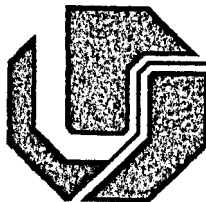


MO1
621.3
1488/0
34/ME4

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA

PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA



ESCALONAMENTO DE SISTEMAS DE PRODUÇÃO

HÍBRIDOS USANDO UMA REDE DE PETRI

P-TEMPORAL T-TEMPORIZADA COM MECANISMO DE

RETROCESSO INTELIGENTE

CLARIMUNDO MACHADO MORAES JÚNIOR

MARÇO

2000

SISBI/UFU



1000205722

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

**ESCALONAMENTO DE SISTEMAS DE PRODUÇÃO
HÍBRIDOS USANDO UMA REDE DE PETRI
P-TEMPORAL T-TEMPORIZADA COM MECANISMO DE
RETROCESSO INTELIGENTE**

Dissertação apresentada por Clarimundo Machado Moraes Júnior à
Universidade Federal de Uberlândia para obtenção do título de Mestre
em Engenharia Elétrica aprovada em 22/03/00 pela Banca Examinadora:

Professor Stéphane Julia, Dr. (UPS/LAAS) - Orientador

Professor Haroldo Rodrigues de Azevedo, Dr. (UFU)

Professor Paulo Eigi Miyagi, Dr. (USP)

Professor Márcia Aparecida Fernandes, Dra. (UFU)

**ESCALONAMENTO DE SISTEMAS DE PRODUÇÃO
HÍBRIDOS USANDO UMA REDE DE PETRI
P-TEMPORAL T-TEMPORIZADA COM MECANISMO DE
RETROCESSO INTELIGENTE**

CLARIMUNDO MACHADO MORAES JÚNIOR

Dissertação apresentada por Clarimundo Machado Moraes Júnior
à Universidade Federal de Uberlândia como parte dos requisitos
para obtenção do título de Mestre em Engenharia Elétrica.

Prof. Dr. Stéphane Julia
Orientador

Prof. Dr. Luiz Carlos de Freitas
Coordenador do Curso de Pós-Graduação

DEDICATÓRIA

**Aos meus queridos pais,
Clarimundo e Glória, por
todo amor, sacrifício e
incentivo que sempre
demonstraram.**

*“Se um dia, já homem feito e realizado,
sentires que a terra cede a teus pés e
que não há ninguém à tua volta para te
estender a mão, esquece a tua
maturidade, passa pela tua mocidade,
volta à tua infância e pense nas últimas
palavras que sempre te restarão na tua
alma: Meu pai, minha mãe.”*

AGRADECIMENTOS

À Deus, pelo amor incondicional e presença constante em minha vida.

Às minhas irmãs, Cláudia e Cristina, pelo amor e amizade que sentem por mim.

Ao meu afilhado e sobrinhas, Guilherme, Flávia e Ana Luíza , por existirem.

Aos amigos, Celestino e Donizete, pela amizade sincera e sólida.

Às minhas tias, Elite e Edite, pelo otimismo, e amor que sentem por mim.

Ao Departamento de Engenharia Elétrica da Universidade Federal de Uberlândia, pelos recursos oferecidos para a execução deste trabalho.

Ao meu orientador Prof. Dr. Stéphane Julia, por sua orientação, dedicação, paciência, e principalmente, pela a amizade que demonstrou durante a produção deste trabalho.

À CAPES Fundação Coordenação de Aperfeiçoamento de Pessoal de Nível Superior, pela ajuda financeira recebida no decurso do meu trabalho de pós-graduação.

À secretária Joana, pela ajuda que sempre ofereceu.

Aos colegas do Laboratório de Automática, em especial ao Robson e Adnei, o meu muito obrigado.

Abstract

The objective of this work is to deal the scheduling problem of the batch systems, with cyclic feeding policy, based on the representation of the system constraints using a p-time t-timed Petri net model and on a token player algorithm with a intelligent backtrack mechanism to find a feasible sequence consistent with the set of constraints.

The p-time t-timed Petri net is more general than an event graphs, it allows on the hand to separate the production routes constraints and the resource allocation constraints and on the other hand to modelize the operation durations of treatment of the batch associated with the places and the operation durations of transference of the batch associates with the transitions.

Furthermore, the backtrack mechanism used in this work not is sistematic but intelligent in accord with analyse of the net's present state.

Resumo

O objetivo deste trabalho é tratar o problema de escalonamento dos sistemas de produção por lotes, com política de produção cíclica, baseado na representação formal das restrições do sistema através de um modelo de rede de Petri p-temporal t-temporizado e usando um “jogador” de rede de Petri com um mecanismo de retrocesso (“backtrack”) inteligente, que busca uma sequência admissível que satisfaça as restrições.

O modelo de rede de Petri p-temporal t-temporizado é mais geral que um grafo de eventos, pois, permite de um lado separar as restrições de roteiros das restrições de recursos, e de outro lado modelar as durações das operações de tratamento dos lotes associadas aos lugares e às durações das operações de transferência dos lotes associadas às transições.

Vale ressaltar, que o mecanismo de retrocesso utilizado neste trabalho não é feito de forma sistemática, e sim de maneira “inteligente” de acordo com uma análise prévia do estado corrente da rede.

Conteúdo

1	Introdução	7
2	Sistemas de Produção Híbridos	9
2.1	Sistemas de Produção	9
2.1.1	Sistemas Flexíveis de Manufatura (SFM)	13
2.1.2	Semelhanças entre SFM e Sistemas de Produção por Lotes	15
2.1.3	Ciclo de Vida de um Sistema de Produção	15
2.1.4	Sistema de Computação Integrada (CIM)	20
2.2	Redes de Petri para Sistemas de Produção Híbridos	22
2.2.1	Modelo Contínuo Estendido com Variáveis Booleanas	22
2.2.2	Modelo Discreto Estendido	24
2.2.2.1	Autômatos Finitos	25
2.2.2.2	Redes de Petri (RdPs)	31
2.2.2.3	RdPs e Sistemas de Produção Híbridos	36
2.3	Problema do Escalonamento	50
2.3.1	Definição do Problema do Escalonamento	50
2.3.2	Abordagem Clássica	53
2.3.3	RdPs para o Problema do Escalonamento	55
2.3.3.1	Para a Manufatura	55
2.3.3.2	Para Sistemas de Produção por Lotes	58
2.3.3.3	Conclusões do Capítulo	62

3	Modelagem de Sistemas de Produção por Lotes	64
3.1	Modelagem das Restrições de Sincronização	65
3.1.1	Receitas	65
3.1.1.1	Roteiros	65
3.1.1.2	Sub-roteiros	66
3.1.1.3	Restrição Kanban	68
3.1.2	Alocação de Recursos	69
3.1.3	Política Cíclica	72
3.1.4	Construção do Modelo Global	74
3.2	Modelo Temporal/Temporizado	79
3.2.1	Modelo t-temporizado	79
3.2.2	Modelo p-t-temporizado	81
3.2.3	Modelo p-temporal	82
3.2.4	Modelo p-temporal t-temporizado	83
3.3	Restrições Temporais e Sistemas de Produção por Lotes	86
3.3.1	Restrições Temporais para uma Restrição Kanban	86
3.3.2	Restrições Temporais para uma Restrição de Recurso	88
4	Princípio Geral para o Escalonamento Cíclico	89
4.1	RdP e Mecanismo de Ajuda à Tomada de Decisão	89
4.2	Noção de Conflito numa RdP p-temporal t-temporizada	90
4.2.1	Conflito para uma RdP Autônoma	91
4.2.2	Conflito para uma RdP t-temporizada	92
4.2.3	Conflito para uma RdP p-temporal t-temporizada	93
4.3	Resolução de Conflito para o Escalonamento	96
4.3.1	Obtenção de um Estado de Conflito	96
4.3.2	Estratégia de Resolução de Conflito	99
4.4	“Jogador” de RdP p-temporal t-temporizado	102
4.4.1	Algoritmo	102

4.4.2	Mecanismo de Retrocesso	105
4.5	Princípio Geral da Pilotagem em Tempo Real	111
5	Exemplo	115
5.1	Modelagem	115
5.1.1	Restrições de Sincronização	116
5.1.2	Restrições Temporais do Exemplo	118
5.2	Implementação da RdP p-temporal t-temporizada	122
5.3	Simulação e Resultados	127
6	Conclusão	131

Lista de Figuras

2.1	Exemplo de Sistema de Produção por Lotes	11
2.2	Sistema Dinâmico de Variáveis Contínuas	12
2.3	O reator	23
2.4	Realimentação de eventos para autômato controlado	24
2.5	Autômato Finito	25
2.6	Autômato de um Reservatório Fechado	25
2.7	Autômato Finito Determinístico	26
2.8	Autômato Finito não Determinístico	27
2.9	Eliminação de Concatenações	27
2.10	Eliminação das Transições não Exclusivas	28
2.11	Autômato de um Termostato	29
2.12	Autômato de um Tanque de Água com Bomba	30
2.13	Rede de Petri	32
2.14	Rede de Petri Marcada	32
2.15	Exemplos de Disparos de Transições numa RdP	33
2.16	Modelos Diferentes para um mesmo Sistema	36
2.17	Tipos de Modelagens numa RdP	37
2.18	Conflito Estrutural	37
2.19	Conflito Efetivo	38
2.20	Um fragmento de receita	38
2.21	Modelo de rede de Petri temporizada	38

2.22	Modelo de rede de Petri híbrida	39
2.23	Modelo de rede de Petri e equações diferenciais	40
2.24	(a) Sistema (b) Rede de Controle	42
2.25	Modelo de Referência	43
2.26	Exemplo de um sistema híbrido	44
2.27	Exemplo de RdP de um Sistema de Produção por Lotes	48
2.28	Exemplo de RdP Colorida com a Declaração das Cores	49
2.29	Exemplo de RdP orientada a objetos	50
2.30	Descrição da planta	54
2.31	Modelagem: a) da planta b) de duas receitas c) de uma máquina	54
2.32	Escalonamento flexível	57
2.33	Sistema do “guindaste”	59
2.34	Exemplo de Rede de Petri p-temporal	60
2.35	Duas unidades de uma planta de produção por lotes	60
2.36	Espaço de busca	61
2.37	Espaço de busca reduzido	62
3.1	Roteiro r	66
3.2	Sub-roteiros $s1r$ e $s2r$	67
3.3	Restrição kanban C_k	69
3.4	Alocação de Recurso C_{R2}	71
3.5	Política Cíclica C_c	74
3.6	Fusão de C_{k1} com C_{R2}	75
3.7	Exemplo 1 de utilização de Sifon	77
3.8	Exemplo 2 de utilização de Sifon	78
3.9	Evolução Dinâmica da Rede de Petri t-temporizada	80
3.10	Evolução Dinâmica da Rede de Petri p-t-temporizada	81
3.11	Evolução Dinâmica da Rede de Petri p-temporal	83
3.12	Evolução Dinâmica da RdP p-temporal t-temporizada	84

3.13	Exemplo 1 de Restrições Temporais Associadas à uma Restrição Kanban . . .	87
3.14	Restrições Temporais Associadas à uma Restrição de Recurso	88
4.1	Noção de Conflito para uma Rede de Petri Autônoma	91
4.2	Exemplo 1 de um Intervalo de Sensibilização de uma Transição	94
4.3	Exemplo 2 de um Intervalo de Sensibilização de uma Transição	94
4.4	Sequência dos Possíveis Intervalos	95
4.5	Noção de Conflito para uma Rede de Petri p-temporal t-temporizada . . .	95
4.6	Exemplo de Cálculo dos Intervalos de Visibilidade	98
4.7	Resolução de Conflito	100
4.8	Exemplo de Uso do Mecanismo de Retrocesso	106
4.9	Exemplo de Mudança da Marcação Inicial	110
4.10	Princípio Supervisão/Pilotagem	112
4.11	Associação de uma Operação a uma Transição	113
5.1	Exemplo de Sistema de Produção por Lotes	115
5.2	Modelagem das receitas Utilizando Rede de Petri	116
5.3	Modelagem dos lugares Kanban Utilizando Rede de Petri	117
5.4	Modelagem dos reatores R_1 , R_2 e R_3 Utilizando Rede de Petri	118
5.5	Modelagem dos lugares cíclicos Utilizando Rede de Petri	119
5.6	Escolha das Bordas Máx. dos Intervalos Estáticos dos Lugares Kanbans . .	120
5.7	Modelagem do Sistema de Produção por Lotes Utilizando Rede de Petri . .	123
5.8	Estrutura de Dados local da rede de Petri	124
5.9	Rede de Petri com as novas condições iniciais	129

Capítulo 1

Introdução

Antes da Revolução Industrial os sistemas de produção eram mais simplificados e, conseqüentemente, mais limitados (linhas de produção que produziam em série e em grande escala). Hoje o que se tem são sistemas de produção mais complexos que podem produzir vários tipos de produtos numa mesma planta. A complexidade dos sistemas de produção se deve à evolução das ciências como a informática industrial ou a robótica, cujo principal objetivo é o de conciliar a eficácia e a flexibilidade que geralmente são considerados critérios contraditórios.

Os sistemas de produção atuais, também chamados de sistemas flexíveis, possuem um ciclo de desenvolvimento extremamente complexo. Devido ao seu caráter híbrido, a visão discreta e a contínua sempre exigem uma técnica de análise específica para cada uma delas, devendo estas técnicas estarem integradas entre si. Com isto, o principal problema que se encontra é obter a capacidade de propor uma metodologia completa de concepção, de análise e de comando dos sistemas flexíveis. Uma regra imperativa a ser observada a fim de garantir, na medida do possível, um desenvolvimento coerente e rigoroso é a utilização de uma metodologia formal. Este trabalho se limitará aos sistemas de produção por lotes, que são modelos híbridos e flexíveis. Em particular, será considerado o problema do escalonamento cíclico de tais sistemas.

O problema do escalonamento tratado neste trabalho será o de buscar seqüências de

operações que sejam admissíveis e não necessariamente ótimas. Para tal abordagem, será utilizada uma técnica híbrida que deve combinar de um lado as redes de Petri para a modelagem e para a análise quantitativa e quantitativa e de outro lado certas técnicas de resolução de Inteligência Artificial dentro do contexto da análise sob restrição.

No capítulo 2, serão apresentados os sistemas de produção híbridos. Inicialmente, serão mostrados os diversos tipos de sistemas de produção que existem. Em particular, será feito um comparativo entre os sistemas flexíveis de manufatura e os sistemas de produção por lotes. Em seguida, serão apresentados os modelos de redes de Petri para sistemas de produção híbridos, fazendo uma distinção dos aspectos discretos e contínuos que são levados em conta. E finalmente, definirá-se o problema do escalonamento quando forem considerados os sistemas de produção por lotes.

No capítulo 3, serão apresentadas todas as considerações a serem feitas na modelagem de sistemas de produção por lotes, quanto às restrições de sincronização. Em seguida, serão apresentados os tipos de modelos temporal/temporizado, bem como, suas respectivas dinâmicas.

No capítulo 4, será apresentado o princípio geral para o escalonamento. Inicialmente, citará-se a noção de conflito para uma rede de Petri p-temporal t-temporizada e em seguida, explicará-se a estratégia de resolução de conflito para o escalonamento do sistema de produção por lotes através da definição de um algoritmo do jogador de rede de Petri p-temporal t-temporizada e de um mecanismo de retrocesso "inteligente". Por último, será feita uma breve abordagem ao princípio geral para pilotagem em tempo real.

No capítulo 5, será tratado especificamente o exemplo prático deste trabalho. Inicialmente, mostrará-se a modelagem do sistema considerado e em seguida, serão definidas as estruturas de dados utilizadas na implementação e finalmente, serão citados os resultados práticos da simulação feita no laboratório.

No capítulo 6, será feita a conclusão do trabalho além de algumas sugestões futuras de estudo.

Capítulo 2

Sistemas de Produção Híbridos

2.1 Sistemas de Produção

Atualmente, existe um interesse crescente no domínio dos sistemas de produção híbridos e das técnicas de modelagem de tais sistemas. Uma das principais motivações disto é o desenvolvimento do controle supervisão para os processos industriais contínuos. Por outro lado, as mudanças de comportamento na sociedade de consumo fazem com que os processos industriais contínuos de produtos únicos sejam substituídos por sistemas de produção por lotes (“batch systems”) mais flexíveis. Os sistemas por lotes permitem a produção simultânea de vários produtos, o que é particularmente interessante na indústria agro-alimentar, por exemplo.

Os processos industriais por lotes operam em matéria prima contínua (geralmente fluidos). Eles são compostos por equipamentos do tipo contínuo (as canalizações, por exemplo) e por equipamentos do tipo discreto (reatores, estoques intermediários etc.). Os lotes são quantidades de matérias contínuas caracterizadas por números reais (volumes etc.). O fato de manipular vários lotes que podem ser tratados simultaneamente e que são transferidos de reator em reator, faz com que estes sistemas assemelhem-se muito aos sistemas flexíveis de manufatura (Kusiak 1990). Em particular, em ambos os tipos de sistema aparecem problemas semelhantes de alocação de recursos. Por exemplo, o uso

de um mesmo reator pode ser necessário para o tratamento de vários lotes de matéria prima. Se o reator funciona em modo fechado (um único produto ao mesmo tempo, em um único reator), de certa forma ele pode ser considerado como um recurso discreto análogo ao de uma célula flexível de manufatura, ainda que certas diferenças existam e devam ser consideradas. Neste contexto, parece interessante adaptar as técnicas de resolução já existentes no campo da manufatura aos sistemas de produção por lotes. Se em um sistema flexível de manufatura a preocupação principal é a obtenção do escalonamento ótimo (Lee and DiCesare 1994b), nos sistemas de produção por lotes, a obtenção de um escalonamento admissível (que respeite um conjunto de restrições) é mais relevante. Assim sendo, nestes últimos, o problema de escalonamento deve ser visto como um problema de satisfação de restrições, e não como um simples problema de otimização. Por exemplo, nos sistemas de produção por lotes, um tempo de espera grande demais entre duas operações pode estragar o lote de produto, o que não acontece no caso de uma peça tratada por um sistema flexível de manufatura.

A figura 2-1 é um exemplo de sistema de produção por lotes. As especificações do sistema são:

- Há 2 buffers de entrada ($E1, E2$), 2 buffers de saída ($S1, S2$) e nenhum buffer intermediário;
- Há 2 reatores ($R1, R2$) que funcionam em modo fechado (um único produto ao mesmo tempo, em um único reator);
- O sistema realiza 2 receitas, sendo estas realizadas com as seguintes sequências de reatores:
 $RECEITA\ 1 \rightarrow R1\ R2$ e $RECEITA\ 2 \rightarrow R2$;

Observe que o reator $R1$ é utilizado somente pela RECEITA 1 e o reator $R2$ é utilizado pelas duas receitas. Portanto, em $R2$ tem-se um compartilhamento, ou seja, um recurso que pode ser solicitado por mais de uma receita. Outro ponto a ser observado é que para cada reator e para cada buffer temos um estado discreto, onde a abertura e fechamento das válvulas representa a mudança de estados. O aspecto contínuo está nos lotes que são, geralmente, fluidos cuja variável contínua é o volume ou massa molar representados por

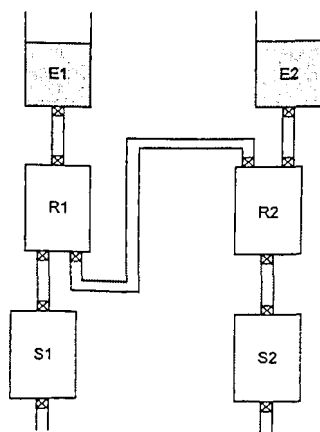


Figura 2.1: Exemplo de Sistema de Produção por Lotes

números reais. Ao serem tratados os lotes são transferidos de modo contínuo através das tubulações. O tempo de tratamento e transferência de cada lote pode também fornecer o aspecto contínuo do sistema.

Neste caso, como será importante ter uma abordagem híbrida que trate os aspectos discretos e contínuos dos sistemas de produção por lotes, deve-se utilizar um modelo que comporte tais aspectos. A seguir, são dadas definições de modelos que tratam apenas os aspectos contínuos ou que tratam apenas os aspectos discretos e, posteriormente, será mostrado um modelo com abordagem híbrida.

Sistemas Contínuos

São sistemas cujas variáveis que representam o estado do sistema mudam continuamente com o tempo e cujos modelos são contínuos. Para acompanhar o comportamento dinâmico de tais sistemas, utiliza-se modelos matemáticos descritos por equações álgebra-diferenciais. (Ho 1989);(Cassandras and Ramadge 1990).

Na figura 2.2, o estado X do sistema está constantemente mudando com o tempo.

Sistemas Discretizados

São sistemas estudados somente em instantes precisos, ou melhor, em tempos discretos

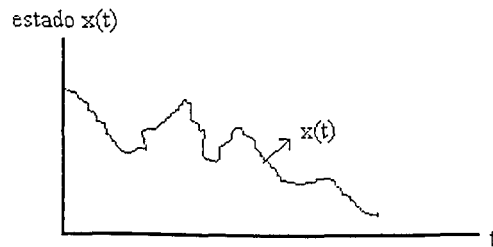


Figura 2.2: Sistema Dinâmico de Variáveis Contínuas

(sistemas amostrados). Nestes sistemas as variáveis de estado mudam continuamente no tempo, sem mudança brusca de comportamento, mas é somente em instantes discretos do tempo que há interesse em conhecer seu valor.

Sistemas Discretos

São sistemas cujas variáveis de estado, ou ao menos algumas delas, mudam bruscamente a certos instantes. Entretanto, estes instantes não podem, necessariamente, ser previstos e o conhecimento do estado a um dado instante não permite que, sem cálculo, se conheça o estado seguinte. Em outras palavras, sistemas discretos são sistemas com estados discretos e tempo contínuo.

Sistemas a Eventos Discretos

São sistemas cujas variáveis de estado variam bruscamente em instantes determinados e que os valores das variáveis nos estados seguintes podem ser calculados diretamente a partir dos valores precedentes e sem ter que considerar o intervalo de tempo entre estes dois instantes. Em outras palavras, sistemas a eventos discretos são sistemas com estados discretos e tempo discreto. **Sistemas Híbridos**

Sistemas de produção híbridos são sistemas com comportamento discreto e/ou contínuo. Quando um sistema possui sub-sistemas com aspectos discretos e outros sub-sistemas com

aspectos contínuos, então, trata-se de um sistema híbrido. Vale ressaltar que o aspecto discreto e o contínuo não devem ser distinguidos de modo absoluto, ou seja, a caracterização de um e de outro depende da análise feita. Por exemplo, numa indústria química, uma quantidade de fluido tratada no sistema, pode ser vista como discreta se for vista como um lote bem definido e pode ser vista como contínua se for considerado o fluxo contínuo do material pelo sistema. No primeiro caso, a análise é feita de maneira local (lote sendo tratado localmente), e no segundo caso, a análise é feita de maneira global com cada equipamento sendo representado pela mesma função durante todo o ciclo do processo (Silva *et al.* 1998).

Os sistemas híbridos são sistemas cujas variáveis de estado e cujo tempo podem ser dos dois tipos (discreto e contínuo).

Como já foi dito, os sistemas de produção por lotes são sistemas híbridos e que possuem características flexíveis. Uma vez que associa-se a flexibilidade de tais sistemas ao fato de manipularem vários lotes que podem ser tratados simultaneamente e que são transferidos de reator em reator, faz com que estes sistemas assemelhem-se muito aos sistemas flexíveis de manufatura. Em particular, em ambos os tipos de sistema aparecem problemas semelhantes de alocação de recursos. Por exemplo, o uso de um mesmo reator pode ser necessário para o tratamento de vários lotes de matéria-prima.

2.1.1 Sistemas Flexíveis de Manufatura (SFM)

São sistemas que fazem parte de uma classe de sistemas de manufatura automatizada cuja característica principal, que a diferencia dos outros sistemas é a sua flexibilidade para processar eficientemente mais de um tipo de peça em quantidades médias. Aqui o conceito de flexibilidade vai se referir não apenas à possibilidade de processar mais de um tipo de produto ou peça, mas também à capacidade de mudar rapidamente o tipo de peça a ser produzido, além da possibilidade de manufaturar simultaneamente diferentes tipos de peças. Assim os SFM surgiram como uma tentativa para reconciliar a eficiência da linha de produção do tipo flow-shop, com a flexibilidade de um sistema do tipo job-

shop com a finalidade de satisfazer uma demanda versátil a um baixo custo. A alta produtividade neste tipo de sistema é alcançada, portanto, pela incorporação eficiente de princípios de tecnologia de grupo, controle total de qualidade e estratégias de produção, tais como MRP II, JIT (Viswanadham and Narahari 1987), etc.

Um SFM é composto de:

1. Um conjunto de máquinas e ferramentas flexíveis de comando numérico;
2. Um sistema automático de transporte e manuseio de peças e ferramentas;
3. Um sofisticado sistema de tomada de decisão para decidir a cada instante quais são as operações que devem ser executadas nas máquinas;

Uma máquina flexível tem a capacidade de executar várias operações, possui um sistema de armazenagem e restauração automático, e permite a adição de programas de máquinas. Esta flexibilidade pode ser chamada de flexibilidade física.

Frequentemente, um SFM é composto de várias células de manufatura. Uma célula é um sistema elementar de manufatura consistindo de uma máquina flexível (ou um esquema de montagem, ou algum esquema dedicado às operações complexas da manufatura). Uma célula permite o agrupamento das peças em famílias de peças, segundo suas características de projeto.

Os sistemas de manufatura têm sido freqüentemente modelados como sendo discretos, os quais devido a sua complexidade não podem ser modelados por um modelo que utilize o conceito de estado total (filas de espera, por exemplo), pois, a partir de um estado pode acontecer muitos eventos provocando diversas alterações de estado. Portanto, para modelar um SFM é necessário procurar métodos que possuam de modo inerente o conceito de estado parcial. Outro aspecto importante é o fato de que durante a vida útil dos SFMs estão envolvidos especialistas de diferentes formações. Portanto, para que a contribuição destas pessoas seja aproveitada eficientemente, a metodologia e a linguagem de programação devem ser de fácil aprendizado e compreensão, além de possuírem boa legalidade através de representações gráficas simples (Rillo 1988).

