8), a tendência do menor caminho distingue entre essas duas situações. Pela figura 4.21, é possível notar que as operações TC1/1 e

TA1/8 estão menos próximas de possuir um ordenamento obrigatório. Portanto, de acordo com a heurística, o segundo ordenamento é escolhido. Fixando essa ordem obtêm-se as janelas da figura 4.22.

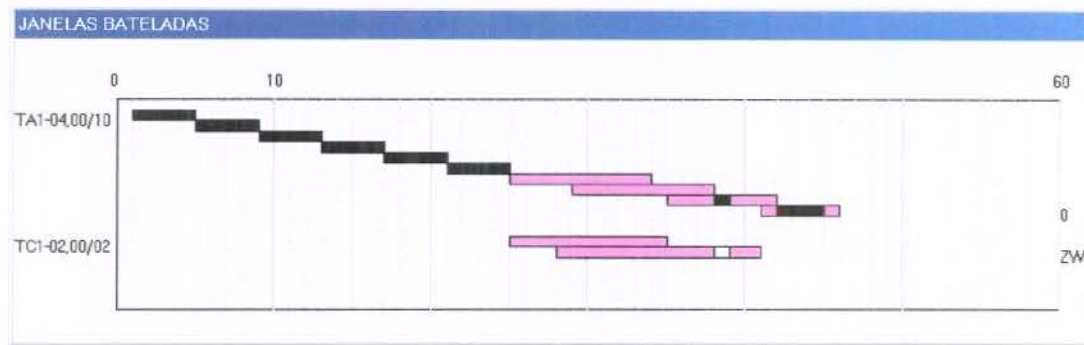


Figura 4.21 –Detalhe das operações TA1 e TC1

O interessante neste caso é quando são utilizados os ordenamentos contrários aos sugeridos. O ordenamento TC1/1 precede TA1/8 fornece um resultado factível, porém o ordenamento TA1/8 precede TC1/1 fornece um resultado infactível. Isso mostra que realmente o segundo par de operações constitui o gargalo mais forte.

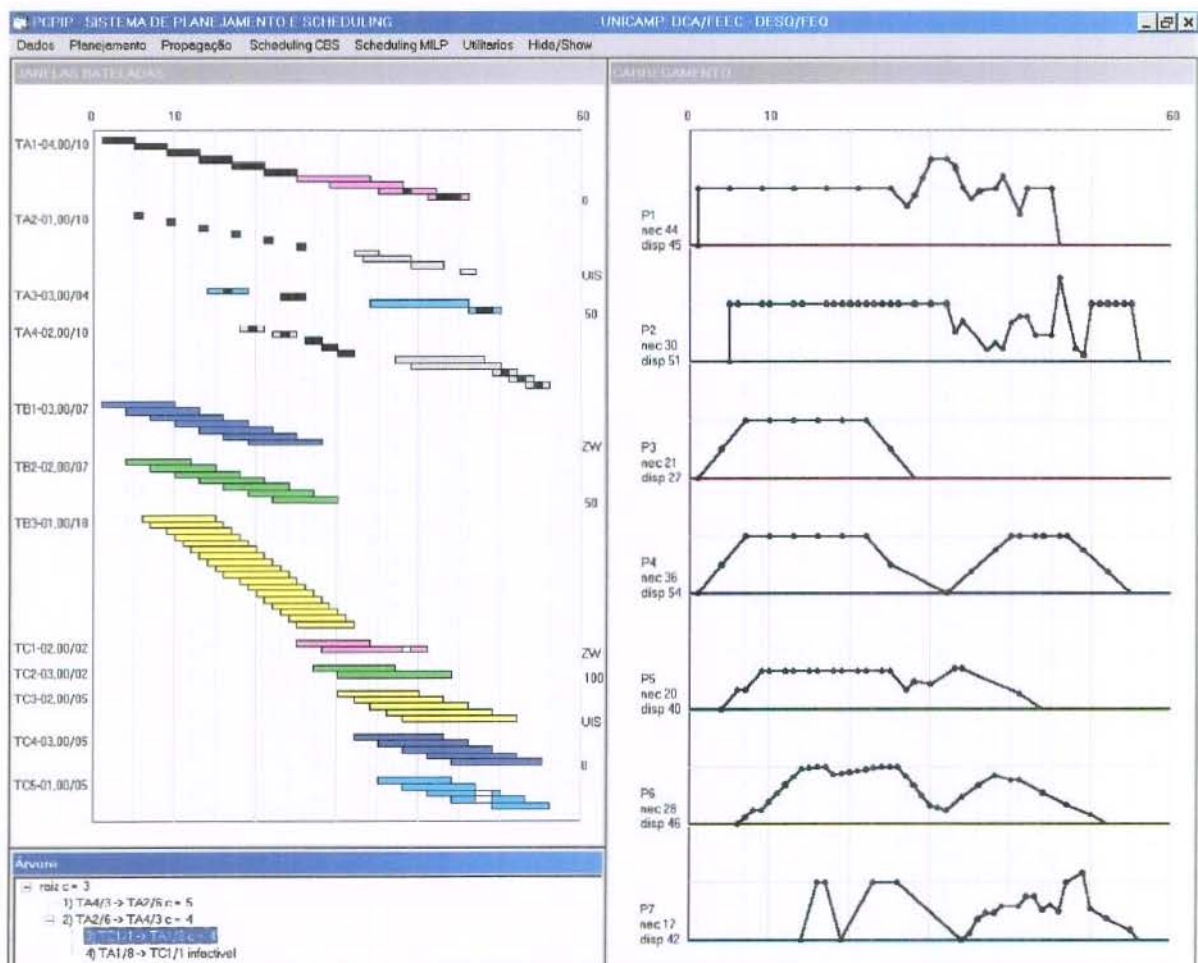


Figura 4.22 - Janelas após ordem TC1/1 precede TA1/8

A identificação precoce desse nó inactível através da detecção de gargalos é vantajosa pois, caso a decisão que gerou esse nó fosse tomada mais tarde, seriam investidos mais tempo e recursos computacionais em um ramo que não resultaria em uma solução.