Desta forma, todos os conceitos anteriores podem considerar um SFM como um sistema dinâmico de eventos discretos (DEDS), do qual, as partes a serem processadas de vários tipos ou classes de trabalho entram no sistema de uma maneira assíncrona, e são processados concorrentemente, compartilhando os recursos limitados (estações de trabalho, robôs, sistemas de manuseio de materiais, buffers, etc).

2.1.2 Semelhanças entre SFM e Sistemas de Produção por Lotes

Apesar dos sistemas flexíveis de manufatura serem representados como sendo discretos e os sistemas de produção por lotes como sendo híbridos, o segundo tipo pode assemelhar-se ao primeiro, se for analisado sob certos aspectos. Em relação aos seus funcionamentos, em ambos os casos podem ocorrer: paralelismo (eventos que ocorrem simultaneamente), sincronismo (eventos comuns a várias evoluções) e recursos compartilhados (recursos que podem ser solicitados para mais de uma operação distinta, como por exemplo, um robô pode ser utilizado para dois tipos de operações distintas ou um reator pode ser utilizado para tratar lotes de receitas diferentes).

Devido à semelhança dos dois sistemas, será interessante fazer uma apresentação de dois aspectos considerados importantes nos SFM e que, conseqüentemente, são importantes nos sistemas de produção por lotes. Uma vez considerados estes aspectos, podemos lançar mão de técnicas de modelagem e análise utilizadas nos SFM e que poderão ser adaptadas aos sistemas de produção por lotes. O primeiro aspecto é com relação ao ciclo de vida dos sistemas de produção por lotes e o segundo aspecto é com relação a aplicação da informática industrial (CIM) ao setor industrial e, particularmente, aos sistemas de produção por lotes.

2.1.3 Ciclo de Vida de um Sistema de Produção

O ciclo de vida de um sistema de produção mostrado em (Silva and Valette 1989) e (Julia 1998), é geralmente constituído pelo seguinte conjunto de etapas:

- Especificação dos produtos.
- Concepção preliminar.
- Concepção detalhada.
- Produção.
- Desmembramento ou evolução do sistema.

Especificação dos Produtos

A especificação dos produtos é o resultado direto de um estudo de mercado e é a origem do levantamento de um projeto de concepção de um sistema de produção. Vale salientar que, com a introdução da noção de flexibilidade, os sistemas de produção devem ser capazes não somente de realizar diversos tipos de produtos, como também de seguir a evolução do mercado e, conseqüentemente, de modificar progressivamente os produtos produzidos inicialmente dentro do sistema de produção. Tudo isso tem por conseqüência deixar esta etapa muito mais complexa do que quando a problemática era produzir um único tipo de produto em grande escala e em um certo período de tempo. É preciso ser capaz de definir uma gama de produtos com uma certa antecipação a fim de prever a funcionalidade que o sistema deverá ter para efetuar, da melhor maneira possível, a produção. Uma das funções importantes desta etapa é a definição, para cada tipo de produto, do conjunto das seqüências de operações possíveis para a sua realização, isto é, a definição dos roteiros de produção.

Concepção Preliminar

Durante esta fase, efetua-se as seguintes tarefas:

- Escolher os recursos disponíveis (máquinas, sistemas de transporte, reatores, etc) para que se realize a produção definida na etapa precedente.

- Estudar a melhor disposição dos recursos sobre a superfície disponível (disposição do sistema).
- Escolher o sistema de gestão de produção que será aplicado ao sistema físico.
- Avaliar o comportamento dinâmico do sistema a fim de obter um conjunto de índices de desempenho que vai permitir verificar se o sistema corresponde às especificações que foram definidas na etapa de especificação dos produtos. Geralmente, neste nível utilizam-se técnicas de simulação e, eventualmente, técnicas analíticas para a avaliação de desempenho.

Durante toda esta fase de concepção preliminar, é importante que se tente, na medida do possível, otimizar um conjunto de critérios que caracterize uma produção eficiente, de boa qualidade e que corresponda a um custo dos mais competitivos possíveis. Vale ressaltar que este problema não é trivial, visto que ele corresponde a um problema de otimização multicritérios.

Concepção Detalhada

A concepção detalhada consiste em realizar a implementação física do sistema de produção. Depois de ter formado os operadores que vão definir uma boa atuação do sistema, deve-se colocar o sistema em regime moderado de funcionamento a fim de efetuar os últimos ajustes que devem anteceder o funcionamento em regime pleno.

Produção

A produção corresponde ao funcionamento normal do sistema. Aparecem aqui os problemas de gestão de produção, de conduta do sistema em tempo real, de monitoramento, de manutenção do sistema e de supervisão. A complexidade de um sistema de produção flexível torna difícil um estudo global do sistema, fazendo com que a estrutura de decisão seja geralmente decomposta em uma hierarquia de níveis de abstração (Erschler and Esquirol 1986); (Silva and Valette 1989). O objetivo de tal decomposição é transformar

a resolução de um problema de grande dimensões, com características variadas, a uma sequência de resolução de problemas de tamanhos razoáveis e com características mais homogêneas. O objetivo aqui não é mais de escolher uma solução dentro de um conjunto de alternativas, mas, de preferência, de restringir progressivamente o conjunto das alternativas possíveis usando o conhecimento próprio a cada nível. Os níveis de decisão clássicos são os seguintes:

Nível de Planificação

Neste nível de produção, deve-se estabelecer um plano de produção agregado em função de uma estimativa do pedido dos clientes e da previsão da evolução do mercado. Em particular, é aqui que se define as diversas taxas de produção, isto é, o número de produtos a serem tratados por semana ou por mês. Faz-se também, neste nível, uma pré-alocação dos diversos recursos, visando a reduzir o risco de uma explosão combinatorial que pode acontecer, posteriormente, no nível de escalonamento. O risco desta explosão provém da flexibilidade das máquinas, que podem, geralmente, ser utilizadas para mais de um tipo de operação.

Nível do Escalonamento

É neste nível que deve ser elaborado o plano de fabricação detalhado. Define-se a ordem de execução de cada operação, sobre cada máquina, considerando-se as restrições impostas no nível de decisão anterior (planificação). Em particular, as restrições estabelecem o respeito às taxas de produção, bem como uma pré-alocação dos recursos. Nos sistemas de produção, o fato de cada máquina poder executar diversas operações gera numerosas alternativas possíveis que aumentam o risco de uma explosão combinatorial durante o escalonamento. O escalonamento é, às vezes, dado como uma sequência de operações ordenadas. Neste caso, considera-se que foi definido um escalonamento explícito. Deve-se evitar esta estratégia porque ela requer tempos de cálculos significativos e não é de modo algum flexível. Uma outra solução é a de elaborar um conjunto de regras que definem a

operação seguinte a cada vez que uma máquina torna-se disponível. Tal estratégia pode ser considerada como a definição de um escalonamento implícito. Geralmente, é uma abordagem mista, que combina as duas estratégias, que proporciona o melhor resultado, ou seja, utilizar um escalonamento explícito em partes do sistema onde não há possibilidade de mudança na sequência de operações e um escalonamento implícito onde a flexibilidade é exigida pelo sistema.

Nível da Supervisão e Pilotagem em Tempo Real

Suas duas funções principais são o monitoramento do sistema e a tomada de decisão em tempo real.

A função de monitoramento permite detectar, em tempo real, todos os comportamentos anormais (pane de máquina, etc.) (Sahraoui 1987); (Courvoisier *et al.* 1989) e efetuar um diagnóstico eficiente, a fim de dar a melhor orientação possível ao grupo de manutenção.

A função tomada de decisão em tempo real deve resolver, em tempo real, eventuais indeterminações que ainda existam. O sistema deve tentar seguir o escalonamento provisional que foi calculado na etapa anterior e deve ser capaz, em caso de um comportamento anormal, de redistribuir as tarefas a serem efetuadas em tempo real para que a produção não seja interrompida. Vale salientar que, quando o escalonamento é definido de maneira implícita através de regras de decisão, poderá ser chamado de escalonamento em tempo real.

Nível de Coordenação dos Sub-sistemas

A função principal deste nível é a de cuidar do comando em tempo real dos sub-sistemas. Em particular, isto se relaciona com a gestão do sistema de transporte que permite fazer a ligação entre as diversas células.

Nível do Comando Local

É neste nível que é implementado o comando em tempo real das máquinas, dos reatores, etc., principalmente através de autômatos programáveis (Courvoisier and Valette 1986) e de comandos numéricos. Nenhuma decisão é tomada neste nível e as restrições de tempo são geralmente expressivas, visto que este é o nível mais baixo de decisão.

Desmembramento ou Evolução do Sistema

Esta etapa pode ser considerada como o fim do ciclo de vida do sistema de produção por lotes. Geralmente, ela torna-se necessária quando as perturbações no nível de mercado são tão importantes que o sistema implementado não é mais capaz de responder com eficácia às necessidades de produção. Neste caso, o sistema será modificado a fim de se adaptar à nova produção ou, então, ele será desmontado e um novo projeto de concepção de sistema de produção, mais atualizado, será lançado.

2.1.4 Sistema de Computação Integrada (CIM)

Os progressos consideráveis da Informática nestes últimos anos têm permitido a introdução de um alto grau de automação na indústria cujas consequências são:

- uma produção grande e regular;
- a melhoria da qualidade dos produtos;
- a melhoria das condições de trabalho e de segurança;

Hoje os sistemas são equipados com um sistema de computação (*CIM: Computer Integrated Manufacturing*) que integra um conjunto de funções que permitem automatizar ou auxiliar o conjunto das etapas do ciclo de desenvolvimento. Em geral, as diversas funções de um CIM são divididas em três módulos distintos, que são:

- a *Concepção Assistida por Computador (CAD: Computer Aided Design)*,

- a *Engenharia Assistida por Computador (CAE: Computer Aided Engineering)*,
- a *Fabricação Assistida por Computador (CAM: Computer Aided Manufacturing)*.

Concepção Assistida por Computador (CAD)

A *Concepção Assistida por Computador* atua na fase de especificação dos produtos. Este módulo é utilizado para a descrição dos produtos, para a escolha do material que vai ser utilizado e para definir a maneira em que o material vai ser utilizado para a fabricação dos produtos (Nau *et al.* 1996). Em particular, é neste nível que são definidas as diversas famílias de produtos em função das semelhanças das operações que vão ser aplicadas a elas. A partir desta definição serão definidos os diversos roteiros de produção possíveis. Posteriormente, a definição simples e eficaz de uma sequência admissível (escalonamento) dependerá, essencialmente, da maneira em que foram reagrupados os produtos dentro de famílias distintas. Kusiak e He (Kusiak and He 1994), apresentam regras de especificação dos produtos que deverão permitir a simplificação do escalonamento da produção.

Engenharia Assistida por Computador (CAE)

A *Engenharia Assistida por Computador* serve para auxiliar na fase de *concepção preliminar*. A principal função deste nível é a de analisar o desempenho do sistema de produção usando técnicas de simulação ou métodos analíticos. O desempenho do sistema dependerá, geralmente, do tipo de gestão de produção que será aplicada.

Fabricação Assistida por Computador (CAM)

A *Fabricação Assistida por Computador* tem uma implicação na fase de *produção*. Pertencem a este módulo as funções de gestão de produção, de comando em tempo real e de monitoramento. A elaboração dos programas para as máquinas-ferramenta com comando numérico faz também parte da fabricação assistida por computador.

2.2 Redes de Petri para Sistemas de Produção Híbridos

Devido à característica híbrida dos sistemas de produção por lotes e por possuírem semelhanças aos sistemas flexíveis de manufatura que possuem aspecto discreto e cujas técnicas de análises são bem conhecidas, nesta dissertação será dada uma ênfase ao aspecto discreto de tais sistemas. Os modelos usados para o escalonamento (objetivo deste trabalho), serão baseados em redes de Petri cujas vantagens serão mostradas ao longo desta secção. Inicialmente, será introduzido o modelo contínuo estendido e posteriormente o modelo discreto estendido.

2.2.1 Modelo Contínuo Estendido com Variáveis Booleanas

A modelagem de um sistema físico pode ser feita por equações diferenciais, onde o estado do sistema é representado pelas variáveis contínuas (números reais) em função do tempo (contínuo). No exemplo da figura 2.3, a equação diferencial que descreve a evolução do volume do material (V) no reator é:

$$\frac{dV}{dt} = q_i - q_o$$

Na equação, V é a variável de estado e o tempo é contínuo. O modelo do recipiente é dado pela seguinte equação:

$$V(t_{n+1}) - V(t_n) = [q_i(t_n) - q_s(t_n)](t_{n+1} - t_n)$$

onde o tempo é discreto, mas V é contínuo porque nos dá um conjunto de números reais. Esta equação permite obter em um dado instante, o volume do material no recipiente sendo que q_i e q_o são, respectivamente, os fluxos de entrada e saída do material no reator e t_n e t_{n+1} são, respectivamente, os instantes atual e posterior analisados.

Com o intuito de associar aspectos discretos que certamente existem em tais sistemas, criou-se um modelo contínuo estendido no qual foram introduzidas variáveis booleanas às equações contínuas.

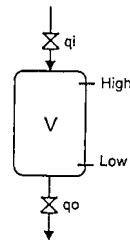


Figura 2.3: O reator

Ao representar o exemplo do reator citado na figura 2.3 em um modelo contínuo estendido com variáveis booleanas, obtém-se:

$$\frac{dV}{dt} = b_i \cdot q_i - b_o \cdot q_o$$

onde b_i e b_o são variáveis booleanas que são usadas para fornecerem a dinâmica do volume do reator durante toda a configuração. Durante o enchimento, $b_i = 1$ e $b_o = 0$ e a equação anterior torna-se:

$$\frac{dV}{dt} = q_i$$

Durante a configuração inicial, $b_i = 0$ e $b_o = 0$.

Durante o esvaziamento, $b_i = 0$ e $b_o = 1$ e a equação anterior é dada por:

$$\frac{dV}{dt} = -q_o$$

No modelo proposto, a parte contínua é representada por V , q_i e q_o , e a parte discreta pelas variáveis booleanas b_i e b_o .

Em casos onde o recipiente não é relevante, ou seja, interessa apenas uma única mudança de configuração (antes \rightarrow depois), este modelo é suficiente. Isto porque se forem considerados os períodos em que o recipiente está sendo enchido ou esvaziado, será preciso considerar volumes específicos (intermediários) que dependem do estado imediatamente anterior ao atual.

2.2.2 Modelo Discreto Estendido

Antes de abordar as redes de Petri, será feita uma breve introdução aos autômatos finitos (Hopcroft and Ullman 1979), que foi a ferramenta que deu origem às redes de Petri (W.Reisig 1985);(David and Alla 1992). O autômato finito possui uma extensão que possibilita a abordagem contínua no seu modelo discreto que é o autômato híbrido mostrado posteriormente.

A problemática do controle supervisorio, inicialmente, foi amplamente estudada com base nos autômatos finitos que permitem realizar a síntese de um sistema respeitando certas especificações. O autômatos finitos controláveis possibilitam obter sequências de estados desejáveis, desconsiderando os estados indesejáveis. Ramadge e Wonham (1989) criaram um controlador realimentado baseado em autômatos controlados cuja característica básica é de que após a cada evento ocorrido, o supervisor considerará apenas os eventos controláveis (figura 2.4).

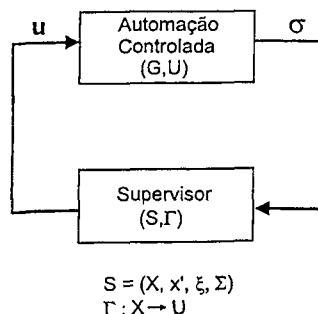


Figura 2.4: Realimentação de eventos para autômato controlado

Neste controlador, a supervisão das transições é ativada pelos eventos da planta (linguagem controlada). A linguagem gerada pelo sistema realimentado é $L(S/G)$. A linguagem dita, é a sequência de eventos controláveis u que foi gerada pelo supervisor mostrado na figura 2.4. O bloco “Automação Controlada” da figura 2.4 recebe os eventos controláveis do supervisor os associa aos novos eventos que porventura possam aparecer, fazendo em seguida uma realimentação com a nova sequência de eventos σ .

A seguir será introduzida uma noção dos autômatos finitos.

2.2.2.1 Autômatos Finitos

O autômato finito é um modelo matemático de um sistema, que contém entradas discretas e saídas discretas.



Figura 2.5: Autômato Finito

O sistema possui um número finito de estados internos, denominados: Estados (ver figura 2.5).

O estado do sistema possui informação suficiente para determinar o comportamento futuro do sistema. Uma aplicação, por exemplo, seria representar as discontinuidades em sistemas de processos que apresentam histereses, tais como, recipientes que possuem válvulas que abrem e fecham em diferentes pressões. Na figura 2.6, o sistema possui dois estados (válvula aberta e válvula fechada), dois arcos que ligam um estado a outro e duas situações que implicam na transição de um estado a outro ($Press. > Press_{permitida}$ e $Press. < Press_{permitida}$). Dá-se às condições que implicam nas mudanças de estados o nome de: Etiquetas.

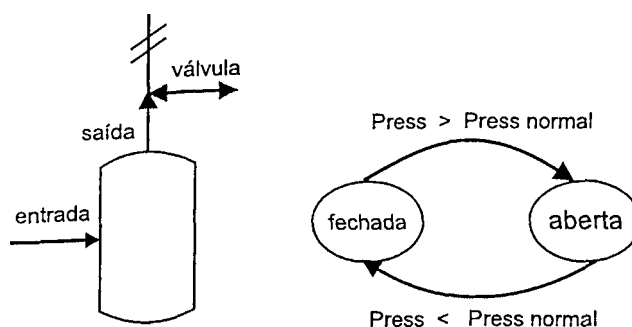


Figura 2.6: Autômato de um Reservatório Fechado

Os autômatos finitos são classificados por: determinísticos e não-determinísticos.

Autômatos Finitos Determinísticos

Um autômato finito determinístico (AFD) é uma quintupla $M = (Q, \Sigma, \delta, q_0, Q_m)$, onde:

Q = conjunto dos estados finitos

Σ = alfabeto que representa as etiquetas ($\Sigma = \{a, b\}$, na figura 2.7)

δ = função de transição $Q \times \Sigma \rightarrow Q$

q_0 = estado inicial

$Q_m \subset Q$ = conjunto dos estados marcados

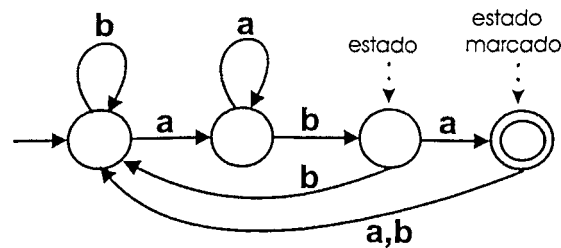


Figura 2.7: Autômato Finito Determinístico

A figura 2.7 mostra claramente que cada etiqueta explicita exatamente qual o estado será alcançado. O estado marcado indica o fim da evolução, ou seja, a partir dele ou o autômato é finalizado ou os estados ocorridos serão novamente executados. A etiqueta (a, b) indica que tanto a ocorrência de a quanto a de b levarão ao mesmo estado.

Autômatos Finitos não Determinísticos

Os autômatos finitos não determinísticos (AFnD) têm a mesma definição dos AFD, exceto no fato de que no primeiro o que se tem é uma **relação de transição** e não uma **função de transição**. Assim, definimos da seguinte forma: $M = (Q, \Sigma, \Delta, q_0, Q_m)$, onde:

Q, Σ, q_0, Q_m = definido igualmente para os AFDs

$\Delta(Q \times \Sigma^* \times Q)$ = relação de transição

e, Σ^* é a iteração do alfabeto ou número de ocorrência de cada evento

Na figura 2.8 podemos verificar por exemplo o não determinismo quando estamos no estado q_0 . A partir deste estado, ao ocorrer a etiqueta b , podemos atingir o estado q_1 ou

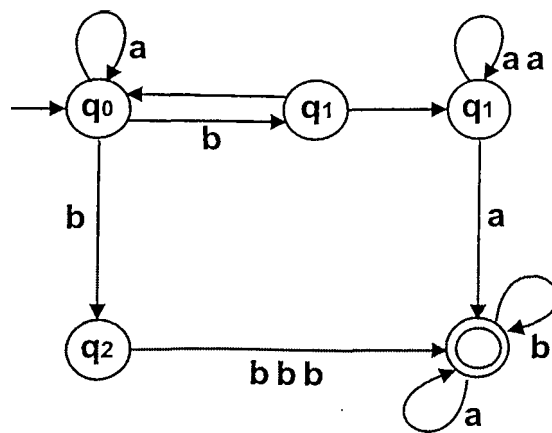


Figura 2.8: Autômato Finito não Determinístico

o estado q_2 . Neste caso não temos de maneira explícita qual o estado deve ser atingido.

Eliminação do não Determinismo

É possível eliminarmos o não determinismo dos autômatos. Tendo em mente que num autômato, a sequência das transições de estado é chamada de linguagem do autômato " $L(M)$ ". então, podemos dizer que um autômato (M_1) é equivalente a outro autômato (M_2), se, $L(M_1) = L(M_2)$.

Para cada AFnD, é possível construir um AFD equivalente desde que se elimine as duas possibilidades que os diferencia:

- 1) Eliminação da concatenação de etiquetas entre dois estados (figura 2.9):

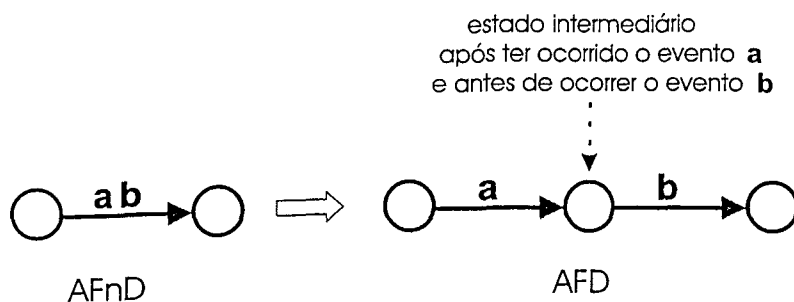


Figura 2.9: Eliminação de Concatenações

- 2) Eliminação de transições não exclusivas (figura 2.10) :

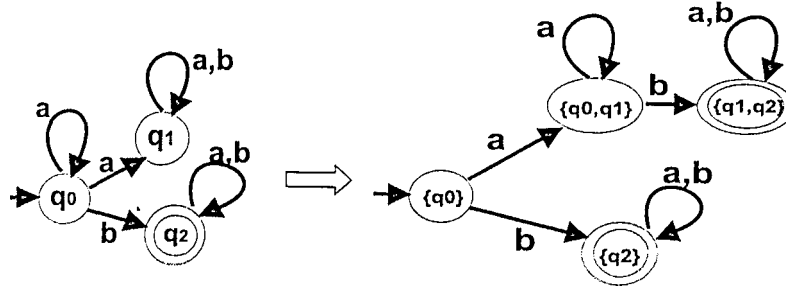


Figura 2.10: Eliminação das Transições não Exclusivas

Autômatos Híbridos

Um autômato híbrido (Seibel and Farines 1997) é constituído por um conjunto finito de variáveis reais X e por um multigrafo etiquetado (V, E) , sendo o conjunto das derivadas primeiras de X denotado por \dot{X} . Os arcos E representam ações discretas e são etiquetados com restrições sobre os valores de X antes e após a execução das ações correspondentes. Os vértices V representam atividades contínuas e são etiquetados com restrições sobre os valores de X e \dot{X} durante a realização das atividades associadas.

Formalmente, temos: $H = (X, V, E, syn, act, inv)$, onde:

- $X = (x_1, x_2, \dots, x_n)$, é um conjunto finito de variáveis, com valores reais, cujo tamanho n de X é a dimensão de H . Sendo S o conjunto das avaliações possíveis do autômato H , então, $s = (x_1 = a_1, x_2 = a_2, \dots, x_n = a_n)$ é um ponto no espaço n -dimensional real \mathcal{R}^n (estado das variáveis contínuas do autômato).
- V , é um conjunto finito de vértices (nós). Sendo $v \in V$ um nó do autômato H , então, (v, s) é o estado de H com $s \in \mathcal{R}^n$. O conjunto de estados é chamado de região. As avaliações s do nó v em uma região W são tais que $(v, s) \in W$.
- E , é um conjunto finito de transições. Cada transição é constituída de $e = (v, a, \mu, v')$, onde $a \in syn$ é um rótulo de sincronização e $\mu \subseteq S^2$ é uma relação de transição. A transição e é dita habilitada em um estado (v, s) se para alguma avaliação $s' \in S$, $(s, s') \in \mu$. O estado (v', s') é dito um sucessor transitivo do estado (v, s) .
- syn , é um conjunto finito de etiquetas de sincronização usadas para definir a composição

paralela de dois autômatos: se dois autômatos partilham a etiqueta α , então, cada α -transição de um dos autômatos é acompanhada de uma α -transição do outro autômato.

- *act.*, é uma função que atribui a cada nó $v \in V$ um conjunto de atividades. Cada atividade é uma função do conjunto de reais não negativos $\mathcal{R}^{\geq 0}$ para S . O modelo exige que as atividades de cada nó sejam invariantes no tempo: para todos os nós $v \in V$.
- *inv.* é uma função que atribui a cada nó $v \in V$ um invariante $inv(v) \subseteq S$. O autômato pode permanecer no estado v apenas enquanto o invariante do nó for verdadeiro.

A figura 2.11 mostra um exemplo de autômato híbrido usado para representar o funcionamento de um termostato (Alur *et al.* 1995). A temperatura de uma sala é controlada pelo termostato, que mede continuamente a temperatura e devolve um sinal **on** e **off** a um aquecedor. A variação da temperatura é representada por equações diferenciais. Quando o sinal é **off**, a temperatura (x) diminui exponencialmente ($x(t) = \theta e^{-kt}$), onde t é o tempo. θ é a temperatura inicial e k é uma constante específica da sala. Quando o sinal é **on**, a temperatura segue a função ($x(t) = \theta e^{-kt} + h(1 - e^{-kt})$), onde h é uma constante que depende da potência do aquecedor. Manter a temperatura entre m e M graus, o aquecedor deve estar em **on** e **off** de acordo com a situação.

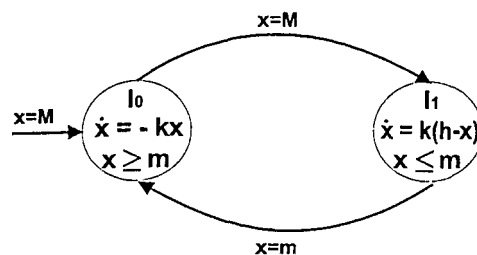


Figura 2.11: Autômato de um Termostato

A verificação de problemas para sistemas híbridos é intrinsecamente difícil, o que nos leva a usar sistemas híbridos lineares para análises algorítmicas de sistemas híbridos. Esta adaptação se adequa muito bem na prática. Em um sistema híbrido linear, para cada variável, a taxa de mudança é constante, apesar desta constante poder variar de um lugar a outro e os termos envolvidos nos invariantes serem requeridos na forma linear.

Na sequência X de um autômato H , um termo linear é uma combinação linear das variáveis em X com coeficientes inteiros. Uma fórmula linear é uma combinação booleana de inequações entre os termos lineares.

A figura 2.12 mostra um exemplo de sistema linear híbrido, que é um monitor de nível de água. O princípio de funcionamento é o seguinte: o nível de água num tanque é controlado por um monitor, o qual mede continuamente o nível de água e devolve um sinal **on** e **off** a uma bomba d'água. O nível da água muda segundo uma função linear temporal específica. Quando a bomba está em **off**, o nível da água (y) cai a 2 unidades de comprimento por segundo; quando a bomba está em **on**, o nível da água sobe a 1 unidade de comprimento por segundo. Supondo que inicialmente o nível da água está em 1 unidade de comprimento e a bomba em **on**. Nós manteremos o nível da água entre 1 e 12 polegadas. O monitor tem um atraso de 2 segundos para mudar efetivamente o estado da bomba. Assim, o monitor deve mudar o sinal da bomba em **on** antes do nível da água cair para 1 unidade de comprimento e deve mudar em **off** antes do nível da água subir para 12 unidades de comprimento.

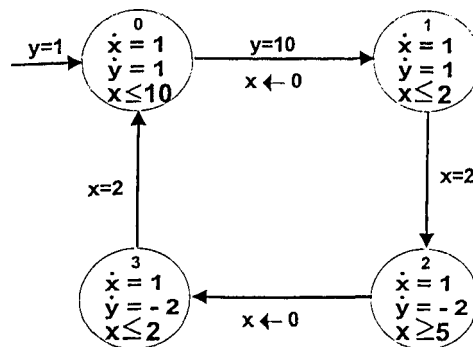


Figura 2.12: Autômato de um Tanque de Água com Bomba

Em geral os autômatos híbridos se adequam bem aos sistemas híbridos, mas possuem suas limitações. Uma delas é o fato de não ser uma representação agregada, isto é, nos sistemas reais que normalmente possuem um número grande de estados, há uma grande possibilidade de ocorrer uma explosão combinatória na representação dos autômatos (número de possíveis estados). Outra limitação é o fato dos autômatos não permitirem a repre-

sentação do paralelismo de um sistema, que é inerente aos sistemas de produção híbridos. Para tentar resolver estes problemas, criou-se uma forma de representação agregada em relação aos estados e que permite a representação do paralelismo de um sistema, que são as redes de Petri.

2.2.2.2 Redes de Petri (RdPs)

As redes de Petri (RdP) têm por origem a tese de *Carl Adam Petri* intitulada *Comunicação com Autômatos*, apresentada em 1962 à Universidade de Darmstadt. É a partir deste trabalho que *Anatol W. Holt*, com a ajuda de um grupo de pesquisadores do MIT (Massachusetts Institute of Technology), desenvolveu as bases da teoria das RdPs entre 1968 e 1976. As RdPs são hoje consideradas como uma ferramenta de representação formal que permite a modelagem, a análise e o controle de sistemas a eventos discretos que comportam atividades paralelas, concorrentes e assíncronas. Atualmente no Brasil, existem algumas publicações envolvendo RdPs, dentre elas está o livro de *Paulo Eigi Miyagi* (Miyagi 1999).

As RdPs possuem basicamente os seguintes elementos:

- a) Lugares: são representados por círculos
- b) Transições: são representadas por barras ou retângulos
- c) Fichas ou Marcas: são representadas por pontos nos lugares
- d) Arcos: ligam transições a lugares e vice-versa, possuindo uma orientação explícita

Um lugar nunca é ligado a outro lugar e nenhuma transição é ligada a outra transição.

Veja o exemplo da figura 2.13.

Quando se considera uma RdP marcada (ver figura 2.14), cada lugar tem um número inteiro positivo ou nulo de marcas. A marcação da RdP (o número de marcas em cada lugar da rede) em um instante particular define o estado da RdP.

A evolução do estado da rede corresponde à evolução da marcação. A evolução da marcação é a consequência direta do disparo das transições. Para disparar uma transição, retira-se uma marca de cada um dos lugares de entrada da transição e acrescenta-se uma

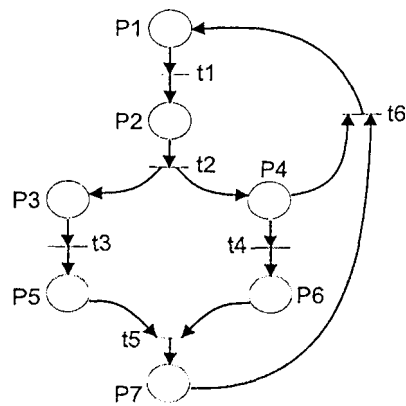


Figura 2.13: Rede de Petri

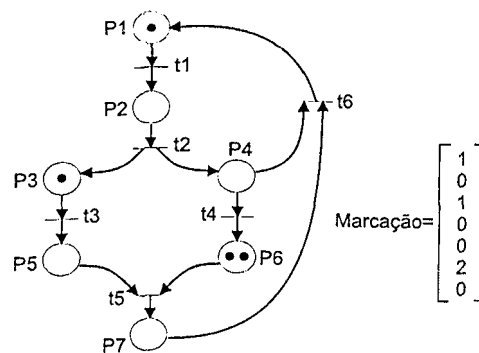


Figura 2.14: Rede de Petri Marcada

marca em cada um dos lugares de saída da transição. Veja a ilustração da figura 2.15 para disparos de transições.

É importante notar que o disparo de uma transição é instantâneo (“*tem uma duração nula*”) e que, no nível da RdP global, só pode acontecer um disparo de cada vez.

Transição Habilitada ou Sensibilizada: Diz-se que uma transição está habilitada ou sensibilizada para um disparo, quando o número de marcas de cada lugar de entrada for maior ou igual ao peso (número associado ao arco, indicando quantas marcas serão necessárias ao disparo) de seus respectivos arcos de saída.

Formalmente, uma RdP é uma quádrupla $R = \langle P, T, Pre, Post \rangle$, onde:

- P é um conjunto finito de lugares de dimensão n ;
- T é um conjunto finito de transições de dimensão m ;

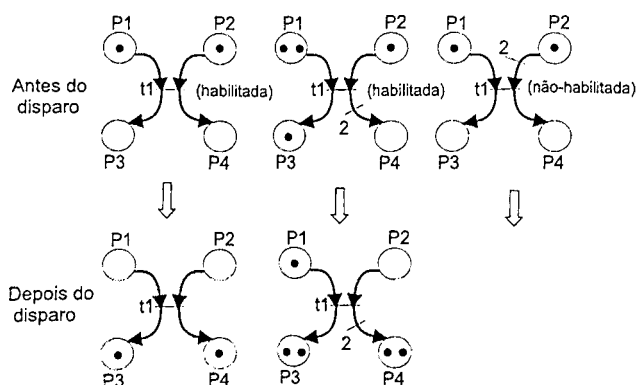


Figura 2.15: Exemplos de Disparos de Transições numa RdP

- $Pre : PXT \rightarrow \mathcal{N}$ é a aplicação de *entrada* (lugares precedentes ou incidência anterior à transição analisada), com \mathcal{N} sendo o conjunto dos números naturais;
- $Post : PXT \rightarrow \mathcal{N}$ é a aplicação de *saída* (lugares procedentes ou incidência posterior à transição analisada).

Formalmente, uma RdP marcada N é uma dupla $N = \langle R, M \rangle$, onde:

- R é uma rede de Petri;
- M é a marcação inicial dada pela aplicação $M : P \rightarrow \mathcal{N}$.

$M(p)$ é o número de marcas contidas no lugar p . A marcação M é a distribuição das fichas nos lugares, sendo representada por um vetor coluna cuja dimensão representa o número de lugares e elementos $M(p)$. Para o exemplo da figura 2.14, a notação vetorial da marcação fica: $M^T = [1010020]$ (M^T é o transposto do vetor M).

A notação matricial da RdP do exemplo da figura 2.14 é dada a seguir. Vale ressaltar que nas matrizes Pre , $Post$ e C , as **linhas representam os lugares** da rede e as **colunas representam as transições** da rede.

$$Pre = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$Post = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

A partir de Pre e $Post$ define-se a matriz de incidência C : $C = Post - Pre$. Neste exemplo, tem-se:

$$C = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 1 \\ 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & -1 \\ 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix}$$

Na primeira coluna da matriz C , por exemplo, após o disparo da transição t_1 , uma ficha será retirada do lugar P_1 (valor -1) e uma ficha será colocada no lugar P_2 (valor 1).

Utiliza-se notação $Pre(., t)$ para a coluna da matriz Pre associada a uma transição t . A dimensão deste vetor é dada pelo número de lugares. Da mesma forma, define-se os vetores coluna $Post(., t)$ e $C(., t)$ em relação às matrizes $Post$ e C , respectivamente.

Para representarmos o número de ocorrências das transições numa sequência de disparo s , usamos o *vetor característico* s . No exemplo anterior, podemos ter a seguinte sequência:

$$s^t = \begin{bmatrix} 1 & 1 & 2 & 0 & 2 & 1 \end{bmatrix}$$

Vale ressaltar que o vetor característico não é único, pois, representa uma sequência de disparo particular. Além disso, o vetor característico não representa a *ordem de disparo de cada transição*, mas sim o *número de ocorrência de cada transição*.

No vetor característico mostrado anteriormente, a sequência mostrada indica que a transição t_1 será disparada uma vez, a transição t_2 será disparada uma vez, a transição t_3 será disparada duas vezes, a transição t_4 não será disparada, a transição t_5 será disparada duas vezes e a transição t_6 será disparada uma vez. Vale ressaltar que os disparos não serão necessariamente nesta ordem.

A evolução da marcação de uma RdP, com marcação inicial M , para uma sequência s , é representada pela *equação fundamental*: $M' = M + Cs^t$ com $M \geq 0$, $s^t \geq 0$ e s^t o vetor transposto de s .

A semântica associada a cada elemento de uma RdP pode ser completamente diferente de uma rede para outra. Um lugar pode ser interpretado como uma condição, um procedimento, um recurso, um estoque, etc. Uma transição pode ser interpretada como um indicador. Por exemplo, as redes da figura 2.16 representam um mesmo sistema com duas semânticas diferentes associadas a cada rede.

Com uma RdP será possível modelarmos diversas características encontradas nos sistemas de produção híbridos, veja os exemplos da figura 2.17.

No caso dos sistemas de produção híbridos, a existência de recursos divididos (reatores) faz com que apareçam conflitos cujas definições são as seguintes:

Conflito Estrutural: n transições, com $n \geq 2$, estão em conflito estrutural, se e

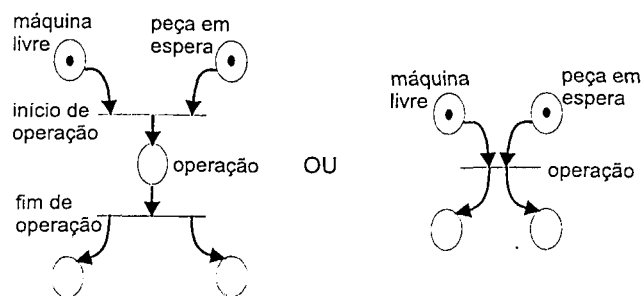


Figura 2.16: Modelos Diferentes para um mesmo Sistema

somente se, ambas têm ao menos um lugar de entrada em comum (ver figura 2.18).

Conflito Efetivo: n transições, com $n \geq 2$, estão em conflito efetivo, se e somente se, ambas estão em conflito estrutural e estão sensibilizadas (ver figura 2.19).

Uma apresentação detalhada das RdPs encontra-se em (Petri 1966);(Brams 1983); (David and Alla 1992);(Murata 1989);(Peterson 1981);(Reutenauer 1990);(W.Reisig 1985); (Cardoso and Valette 1997).

2.2.2.3 RdPs e Sistemas de Produção Híbridos

Nesta dissertação, estudaremos os sistemas de produção por lotes (Andreu *et al.* 1996), que possuem um comportamento discreto (estado dos lotes, válvulas, etc) e contínuo (evolução temporal do sistema, como tempo de transferência de um lote, tempo de permanência do lote no reator, etc). Normalmente, o aspecto contínuo destes sistemas é descrito por equações álgebra-diferenciais, e o aspecto discreto por máquinas de estado finito, grafcet ou redes de Petri.

Vários trabalhos foram feitos por diferentes pesquisadores para tratar os sistemas híbridos representados por RdPs. Veja alguns deles:

Em (Champagnat *et al.* 1996) apresentou-se alguns modelos híbridos para representar os sistemas de produção híbridos. Um primeiro modelo sugerido foi o modelo contínuo adicionado a variáveis lógicas, são os chamados “Modelos Contínuos Extendidos com

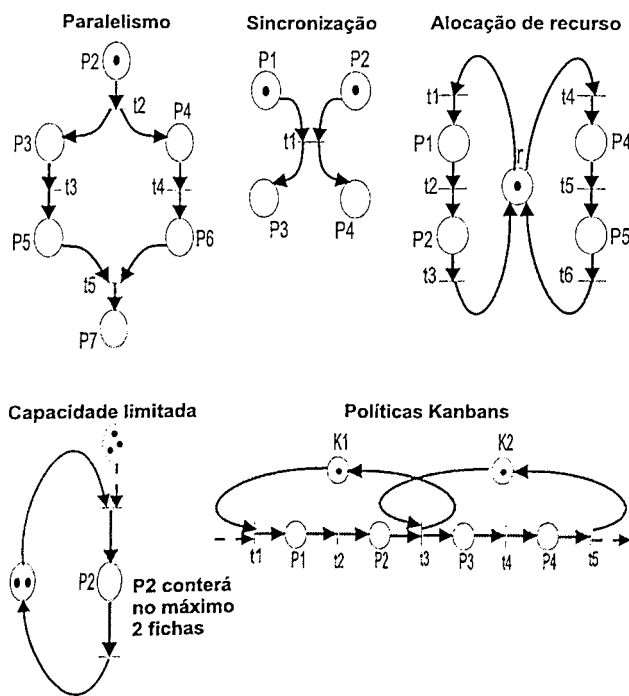


Figura 2.17: Tipos de Modelagens numa RdP

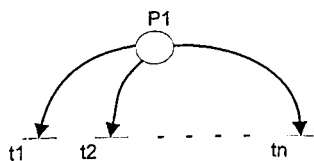


Figura 2.18: Conflito Estrutural

Variáveis Booleanas”, discutidos na sub-seção 2.2.1 deste capítulo. A partir das equações contínuas que regem o sistema analisado, pode-se obter uma determinada representação do modelo discreto (rede de Petri autônoma).

A figura 2.20 mostra um modelo de rede de Petri autônoma que representa o reator da figura 2.3 da seção 2.2.1 no estado vazio ($b_i = b_o = 0$), sendo enchido com 1000 l do produto A ($b_i = 1; b_o = 0$), sendo enchido com 500 l do produto B ($b_i = 1; b_o = 0$) e cheio ($b_i = b_o = 0$), sendo que a equação do volume do material é dada por:

$$\frac{dV}{dt} = b_i \cdot q_i - b_o \cdot q_o$$

Uma limitação deste modelo é que ele não permite a especificação de estados interme-

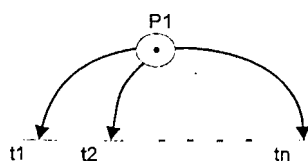


Figura 2.19: Conflito Efetivo

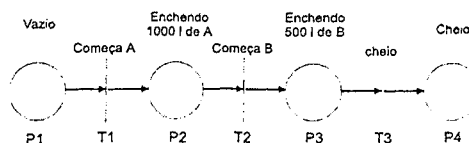


Figura 2.20: Um fragmento de receita

diários, ou seja, ele permite que se conheça apenas antes e após uma única mudança de configuração.

Um segundo modelo sugerido foi o modelo de rede de Petri temporizada. Neste modelo as durações das operações são valores contínuos e são associados às transições ou aos lugares (Genrich *et al.* 1994);(Hanish 1993). Usando o mesmo sistema mencionado no modelo anterior (reator da figura 2.3 da seção 2.2.1) podemos representá-lo através de uma rede de Petri t-temporizada como mostra a figura 2.21.

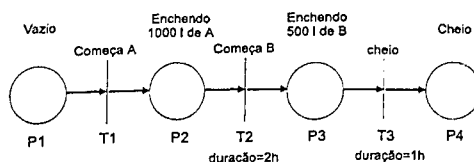


Figura 2.21: Modelo de rede de Petri temporizada

Pela figura 2.21 podemos observar que as durações das operações de enchimento do reator são durações associadas às transições T_2 para o enchimento com o produto “A” e T_3 para o enchimento com o produto “B”. Neste modelo de rede de Petri t-temporizada a parte discreta é representada pelos lugares e transições e a única variável contínua é o tempo, permitindo obter-se assim por meio de equações diferenciais, todas as variáveis de estado contínuas. No sistema “reator” mostrado anteriormente se considerarmos o fluxo de entrada q_i como sendo constante, por meio de interpolação linear é possível

calcularmos V em qualquer instante. Por exemplo durante a configuração associada ao lugar P_2 , tem-se (evento “começa A” sendo a origem do tempo):

$$V(t) = \frac{1000}{2} \cdot t$$

Uma limitação deste modelo é que é necessário uma boa avaliação das durações de todas as configurações, e estas durações têm de ser independentes de evoluções passadas do sistema.

Um terceiro modelo sugerido foi o modelo de rede de Petri híbrido que possui uma representação unificada das variáveis contínuas com lugares contínuos (as fichas são representadas por números reais) e das variáveis discretas com lugares discretos (as fichas são representadas por números inteiros) (Alla 1995); (Bail *et al.* 1991a); (David and Alla 1987).

Usando o mesmo sistema mencionado no modelo anterior (reator da figura 2.3) podemos representá-lo por uma rede de Petri híbrida como mostra a figura 2.22.

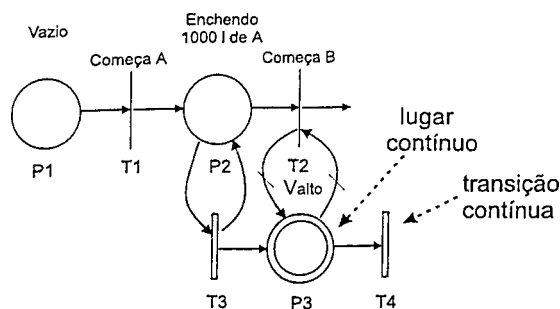


Figura 2.22: Modelo de rede de Petri híbrida

Na figura 2.22, a velocidade de disparo da transição T_3 representa o fluxo de entrada q_i , e a velocidade de disparo da transição T_4 representa o fluxo de saída q_o . A quantidade contínua da(s) ficha(s) no lugar P_3 representa o volume V do material no reator. Os estados do reator são representados pelos lugares discretos e pelas transições da rede: entre o disparo de T_1 e T_2 , o reator está sendo enchido. A interação da parte discreta na parte contínua é denotada pelos arcos que ligam P_2 e T_3 . Quando há pelo menos uma fração de ficha no lugar P_2 , a velocidade de disparo de T_3 é q_i , caso contrário T_3 não é habilitada e não pode ser disparada. A interação da parte contínua na parte discreta é denotada pelos

arcos que ligam P_3 e T_2 . Quando a fração de ficha em P_3 é $V = V_{alto} = 1000l$, então T_2 é disparado, e a fração de ficha em P_2 é removida.

Neste modelo somente variáveis contínuas do tipo linear é que podem ser representadas (velocidade constante). Vale ressaltar que este modelo tem sido estendido para representar outros tipos de evoluções, mas para cada tipo de evolução um novo tipo de lugar contínuo tem de ser introduzido.

Um quarto modelo sugerido foi o modelo de rede de Petri autônoma associado com equações diferenciais. Nesta técnica a rede de Petri trata a parte discreta e as equações diferenciais tratam a parte contínua do sistema (Andreu *et al.* 1995b); (Andreu *et al.* 1995a);(Daubas *et al.* 1994);(Pagès and Pingaud 1995);(Valette 1995). Para o exemplo do reator, a rede de Petri é igual a rede de Petri utilizada no modelo contínuo estendido com variáveis booleanas com a diferença de que a parte contínua é tratada por um sistema álgebra-diferencial sem nenhuma restrição, como mostrado na figura 2.23.

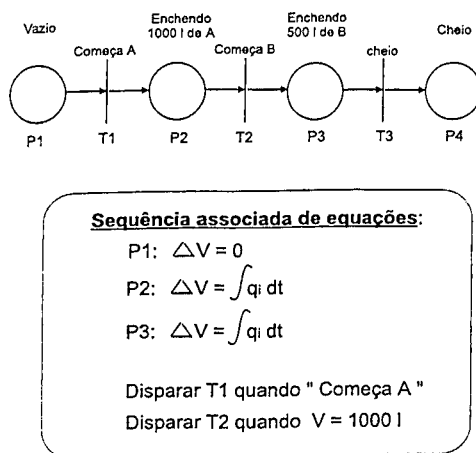


Figura 2.23: Modelo de rede de Petri e equações diferenciais

A interação entre a parte discreta e contínua é feita da seguinte forma: uma ficha é colocada num lugar da rede que representa o início de integração das equações correspondentes. Concorrentemente com a integração, um certo número de sinais são monitorados. Cada sinal é associado com uma transição a qual é uma transição de saída do lugar marcado. Quando o sinal é detectado, significa que o evento correspondente está ocorrendo e

a transição associada é disparada. A nova marcação é obtida e começa a integração de um novo sistema. No exemplo do reator da figura 2.3 os sinais associados aos lugares P_2 e P_3 da figura 2.23 são diferenciados, pois, para P_2 com o sinal $V = 1000l$, a transição T_2 é disparada e para P_3 com o sinal $V = 500l$, a transição T_3 é disparada. Podemos dizer que a rede de Petri (parte discreta do modelo) supervisiona o sistema de equações álgebra-diferenciais cuja estrutura muda a cada vez que uma transição é disparada e podemos dizer também que o sistema de equações (parte contínua do modelo) controla a evolução da rede de Petri provocando o disparo de suas transições.

Neste modelo não há restrição sobre o tipo de evoluções contínuas que podem ser representadas.

Em (Andreu *et al.* 1996) propôs-se um modelo híbrido baseado na cooperação de uma parte de eventos discretos (redes de Petri autônoma) e uma parte contínua (equações álgebra-diferenciais), tratando concorrentemente com modelos contínuos e discretos. No modelo de eventos discretos, ignorando longas evoluções contínuas do sistema, considera-se somente as descontinuidades (especificações contínuas descritas pelos predicados). A satisfação destas condições na evolução do processo (por exemplo, uma certa temperatura é atingida), detectada por um gerador de eventos, são eventos discretos que causam uma mudança do controle do estado discreto do sistema. Desta forma, assim que estas condições são satisfeitas, o controle do sistema muda imediatamente de modo de controle e automaticamente obtém-se a seleção da sequência ativa de equações. Esta seleção é obtida por meio de um modelo de referência: um modelo híbrido do processo mostrado na figura 2.25. Este modelo de referência permite implementar um comportamento do processo monitorando uma função baseada num mecanismo de comparação. Este mecanismo, por sua vez, permite monitorar o comportamento do processo detectando possíveis falhas, diagnosticando as falhas detectadas e recuperando o processo. Na figura 2.24.a temos um sistema de tanque de armazenagem que possui basicamente duas atividades 2.24.b : a operação de enchimento (expressada por LO) e a operação de esvaziamento (expressada

por DO). As condições de fim de operações depende do volume a ser transferido. A figura 2.25 representa o modelo de referência do sistema da figura 2.24.a. O começo da operação de armazenagem corresponde ao disparo da transição t_a na rede de controle mostrada na figura 2.24.b e o fim da operação de enchimento corresponde ao disparo da transição t_b (similarmente para a operação de esvaziamento). O disparo de alguma destas transições implica no disparo de uma transição correspondente no modelo de referência da figura 2.25.