Repetindo o processo para as janelas resultantes, os gargalos existentes são mostrados na tabela 4.9. O gargalo com menor folga ainda é o primeiro. Os cálculos da tendência do menor caminho são mostrados na tabela 4.10.

Tabela 4.9 - Intervalos, operações envolvidas e folga dos gargalos da figura 4.22

Equipamento	Intervalo	Operações envolvidas	Folga do conjunto de operações
P1	29 - 34	TA1/7 e 8 e TC1/1 e 2	25,0 %
P1	38 - 40	TA1/9 e TC1/2	57,1 %
P2	45 - 47	TA2/10 e TA4/6 e 7	61,5 %
P7	47 - 49	TA3/4 e TC5/3 a 5	58,3 %

Tabela 4.10 - Tendência do menor caminho dos pares de operações

Opr A	Opr B	spAB	spBA	S	bspAB	bspBA
TA1/7	TC1/1	9	9	1,00	9,00	9,00
TA1/7	TC1/2	6	16	0,38	9,80	26,13
TA1/8	TC1/2	10	12	0,83	10,95	13,15

A tendência do menor caminho fornece duas ordens igualmente promissoras para o mesmo par de operações, TA1/7 e TC1/1. Testando ambos os ordenamentos, verifica-se que somente uma é factível, conforme mostrado na figura 4.23.

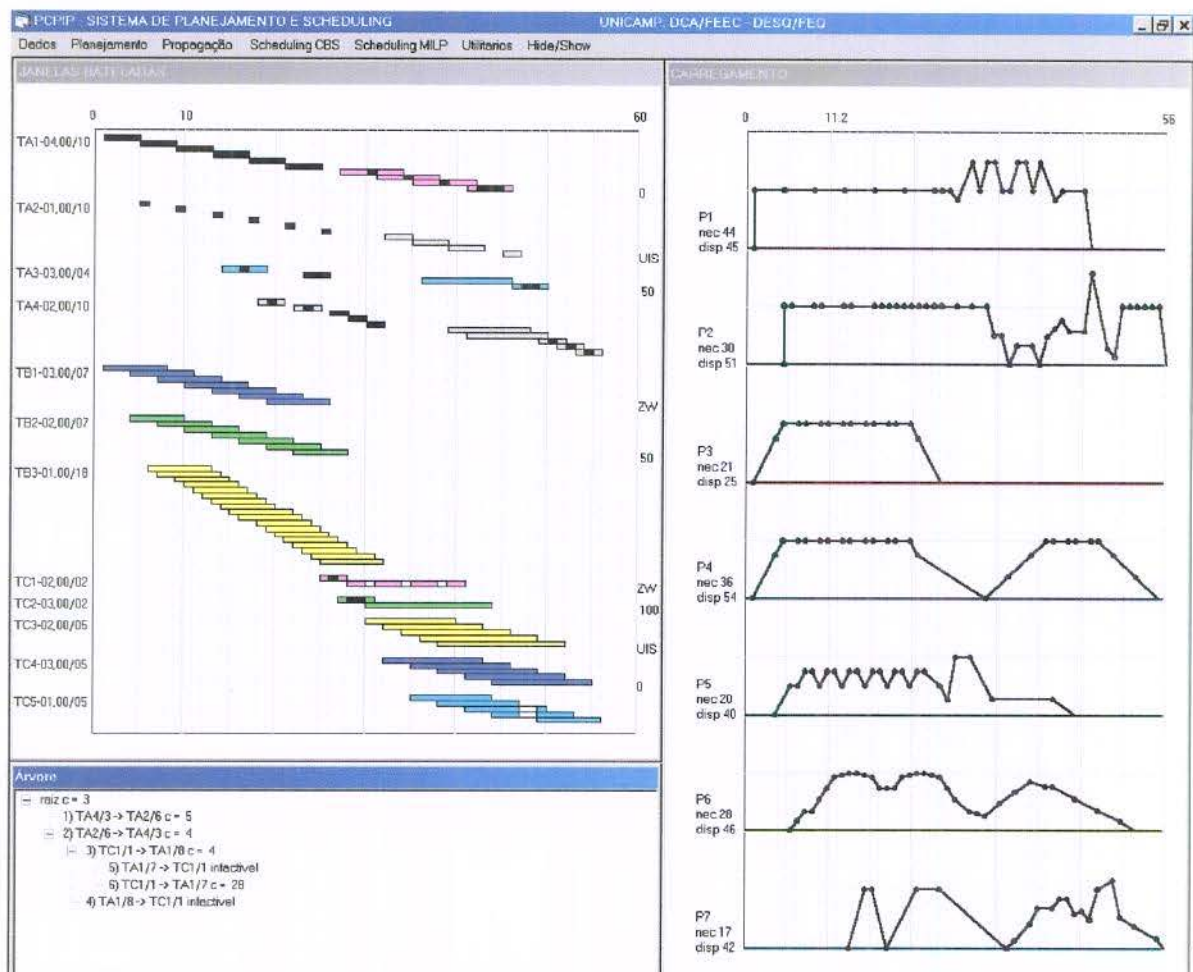


Figura 4.23 - Janelas após a ordem TC1/1 precede TA1/7

Repetindo esse procedimento até que todos os ordenamentos estejam definidos, obtêm-se como solução final as janelas ilustradas na figura 4.24. Apesar de muitas bateladas ainda serem apresentadas na forma de janelas, tem-se uma solução completa do problema de *scheduling* pois todos os ordenamentos estão definidos. Os instantes de término mais tarde de todas as bateladas estão definidos e portanto o valor da função de custo não será alterado se as operações que influenciam diretamente na função de custo (as que produzem produtos finais) forem alocadas no fim de suas respectivas janelas. Após a obtenção desta solução completa procede-se a exploração dos ramos com custo menor do que o limitante superior, com o objetivo de obter uma solução de menor custo. Caso o custo do nó que se esteja investigando ultrapasse o custo obtido como limitante superior, a exploração deste ramo é suspensa e parte-se para outro nó.