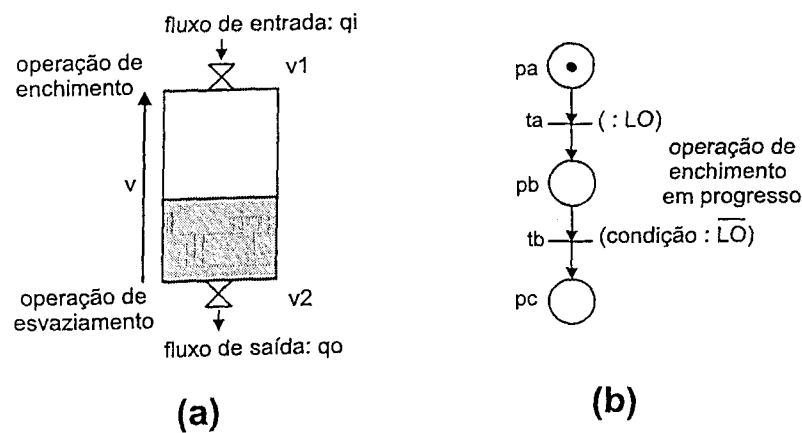


Figura 2.24: (a) Sistema (b) Rede de Controle

Em (Barros and Perkusich 1998) mostrou-se um procedimento de modelagem baseado nos possíveis estados alcançados pelos recursos de um sistema de produção híbrido. Neste trabalho utilizou-se redes de Petri autônomas para modelar tais sistemas e através de seus invariantes de lugar, determinou-se o chaveamento entre os diversos campos vetores das equações diferenciais que descrevem o comportamento dinâmico do sistema de produção híbrido. A figura 2.26.a mostra o exemplo de um sistema de produção híbrido utilizado no trabalho.

Na figura 2.26.b temos a modelagem do tanque T_a da figura 2.26.a. O tanque T_b , as válvulas e as vias de acesso mostrados na figura 2.26.a, também foram modelados por redes de Petri autônomas. Ao fazer todas as modelagens necessárias, obteve-se o modelo

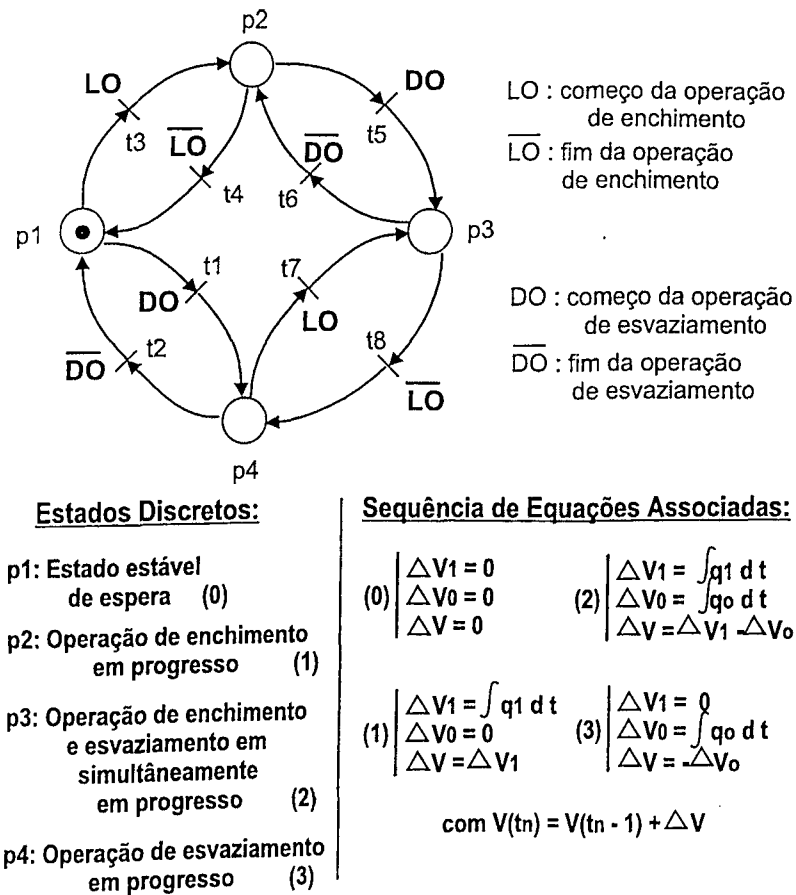


Figura 2.25: Modelo de Referência

da planta do sistema através da fusão de transições comuns aos modelos obtidos até o momento. Posteriormente, obteve-se o modelo de controle através das especificações dadas no sistema e finalmente, através da fusão das transições comuns do modelo da planta com o modelo de controle. O modelo do controle é uma rede de Petri autônoma, onde algumas de suas transições são sensibilizadas a serem disparadas dependendo da ocorrência ou não de certos eventos. A detecção destes eventos é representada por equações álgebra-diferenciais. A inovação do trabalho está no fato de que, uma vez que os invariantes de lugar são preservados após as fusões dos modelos, estes podem ser utilizados para determinar o chaveamento dos campos vetores das equações diferenciais que descrevem o comportamento dinâmico de cada recurso no sistema. Por exemplo, o invariante de lugar relacionado ao modelo do tanque T_a é $Y_a = [1, 1, 1, 1]^T$ (vetor transposto de Y_a). Portanto,

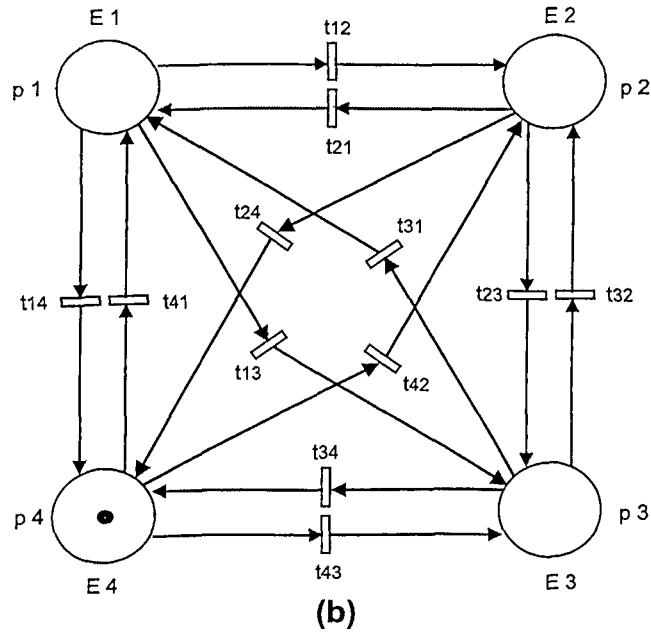
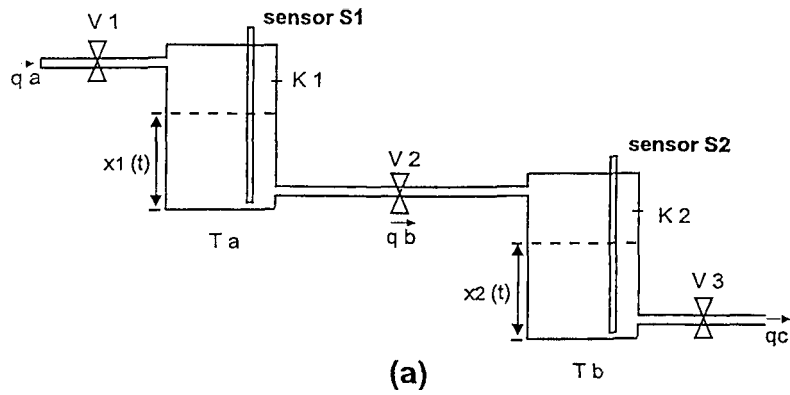


Figura 2.26: Exemplo de um sistema híbrido

temos:

$$M(p1) + M(p2) + M(p3) + M(p4) = 1$$

Aplicando o chaveamento entre os campos vetores (ξ) da equação obtida anteriormente, chegamos na equação que representa a parte da dinâmica contínua da planta do sistema:

$$x_1(t) = M(p1)\xi_1(t) + M(p2)\xi_2(t) + M(p3)\xi_3(t) + M(p4)\xi_4(t)$$

Em (Champagnat *et al.* 1998a) apresentou-se dois projetos de pesquisa sobre sistemas híbridos: o projeto ANVAR e o projeto MENESR. O projeto ANVAR tem como objetivo prático especificar e implementar uma ferramenta de software para simulação e, consequentemente, avaliação do desempenho de um sistema híbrido. O modelo deve ser num nível detalhado para avaliar uma trajetória específica de possíveis estados ou num nível global para avaliar a sua performance (simulação Monte-Carlo). Um ponto analisado pelo projeto é que se a simulação de sistemas contínuos puros e de sistemas discretos puros é concorrentemente destinada por um grande número de ferramentas, a simulação de sistemas combinados destinados a processos contínuos são fracos para apresentar aspectos discretos e vice-versa. Neste projeto foi apresentado um exemplo de unidade de estocagem de gás derivado. Em síntese, procedeu-se da seguinte forma: foram descritas as receitas em geral, configurou-se todas as unidades de produção; modelou-se através de equações álgebra-diferenciais a estocagem do gás, o compressor de entrada, a válvula de dissipação da entrada, o compressor de saída e a válvula de dissipação da saída; citou-se as restrições do sistema; e no final especificou-se a política de produção e a demanda de consumo. O projeto MENESR é puramente acadêmico e trata da modelagem para eventos discretos de processos industriais, que são processos nos quais incluem controladores contínuos mas são dirigidos por eventos discretos numa série de configurações para garantir, por exemplo, a execução de uma receita. Tais sistemas são encontrados na indústria alimentícia ou numa planta da indústria química. Os pesquisadores defendem a idéia de que a modelagem da parte discreta pode ser feito por RdPs ou grafets e que a parte contínua pode ser adicionada ou integrada em tais modelos. Quando a parte contínua é simples, ela pode ser considerada por meio de restrições temporais. Neste caso, utiliza-se RdPs temporais ou temporizadas (Sifakis 1977);(Merlin 1979);(Hanish 1993). Quando isto não é possível, há duas possibilidades:

- A dinâmica da parte contínua é otimizada com o mesmo formalismo da parte discreta (RdPs autônomas) e certos lugares específicos são usados como contínuos para tratar das variáveis contínuas. Neste caso, utiliza-se fichas com valores reais ao invés de

inteiros (David and Alla 1987);(Bail *et al.* 1991b);(Flaus 1996).

- A dinâmica da parte contínua é expressada por equações álgebro-diferenciais cuja interação com a RdP autônoma ou grafcet tem de ser claramente especificada (Andreu *et al.* 1995b);(Daubas *et al.* 1994);(Valentin and Ladet 1993);(Valentin 1994).

O trabalho cita ainda que em (Alla and Flaus 1998) o exemplo da estocagem de gás é dado por meio de redes de fluxo híbrido. (Champagnat *et al.* 1998c) baseia-se na integração de uma RdP Predicado/transição com sequência de equações álgebro-diferenciais e em (Valentin-Roubinet 1998) apresenta-se RdPs e equações álgebro-diferenciais.

(Moncelet *et al.* 1998) preocuparam-se com a modelagem e simulação de sistemas híbridos conhecidos como sistemas automotivos de mecatrônica. São sistemas que possuem tecnologia elétrica, mecânica, hidráulica e eletrônica e usa um controle por computador (Guyot 1994). Há dois tipos de ação: discreta e contínua. O artigo trata apenas o processo de controle discreto, onde detecta-se certos eventos indesejáveis e obtém-se um novo estado discreto do sistema e uma nova configuração é analisada posteriormente. Ao projetar um sistema mecatrônico, os projetistas tratam da avaliação da dependabilidade qualitativa e quantitativa (Hénault 1996) que é feita pelo método da árvore de falhas. A árvore de falhas dá a condição booleana de ocorrência de um evento indesejável. Sua deficiência está no fato de ser estática (não prevê reconfigurações) e gerar explosão combinatorial de possíveis estados. O artigo dá um exemplo de sistema mecatrônico cuja proposta é manter o nível de pressão (P) do tanque no intervalo $[P_{min}, P_{max}]$. Inicialmente, um modelo funcional e disfuncional é obtido pelas equações álgebro-diferenciais que descrevem a dinâmica contínua do sistema. Para descrever, em cada tempo de amostragem, as interações entre a parte operativa e o controle, um modelo de RdP colorida é obtido (modelo de especificação). Se a partir dele for possível definir intervalos que limitem os possíveis valores permitidos pelas variáveis envolvidas (Pressão, no caso do exemplo), então, podemos obter um modelo abstrato mais simples. Ao avaliar a dependabilidade do

sistema, é necessário uma modelagem quantitativa e qualitativa. Na modelagem quantitativa, é possível fazer um modelo abstrato que considere a ocorrência das falhas e os possíveis reparos de tais ocorrências. A simulação é feita através do simulador Monte-Carlo e a partir dele, obtém-se uma sequência de parâmetros (período de amostragem, taxa de falhas, tempos de espera, etc.) que definem as configurações do sistema. Na modelagem qualitativa, é possível fazer um modelo abstrato que considere somente a sequência de eventos e não mais o tempo de forma explícita. As propriedades dinâmicas de vivacidade e limitabilidade do modelo são verificadas pelo grafo de ocorrência (um arco representa um disparo de transição ou mudança de evento, um lugar representa uma marcação da RdP ou estado do sistema). Este grafo mostra todos os possíveis cenários, particularmente os cenários raros os quais não podem ser facilmente mostrados pela simulação Monte-Carlo.

Na tese de Gilles Moncelet, abordou-se um modelo de RdP colorida para tratar da modelagem dos sistemas de produção híbridos. Normalmente, em sistemas de produção por lotes onde há um número considerável de receitas, obtemos uma estrutura de tamanho considerável se utilizarmos uma RdP (lugar/transição). Se utilizarmos uma RdP colorida (Caradec and Prunet 1997), obtemos um modelo mais compacto. Considere a figura 2.27, onde:

$E1, E2$, entradas de lotes nas receitas 1 e 2, respectivamente

$S1, S2$, saídas de lotes nas receitas 1 e 2, respectivamente

$Op1a$, tratamento sobre a receita 1 através do reator a

$Op2a$, tratamento sobre a receita 2 através do reator a

$Op1b$, tratamento sobre a receita 1 através do reator b

a , reator a

b , reator b

Observe que as receitas 1 e 2 possuem tratamentos diferentes e que se soubermos o tipo de lote, então, saberemos qual a sequência de tratamentos será feita. Assim, podemos

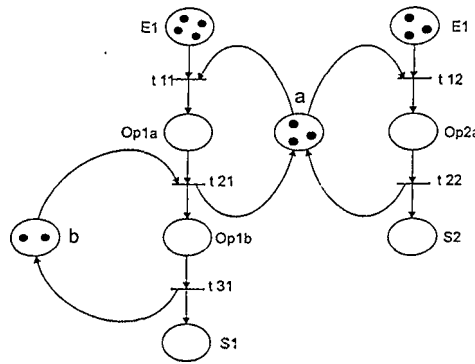


Figura 2.27: Exemplo de RdP de um Sistema de Produção por Lotes

utilizar uma RdP colorida correspondente que diferenciará os lotes e os reatores através de cores (etiquetas).

Considere as seguintes cores:

$l1$, lote tratado segundo a receita 1

$l2$, lote tratado segundo a receita 2

ra , reator a

rb , reator b

A representação é dada pela figura 2.28. Observe que o disparo das transições depende do tipo de cor a ser usada, ou seja, tipo de lote *e/ou* reator usado. No final, o que se tem é um modelo mais agregado.

Para que o tempo seja considerado, é possível associar às RdP coloridas, as técnicas usadas nas RdPs temporizadas.

Em (Caradec and Prunet 1997) utilizou-se um modelo estendido das RdPs coloridas, associando-as às RdPs por Lotes. Além de utilizar as vantagens das RdPs coloridas, utilizou-se a característica das RdPs por lotes que é a de considerar os fenômenos de acumulações e atrasos frequentes nos transportadores de sistemas por lotes. Este trabalho objetivou sintetizar a modelagem do sistema híbrido bem como prever os fenômenos citados acima, para a *simulação* dos mesmos.

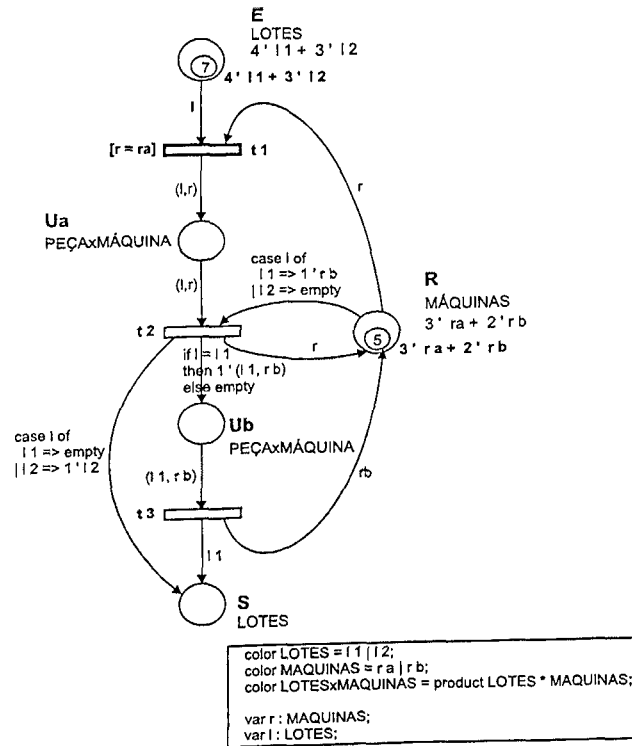


Figura 2.28: Exemplo de RdP Colorida com a Declaração das Cores

Em (M.J.Sanders 1998) mostrou-se uma RdP de alto nível (RdP Orientada a Objetos) para modelar sistemas de produção complexos, representando problemas baseados em restrições e usando uma sequência de restrições primitivas. A dependência causal da representação de RdPs, sua concorrência, habilidade para modelar o controle de fluxo e sua notação gráfica, fazem delas um recurso promissor para programação de restrições. As RdPs orientadas a objetos somam os aspectos de dinamismo e modularidade, e potencialmente a habilidade de definir ao usuário restrições sobre os estados inteiros de sub-redes representando objetos. O trabalho mostrou uma correspondência entre problemas de satisfação de restrições convencionais e a representação de RdPs orientadas a objetos, e discutiu como algoritmos de busca avançados devem ser aplicados para o problema de avaliação de ligações das transições. No final do trabalho, foi apresentada uma ferramenta baseada na técnica de objetos comunicativos (OCELOT). A implementação de OCELOT consiste em gerar uma linguagem textual para a definição da rede e,

posteriormente, desenvolver um software para produção eficiente de modelos executáveis. Uma representação do modelo da rede é produzida por um compilador $C++$. O modelo é compilado e linkado com um tempo de execução que contém, além de outras coisas, a resolução das restrições. A figura 2.29 mostra um exemplo de RdP orientada a objetos, onde TEMPERATURA é a propriedade interna e COR é a propriedade externa. A transição RESOLVE seleciona um ou mais objetos cujas propriedades privadas apresentam critérios específicos.

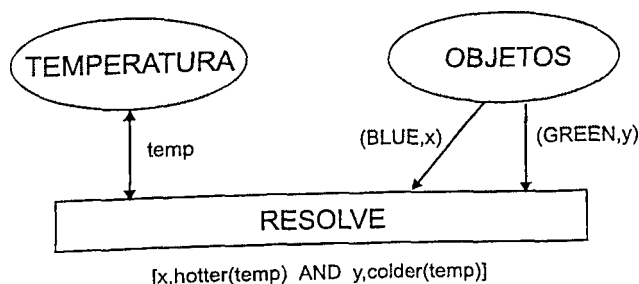


Figura 2.29: Exemplo de RdP orientada a objetos

2.3 Problema do Escalonamento

2.3.1 Definição do Problema do Escalonamento

O problema do escalonamento consiste em organizar no tempo a realização de tarefas considerando restrições temporais (datas de início de tarefas, restrições de precedência) e restrições relativas à disponibilidade de recursos requeridos.

Um escalonamento constitui uma das soluções do problema do escalonamento. Ele é definido pelo planejamento de execução das tarefas (ordem das tarefas e datas de início e fim das tarefas) e de alocação dos recursos envolvidos. Geralmente, um escalonamento visa satisfazer diversos objetivos.

Uma tarefa é uma entidade elementar localizada no tempo por uma data de início e/ou uma data de fim, cuja realização necessita da utilização de um ou diversos recursos.

Dependendo do problema, as tarefas podem ser divididas em diversas subtarefas (problemas preemptivos) ou devem ser executadas sem interrupção (problemas não preemptivos).

Um recurso é um meio técnico ou humano necessário a ser realizado para a realização de uma tarefa e disponível em quantidade limitada (a sua capacidade).

Diversos tipos de recursos podem ser considerados. Um recurso é renovável se depois de ter sido alocado à uma ou diversas tarefas, ele é disponível novamente na mesma quantidade (os homens, as máquinas, etc.). No caso contrário, o recurso é consumável (matéria bruta, dinheiro, etc...) e sua utilização no tempo é limitada.

A disponibilidade de um recurso pode variar no tempo. No caso dos recursos renováveis, podem ser distinguidos os recursos disjuntivos que só podem executar uma tarefa por vez (máquinas-ferramenta, robôs manipuladores, etc.) e os recursos cumulativos que podem ser utilizados por diversas tarefas simultaneamente, mas em número limitado (equipe de operários, por exemplo).

As restrições representam os conjuntos dos valores que podem ter as variáveis de decisão. Podemos distinguir as restrições em:

- Restrições temporais:

- As restrições de tempo alocadas que são derivadas geralmente de imperativos relacionados à gestão dos projetos e relativos às datas limites das tarefas (prazos de entrega, disponibilidade de abastecimento, etc.) ou à duração total do projeto.
- As restrições de coerência tecnológica ou restrições de roteiros, que descreve as relações de ordem relativa entre as diversas tarefas.

- Restrições de recursos:

- As restrições de utilização de recurso que exprimem a natureza e a quantidade dos meios utilizados para as tarefas, assim como as características de utilização destes meios.

- As restrições de disponibilidade dos recursos que precisam a natureza e a quantidade dos meios disponíveis no tempo.

Todas estas restrições podem ser formalizadas, baseadas nas distâncias entre os inícios de tarefas.

Na resolução de um problema de escalonamento pode-se optar por dois grandes tipos de estratégias, cujo objetivo é a otimização das soluções ou admissibilidade das soluções.

A abordagem por otimização supõe que as soluções candidatas a um problema possam ser ordenadas de maneira racional segundo um ou diversos critérios de avaliação numérica, que são vistos como indicadores de desempenho. Procura-se, então, minimizar ou maximizar tais critérios. Exemplos de critérios são:

- relacionados ao tempo
 - o tempo total de execução ou o tempo médio de finalização de um conjunto de tarefas
 - o estoque de tratamento em andamento
 - diferentes atrasos em relação às datas limites fixadas
- relacionados aos recursos
 - a quantidade de recursos necessários para realizar um conjunto de tarefas
- relacionados ao potencial ou ao fluxo do sistema
- relacionados aos custos de lançamentos, de produção, de transporte, etc., mas também relacionados aos retornos de investimentos

O problema de escalonamento pode ser estático ou dinâmico (Esquirol and Lopez 1999). O escalonamento estático é aquele onde o conjunto das informações necessárias à sua resolução é fixo (conjunto de tarefas, de recursos, e suas características). A solução é um plano previsional onde o conjunto dos intervalos de execução das tarefas caracteriza

o horizonte temporal do plano. A solução possui uma duração de validade limitada devido ao aparecimento de novas informações que ocorrem na execução do plano, como por exemplo, o aparecimento de novas tarefas, panes, atrasos, e outros. Um controle da execução é necessário, sendo este constituído de uma função de decisão em tempo real que realizará os ajustes e como consequência irá satisfazer os objetivos globais do plano. Globalmente, a geração de um plano e o controle de sua execução, é um problema de escalonamento dinâmico. A solução é um conjunto de soluções de problemas estáticos, com informações que permitem atualizar o modelo do problema a ser resolvido.

Nesta dissertação, a meta é abordar o problema do escalonamento previsional dos sistemas de produção por lotes utilizando como modelo do sistema uma rede de Petri. Num sistema de produção, escalonamento quer dizer a busca de uma sequência de operações que respeite determinadas condições estabelecidas pelo usuário que vão de interesse com o sistema analisado. Tais condições podem ser, por exemplo, um tempo (π) de funcionamento cíclico dentro do intervalo de tempo permitido ($\pi_{min} \leq \pi \leq \pi_{max}$), ausência de deadlock, tempo de permanência de um lote num reator dentro do intervalo de tempo permitido (intervalo de visibilidade), alocação de recurso sem o aparecimento de conflito efetivo, etc...

Apresentaremos as abordagens clássicas de modelos e algoritmos usados especificamente para tratar o problema do escalonamento e posteriormente, alguns trabalhos que utilizam modelos de RdPs associados a alguma técnica para tratar o escalonamento na manufatura e nos sistemas de produção por lotes.

2.3.2 Abordagem Clássica

Originalmente, a abordagem clássica do escalonamento de um sistema de produção foi feita a partir dos grafos. Tais modelos podem ser usados nos escalonamentos dos sistemas de produção cujas políticas não são cíclicas. Neles, existem os nós e os arcos orientados. As durações das operações são associadas geralmente com os arcos e o começo e fim das operações são associados aos nós. As relações de precedências entre as operações de um

roteiro são representadas de modo quantitativo pelos arcos (Elmaghraby 1977);(Gondran and Minoux 1984);(GOTHA 1993);(Prins 1994). Cada nó será ativado uma única vez durante um horizonte temporal no qual o plano tem sido elaborado. A cada vez que uma violação (evento indesejável) é encontrada, um novo grafo tem que ser criado.

Vamos ver um exemplo de modelagem a partir dos grafos. Dada uma planta composta de um buffer de entrada (IB), dois reatores (Ra,Rb), um buffer intermediário (B) e um buffer de saída (OB), temos o sistema da figura 2.30.

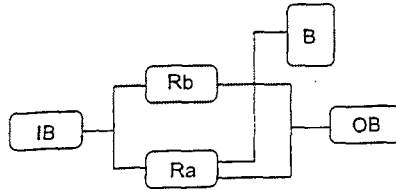


Figura 2.30: Descrição da planta

Seja IB e OB com capacidade infinita, Ra e Rb idênticos e com capacidade finita, B com capacidade de um lote, o sistema produz somente um tipo de produto, tem-se a representação da figura 2.31.

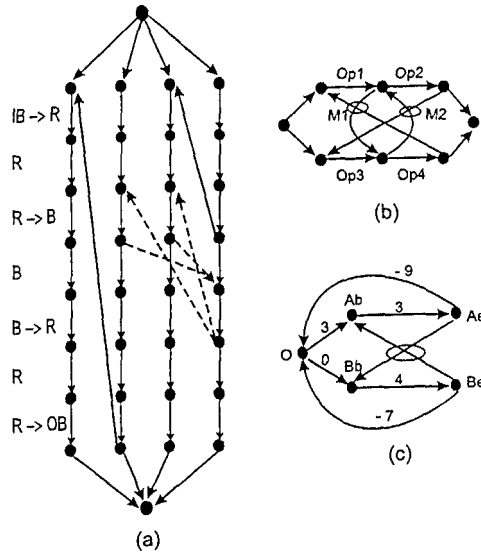


Figura 2.31: Modelagem: a) da planta b) de duas receitas c) de uma máquina

Com relação aos algoritmos utilizados no escalonamento, foram desenvolvidas técnicas

de pesquisa operacional para obter uma otimização do sistema, técnicas de IA e heurísticas.

A seguir veremos os trabalhos feitos para tratar o escalonamento para sistemas de manufatura e para sistemas de produção por lotes (batch systems) usando RdPs.

2.3.3 RdPs para o Problema do Escalonamento

2.3.3.1 Para a Manufatura

Se um escalonamento é visto como uma sequência de operações a serem seguidas para tratar uma peça a fim de obter um produto, então as redes de Petri t-temporizadas são particularmente bem adaptadas à modelagem do escalonamento de um sistema flexível de manufatura. De fato, é natural se associar as transições da rede às operações ou tarefas, bem como se representar os recursos compartilhados por marcas que se acham nos lugares.

Quando se fala em escalonamento, faz-se, geralmente, referência a uma sequência particular de operações. Se este é o caso, então o grafo de eventos constitui um modelo suficiente para descrever uma solução do problema de escalonamento, particularmente, no caso de escalonamentos cíclicos. Uma vez que o escalonamento tenha sido modelado pela rede, graças ao cálculo de tempo de ciclo é possível se obter um desempenho para o funcionamento estacionário periódico (Ramammoorthy and Ho n.d.). Considerando um modelo geral que permita a representação de diversas escolhas no nível dos roteiros de fabricação que foram propostos pela parte CAD do CIM, quando as taxas de produção são fixadas, se aplicarmos algoritmos de otimização à busca do ciclo ótimo, conseguiremos eliminar, progressivamente, todas as indeterminações e transformar o modelo geral em um grafo de eventos completamente determinístico (Cohen *et al.* 1989). Esta representação algébrica permite, em particular, encontrar a marcação inicial da rede de modo tal que o tempo de ciclo seja mínimo para um funcionamento estacionário periódico. A limitação de tal abordagem é que ela permite estudar somente os sistemas com um funcionamento completamente cíclico e repetitivo. Entretanto, um sistema flexível nunca terá um comportamento estritamente cíclico e repetitivo. Também, os grafos de eventos não permitem

a representação de recursos compartilhados, que é uma das noções fundamentais dentro do conceito de flexibilidade em um sistema flexível de produção. Consequentemente, no caso de sistemas flexíveis, é preferível adotar um modelo mais geral do que um grafo de eventos.

Na medida em que um modelo mais geral do que um grafo de eventos represente de maneira agregada um conjunto de sequências admissíveis, poderemos falar em modelagem de um escalonamento flexível e, neste caso, a abordagem mais natural será destacada por duas etapas fundamentais (Lee and DiCesare 1994b);(Lee and DiCesare 1994a);(Azzopardi and Loyd 1994a):

- A modelagem do escalonamento flexível do sistema de manufatura graças a uma rede de Petri geral t-temporizada (Lee and DiCesare 1994b). Esta etapa tem de representar, em particular, os roteiros de produção e os recursos flexíveis (figura 2.32).
- A busca do escalonamento ótimo graças a um algoritmo do tipo *Branch and Bound*. Este algoritmo, percorrendo o grafo das marcações acessíveis, faz uma busca à solução que minimiza o tempo de produção. A rápida explosão combinatorial de estados geralmente exige a utilização de heurísticas que reduzam o tempo de cálculo. A solução obtida é, então, ótima ou sub-ótima.

O resultado destas duas etapas é um escalonamento explícito que, infelizmente, dificilmente será passível de realização quando se considerar o nível "coordenação global" do sistema. Salientemos também que, neste tipo de abordagem as transições são geralmente disparadas o mais cedo possível, o que pode conduzir a um escalonamento que não seja ótimo, conforme foi mostrado em (Silva and Valette 1989) num exemplo particular.

Uma outra solução que parece melhor adaptada à natureza dos sistemas flexíveis é a elaboração de um conjunto de regras "heurísticas" que definem o que deve ser feito a cada vez que uma máquina se torna disponível para a operação seguinte a se realizar. Estas regras substituem o algoritmo *Branch and Bound* e podem ser aplicadas diretamente, em

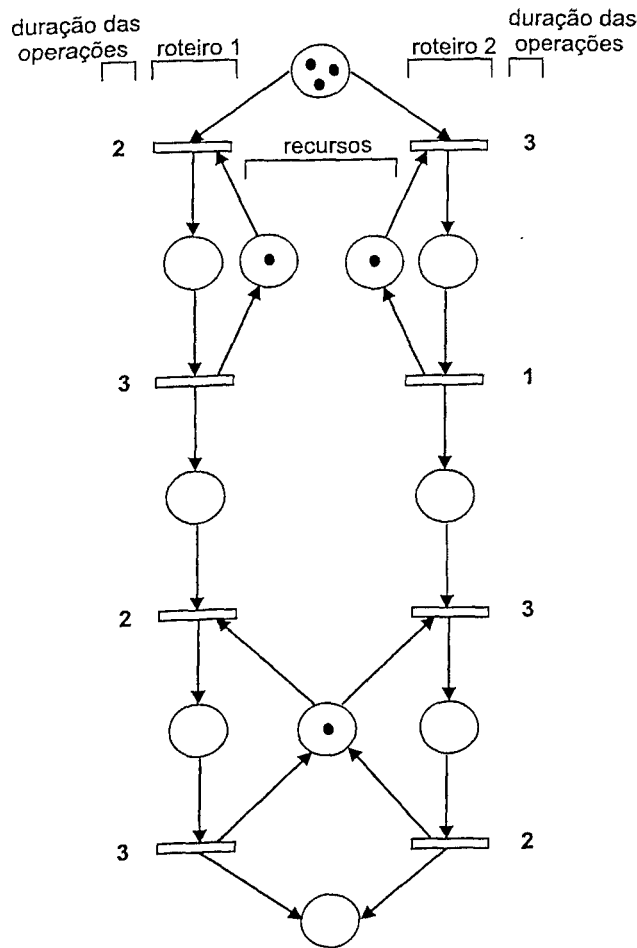


Figura 2.32: Escalonamento flexível

tempo real, à rede de Petri t -temporizada no nível da *coordenação global* do sistema. Fala-se, então, de escalonamento implícito. A limitação de tal abordagem vem da dificuldade de se obter um bom desempenho do sistema, ou, em outras palavras, de comparar o desempenho obtido com o da solução ótima.

Vale ressaltar ainda que, é geralmente uma abordagem mista, combinando as duas estratégias (método de otimização e conjunto de regras a aplicar em tempo real), que dá o melhor resultado. Em (Julia *et al.* 1995), por exemplo, apresentou-se uma abordagem baseada em uma rede de Petri temporizada com as durações de espera associadas com os lugares (tempo de espera de uma peça em um estoque intermediário entre duas operações) e com as durações das operações associadas com as transições. Em uma primeira fase, a

programação linear (Dunod 1983) é utilizada para calcular valores mínimos e máximos que representam um funcionamento mais ou menos flexível do sistema. Em uma segunda fase, um algoritmo baseado em um "jogador" de rede de Petri (mecanismo de inferência especializado) associado a um mecanismo de retrocesso (*backtrack*) permite calcular um escalonamento admissível que respeita os valores obtidos a partir da programação linear. O objetivo é obter um bom equilíbrio entre flexibilidade e eficácia a fim de poder explorar a solução obtida em tempo real usando-se um "jogador" de rede de Petri sem mecanismo de retrocesso, de maneira tal a suportar as possíveis perturbações (chegada atrasada das peças nos estoques de entradas das operações seguintes etc.) durante o funcionamento real do sistema.

De um ponto de vista geral, as redes de Petri são adaptadas à modelagem do problema de escalonamentos flexíveis. Por outro lado, há ainda muito trabalho a se fazer para se poder calcular, em tempo real, um escalonamento que seja ao mesmo tempo robusto e eficaz. Enfim, sendo o interesse das redes de Petri justamente o de poder conservar uma certa coerência entre os diversos níveis de um *CIM*, será sempre interessante que os resultados obtidos na etapa de avaliação de desempenho do sistema sejam melhor explorados na etapa de escalonamento. Isso reduzirá os cálculos a serem efetuados.

2.3.3.2 Para Sistemas de Produção por Lotes

Em (Champagnat *et al.* 1998b), o método utilizado em sistemas de produção por lotes, foi o de modelar o nível de escalonamento através de um grafo "activities on arcs" (AOA-graphs) e modelar o nível de supervisão através de RdPs. Com isto, é possível obter sequências que não possuem inconsistências (estrutural, decisional e quantitativa).

As RdPs usadas para supervisão são tipicamente agregadas e as durações das operações são indicadas nas transições (RdP t-temporizada) e são repetitivas. Além disso, elas devem ser controladas, ou seja, o disparo das transições correspondentes ao início das operações será causado pelo plano. Outro ponto importante é que um lugar ou uma transição não representa uma ocorrência única de uma operação, mas um tipo de operação o qual será

executado cíclicamente. Na detecção das inconsistências, a RdP diferencia a forma de representá-las considerando o aspecto estrutural (deadlock ou recurso compartilhado) e o quantitativo (durações das transições).

Em (Dutilleul and Denat 1998), é tratado o problema do escalonamento do “guindaste” mostrado na figura 2.33. Tais sistemas possuem a mesma problemática dos sistemas de produção por lotes: tempos imprecisos. O sistema consiste, basicamente, de uma esteira, de vários tanques, sendo que cada um possui um único produto químico diferenciados entre si (soluções), e de um guindaste cuja função é tirar a peça de um tanque e colocá-la em outro tanque. Não há buffers no sistema e o funcionamento é repetitivo.

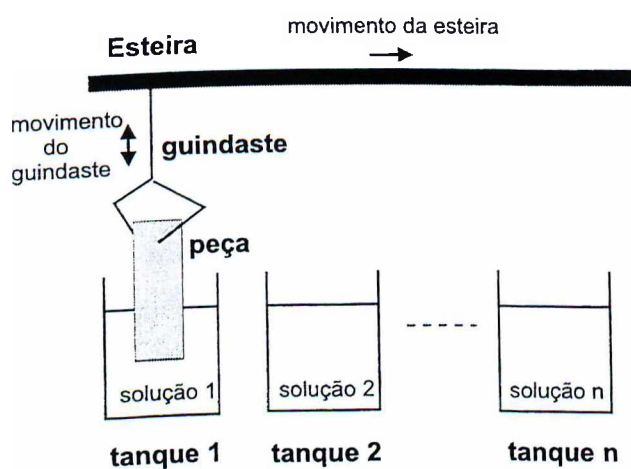


Figura 2.33: Sistema do “guindaste”

Para estes tipos de sistemas, o tempo de tratamento das peças não é bem definido. Assim, ao invés de tratarmos durações específicas no escalonamento, usamos intervalos de tempo associados aos lugares.

Inicialmente, o método propõe o uso de uma ferramenta gráfica (RdP p-temporal, ver figura 2.34) para modelar as diferentes especificações do problema. Os lugares representam as operações de tratamento químico das peças em cada tanque e as transições representam as operações de transferência da peças de uma tanque a outro. Uma segunda parte do problema é encontrar um escalonamento e uma terceira parte consiste em determinar um

controle local. As duas últimas partes, são resolvidas usando propriedades matemáticas do modelo gráfico.

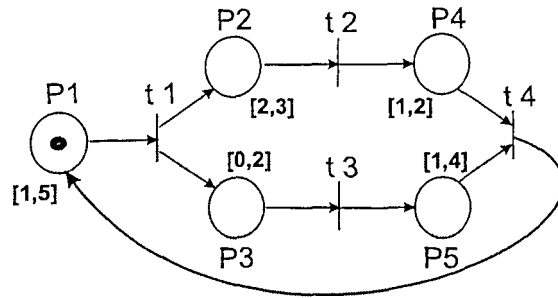


Figura 2.34: Exemplo de Rede de Petri p-temporal

Em (Azzopardi 1994), mostrou-se que a modelagem de um processo por lotes através de uma rede de Petri temporizada facilita o escalonamento e a simulação da planta. O trabalho mostrou ainda que através das restrições quantitativas e estruturais (tais como a sequência de produção dos produtos, concorrência e conflitos) da rede, pode-se utilizar técnicas de pesquisa operacional “algoritmo de busca Branch and Bound”, com o intuito de resolver os problemas para o escalonamento dos sistemas de produção por lotes. Veja na figura 2.35 um exemplo de sistema de produção por lotes considerado no trabalho (Azzopardi 1994).

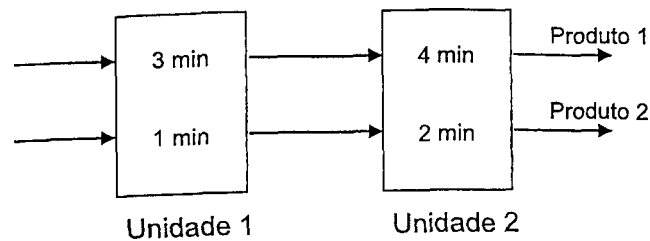


Figura 2.35: Duas unidades de uma planta de produção por lotes

Ao modelar esta planta através de uma rede de Petri temporizada, obtém-se através do algoritmo “Branch and Bound” as possíveis sequências de disparo das transições da rede considerando suas restrições temporais. Veja uma parte do respectivo espaço de

busca na figura 2.36, obtido inicialmente pelo algoritmo, referente ao sistema da figura 2.35.

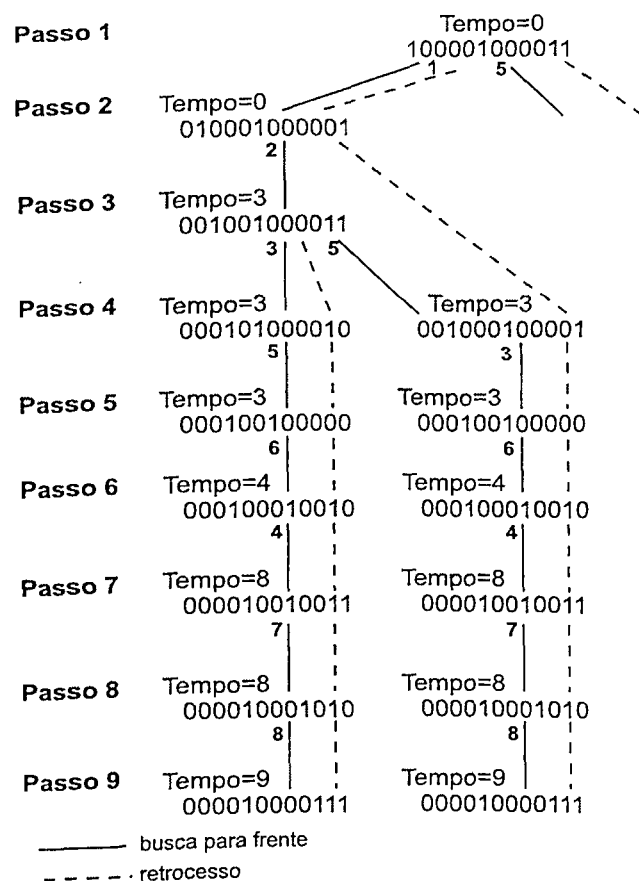


Figura 2.36: Espaço de busca

Na figura 2.36 cada passo representa um estado de marcação da rede de Petri e as sequências de 0's e 1's representam as marcações. Podemos verificar que a partir do *Passo 1*, existem dois caminhos a serem seguidos, sendo que a figura mostra apenas um deles por completo. Do *Passo 2* aparecem dois novos caminhos que são mostrados por completo.

Num sistema de produção por lotes, o número de permutações que tem de ser investigado resulta em um espaço de busca muito grande para o algoritmo "Branch and Bound". Para melhorar este fato, mostrou-se uma técnica que reduz o espaço de busca original do sistema (figura 2.36) através do aperfeiçoamento do algoritmo de busca melhorando sua taxa de convergência (Azzopardi and Loyd 1994a). A figura 2.37 mostra o espaço de

busca reduzido do exemplo da figura 2.36.

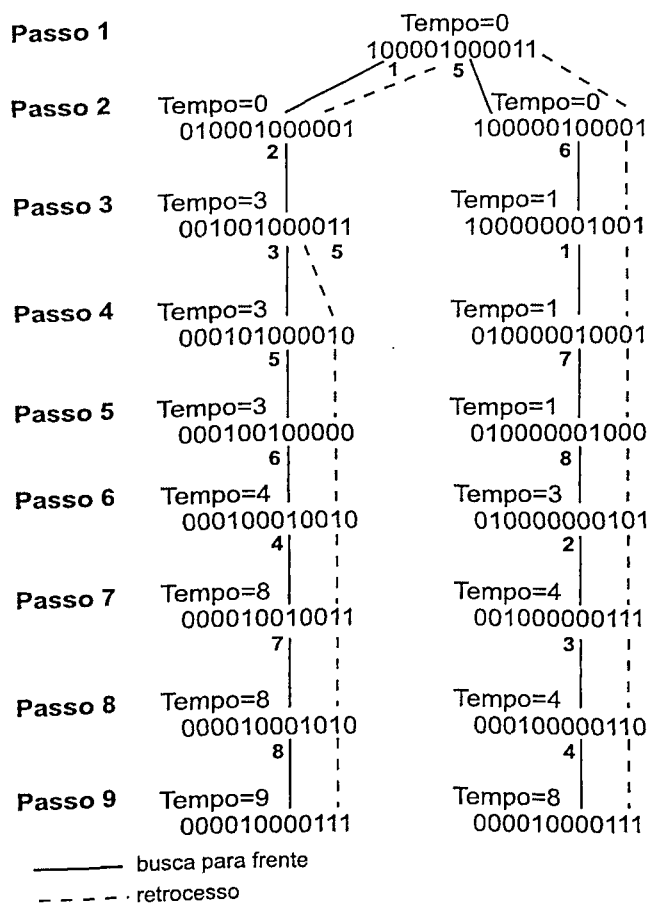


Figura 2.37: Espaço de busca reduzido

Encontrado o espaço de busca reduzido, utilizou-se um algoritmo de simulação para obter um escalonamento otimizado das operações da planta.

Vale ressaltar que neste trabalho, utilizou-se um Diagrama de Gant para ilustrar as operações otimizadas do sistema. Este diagrama auxilia na visualização das sequências de operações ocorridas num determinado espaço de tempo.

2.3.3.3 Conclusões do Capítulo

- Os grafos não são modelos compactados e por isso exigem uma considerável dedicação na sua construção, principalmente em sistemas flexíveis. Além disso, os grafos são pouco práticos no que diz respeito a mudanças que certamente ocorrem

no sistema durante a sua evolução. Vale ressaltar que o interfaceamento entre os grafos e as RdPs não é muito fácil, quando ambos estão modelando níveis diferentes (escalonamento e supervisão).

- As RdPs são em geral modelos bem adaptados para representar o problema do escalonamento dos sistemas, sejam eles de manufatura ou de produção por lotes. São modelos compactos, flexíveis e interativos.
- As RdPs t-temporizadas se adaptam melhor aos sistemas de manufatura, por tratarem o tempo de maneira precisa. Já as RdPs p-temporais se adaptam bem aos sistemas híbridos por tratarem o tempo de maneira imprecisa, exibindo um intervalo de tempo a cada lugar da rede.
- Devido ao fato dos sistemas de produção por lotes possuírem tempos imprecisos, utilizaremos RdPs com tempos imprecisos. Os tempos de transferência dos lotes (fluxos nas canalizações) são indicados pelas temporizações das transições e os tempos de tratamentos dos lotes (tratamento nos reatores) são indicados pelos intervalos de tempos associados aos lugares. Isto porque nos sistemas de produção por lotes o tempo de transferência do lote não é controlável, ou seja, uma vez iniciada a transferência, não podemos interferir na operação de modo a acelerá-la ou retardá-la. Na verdade o que se tem é um intervalo contínuo com os possíveis valores de tempo de transferência dos lotes. Para não termos que considerar uma infinidade de valores de tempo, o que fatalmente provocaria uma explosão combinatorial de estados, consideramos um valor médio deste intervalo que será bem adaptado ao funcionamento real do sistema. Daí a justificativa de se usar uma temporização para cada duração de transferência de um lote. Já o tempo de tratamento de cada lote é controlável, pois, apesar de também termos um intervalo contínuo de tempo de tratamento, podemos decidir dentro deste intervalo qual o tempo necessário. Daí o fato de usarmos um intervalo contínuo para cada tempo de tratamento de um lote.

A seguir, veremos como os sistemas de produção por lotes podem ser modelados.

Capítulo 3

Modelagem de Sistemas de Produção por Lotes

Como já foi dito anteriormente, os sistemas de produção por lotes assemelham-se muito aos sistemas flexíveis de manufatura quando consideramos o fato de que ambos possuem sequências de operações ou tratamentos bem definidos, recursos compartilhados, produção cíclica ou acíclica. O que basicamente diferencia um do outro, são as restrições temporais. Com isto, é possível utilizar os modelos usados no campo da manufatura flexível para os sistemas de produção por lotes levando-se em conta suas diferenças.

Os sistemas tratados nesta dissertação são sistemas de produção por lotes que possuem receitas pré-definidas, reatores funcionando em modo fechado (um reator é usado por apenas uma operação ao mesmo tempo), funcionamento cíclico, tempo de transferência dos lotes de um reator para outro com valor específico e tempo de permanência dos lotes num reator com valor variável dentro de um intervalo de tempo conhecido.

Veremos agora como modelaremos as restrições de sincronização e as restrições temporais, cuja referência é o trabalho de Tese de Doutorado (Julia 1997a).

3.1 Modelagem das Restrições de Sincronização

Quando falamos em restrições, no nível da modelagem, estamos nos referindo a sub-redes de Petri que representam somente um aspecto parcial do sistema. O modelo global é obtido da união das sub-redes e é uma rede de Petri mais geral que um grafo de eventos, sendo mantidas as boas propriedades obtidas das sub-redes.

Veremos a seguir, a modelagem de cada restrição e seu respectivo formalismo.

3.1.1 Receitas

3.1.1.1 Roteiros

Dentro de um sistema de produção por lotes existem as chamadas receitas. As receitas são os roteiros de produção definidos pelas sequências de tratamentos dos lotes feitos pelos diversos reatores.

Formalmente, no nível da rede de Petri, definimos um roteiro da seguinte forma: numa rede de Petri ordinária e não marcada $r = \langle O_r, T_r, Pre_r, Post_r \rangle$ (David and Alla 1992), r define um roteiro se e somente se:

- $T_r = \{t_{r,1}, t_{r,2}, \dots, t_{r,n_r}\}$ é uma sequência finita de n_r transições, com n_r sendo o número de transferências que o lote sofrerá de um reator a outro associadas ao roteiro r .
- $O_r = \{o_{r,1}, o_{r,2}, \dots, o_{r,n_r+1}\}$ é uma sequência finita de $n_r + 1$ lugares, com $n_r + 1$ sendo o número de tratamentos que o lote sofrerá dos reatores associados ao roteiro r .
- $Pre_r : O_r \times T_r \rightarrow \{0,1\}$ é a aplicação de incidência anterior sendo que para $i \neq j$, $Pre_r(o_{r,i}, t_{r,i}) = 1$ e $Pre_r(o_{r,j}, t_{r,i}) = 0$.

Isto significa que há um lugar $o_{r,i}$ na entrada da transição $t_{r,i}$.

- $Post_r : O_r \times T_r \rightarrow \{0,1\}$ é a aplicação de incidência posterior sendo que para $i \neq j$, $Post_r(o_{r,i+1}, t_{r,i}) = 1$ e $Post_r(o_{r,j+1}, t_{r,i}) = 0$.

Isto significa que há um lugar $o_{r,i+1}$ na saída da transição $t_{r,i}$.

Interpreta-se a transição $t_{r,i}$ como sendo a $i^{\text{ésima}}$ transferência de um lote de um reator a outro num roteiro r e o lugar $o_{r,i}$ como sendo o $i^{\text{ésimo}}$ tratamento de um lote por um reator no roteiro r .

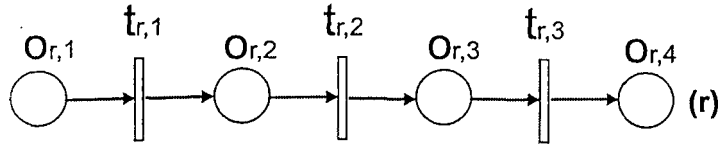


Figura 3.1: Roteiro r

A rede da figura 3.1 representa um roteiro r onde, um lote inicialmente se encontra num estoque de entrada $o_{r,1}$ e no final ele se encontra num estoque de saída $o_{r,4}$, tendo o lote recebido os tratamentos $o_{r,2}$ e $o_{r,3}$ nesta ordem.