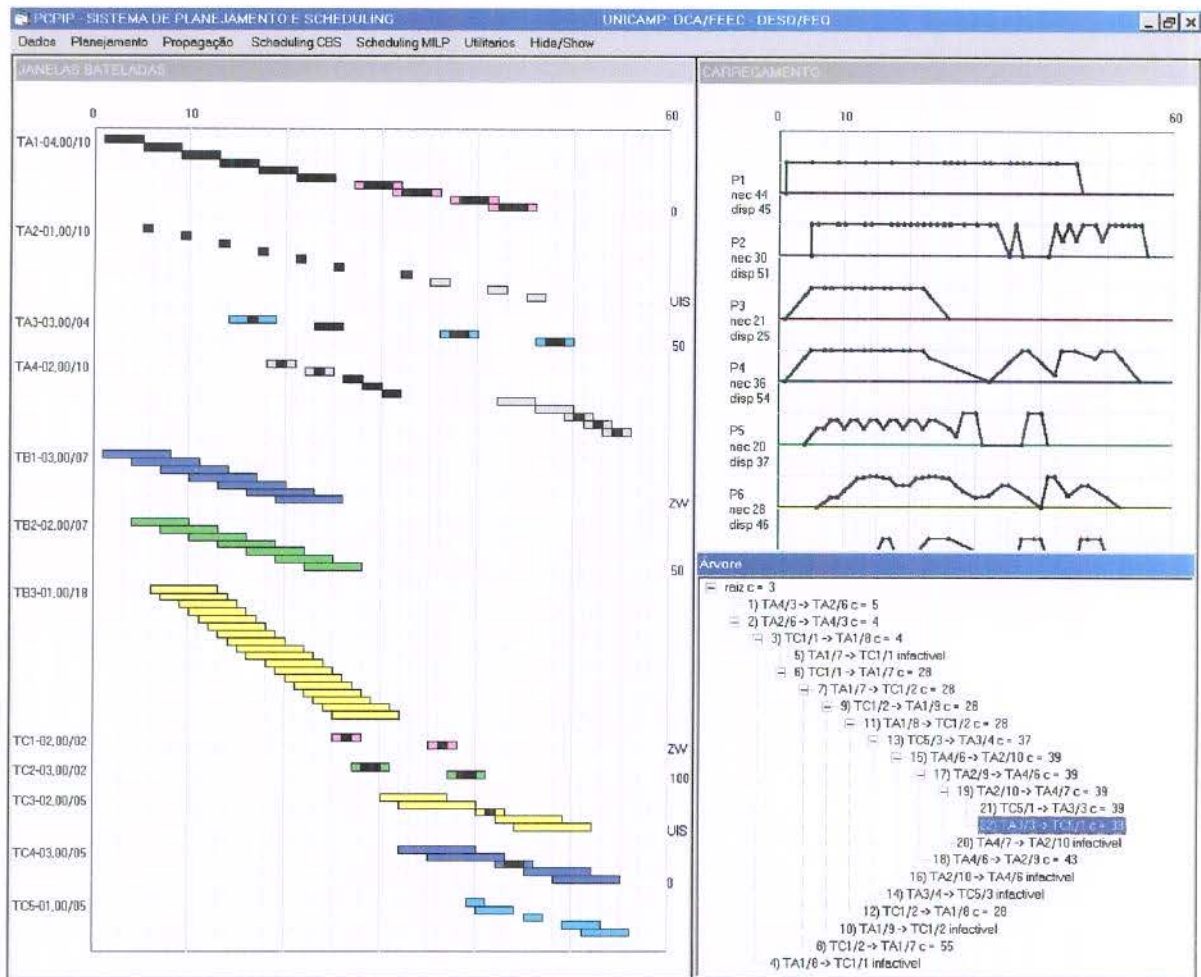


Figura 4.24 - Janelas finais da busca em profundidade e limitante superior do custo

Após investigar toda a árvore de busca se obtêm uma segunda solução de custo igual ao limitante superior, conforme mostrado na figura 4.25. Não foi encontrada qualquer solução de custo menor, o que indica que a primeira solução obtida é a solução ótima do problema. A obtenção da solução ótima como limitante superior pelo algoritmo usado não ocorre necessariamente em qualquer caso, entretanto o limitante superior obtido dessa maneira é certamente de melhor qualidade do que os obtidos pelos métodos usados na seção anterior.

Deste modo, em alguns problemas, o tempo extra gasto na investigação de ambos os nós em cada ponto da árvore de busca pode ser compensado pelo tempo salvo na investigação e *backtracking* de ramos cujo limitante superior foi obtido por métodos menos eficazes.