3.1.1.2 Sub-roteiros

Formalmente, no nível da rede de Petri, definimos um sub-roteiro da seguinte forma: numa rede de Petri ordinária e não marcada $sr = \langle O_{sr}, T_{sr}, Pre_{sr}, Post_{sr} \rangle$, sr define um sub-roteiro de um roteiro r se e somente se:

- $T_{sr} = \{t_{r,m_{sr}+1}, t_{r,m_{sr}+2}, \dots, t_{r,m_{sr}+n_{sr}}\}$ é uma sequência finita de n_{sr} transições, com $m_{sr} \in \mathbb{N}$ e $n_{sr} \in \mathbb{N}^*$ ($T_{sr} \subseteq T_r$).

n_{sr} representa o número de operações discretas associadas ao sub-roteiro sr .

$m_{sr} + 1$ representa o número da primeira operação do sub-roteiro sr dentro do roteiro r .

- $O_{sr} = \{o_{r,m_{sr}+2}, o_{r,m_{sr}+3}, \dots, o_{r,m_{sr}+n_{sr}}\}$ é uma sequência finita de $n_{sr} - 1$ lugares de O_r ($O_{sr} \subseteq O_r$).

$n_{sr} = 1$ corresponde ao caso em que o sub-roteiro é vazio.

$n_{sr} - 1$ representa o número de lugares intermediários do sub-roteiro sr .

$m_{sr} + 2$ representa o número do primeiro lugar intermediário do sub-roteiro sr dentro do roteiro r .

- $Pre_{sr} : O_{sr} \times T_{sr} \rightarrow \{0,1\}$ é a aplicação de incidência anterior sendo que para $i \neq j$, $Pre_{sr}(o_{r,i}, t_{r,i}) = 1$ e $Pre_{sr}(o_{r,j}, t_{r,i}) = 0$.

Isto significa que há um lugar $o_{r,i}$ na entrada da transição $t_{r,i}$, exceto para a primeira transição $t_{r,m_{sr}+1}$ do sub-roteiro sr .

- $Post_{sr} : O_{sr} \times T_r \rightarrow \{0,1\}$ é a aplicação de incidência posterior sendo que para $i \neq j$, $Post_{sr}(o_{r,i+1}, t_{r,i}) = 1$ e $Post_{sr}(o_{r,j+1}, t_{r,i}) = 0$.

Isto significa que há um lugar $o_{r,i+1}$ na saída da transição $t_{r,i}$, exceto para a última transição $t_{r,m_{sr}+n_{sr}}$ do sub-roteiro sr .

A diferença fundamental entre um roteiro e um sub-roteiro é que um roteiro começa por um lugar “estoque de entrada” e termina por um lugar “estoque de saída” enquanto que um sub-roteiro começa por uma transição e termina por uma outra transição (as transições de entrada e de saída são as mesmas quando o sub-roteiro se resumir à uma operação de transferência somente). Quando considerarmos mais de um sub-roteiro associado ao roteiro $r1$ por exemplo, notaremos o primeiro sub-roteiro como sendo $s1r1$, o segundo $s2r1$ etc.

A rede da figura 3.2 representa dois exemplos de sub-roteiros $s1r$ e $s2r$ obtidos a partir do roteiro r da figura 3.1.

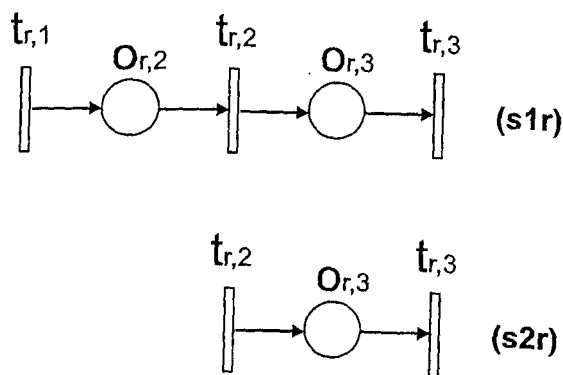


Figura 3.2: Sub-roteiros $s1r$ e $s2r$

3.1.1.3 Restrição Kanban

A política do tipo kanban permite que se controle o número de lotes a serem tratados num sub-roteiro simultaneamente.

Formalmente, no nível da rede de Petri, definimos uma política kanban da seguinte forma: numa rede de Petri ordinária e marcada $C_k = \langle O_k, T_k, Pre_k, Post_k, M_k \rangle$, C_k associa uma política kanban a um sub-roteiro sr se e somente se:

- $O_k = O_{sr} \cup \{o_k\}$

- $T_k = T_{sr}$

-

$$Pre_k \begin{cases} Pre_{sr} : O_{sr} \times T_{sr} \rightarrow \{0, 1\} \\ Pre_{ok} : \{o_k\} \times T_{sr} \rightarrow \{0, 1\} \end{cases}$$

é a aplicação de incidência anterior tal que para $i \neq m_{sr} + 1$,

$$Pre_{ok}(o_k, t_{r, m_{sr}+1}) = 1 \text{ e } Pre_{ok}(o_k, t_{r, i}) = 0$$

-

$$Post_k \begin{cases} Post_{sr} : O_{sr} \times T_{sr} \rightarrow \{0, 1\} \\ Post_{ok} : \{o_k\} \times T_{sr} \rightarrow \{0, 1\} \end{cases}$$

é a aplicação de incidência posterior tal que para $i \neq m_{sr} + n_{sr}$,

$$Post_{ok}(o_k, t_{r, m_{sr}+n_{sr}}) = 1 \text{ e } Post_{ok}(o_k, t_{r, i}) = 0$$

- $M_k : O_k \rightarrow N$ é a marcação inicial tal que

$$M_k(o_k) = m_{ok} \text{ e } M_k(o_{r, i}) = 0 \text{ com } i \in \{m_{sr} + 2, \dots, m_{sr} + n_{sr}\}.$$

m_{ok} é o número de marcas kanban que se encontra dentro do lugar o_k .

Damos a notação para o primeiro lugar kanban da rede como sendo o_{k1} , o segundo como sendo o_{k2} , etc.

Para que a primeira operação de transferência $t_{r,m_{sr}+1}$ do sub-roteiro sr seja executada, é necessário que haja um número suficiente de fichas dentro do estoque de entrada e do kanban o_k . A marcação inicial de o_k é que vai definir a capacidade dos lugares intermediários do sub-roteiro sr .

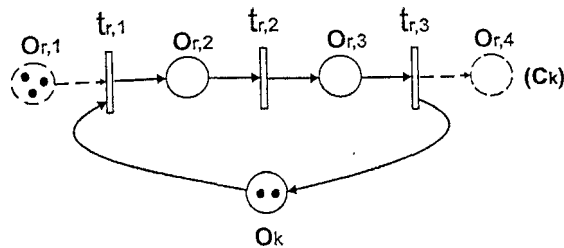


Figura 3.3: Restrição kanban C_k

Na figura 3.3, considera-se que as duas fichas dentro do lugar o_k representam o número de lotes do estoque de entrada $o_{r,1}$ do roteiro r que podem ser tratados simultaneamente. As fichas kanban podem ser vistas como sinais de controle que autorizam o tratamento de novos lotes em função das capacidades dos lugares intermediários $o_{r,2}$ e $o_{r,3}$ do sub-roteiro $s1_r$. A soma das marcas nos lugares $o_{r,2}$ e $o_{r,3}$ está limitada em 2.

3.1.2 Alocação de Recursos

Em um grafo de eventos, a ordem das operações é completamente definida (Hillion and Proth 1989);(Ramammoorthy and Ho n.d.): cada lugar possui somente um arco de entrada e um arco de saída. Nesta situação, não podemos obter sucesso se o sistema modelado possuir problemas envolvendo situações de conflitos, ou seja, existência de sequências que serão definidas com a evolução do sistema. É o caso de sistemas flexíveis como por exemplo o sistema de produção por lotes, que possui reatores que podem ser solicitados por mais de uma receita ao mesmo tempo. Neste tipo de situação, o modelo deve exibir situações de conflitos através de alocação de recursos compartilhados.

Formalmente, no nível da rede de Petri, definimos uma alocação de recurso da seguinte forma: numa rede de Petri ordinária e marcada $C_R = \langle O_R, T_R, Pre_R, Post_R, M_R \rangle$, C_R

define uma alocação de recurso se e somente se:

- $O_R = \bigcup_{\alpha=1}^{n_R} O_{(sr)_\alpha} \cup \{o_R\}$

com o_R sendo um lugar de um recurso compartilhado e n_R sendo o número de sub-roteiros $(sr)_\alpha$ associados ao lugar o_R .

Um recurso pode ser compartilhado por mais de um sub-roteiro, estando no mesmo roteiro, desde que ambos sejam disjuntos. Desta forma temos:

$$O_{(sr)_i} \cap O_{(sr)_j} = \emptyset \text{ para } i \neq j.$$

- $T_R = \bigcup_{\alpha=1}^{n_R} T_{(sr)_\alpha}$

com $T_{(sr)_i} \cap T_{(sr)_j} = \emptyset$ para $i \neq j$.

-

$$Pre_R \left\{ \begin{array}{l} Pre_{(sr)_1} : O_{(sr)_1} \times T_{(sr)_1} \rightarrow \{0, 1\} \\ Pre_{(sr)_2} : O_{(sr)_2} \times T_{(sr)_2} \rightarrow \{0, 1\} \\ \vdots \\ Pre_{(sr)_{n_R}} : O_{(sr)_{n_R}} \times T_{(sr)_{n_R}} \rightarrow \{0, 1\} \\ Pre_{o_R} : \{o_R\} \times T_R \rightarrow \{0, 1\} \end{array} \right.$$

é a aplicação de incidência anterior, com $i \neq m_{(sr)_\alpha} + 1$,

$$Pre_{o_R}(o_R, t_{(r)_\alpha, m_{(sr)_\alpha} + 1}) = 1 \text{ e } Pre_{o_R}(o_R, t_{(r)_\alpha, i}) = 0.$$

Isto significa que o_R é o lugar de entrada da primeira transição (transferência) do sub-roteiro $(sr)_\alpha$.

$(r)_\alpha$ é o roteiro que contém o sub-roteiro $(sr)_\alpha$.

-

$$Post_R \left\{ \begin{array}{l} Post_{(sr)_1} : O_{(sr)_1} \times T_{(sr)_1} \rightarrow \{0, 1\} \\ Post_{(sr)_2} : O_{(sr)_2} \times T_{(sr)_2} \rightarrow \{0, 1\} \\ \vdots \\ Post_{(sr)_{n_R}} : O_{(sr)_{n_R}} \times T_{(sr)_{n_R}} \rightarrow \{0, 1\} \\ Post_{o_R} : \{o_R\} \times T_R \rightarrow \{0, 1\} \end{array} \right.$$

é a aplicação de incidência posterior, com $i \neq m(sr)_\alpha + n(sr)_\alpha$,

$$Post_{o_R}(o_R, t_{\alpha, m(sr)_\alpha + n(sr)_\alpha}) = 1 \text{ e } Post_{o_R}(o_R, t_{\alpha, i}) = 0.$$

Isto significa que o_R é o lugar de saída da última transição do sub-roteiro $(sr)_\alpha$.

- $M_R : O_R \rightarrow N$ é a marcação inicial tal que

$$M_R(o_R) = m_{o_R} \text{ e } M_R(s_{\alpha, i}) = 0$$

para $i \in \{m(sr)_\alpha + 2, \dots, m(sr)_\alpha + n(sr)_\alpha\}$.

m_{o_R} é o número de recursos do mesmo tipo que se encontram dentro do lugar o_R .

Damos a notação do primeiro lugar de recurso compartilhado do sistema de produção por lotes como sendo o_{R1} , o segundo como sendo o_{R2} etc.

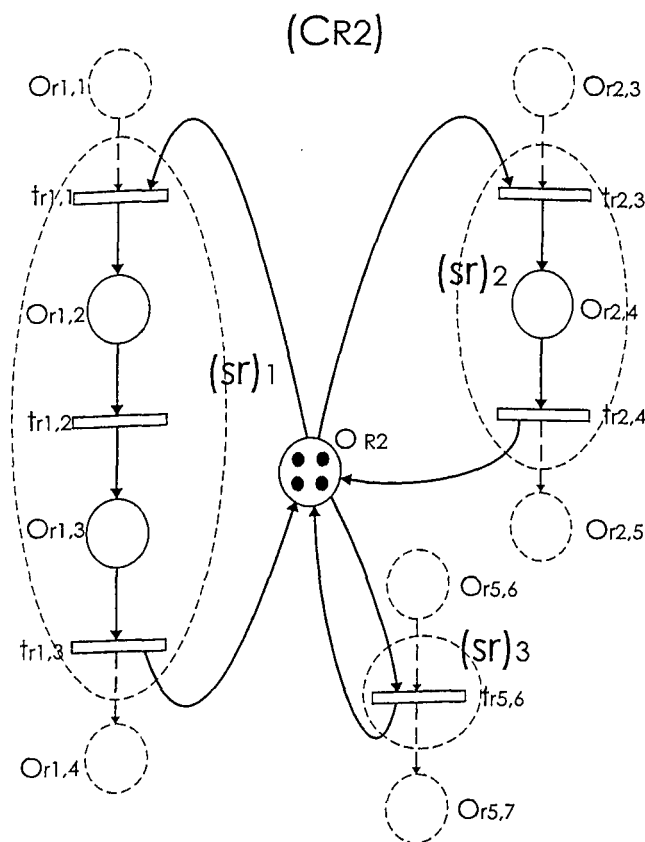


Figura 3.4: Alocação de Recurso C_{R2}

As fichas de o_{R2} da figura 3.4 representam os recursos que por ventura podem ser

utilizados para possibilitar as operações de transferência $t_{r1,1}$, $t_{r1,2}$ e $t_{r1,3}$ do sub-roteiro $(sr)_1$, ou as operações de transferência $t_{r2,3}$ e $t_{r2,4}$ de $(sr)_2$, ou a operação de transferência $t_{r5,6}$ de $(sr)_3$. O lugar o_{R2} que é compartilhado por três sub-roteiros, permite representar um escalonamento flexível, ou seja, a ordem das alocações pelos sub-roteiros é variada.

Vale ressaltar que a definição sintática da restrição C_k é uma caso particular da definição da restrição C_R . Com efeito, as capacidades dos lugares intermediários dos sub-roteiros são relacionadas diretamente com os recursos necessários à realização da produção. Entretanto, podemos associar uma semântica diferente para cada uma das duas restrições do ponto de vista da produção. Quanto mais fichas houver dentro dos lugares Kanban, maior é a chance para que as fichas dos recursos compartilhados se acabem e, conseqüentemente, o número de recursos que estão sendo esperados para serem utilizados aumenta. Em contra partida, se diminuirmos o número de fichas nos lugares Kanban, a flexibilidade do sistema e o número de recursos a serem utilizados diminui. Portanto, a importância de diferenciar estes dois tipos de restrições é de definir um número de fichas Kanban aceitável que não comprometa nem a flexibilidade do sistema e nem o número de recursos em curso.

3.1.3 Política Cíclica

A política cíclica é a política na qual se tem o controle de quantos lotes serão tratados em cada receita durante um mesmo período, sendo assim, a restrição descreverá as exigências da produção. Esta política cíclica foi apresentada pela primeira vez por Ohl (Ol *et al.* 1994).

Formalmente, no nível da rede de Petri, definimos uma política cíclica da seguinte maneira: sejam Nr roteiros descritos pela rede r_i com $i \in \{1, \dots, Nr\}$, a rede de Petri generalizada e marcada (David and Alla 1992) $C_c = \langle O_c, T_c, Pre_c, Post_c, M_c \rangle$, C_c define uma política de produção cíclica se e somente se:

- $O_c = \{c_{r1,1}, c_{r1,2}, \dots, c_{ri,1}, c_{ri,2}, \dots, c_{rNr,1}, c_{rNr,2}\}$ é uma seqüência finita e não nula de $2Nr$ lugares.

- $T_c = \{t_{r1,1}, t_{r2,1}, \dots, t_{rN_r,1}\} \cup \{t_s\}$ é uma sequência finita de $N_r + 1$ transições.
Os elementos da sequência representam as primeiras operações de transferência dos N_r roteiros mais a transição de sincronização t_s .
- $Pre_c : O_c \times T_c \rightarrow N$ é a aplicação de incidência anterior tal que:

$$Pre_c(t_s, c_{ri,2}) = lb_{ri}(\text{lotes})$$

$$Pre_c(t_s, c_{ri,1}) = 0$$

$$Pre_c(t_{ri,1}, c_{ri,1}) = 1$$

$$Pre_c(t_{ri,1}, c_{rj,1}) = 0 \text{ se } j \neq i$$

$$Pre_c(t_{ri,1}, c_{rj,2}) = 0 \text{ se } j \in \{1, \dots, N_r\}$$
- $Post_c : O_c \times T_c \rightarrow N$ é a aplicação de incidência posterior tal que:

$$Post_c(t_s, c_{ri,1}) = lb_{ri}(\text{lotes})$$

$$Post_c(t_s, c_{ri,2}) = 0$$

$$Post_c(t_{ri,1}, c_{ri,2}) = 1$$

$$Post_c(t_{ri,1}, c_{rj,2}) = 0 \text{ se } j \neq i$$

$$Post_c(t_{ri,1}, c_{rj,1}) = 0 \text{ se } j \in \{1, \dots, N_r\}$$
- $M_c : O_c \rightarrow N$ é a marcação inicial tal que

$$M_c(c_{ri,1}) \geq lb_{ri} \text{ e } M_c(c_{ri,2}) = 0.$$

A rede da figura 3.5 representa um exemplo de política de produção cíclica. Nesta restrição modela-se o fato de que o sistema de produção por lotes irá tratar lb_{r1} lotes do estoque de entrada $o_{r1,1}$, lb_{r2} lotes do estoque de entrada $o_{r2,1}$ e lb_{r3} lotes do estoque de entrada $o_{r3,1}$. O sinal "+" dentro do lugar de entrada dos roteiros $r1$, $r2$ e $r3$ significa que os estoques de entrada do sistema são alimentados permanentemente e t_s é a transição de sincronização que impõe a política cíclica.

É possível considerar, no exemplo, que os estoques de entrada $o_{r1,1}$ e $o_{r2,1}$ são alimentados por lotes de mesmo tipo. Assim, pode-se representar os dois estoques como sendo um só estoque de entrada que alimenta dois sub-roteiros distintos. Quanto às taxas de

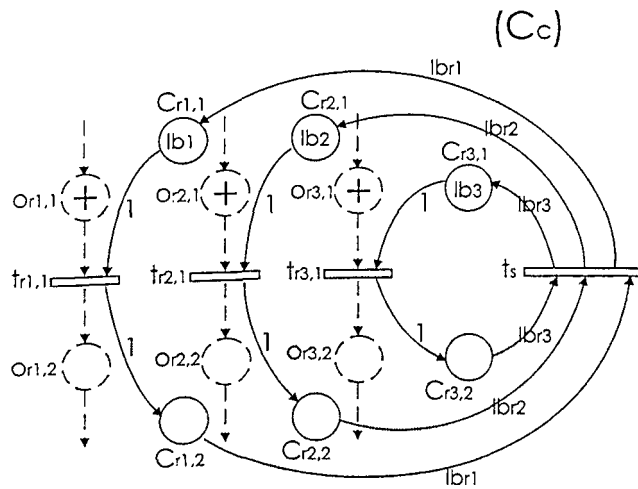


Figura 3.5: Política Cíclica C_c

produção associadas aos lotes dos estoques de entrada $(o_{r1,1}, o_{r2,1})$ de uma parte, e ao lote de entrada $o_{r3,1}$ de outra parte da produção, são definidas da seguinte forma:

$$\tau_{(o_{r1,1}, o_{r2,1})} = \frac{lb_{r1} + lb_{r2}}{lb_{r1} + lb_{r2} + lb_{r3}}$$

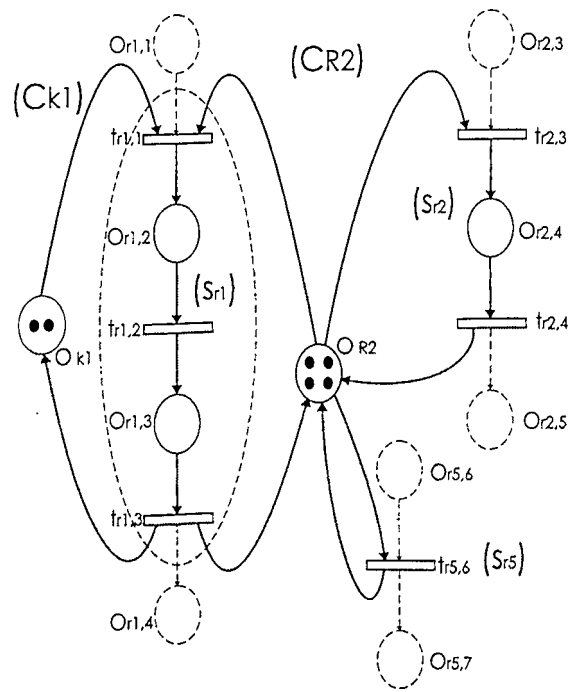
$$\tau_{(o_{r3,1})} = \frac{lb_{r3}}{lb_{r1} + lb_{r2} + lb_{r3}}$$

3.1.4 Construção do Modelo Global

O modelo global do sistema de produção por lotes é obtido pela fusão de sub-roteiros das restrições do tipo C_k , C_R e C_c .

A figura 3.6 é um exemplo de fusão de uma restrição C_{k1} com a restrição C_{R2} segundo o sub-roteiro (S_{r1}) .

Em (Julia 1997a), mostrou-se em detalhe no nível dos invariantes de lugar e de transição o efeito das fusões das restrições. Em síntese o que se obtém é que cada uma das restrições que definimos (restrição kanban, restrição de recurso, restrição cíclica) aparece

Figura 3.6: Fusão de C_{k1} com C_{R2}

dentro do modelo global sob a forma de um invariante de lugares. O modelo global do sistema é um invariante de transições, além de que se para um ciclo a transição t_s é disparada uma vez, então as transições dos roteiros r do modelo global serão disparadas lb_r vezes cada uma. Isto quer dizer que as restrições que definem o modelo do sistema possuem frequências de ocorrência de suas transições que são impostas pela restrição cíclica C_c .

Existe no nível do modelo global do sistema de produção por lotes uma associação de componentes conservativos e repetitivos estacionários. Deste resultado, podemos deduzir de um lado que a rede que modela o sistema é k -limitada (todos os lugares da rede são k -limitados, ou seja, o número de marcas de cada lugar é menor ou igual a um número k). Por outro lado, a rede é coberta por um invariante de transições possibilitando um regime cíclico que induz a sub-rede seguir a política cíclica, em relação às proporções entre os lotes dos diversos roteiros (a transição t_s sincroniza as diversas execuções).

Como nos restringimos ao caso de uma rede global viva, é importante verificar que

não existe um entrelaçamento indesejado de utilização de recursos. A ocorrência deste fato provoca os chamados bloqueios mortais ("Deadlock"), que são traduzidos pelo não disparo de determinadas transições em definitivo. Existe algumas técnicas utilizadas para análise do modelo do sistema para verificar sua vivacidade, como por exemplo o uso de regras de redução (Berthelot *et al.* 1980);(Berthelot 1987);(Murata 1989) ou a busca de sifons (Jeng and Xie 1997).

A ausência de "deadlock" numa rede depende da sua marcação inicial, onde o número de fichas pode ser visto como parâmetro para a análise da vivacidade. Fazendo esta verificação através da enumeração das marcações, é quase certo de que depararemos com uma explosão combinatorial de estados se a rede for complexa. Uma outra técnica que pode ser usada para a verificação da vivacidade da rede é o uso de regras de redução que procuram simplificar a rede através da fusão de certas transições e/ou lugares. A partir da rede simplificada, torna-se mais fácil enumerarmos a marcação da rede sem que as boas propriedades iniciais tenham sido perdidas. Dentre todas as regras de redução, uma é de particular importância: a redução de lugares redundantes (implícitos). Entretanto, há casos em que a fusão de uma série de transições não é possível devido à complexidade da rede, e então podemos utilizar um outro método baseado nos chamados "sifons".

Uma sub-sequência de lugares S é chamada de Sifon se $\bullet S \subseteq S^\bullet$, ou seja, qualquer transição de entrada de S é também uma transição de saída de S . Um sifon é dito mínimo se o mesmo não contém outro sifon. Um sifon livre de fichas, implica que todas as transições conectadas em S são não vivas. Assim, para qualquer marcação tal que nenhuma transição é habilitada, a sequência de lugares vazios forma um sifon. Uma rede de Petri está livre de bloqueios mortais se não há sifons mínimos eventualmente vazios.

Suponha um sistema de produção por lotes onde há duas receitas:

receita1 = primeira operação em $R1(O_{11})$, transferência do reator $R1$ para $R2(O_{12})$,

segunda operação em $R2(O_{13})$.

receita2 = primeira operação em $R2(O_{21})$, transferência do reator $R2$ para $R1(O_{22})$,

segunda operação em $R1(O_{23})$.

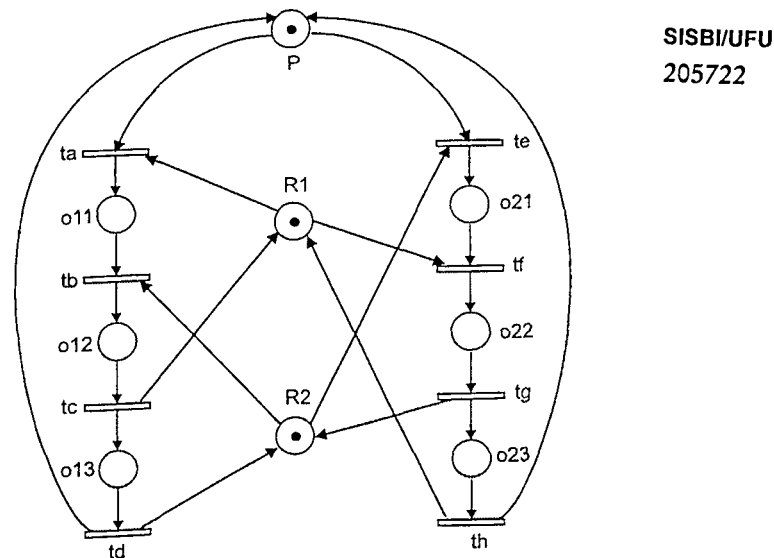


Figura 3.7: Exemplo 1 de utilização de Sifon

Vamos supor que apenas um lote de produto é tratado por vez, então a marcação inicial da respectiva rede de Petri é dada na figura 3.7. Em particular, a rede é viva e portanto livre de “deadlock”. Esta análise pode ser feita por enumeração de estados (somente 7 estados). Supondo agora que ao invés de apenas um lote ser tratado por vez, sejam tratados dois lotes de produtos simultaneamente. Isto equivale a dizer que o lugar p da rede tem duas fichas inicialmente. Desta forma podemos disparar a transição t_a e, imediatamente depois, a transição t_e . Como consequência teremos uma marcação tal que $M(O_{11}) = M(O_{21}) = 1$ e $M(O_{ij}) = 0$ para todos os outros lugares. Isto caracteriza a impossibilidade de disparar qualquer outra transição da rede: existência de “deadlock”. Com o incremento de fichas em p , a vivacidade é perdida.

Considerando na rede da figura 3.7 o conjunto de lugares: $R_1, R_2, O_{12}, O_{13}, O_{22}, O_{23}$; teremos o sifon da figura 3.8.

Ao disparar t_a e t_e consecutivamente, teremos um sifon vazio e portanto a caracterização de um “deadlock”. Esta situação é contornável se considerarmos que:

$$M_0(p) < M_0(R_1) + M_0(R_2)$$

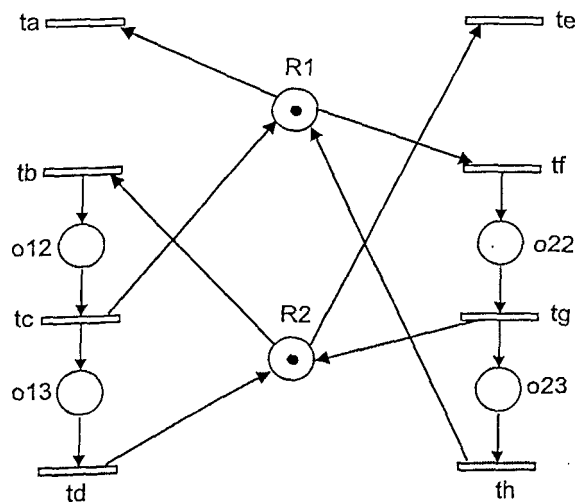


Figura 3.8: Exemplo 2 de utilização de Sifon

o que significa que o número total de reatores do tipo $R1$ $R2$ tem que ser maior que o número total de lotes.

É importante salientar que a tentativa proposta não garante a existência efetiva de uma sequência de habilitações de transições correspondente ao ciclo. Assim, se a sequência de sub-redes define uma sequência de restrições contraditórias, a rede global não será viva e é possível que alguma sequência habilitável não corresponda ao invariante de transições definido no ciclo. É claro, que se impormos certas restrições em relação à ordem de utilização dos recursos dentro dos roteiros, podemos obter uma rede viva. A atitude será então muito restrita e o alvo de nosso trabalho é de caracterizar desde políticas de conduta até uma sequência de condições necessárias. É mais importante introduzirmos o mais cedo possível o tempo de forma explícita para tratarmos as condições necessárias em relação ao processo de decisão, do que provar que uma sequência correspondente ao componente repetitivo estacionário é efetivamente habilitável. Isto porque queremos obter uma sequência admissível de operações. No caso de sistemas de produção por lotes, a problemática do tempo é bem mais crítica do que em sistemas que buscam simplesmente a otimização de sequências de operações, como por exemplo sistemas flexíveis de manufatura. Nos sistemas de produção por lotes, um tempo de espera grande demais de um

lote em um reator poderia comprometer todo o lote (por exemplo, um lote fervendo em um reator por muito tempo poderia estragá-lo), enquanto que num sistema flexível de manufatura, um tempo de espera grande demais num determinado local da célula não seria tão prejudicial. Portanto, o que buscamos é uma sequência de restrições admissíveis e não uma busca otimizada das mesmas.

Vamos agora, introduzir explicitamente o tempo dentro do nosso modelo.

3.2 Modelo Temporal/Temporizado

Um sistema de produção por lotes é um sistema que depende do tempo, pois, possui uma duração de operação não nula. Desta forma, o modelo utilizado para modelar tal sistema tem de ser capaz de incorporar nele as durações de operações. Existem diversos modelos que permitem a representação do tempo: as redes de Petri temporizadas (p-temporizadas: (Sifakis 1977), (Tazza 1987); t-temporizadas: (Ramchandani 1974), (Ramammoorthy and Ho n.d.)), as redes de Petri temporais (t-temporais: (Merlin 1974), (Menasche 1996); p-temporais: (Khansa *et al.* 1996)), as redes de Petri estocásticas (Florin and Natkin 1984) etc...

3.2.1 Modelo t-temporizado

As RdP temporizadas associam uma duração de operação não nula entre o início e o fim de uma operação, conseqüentemente, o que se pode modelar são durações de operações bem definidas. Quando a duração da operação é associada aos lugares temos as chamadas redes de Petri p-temporizadas e quando a duração da operação é associada às transições temos as chamadas redes de Petri t-temporizadas. Considerando o segundo modelo de rede, a política de disparo ao mais cedo não é uma regra obrigatória, ou seja, ao sensibilizar uma transição o disparo pode ser imediato ou não.

Na figura 3.9 após o disparo da transição t_1 , a ficha está no lugar O_2 no estado não reservado. A partir deste instante, podemos decidir disparar t_2 a qualquer momento. Se

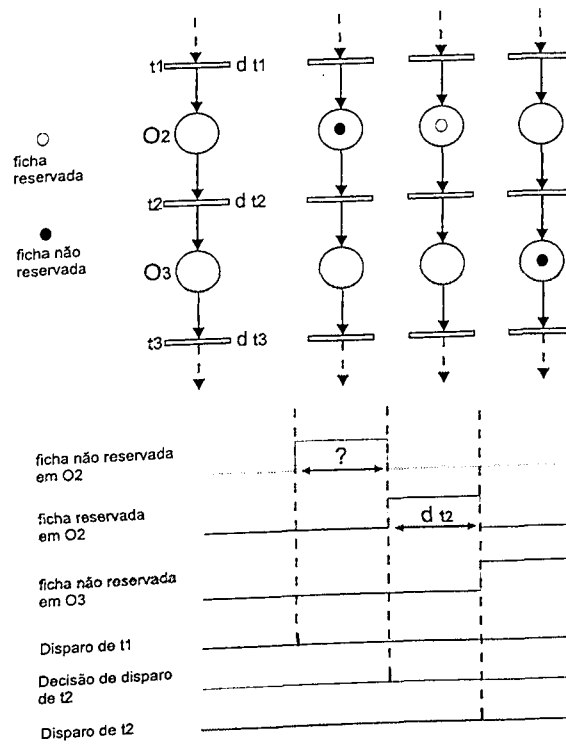


Figura 3.9: Evolução Dinâmica da Rede de Petri t-temporizada

após uma duração arbitrária, duração de espera na qual a ficha do lugar O_2 está no estado não reservado, decidirmos disparar t_2 , então a ficha ficará no estado reservado. Após uma duração d_{t_2} (temporização associada à t_2), a transição é efetivamente disparada, a ficha é retirada de O_2 e uma ficha no estado não reservado é depositada no lugar O_3 .

Nos sistemas de produção por lotes, a operação de transferência do lote de um reator a outro, pode ser obtida fazendo-se uma média sobre os possíveis valores de tempo de transferência do lote e atribuí-la à temporização da transição associada. As durações de transferências são eventos detectáveis e não controláveis, ou seja, uma vez iniciada a operação de transferência o que se deve fazer é esperar que ela termine, sendo que o término de operação é detectado no sistema real através de sensores. Este modelo não é suficiente para representar o funcionamento do sistema de produção por lotes porque a tomada de decisão de quanto tempo uma ficha deve esperar no estado não reservado num lugar não está de forma explícita. Isto quer dizer que o modelo representa de forma explícita o tempo de transferência dos lotes associados às transições, mas não representa as

durações das operações de tratamento dos lotes associadas aos lugares. Assim, considera-se a utilização de um modelo p-t-temporizado.

3.2.2 Modelo p-t-temporizado

Neste modelo as temporizações associam-se aos lugares indicando o tempo de espera de um lote num determinado reator até que ele seja liberado para ser transferido a outro reator. Esta característica é definida nas redes p-temporizadas, mas como vamos preservar as durações associadas às transições, então utilizaremos a rede p-t-temporizada.

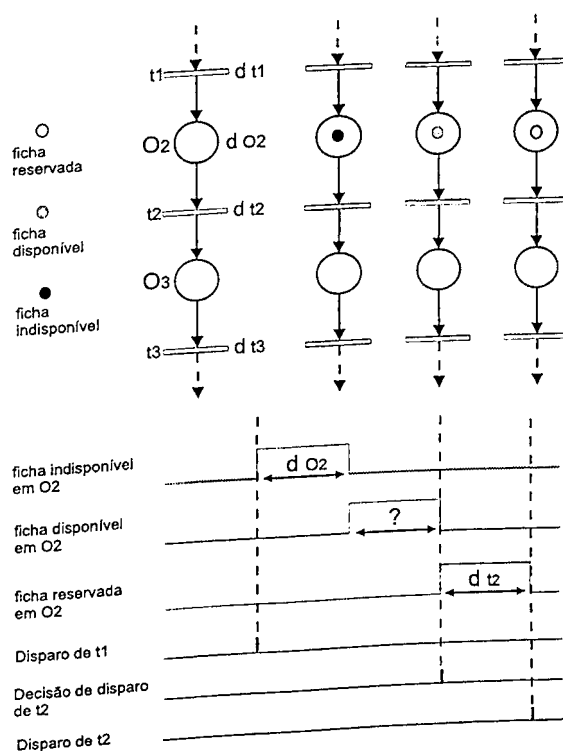


Figura 3.10: Evolução Dinâmica da Rede de Petri p-t-temporizada

Considerando a figura 3.10, após o disparo da transição t_1 , uma ficha é depositada no lugar O_2 no estado indisponível. Após a duração d_{O_2} (temporização associada à O_2), a ficha passa para o estado disponível. A partir deste instante t_2 é sensibilizada. Se após uma duração arbitrária, duração de espera na qual a ficha do lugar O_2 está no estado disponível, decidirmos disparar t_2 , então a ficha passa para o estado reservado. Após uma

duração d_{t_2} (temporização associada à t_2), a transição é efetivamente disparada, a ficha é retirada de O_2 , e uma ficha no estado indisponível é depositada em O_3 .

Neste modelo poderíamos optar pelo disparo ao mais cedo. Mas, isto iria limitar a evolução do sistema no nível de flexibilidade, uma vez que não poderíamos por exemplo retardar a execução de uma operação menos urgente associando uma duração de espera no lugar de entrada da transição correspondente. Portanto, neste modelo optou-se por não restringir o disparo ao mais cedo. Porém, com uma política de tempo de espera para disparar uma transição baseada em temporizações, certamente nos levaria a tempo de retardo indefinido. Assim, o mais realístico é considerar um valor mínimo de espera e um valor máximo de espera que pode ser modelado pelas redes de Petri p-temporais.

3.2.3 Modelo p-temporal

Nos sistemas de produção por lotes, os reatores são responsáveis pelo tratamento dos lotes. O tempo de tratamento de um lote pode ser variado até um certo limite, ou seja, para um tratamento específico existe um tempo mínimo e um tempo máximo em que o lote pode permanecer no reator sem comprometer a qualidade do tratamento. O modelo p-temporal associa aos lugares um intervalo de tempo que possui uma borda mínima antes da qual o lote não é tratado e uma borda máxima após a qual o lote fica danificado.

Na figura 3.11 após o disparo da transição t_1 , a ficha está no lugar O_2 no estado indisponível na data δ_0 . A partir desta data, após uma duração de (d_{o2min}) correspondente à borda mínima do intervalo estático do lugar O_2 , a ficha passa para o estado disponível na data $\delta_1 = \delta_0 + (d_{o2min})$ e podemos decidir disparar t_2 a qualquer momento. Se após uma duração arbitrária, duração de espera na qual a ficha do lugar O_2 está no estado disponível, decidirmos disparar t_2 , então a ficha é retirada de O_2 e uma ficha no estado indisponível é depositada no lugar O_3 na data δ_2 , com $\delta_1 \leq \delta_2 \leq \delta_0 + (d_{o2max})$.

A figura 3.11 mostra um modelo p-temporal. Este modelo se adequa bem aos sistemas de produção por lotes pelo fato de que os tempos de tratamentos podem ser variados e decididos pelo usuário baseando-se no seu conhecimento sobre o sistema real. Em outras

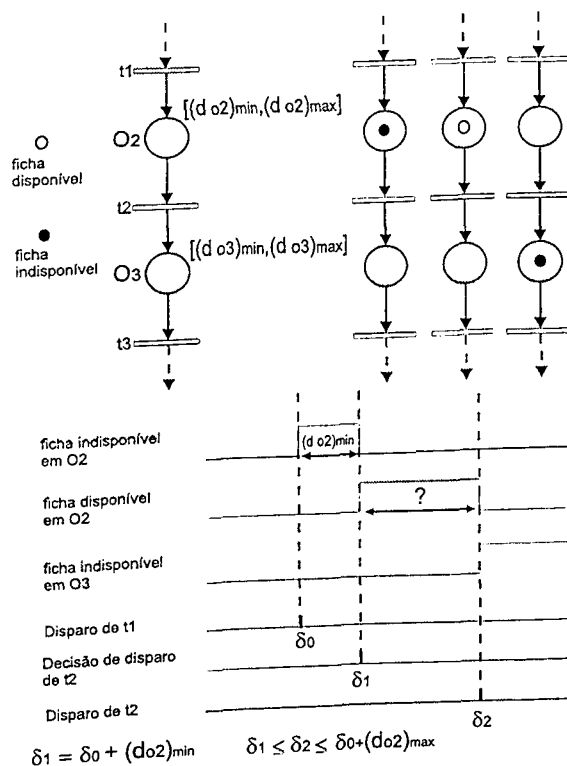


Figura 3.11: Evolução Dinâmica da Rede de Petri p-temporal

palavras, o usuário sabe com certa margem o quanto se pode aumentar o valor da borda mínima ou o quanto se pode diminuir o valor da borda máxima de um intervalo de tempo associado a um lugar sem comprometer o lote a ser tratado.

Como queremos manter as temporizações associadas às transições que representam a duração da operação de transferência de um lote, então o modelo simplesmente p-temporal não é suficiente. Portanto, utilizaremos o modelo de rede de Petri p-temporal t-temporizado (Julia *et al.* 1998).

3.2.4 Modelo p-temporal t-temporizado

Definição: Uma rede de Petri p-temporal t-temporizada é uma tripla

$\langle R, I_p, tempo-t \rangle$ tal que,

- R é uma rede de Petri marcada

- I_p é uma aplicação definida por:

$$I_p : P \rightarrow (Q^+ \cup 0) \times (Q^+ \cup \infty)$$

$$p_i \rightarrow I_{p_i} = [a_i, b_i] \text{ com } 0 \leq a_i \leq b_i$$

- tempo- t é uma aplicação de tempo nas transições dentro de um conjunto de números reais positivos ou nulos

Se t é uma transição então, tempo- $t(t) = d_t$ é a temporização associada à t .

$I_{p_i} = [a_i, b_i]$ define o intervalo estático durante o qual uma ficha está dentro do lugar p_i : a_i é a duração mínima antes da qual a ficha de p_i está no estado indisponível; b_i é a duração máxima antes da qual a ficha de p_i deve ser reservada.

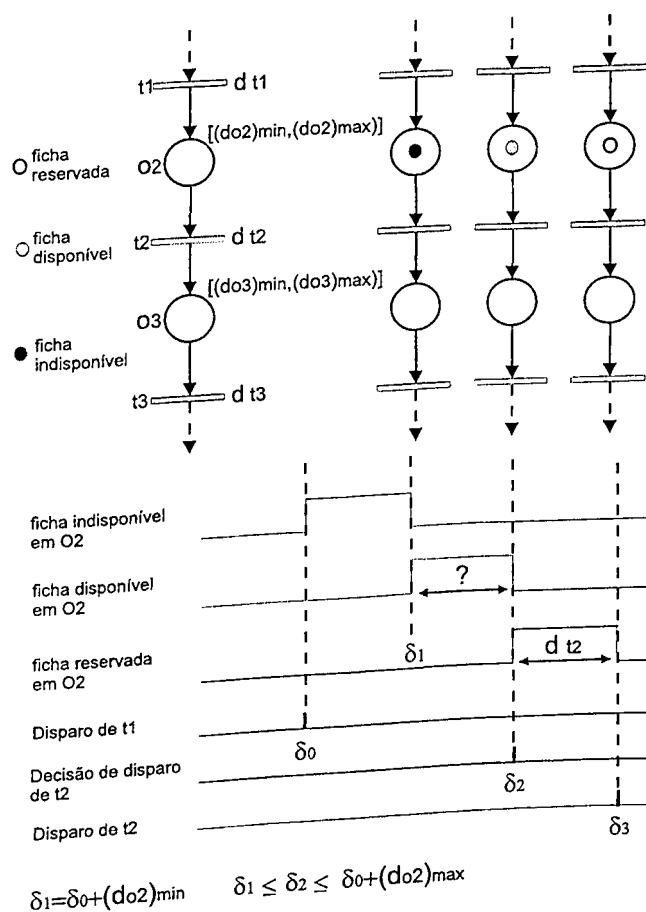


Figura 3.12: Evolução Dinâmica da RdP p-temporal t-temporizada

A figura 3.12 ilustra o comportamento dinâmico de parte de uma rede p-temporal

t-temporizada. O lugar o_2 possui um intervalo estático do tipo: $[(d_{o_2})_{min}; (d_{o_2})_{max}]$. Após o disparo da transição t_1 , a ficha é depositada dentro do lugar o_2 na data δ_0 no estado indisponível. Na data $\delta_1 = \delta_0 + (d_{o_2})_{min}$, a ficha passa para o estado disponível. A partir deste instante, a ficha de o_2 pode ser reservada para o disparo de t_2 porque a ficha passou para o estado disponível. Na data δ_2 , com $\delta_1 \leq \delta_2 \leq \delta_0 + (d_{o_2})_{max}$, decidimos disparar t_2 e a ficha passa para o estado reservado. Na data $\delta_2 + d_{t_2}$ com d_{t_2} sendo a temporização associada a transição t_2 , a transição é efetivamente disparada, a ficha é retirada de o_2 , e uma ficha que se encontra no estado indisponível é depositada dentro de o_3 .

As bordas a_i e b_i do intervalo estático associado a cada lugar da rede consistem nas datas “ao mais cedo” e nas “datas ao mais tarde” na qual as fichas devem permanecer num determinado lugar. No nosso modelo, isto representa um par de restrições para a habilitação das transições. Seja por exemplo, p_i e p_j os dois lugares de entrada de uma transição t . Seja δ_i a data em que aparece uma ficha dentro do lugar p_i . A restrição I_{p_i} impõe a habilitação de t dentro do intervalo $[\delta_i + a_i, \delta_i + b_i]$. Seja δ_j a data em que aparece uma ficha dentro do lugar p_j . A restrição I_{p_j} impõe a habilitação de t dentro do intervalo $[\delta_j + a_j, \delta_j + b_j]$. t deve ser habilitada dentro do intervalo obtido pela intersecção de $[\delta_i + a_i, \delta_i + b_i]$ com $[\delta_j + a_j, \delta_j + b_j]$ porque as duas fichas que se encontram dentro dos lugares de entrada de t devem estar no estado disponível para serem reservadas para o disparo da transição de saída. Se esta intersecção é nula, as restrições são contraditórias. Na prática isto faz com que uma restrição (ao menos) deve estar violada. Veremos mais tarde com mais detalhes a noção de conflito para uma rede p-temporal t-temporizada bem como o algoritmo do “jogador” de uma rede de Petri p-temporal t-temporizada.

No que se refere à semântica dos intervalos estáticos associados às operações de tratamento no modelo de rede p-temporal t-temporizada, podemos verificar que as bordas destes intervalos representam os valores críticos associados aos tempos de tratamento de um lote. Isto é, as bordas dos intervalos estáticos de cada lugar da rede representam a flexibilidade existente no sistema em relação aos tempos de tratamento de um lote. Por exemplo, se consideramos que um lugar da rede possui intervalo estático $[2, 5]_s$, isto nos

implicará que quando o lote chegar no reator associado a este tratamento ele deverá esperar no mínimo 2 unidades de tempo e no máximo 5 unidades de tempo para que se faça o tratamento. Caso o lote demore mais do que o permitido, o mesmo poderá ser prejudicado (lote de leite fervendo demais, por exemplo).

3.3 Restrições Temporais e Sistemas de Produção por Lotes

No cálculo do escalonamento previsional, para obtermos uma boa consistência entre o sistema de produção real e o respectivo modelo, associamos a cada restrição um intervalo de tempo. Com isto, verifica-se que a rede certamente estará sujeita ao aparecimento de conflitos dentro destes intervalos temporais. Ao tentarmos resolver os conflitos, aparecerão várias soluções possíveis. Uma maneira de escolhermos a melhor sequência de operações a ser seguida é fazer todos os grafos de eventos possíveis dando a ordem de utilização dos reatores, e escolhendo a melhor sequência. O resultado desta técnica envolve muito cálculo algébrico e dificilmente explorável no nível da pilotagem em tempo real do sistema. O grafo de eventos não permite mudar a ordem de execução das operações dos reatores (o grafo de eventos é rígido). Se adotarmos uma política de disparo “ao mais cedo” das transições da rede, minimiza-se o espaço de busca de soluções.

Afim de reduzir o espaço de busca total com um grau de flexibilidade susceptível de tornar a política de controle mais robusta, associaremos a cada lugar um intervalo min/max . Na seção 3.2.4 definimos a rede de Petri p-temporal t-temporizada que possui um intervalo estático em cada lugar do modelo.

3.3.1 Restrições Temporais para uma Restrição Kanban

O processo de escolha do intervalos temporais estáticos no lugar Kanban é baseada numa heurística, cuja técnica é atribuir para a borda mínima o valor 0 e para a borda máxima

um valor que dependerá da liberação “ao mais tarde” do recurso necessário para o disparo da transição de saída do lugar Kanban. Isto é, como a transição de saída de cada lugar Kanban também depende de um recurso estar disponível para que ela seja sensibilizada, devemos verificar quanto tempo “ao mais tarde” este recurso estará liberado. Se o recurso envolvido não for um caso típico de compartilhamento de recurso (isto é, um recurso que pode ser solicitado por mais de uma receita), então ele estará disponível sempre e conseqüentemente a ficha do lugar Kanban pode esperar qualquer tempo para ser utilizada (normalmente escolhe-se um tempo mínimo de espera). Mas, quando o recurso envolvido é solicitado por mais de uma receita, nem sempre ele estará imediatamente disponível para sensibilizar a transição de saída do lugar Kanban.

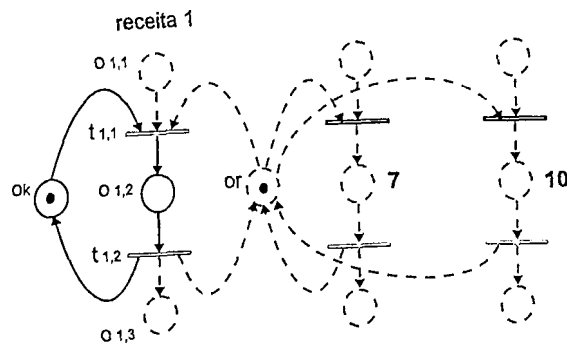


Figura 3.13: Exemplo 1 de Restrições Temporais Associadas à uma Restrição Kanban

Considerando que na figura 3.13 o reator o_r seja utilizado por outra receita antes de ser utilizado pela *receita1* gastando 10 unidades de tempo para ser liberado, então, o intervalo estático do lugar Kanban o_k será

$$[0; 10]_s$$

Se o recurso o_r fosse solicitado apenas pela *receita1*, então poderíamos escolher como borda máxima para o lugar Kanban o_k um valor menor do que 10 dependendo apenas da escolha do usuário que tem informações do sistema de produção.

3.3.2 Restrições Temporais para uma Restrição de Recurso

O intervalo estático associado aos lugares recurso O_r é denotado por $[(d_{o_r})_{min}; (d_{o_r})_{max}]_S$. Assim que uma ficha entra num lugar de recurso o_r , o mesmo é disponibilizado para ser utilizado por alguma receita, portanto, consideraremos como borda mínima dos intervalos estáticos destes lugares como sendo $(d_{o_r})_{min} = 0$. E ainda, sabendo que a ficha que se encontra no lugar o_r pode aguardar um tempo ilimitado sem que haja problema, então consideraremos como borda máxima dos intervalos estáticos destes lugares como sendo $(d_{o_r})_{max} = +\infty$.

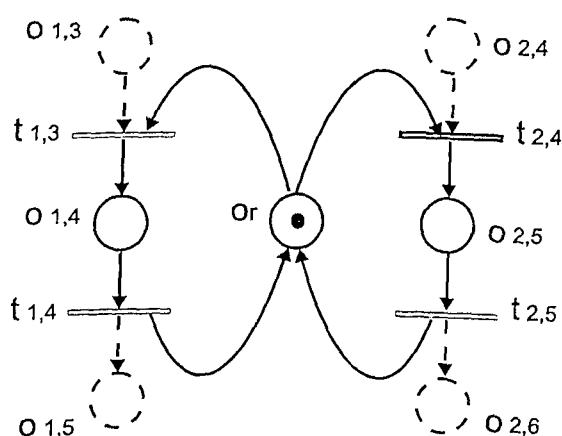


Figura 3.14: Restrições Temporais Associadas à uma Restrição de Recurso

O estado inicial do ciclo para a restrição de recurso representada pela figura 3.14 é tal que o recurso de entrada do ciclo se encontra dentro do lugar o_r no estado não reservado.