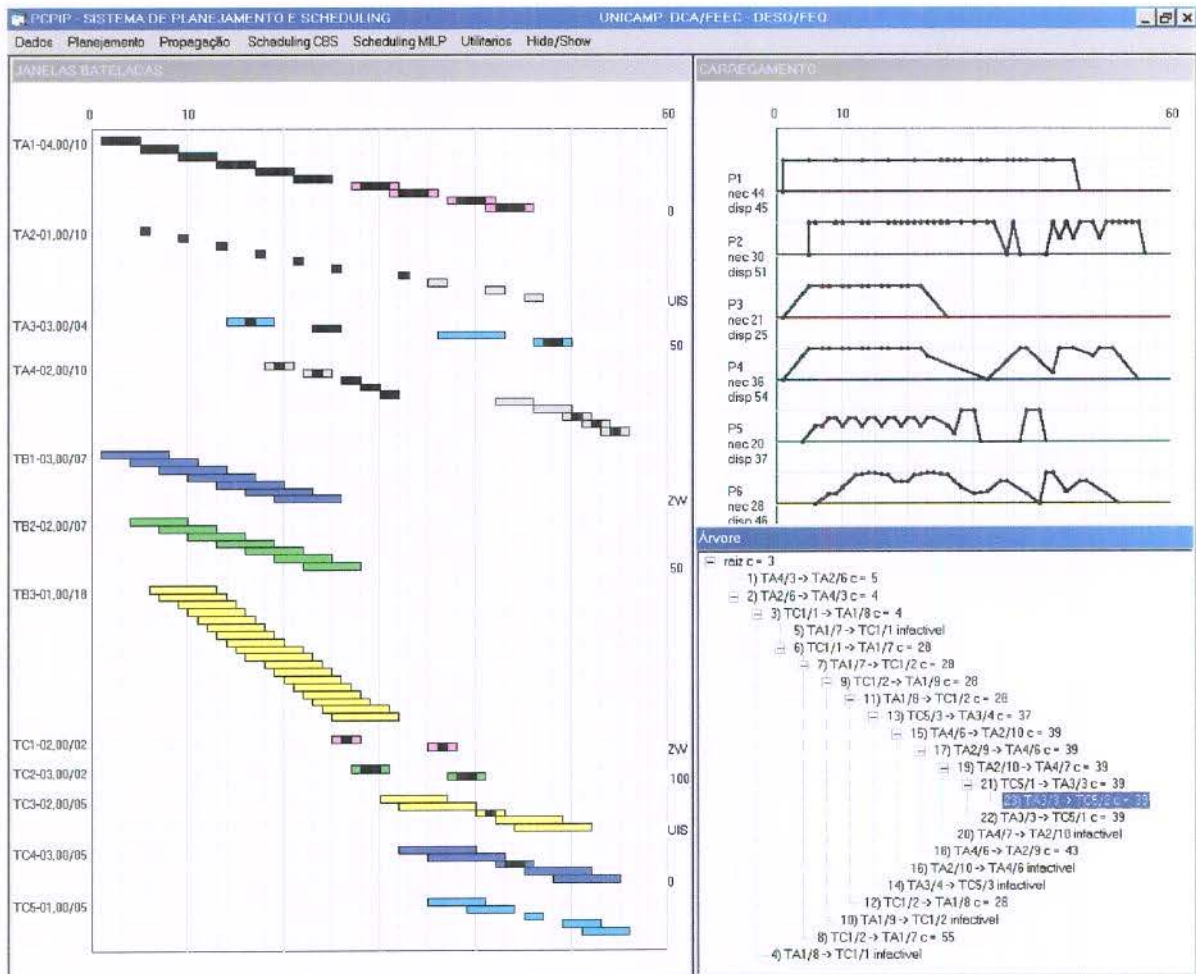


Figura 4.25 - Janelas da segunda solução

A árvore completa de busca é ilustrada na figura 4.26. Como é possível notar, existem muitos nós ineficazes. Esses nós exigiram um gasto em tempo de cálculo na fase de propagação de restrições para serem detectados, além do tempo necessário para o *backtracking* desfazer a decisão tomada e restaurar o estado original das janelas antes da decisão. Attingir um nó ineficaz em um ponto avançado da árvore de busca significa que foram investidos tempo e recursos computacionais em uma solução parcial que não será completada. Por esse motivo, tentar estimar resultados ineficazes o mais cedo possível com pequeno esforço de cálculo sempre é interessante pois não seria desperdiçado tempo em um ramo que não conduzirá a uma solução.

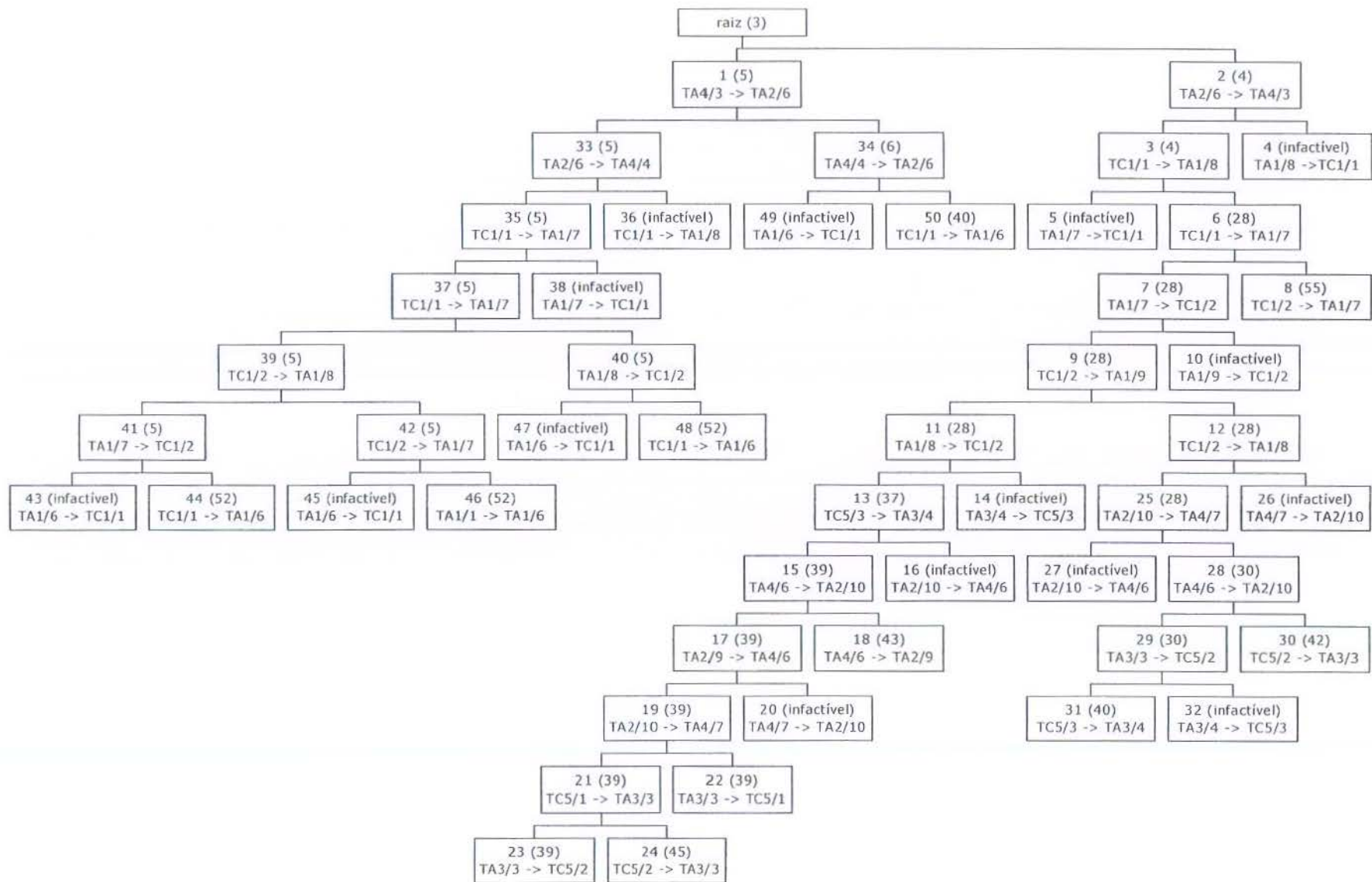


Figura 4.26 – Árvore de busca completa

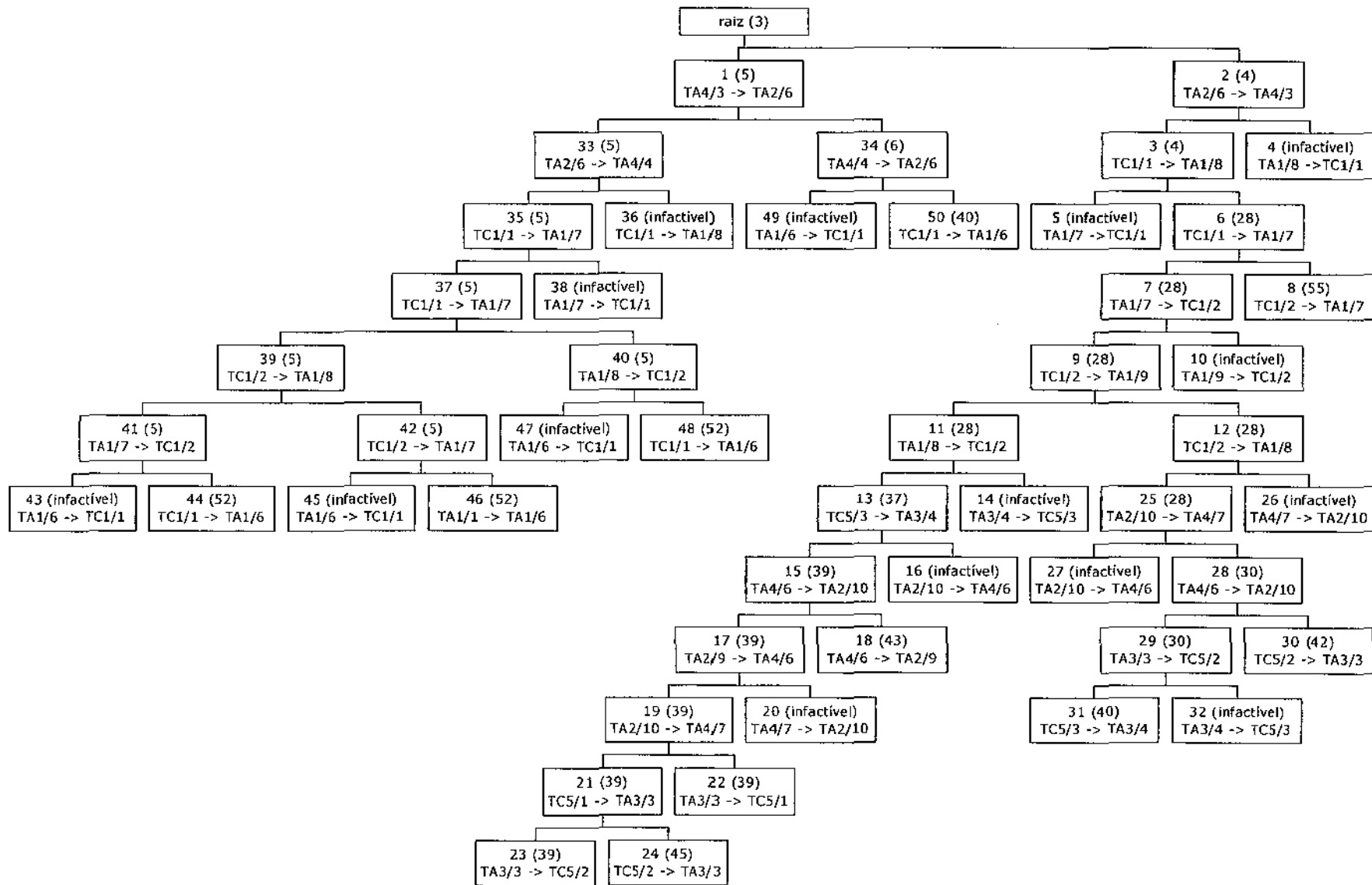


Figura 4.26 – Árvore de busca completa

Capítulo 5 - Conclusões e sugestões

Neste trabalho foi proposto o uso de heurísticas da área de busca orientada por restrições de forma combinada para a resolução de problemas de *scheduling* muito restritos, com o objetivo de obter uma heurística híbrida com maior eficiência de solução e de menor custo computacional. A maior eficiência de solução é obtida através da orientação aos gargalos mais graves, onde as restrições do problema correm maior risco de serem violadas. O menor custo computacional é obtido através de uma escolha inteligente dos elementos que são analisados para tomada de decisão, e também por decisões que permitam atingir uma solução final satisfatória com o mínimo possível de nós abertos.

A orientação aos gargalos mais graves foi alcançada pelo uso da demanda agregada juntamente com tempo de folga, que direcionou a busca para as regiões de concorrência que possuem pouco espaço para acomodar as reduções nas janelas, evitando que os esforços de solução sejam empregados em regiões de concorrência onde as decisões de ordenamento têm pouco impacto sobre a redução das janelas e portanto menor influência sobre a resolução do gargalo. O uso combinado das técnicas citadas também reduziu sensivelmente o número de elementos (operações) sobre as quais foram feitas as análises necessárias pelas heurísticas de escolha de variável e de valor, diminuindo assim o custo computacional para resolução do problema.

O menor custo computacional também foi alcançado através do algoritmo de detecção de ciclos e da heurística de escolha de variável de tendência do menor caminho. A tendência do menor caminho é utilizada com o objetivo de preservar a maior flexibilidade possível das operações, ou seja, preservar a maior folga possível das janelas. Deste modo as janelas podem absorver mais facilmente as reduções propagadas de outras operações. Como principal vantagem dessa maior flexibilidade, diminui a chance de violar as restrições do problema e portanto de obter um nó infactível, o que exigiria *backtracking*. O algoritmo de detecção de ciclos em grafos elimina os ordenamentos que, em virtude de ordenamentos já fixados e das relações de precedência impostas pela receita de produção, já estão realmente definidos, porém não podem ser identificados pelas demais técnicas apresentadas devido à complexidade das relações de receita e de compartilhamento entre as operações.

Uma crítica que deve ser feita sobre a heurística do menor caminho é que, diferente do conceito de folga, não é levado em consideração o tempo de processamento do

par de operações. Assim, podem existir situações em que o par de operações não possui o menor valor para a variável w , mas seus tempos de processamento somam um valor elevado, de forma que pode ser mais difícil estabelecer ambos os ordenamentos do que um par com valor de w baixo onde os tempos de processamento são pequenos, constituindo uma situação menos crítica.

Outro ponto a ser considerado é a eficiência global da heurística proposta. Uma questão relevante nesse sentido é até quando é viável, do ponto de vista de resolução de gargalos, a utilização dessa heurística. Dependendo da natureza do gargalo, as decisões de ordenamento tomadas têm pouco impacto sobre as janelas de forma que se torna mais difícil eliminar o gargalo, e a heurística proposta perde sua eficiência. Uma forma de mensurar até que ponto uma decisão de ordenamento pode ter influência sobre o gargalo pode ser obtida através da folga global do conjunto de operações envolvido no gargalo, informação que é utilizada pela própria heurística. Quanto menor a folga, maior será o impacto de uma decisão de ordenamento sobre as janelas das demais operações, pois essas possuem pouco espaço para acomodar as reduções. Quando essa folga é razoavelmente grande, a redução é pequena e a propagação das restrições pode ter efeito pequeno, de modo que o gargalo permanece.

Propõe-se que, em trabalhos futuros seja estabelecido um novo critério de parada para a heurística de identificação de gargalos proposta. O valor do tempo de folga pode ser usado para este fim, pois quando a folga dos conjuntos de operações é grande, uma decisão de ordenamento terá pouco impacto na redução de janelas. Neste quadro, a identificação dos gargalos mais graves e das operações neles envolvidas trazem pouco ou nenhum ganho sobre a redução das janelas, justamente a principal proposta dessa ferramenta. Assim, podem ser inseridos ordenamentos entre quaisquer pares de operações não ordenados de acordo com critérios que o usuário julgar adequados, inclusive a heurística de tendência do menor caminho, sem a necessidade de realizar a busca por gargalos. Propõe-se também a inserção do tempo de processamento nos cálculos da tendência do menor caminho e o uso de um algoritmo de detecção de ciclos em grafos mais completo, que considere também relações de precedência por balanço de massa.

Com respeito a problemas de otimização, pelos motivos descritos acima, a busca orientada por restrições pode não constituir um método adequado de otimização isoladamente, porém pode ser utilizado em conjunto com outros métodos como o MILP e *simulated annealing*, de modo a gerar cenários que o usuário julgue mais promissores para o uso destes métodos.

Apêndice A

O sistema de busca orientada por restrições desenvolvido realiza a busca de soluções de modo interativo. O sistema provê ao usuário informações a respeito dos gargalos que obedecem ao critério estabelecido por ele, identificando qual o gargalo mais grave. Também são fornecidas informações a respeito das decisões de ordenamento que podem ser tomadas e qual a mais promissora, do ponto de vista da heurística de escolha de valor adotada. Essas informações servem ao propósito de orientar o usuário na tomada de decisão, de forma que cabe a ele o julgamento final sobre que gargalo abordar e que decisão de ordenamento tomar.

A seguir são mostrados os passos envolvidos no estabelecimento de um ordenamento, correspondente a um nó na árvore de busca. Em primeiro lugar, de posse das janelas de tempo das operações, após finalizada a etapa de planejamento, o usuário inicia o procedimento de *scheduling* através do respectivo comando na barra de menus, como mostrado na figura A.1.

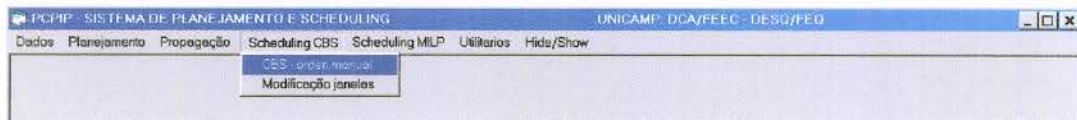


Figura A.1

O sistema então exibe uma caixa de diálogo mostrada na figura A.2, onde o usuário fornece o equipamento e intervalo de tempo onde se encontra o gargalo escolhido. Em um painel à direita é mostrada a folga total relativa de todas as operações atribuídas ao equipamento. O botão *auto análise* faz os cálculos referentes à abordagem proposta nesta tese, de encontrar gargalos onde a demanda agregada ultrapassa o valor unitário e dentre estes escolher o que possui menor folga entre as operações envolvidas, e mostra o resultado (equipamento e intervalo) nos respectivos campos para avaliação do usuário, que pode aceitá-los ou realizar outra escolha. O botão *próximo* conduz à próxima caixa de diálogo.

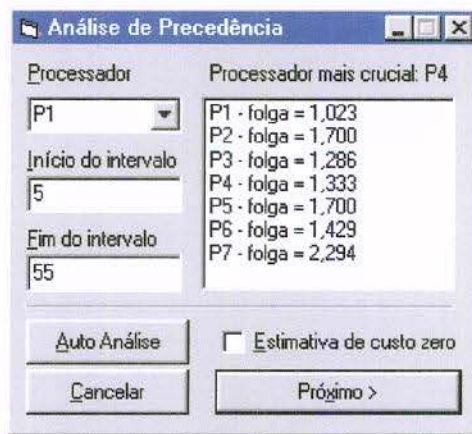


Figura A.2

Na caixa de diálogo seguinte (figura A.3) o usuário fornece um ordenamento entre um par de bateladas, entrando com o nome da operação e número da batelada que irão compor a ordem de precedência, de forma que a operação1/batelada1 precede a operação2/batelada 2. O botão *shortest path* faz os cálculos referentes à heurística de escolha da tendência do menor caminho, escolhendo um par e um ordenamento entre eles, de acordo com a heurística mostrada na seção 3.4.4. O botão *longest path* realiza os cálculos referentes à escolha do par e do ordenamento que irão produzir maior redução das janelas, de acordo com a heurística sugerida na seção 4.2.3. Novamente, os botões mostram os dados nos respectivos campos, cabendo ao usuário a decisão final sobre que par escolher e que ordenamento estabelecer. O botão voltar retorna ao formulário anterior, para uma nova entrada de equipamento/intervalo.



Figura A.3

Caso o usuário deseje entrar com o ordenamento manualmente, é possível saber quais ordenamentos podem ser arbitrados pressionando o botão *mais*, que expande o formulário como exibido na figura A.4. No painel direito são exibidos quais os

ordenamentos livres identificados pelo algoritmo de detecção de ciclos (seção 3.6.1), sendo que no painel esquerdo estão relacionados os ordenamentos fixos.



Figura A.4

Após inserido um ordenamento, caso o usuário deseje realizar *backtracking* na árvore de busca (retroceder a um nó anterior), basta clicar com o mouse sobre o nó correspondente na janela que exibe a árvore. As janelas de tempo das operações e outros dados relevantes são recuperados no estado em que se encontravam no nó em questão.

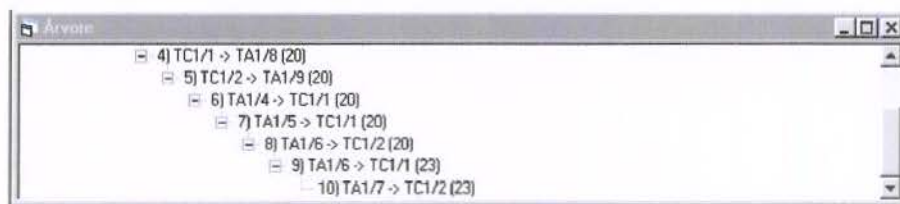


Figura A.5

Referências Bibliográficas

- APPLEQUIST G., SAMINOGLU O., PEKNY J., REKLAITIS G.V. 1997. Issues in use, design and evolution of process scheduling and planning systems. *ISA Transactions*. (vol. 36) n° 2, 81-121
- BAPTISTE P., 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.
- BECK J. C., DAVENPORT A. J., SITARSKI E. M., FOX M. S. 1997a. Texture-based heuristics for scheduling revisited. *Proceedings of American Association of Artificial Intelligence*. Menlo Park, California.
- BECK J. C., DAVENPORT A. J., SITARSKI E. M., FOX M. S. 1997b. Beyond Contention: Extending Texture-Based Scheduling Heuristics. *Proceedings of American Association of Artificial Intelligence*. Menlo Park, California.
- BECK J. C., DAVENPORT A. J., SITARSKI E. M., FOX M. S. 1997c. Five pitfalls of empirical scheduling research. Relatório técnico. Department of Industrial Engineering, University of Toronto.
- BODDY M. S. 1996. Temporal reasoning for planning and scheduling in complex domains. *Advanced Planning Technology Technological Achievements of the ARPA/ROME Laboratory Planning Initiative*. Ed. Austin Tate.
- BRAILSFORD S. C., POTTS C. N., SMITH B. 1999. Constraint Satisfaction Problems: Algorithms and Applications. *European Journal of Operational Research*. n° 199, 557-581
- CASEAU I., LABURTHE F. 1996. Cumulative scheduling with task intervals. *Joint International Conference and Symposium on Logic Programming*. Bonn, Germany.
- CASEAU Y., LABURTHE F. 1995. Improving branch and bound for job shop scheduling with constraint propagation. *Proceedings of the 8th Franco-Japanese 4th Franco-Chinese Conference*.

- CASEAU Y., LABURTHE F. 1994. Improved CLP scheduling with task intervals. *Proceedings of 11th International Conference on Logic Programming*.
- CHENG C. C., SMITH S. F. 1995a. Applying constraint satisfaction techniques to job shop scheduling. Relatório Técnico CMU-RI-TR-95-03, The Robotics Institute, Carnegie Mellon University.
- CHENG C. C., SMITH S. F. 1995b. A constraint-posting framework for scheduling under complex constraints. *Proceedings Point IEEE/INRIA Conference on Engineering Technologies for Factory Automation, Paris, France*.
- CLAIRE *SCHEDULE* 1.0 1999. Reference manual. <http://www.ens.fr/~laburthe>
- EGLI U. M., RIPPIN D. W. T. 1986. Short-term scheduling for multiproduct batch chemical plants. *Computers & Chemical Engineering*. (vol. 10) n° 4 303 - 325
- ERSCHLER J. 1976. Analyse sous contraintes et aide à la décision pour certains problèmes d'ordonnancement. Tese de doutorado, Universidade Paul Sabatier, Toulouse, França.
- FOX M. S. 1983. Constraint-directed search: a case study of job shop scheduling. Tese de doutorado CMU-RI-TR-83-22. School of Computer Science, Carnegie Mellon University, EUA.
- 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.
- HENNING G. P., 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.
- IVENS P., LAMBRECHT M., 1996. Extending the shifting bottleneck procedure to real-life applications. *European Journal of Operational Research*, 252 - 258. Elsevier. Amsterdam, Holanda.

- KENG N. P., YUN D. Y. Y., ROSSI M. 1988. Interaction sensitive planning system for job shop scheduling. *Expert systems and intelligent manufacturing*, 57-69. Elsevier. Amsterdam, Holanda.
- KONDILI E., PANTELIDES C. C., SARGENT 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.
- LE PAPE C., COURONNÉ P., VERGAMINI D., GOSSELIN V. 1995. Time versus capacity compromises in project scheduling. ILOG S.A.
- LE PAPE C. 1994. Implementation of Resource Constraints in ILOG Schedule: A Library for the Development of Constraint-Based Systems. *Intelligent Systems Engineering*. (vol. 3) n° 2. 55 - 66
- LÖHL T., SCHULZ C., ENGELL S. 1998. Sequencing of batch operations for a highly coupled production process: genetic algorithms versus mathematical programming. *Computers and Chemical Engineering*. (vol. 22) 579 - 585
- LÖHL T., SCHULZ C., ENGELL S. 1998. Sequencing of batch operations for a highly coupled production process. *Computers and Chemical Engineering*, 22 (supplement), 579-585.
- MILLARD B. 1998. Just-in-time vs. MRP - Unmasking the great push-pull myth. *APICS*. (vol. 8) n° 3
- MOCKUS L., REKLAITIS G. V. 1999. Continuous time representation approach to batch and continuous process scheduling. 1. MINLP formulation. *Industrial Engineering and Chemical Research*, 38, 197-203.
- MOCKUS L., REKLAITIS G. V. 1999. Continuous time representation approach to batch and continuous process scheduling. *Industrial Engineering and Chemical Research*, 38, 204-210.
- NUIJTEN W. P. M., AARTS E. H. L. 1996. A computational study of constraint satisfaction for multiple capacitated job-shop scheduling. *European Journal of Operational Research*. n° 90, 269-284.
- ORLICKY J. 1975. *Material Requirements Planning*. McGraw-Hill. Nova York, EUA.

- PAPAGEORGIU L. G., PANTELIDES C. C. 1993. A hierarchical approach for campaign planning of multipurpose batch plants. *Computers and Chemical Engineering*, 17 (supplement), 27-32.
- PAPAGEORGIU L. G., 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., 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.
- RICH S. H., PROKOPAKIS G. J. 1986. Scheduling and sequencing of batch operations in a multipurpose plant. *Industrial Engineering Process Design Development*, 25, 979 - 988
- RICH S. H., PROKOPAKIS G. J. 1986. Scheduling and sequencing of batch operations in a multipurpose plant. *Industrial Engineering Chemical Process Des. Dev.*, 25, 979 - 988.
- RODRIGUES L. C. A. 2000d. Planejamento e programação da produção em plantas multipropósito operando em batelada: Proposta de uma estratégia de decomposição usando janelas de tempo. Tese de doutorado, Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas.
- RODRIGUES L. C. A., GRAELLS M., CANTÓN J., GIMENO L., RODRIGUES M. T. M., ESPUÑA A., 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.
- RODRIGUES M. T. M., GIMENO L., PASSOS C. A. S., 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., 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., 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.
- SADEH N. 1991. Look-ahead techniques for micro-opportunistic job shop scheduling, Tese de doutorado, School of Computer Science, Carnegie Mellon University.
- SANMARTI E., FRIEDLER F., PUIGJANER L. 1998. Combinatorial technique for short term scheduling of multipurpose batch plants based on schedule-graph representation. *Computers and Chemical Engineering*. (vol. 22) 847 - 850
- SAP 1999. SAP Advanced Planner & Optimiser. Production Planning and Detailed Scheduling. <http://www.sap.com>.
- SHAH N., CHUNG P. W. H., DAS B. P., HUANG W. 1999. A comparative study of the time-based and activity-based approaches to multipurpose multistage batch process production scheduling. *2nd Conference on Process Integration, Modelling and Optimisation for Energy Saving and Pollution Reduction*.
- SHAH N., PANTELIDES C. C., SARGENT R. W. H. 1993. Optimal periodic scheduling of multipurpose batch plants. *Annals of Operational Research*. n° 42. 193 - 228
- SHAH N., PANTELIDES C. C., SARGENT R. W. H. 1993b. A general algorithm for short term scheduling of batch operations – II. Computational issues. *Computers and Chemical Engineering*, 17 (2), 229-244.
- SHAH N., CHUNG P. W. H., DAS B. P. 1998. Off-line scheduling a simple chemical batch process production plan using ILOG Scheduler. *Computer & Chemical engineering*. (vol. 22) Supplement. S947 - S950.
- SMITH S. F., CHENG C. C. 1993. Slack-based heuristics for constraint satisfaction scheduling. *Proceedings of 11th National Conference on Artificial Intelligence*, Washington DC, 139-144
- SMITH S. F., OW P. S. 1985. The use of multiple problem decompositions in time constrained planning tasks. *Proceedings of the 9th International Joint Conference on Artificial Intelligence*. pág. 1013-1015

- SMITH S. F., PATHAK D. K. 1992. Balancing antagonistic time and resource utilization constraints in over-subscribed scheduling problems. *8th Conference on Artificial Intelligence for Applications*. Monterey, California.
- SUBRAHMANYAM S., PEKNEY J.F., REKLAITIS G.V. 1996. Decomposition approaches to batch plant design and scheduling. *Industrial Engineering and Chemical Research*, 35 (6), 1866-1876.
- TAAL M., WORTMANN J. C. 1997. Integrating MRP and finite capacity planning. *Production Planning & Control*. (vol. 8) n^o 3. 245 - 254