O intervalo estático associado ao recurso o_r é $[0; +\infty[_S$. Nesta situação tanto o lugar $O_{1,4}$ quanto o lugar $O_{2,4}$ poderão utilizar o recurso, cabendo ao “jogador” decidir a melhor opção de modo a obter uma sequência admissível.

Capítulo 4

Princípio Geral para o Escalonamento Cíclico

4.1 RdP e Mecanismo de Ajuda à Tomada de Decisão

Se considerarmos a sequência de operações propostas no nível de um sistema de produção por lotes, então as redes de Petri são modelos bem adaptados, pois, permitem a modelagem dos conflitos existentes em toda sua complexidade. Em outras palavras, elas permitem através de uma representação simples e compacta, enumerar todas as possibilidades de escalonamento das diversas operações do sistema.

Por outro lado, os mecanismos de ajuda à tomada de decisão na presença de um conjunto de restrições temporais explícitas que uma rede de Petri pode representar se resume às simulações. É claro que os mecanismos lógicos de alocação de recursos são formalmente especificados e as situações de conflitos (situações dentro das quais decisões são tomadas) são evidenciadas (Atabakhche 1987). Portanto, as situações de conflitos são analisadas num tempo crescente, ou seja, os primeiros conflitos que aparecem são resolvidos primeiros independente da sua data de ocorrência.

Os mecanismos lógicos utilizados nas diversas técnicas de Inteligência Artificial não abordam a noção de conflito com a mesma riqueza que as redes de Petri, mas eles permitem

construirmos os mecanismos de decisão muito eficazes utilizando os procedimentos de propagação de restrições e de retrocessos ("backtracking").

Da necessidade de resolver os conflitos existentes, salientamos que as decisões tomadas devem essencialmente estar baseadas nas restrições temporais para exercer uma influência positiva sobre a produção, o que justifica a utilização de um modelo temporal/temporizado com análise prévia a fim de que uma certa consistência global possa ser utilizada na resolução de um conflito local.

Isto leva à conclusão de que para o escalonamento de sistemas de produção por lotes, será interessante utilizar uma técnica híbrida que combina as redes de Petri e técnicas de Inteligência Artificial. Utilizaremos o formalismo das redes de Petri (p-temporal t-temporizada) para colocar em evidência as situações de conflito. Com a definição dos intervalos para cada restrição, ao aparecerem os conflitos, decisões serão tomadas de maneira local por um "jogador" de rede de Petri p-temporal t-temporizada, que é um mecanismo de inferência especializado que ordena no tempo os eventos que irão ocorrer durante um ciclo de produção, e em seguida evolui-se a rede de modo a obter uma sequência admissível que irá respeitar a sequência de restrições associadas à rede. Associado ao mecanismo de tomada de decisão, utiliza-se um mecanismo de retrocesso que é acionado toda vez que for necessário retrocedermos em eventos passados de modo a mudar valores que poderão influenciar no estado presente.

4.2 Noção de Conflito numa RdP p-temporal t-temporizada

As redes de Petri permitem representar os conflitos existentes num sistema de produção por lotes dentro de toda sua complexidade, ou seja, permitem enumerar todas as possibilidades de escalonamento das operações. Nas redes de Petri autônomas, o tempo não é representado de maneira explícita e a informação importante encontrada por elas são as operações que podem ser executadas em paralelo e quais são as operações que se ex-

cluem mutuamente (conflito existente entre as transições em conflito para uma marcação). Quando o tempo é representado de maneira explícita, o problema é mais complexo, pois o estado do sistema não é simplesmente uma marcação corrente. A noção de conflito pode mudar segundo o modelo de rede de Petri utilizado. Inicialmente, daremos algumas definições básicas e depois apresentaremos a noção de conflito para uma rede de Petri p-temporal t-temporizada mostrado em (Julia 1997b) que é o modelo a partir do qual será definido o escalonamento das operações do sistema.

4.2.1 Conflito para uma RdP Autônoma

Conflito Estrutural: *Duas transições $t1$ e $t2$ estão em conflito estrutural se e somente se elas têm ao menos um lugar de entrada em comum:*

$$\exists p / Pre(p, t1).Pre(p, t2) \neq 0$$

Conflito Efetivo: *Duas transições $t1$ e $t2$ estão em conflito efetivo para uma marcação M se e somente se elas estão em conflito estrutural e se:*

$$M \geq Pre(., t1)$$

$$M \geq Pre(., t2)$$

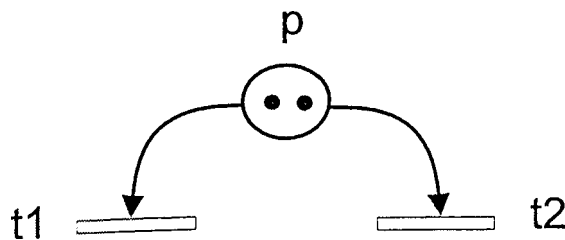


Figura 4.1: Noção de Conflito para uma Rede de Petri Autônoma

Se há um conflito estrutural e a marcação m é tal que as duas transições sejam sensibilizadas, então teremos mais de uma alternativa a considerar. Consideremos por exemplo a rede autônoma da figura 4.1. Como temos duas fichas dentro do lugar de entrada p , temos uma decisão irreversível a tomar: disparar duas vezes t_1 , ou disparar duas vezes t_2 , ou disparar uma vez t_1 e uma vez t_2 . Se a rede de Petri descreve uma sequência de processos compartilhando um conjunto de recursos, teremos sequenciamentos possíveis entre os quais o jogador irá escolher.

4.2.2 Conflito para uma RdP t-temporizada

O estado de um sistema descrito pela rede de Petri t-temporizada é dado pela marcação corrente associada ao tempo gasto para os disparos das transições. Conhecendo os tempos associados às transições, deduzimos as datas de fim de disparo. A dinâmica não será de fato completamente definida até que as fichas estejam reservadas, sensibilizando as transições (disparo “ao mais cedo”), como mostrado na figura 3-9 do capítulo 3. Nesta restrição, a duração pendente onde uma ficha de um lugar p pode estar no estado não reservado antes da habilitação de uma transição t , pertence ao intervalo $[0; +\infty[$. Portanto, para não termos de considerar uma sequência de conflitos existentes num intervalo, geralmente, para os problemas de escalonamento modelados pelas redes de Petri t-temporizadas, utiliza-se a estratégia de disparo “ao mais cedo” das transições (Lee and DiCesare 1994b);(Lee and DiCesare 1994a);(Azzopardi and Loyd 1994b). Desde que uma ficha sensibilize uma transição t , a ficha é imediatamente reservada para o disparo desta transição. Se considerarmos somente as fichas não reservadas de um modelo t-temporizado sob o aspecto de uma estratégia de disparo “ao mais cedo”, então a definição de *conflito efetivo* das redes de Petri autônomas se adapta para as redes de Petri t-temporizadas.

4.2.3 Conflito para uma RdP p-temporal t-temporizada

Na noção de conflito para uma rede p-temporal t-temporizada, os conflitos para um recurso são considerados para um intervalo de tempo e não somente em um instante preciso. Portanto, é possível gerar sequências para as quais as transições não são necessariamente disparadas assim que as fichas estejam presentes nos lugares de entrada. Com as datas de disponibilidades das fichas, podemos variar de modo contínuo a data de sensibilização das transições.

Em (Julia 1997b) definiu-se a noção de conflito de uma rede de Petri p-temporal t-temporizada que será dada a seguir.

Intervalo de Visibilidade de uma Ficha num Lugar: *Um intervalo de visibilidade $[(\delta_p)_{min}; (\delta_p)_{max}]_V$ associado à uma ficha de um lugar p de uma rede p-temporal t-temporizada define a data “o mais cedo” $(\delta_p)_{min}$ à partir da qual a ficha que se encontra no lugar p torna-se disponível e pode ser reservada para o disparo de uma transição t de saída do lugar p . Tal associação define também a data “o mais tarde” $(\delta_p)_{max}$ após a qual se tem a “morte” da ficha que, a partir de então, não pode mais ser utilizada para o disparo de nenhuma transição.*

O modelo de Khansa (Khansa et al. 1996) fala sobre a “morte” de uma ficha de um lugar p , que não poderá habilitar nenhuma transição de saída deste lugar.

Intervalo de Sensibilização de uma Transição: *Se uma transição tem n lugares de entrada que têm, cada um, uma ou várias fichas, então um intervalo de sensibilização desta transição $[(\theta)_{min}; (\theta)_{max}]_t$ é obtido escolhendo-se, para cada um dos n lugares, uma marca e o seu intervalo de visibilidade, bem como fazendo-se a intersecção daqueles intervalos de visibilidade.*

Consideremos a rede da figura 4.2. Se os intervalos de visibilidade associados às fichas de p_1 e de p_2 são

$$[(\delta_{p_1})_{min}; (\delta_{p_1})_{max}] = [3; 6]_V$$

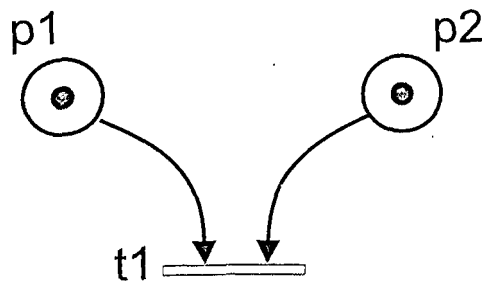


Figura 4.2: Exemplo 1 de um Intervalo de Sensibilização de uma Transição

$$[(\delta_{p2})_{min}; (\delta_{p2})_{max}] = [4; 7]_V$$

então o intervalo de sensibilização de $t1$ é

$$[(\theta_{t1})_{min}; (\theta_{t1})_{max}]_{t1} = [3; 6]_V \cap [4; 7]_V = [4; 6]_{t1}$$

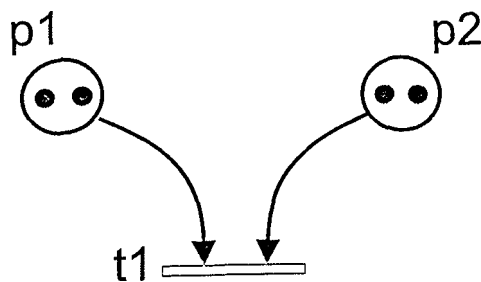


Figura 4.3: Exemplo 2 de um Intervalo de Sensibilização de uma Transição

Consideremos a rede da figura 4.3. Os intervalos de visibilidade são:

$$[(\delta_{p1})_{min}; (\delta_{p1})_{max}]_1 = [3; 6]_V$$

$$[(\delta_{p1})_{min}; (\delta_{p1})_{max}]_2 = [8; 15]_V$$

$$[(\delta_{p2})_{min}; (\delta_{p2})_{max}]_1 = [4; 7]_V$$

$$[(\delta_{p2})_{min}; (\delta_{p2})_{max}]_2 = [5; 10]_V$$

Como temos dois intervalos de visibilidade para cada lugar, devemos analisá-los.

Na figura 4.4, se fizermos a intersecção de $[(\delta_{p1})_{min}; (\delta_{p1})_{max}]_1$ e de $[(\delta_{p2})_{min}; (\delta_{p2})_{max}]_1$, então o intervalo de sensibilização é $[4; 6]$, e se fizermos a intersecção de $[(\delta_{p1})_{min}; (\delta_{p1})_{max}]_2$ e de $[(\delta_{p2})_{min}; (\delta_{p2})_{max}]_2$ tem-se o intervalo de sensibilização $[8; 10]$. Esta é uma sequência de disparo possível.

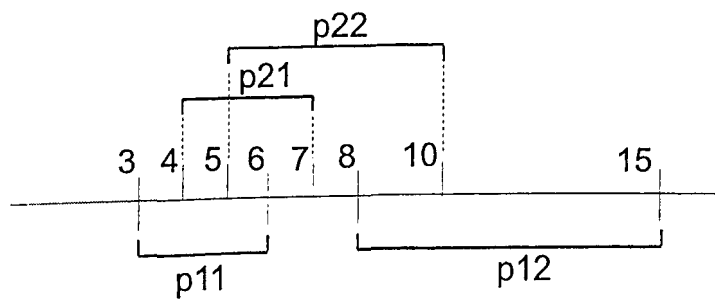


Figura 4.4: Sequência dos Possíveis Intervalos

Se fizermos a intersecção de $[(\delta_{p1})_{min}; (\delta_{p1})_{max}]_1$ e de $[(\delta_{p2})_{min}; (\delta_{p2})_{max}]_2$, então o intervalo de sensibilização é $[5; 6]$, e se fizermos a intersecção de $[(\delta_{p1})_{min}; (\delta_{p1})_{max}]_2$ e de $[(\delta_{p2})_{min}; (\delta_{p2})_{max}]_1$ tem-se o intervalo de sensibilização nulo, o que significa dizer que ocorrerá morte em uma das fichas de $p1$ ou de $p2$. Esta não é uma sequência de disparo possível.

Assim, como nosso objetivo é a busca de uma sequência admissível que respeite as restrições, o primeiro intervalo de sensibilização de $t1$ será $[4; 6]$ e o segundo $[8; 10]$.

Intervalo de Tempo de Conflito: *Se duas transições $t1$ e $t2$ estão em conflito estrutural então o intervalo de tempo de conflito daquelas transições é obtido fazendo-se a intersecção dos intervalos de sensibilização das transições $t1$ e $t2$. Durante o intervalo de tempo obtido, $t1$ e $t2$ estão em conflito efetivo em relação à marcação. Fora deste intervalo, elas não estão mais em conflito efetivo.*

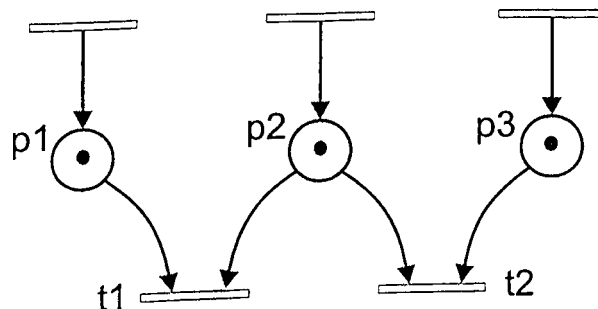


Figura 4.5: Noção de Conflito para uma Rede de Petri p-temporal t-temporizada

Consideremos o exemplo da figura 4.5. Os intervalos estáticos associados aos lugares

são:

$$[(d_{p1})_{min}; (d_{p1})_{max}] = [1; 6]_S,$$

$$[(d_{p2})_{min}; (d_{p2})_{max}] = [0; 7]_S,$$

$$[(d_{p3})_{min}; (d_{p3})_{max}] = [2; 6]_S.$$

Na data 0 uma ficha chega em $p1$, na data 2 uma ficha chega em $p3$, e na data 3 uma ficha chega em $p2$. Nós podemos deduzir os intervalos de visibilidade associados às fichas dos lugares $p1$, $p2$ e $p3$:

$$[(\delta_{p1})_{min}; (\delta_{p1})_{max}] = [1 + 0; 6 + 0]_S = [1; 6]_V,$$

$$[(\delta_{p2})_{min}; (\delta_{p2})_{max}] = [0 + 3; 7 + 3]_S = [3; 10]_V,$$

$$[(\delta_{p3})_{min}; (\delta_{p3})_{max}] = [2 + 2; 6 + 2]_S = [4; 8]_V.$$

Os intervalos de sensibilização são $[1; 6]_V \cap [3; 10]_V = [3; 6]_{t1}$ para $t1$ e $[3; 10]_V \cap [4; 8]_V = [4; 8]_{t2}$ para $t2$. O intervalo de conflito associado à dupla $(t1, t2)$ é $[3; 6]_{t1} \cap [4; 8]_{t2} = [4; 6]_C$.

4.3 Resolução de Conflito para o Escalonamento

4.3.1 Obtenção de um Estado de Conflito

Logo que uma ficha sensibiliza uma transição que está em conflito estrutural com outras transições, o sistema de decisão do “jogador” trabalha sob um horizonte temporal mínimo que é o intervalo de visibilidade da ficha avaliada. Depois compara-se as sequências possíveis sob um certo horizonte.

Se o intervalo de visibilidade da ficha é $[(\delta_p)_{min}; (\delta_p)_{max}]_V$, então, calcula-se neste intervalo as possíveis datas de chegada das outras fichas nos lugares de entrada das transições em conflito estrutural com a transição sensibilizada.

Em função das diversas datas de chegada, é possível definir os diversos intervalos de visibilidade e de estabelecer os possíveis intervalos de conflito.

Um problema é quando certas transições a serem habilitadas estiverem elas mesmas em conflito com outras transições e que o cálculo dos intervalos de visibilidade para resolver um conflito depende da resolução de outros conflitos. Este problema pode aparecer mesmo

no caso de uma estratégia de resolução de conflitos numa ordem temporal crescente. De fato, um estado é definido para uma marcação e para o conjunto de intervalos de visibilidade do conjunto de fichas desta marcação. Para cada transição, o fato da borda mínima do intervalo de sensibilização da transição t_i ser menor que a de uma transição t_j não garantirá o disparo de t_i primeiro. Para o caso da borda máxima de sensibilização da transição t_i ser menor que a borda mínima do intervalo de sensibilização de uma transição t_j , deve-se geralmente dar prioridade à habilitação de t_i antes de t_j o que nem sempre é possível.

Se duas transições t_1 e t_2 estão em conflito estrutural e se uma delas é sensibilizada para uma ficha que se encontra dentro do lugar de entrada p na data $(\delta_p)_{min}$ (borda inferior do intervalo de visibilidade da ficha que sensibiliza a transição), então utiliza-se o seguinte procedimento para estabelecer os intervalos de visibilidade e, conseqüentemente, o estado de conflito:

Etapa 1: Simular num horizonte temporal mínimo $[(\delta_p)_{min}; (\delta_p)_{max}]$, para o disparo "ao mais cedo", a evolução das fichas da rede nas partes que não possuem escolha, ou seja, que não há conflito estrutural. Para as fichas que evoluem nas partes que possuem escolha, considera-se para o disparo ao mais cedo o pior dos casos, onde os recursos são utilizados por outros roteiros primeiro.

Etapa 2: Calcular para as fichas que se encontram dentro dos lugares de entrada das transições em conflito estrutural os intervalos de visibilidade em função das datas de chegada das fichas e dos intervalos estáticos associados aos lugares.

Após este procedimento, os diversos intervalos de visibilidade das fichas são definidos e os intervalos de conflito podem ser calculados.

Considerando a rede da figura 4.6, na data $\delta_1 = 2$ uma ficha chega no lugar o_{31} e uma outra no lugar o_{22} . Os intervalos estáticos associados aos lugares são:

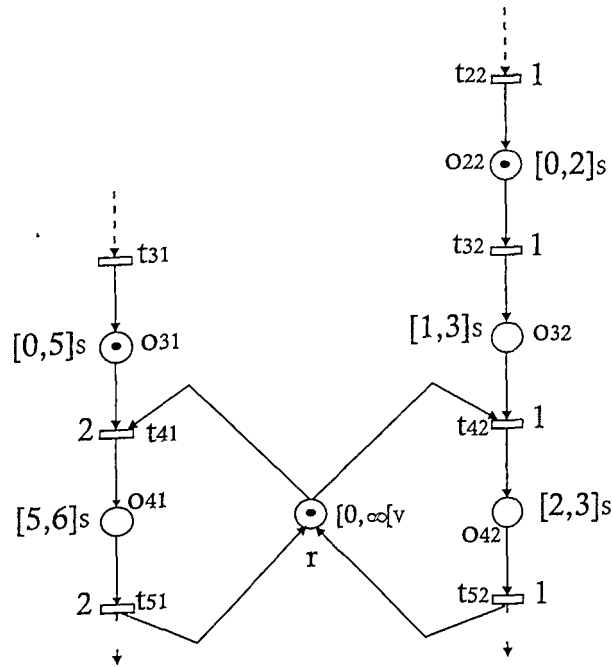


Figura 4.6: Exemplo de Cálculo dos Intervalos de Visibilidade

$$[(d_{o_{31}})_{min}; (d_{o_{31}})_{max}] = [0; 5]_s$$

$$[(d_{o_{22}})_{min}; (d_{o_{22}})_{max}] = [0; 2]_s$$

$$[(d_{o_{32}})_{min}; (d_{o_{32}})_{max}] = [1; 3]_s$$

A ficha de recurso do lugar r possui um intervalo estático igual a:

$$[(\delta_r)_{min}; (\delta_r)_{max}] = [0; +\infty]_s$$

As durações das operações de transferência dos lotes nas transições t_{41} , t_{32} e t_{42} são respectivamente: 2, 1 e 1.

As transições t_{41} e t_{42} estão em conflito estrutural. Os lugares de entrada destas transições são o_{31} e o_{32} , respectivamente, e r para ambos. O intervalo de visibilidade da ficha dentro do lugar o_{31} é:

$$[(\delta_{o_{31}})_{min}; (\delta_{o_{31}})_{max}] = [0 + 2; 5 + 2]_v = [2; 7]_v$$

O intervalo de visibilidade da ficha dentro do lugar o_{22} é:

$$[(\delta_{o_{22}})_{min}; (\delta_{o_{22}})_{max}] = [0 + 2; 2 + 2]_V = [2; 4]_V$$

Se iniciamos o disparo da transição t_{32} na data $\delta_1 = 2$ (simulação com disparo “ao mais cedo”), a ficha que se encontra no lugar o_{22} chegará no lugar o_{32} na data:

$$\delta_2 = \delta_1 + d_{t_{32}} = 2 + 1 = 3$$

e o intervalo de visibilidade desta ficha no lugar o_{32} será:

$$[(\delta_{o_{32}})_{min}; (\delta_{o_{32}})_{max}] = [1 + 3; 3 + 3]_V = [4; 6]_V$$

O intervalo de sensibilização de t_{42} é $[4; 6]_{t_{42}}$ e de t_{41} é $[2; 7]_{t_{41}}$. O intervalo de tempo de conflito associado à dupla (t_{41}, t_{42}) é então $[4; 6]_C$. Este intervalo não é nulo e não podemos ignorá-lo perante o “jogador”, portanto, temos um conflito no sentido de um modelo p-temporal t-temporizado.

Caso considerássemos que no exemplo anterior o recurso r fosse solicitado por mais de duas receitas, neste caso, teríamos que analisar a existência ou não de intervalos de conflito entre a transição t_{41} e todas as outras transições que estivessem em conflito estrutural com ela, duas a duas.

4.3.2 Estratégia de Resolução de Conflito

Vale lembrar que nosso objetivo não é a busca de uma sequência otimizada frente ao tempo de execução, mas sim a busca de uma sequência admissível com o conjunto das restrições temporais definidas anteriormente. A sequência obtida deve ser robusta à pequenas perturbações (como o atraso do início de uma operação, por exemplo) e deve poder

ser calculada em tempo real no nível da pilotagem do sistema. A heurística de busca da sequência não deve ser simples, em particular, o sistema de decisão associado ao “jogador” de rede de Petri deve escalonar as operações de modo a trabalhar num horizonte temporal mínimo definido a partir dos intervalos de visibilidade dos lotes que se encontram nos lugares de entrada das transições em conflito em vez de considerar um único instante para tomar uma decisão.

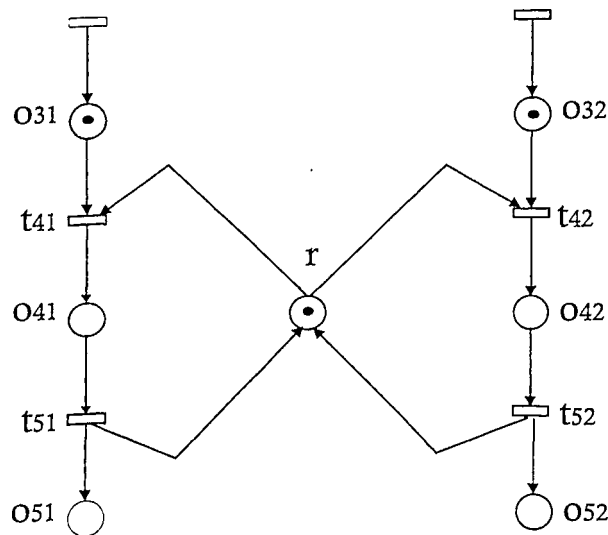


Figura 4.7: Resolução de Conflito

Para mostrar a estratégia de resolução de conflito consideramos o exemplo da figura 4.7. Os intervalos de visibilidade são:

$$\begin{aligned} [(\delta_{o_{31}})_{min}; (\delta_{o_{31}})_{max}] &= [1; 6]_V, \\ [(\delta_{o_{32}})_{min}; (\delta_{o_{32}})_{max}] &= [4; 8]_V, \\ [(\delta_r)_{min}; (\delta_r)_{max}] &= [0; +\infty]_V \end{aligned}$$

Os intervalos de sensibilização são $[1; 6]_{t_{41}}$ para t_{41} e $[4; 8]_{t_{42}}$ para t_{42} . O intervalo de conflito associado à dupla (t_{41}, t_{42}) vale $[4; 6]_C$.

Supondo que os intervalos estáticos dos lugares o_{41} e o_{42} sejam, respectivamente, $[2; 4]_S$ e $[0; 1]_S$. E ainda, que as durações das operações de transferência sejam : $d_{t_{41}} = d_{t_{51}} = d_{t_{42}} = d_{t_{52}} = 1$.

Se t_{41} é a primeira transição a ser disparada, então a data de fim de disparo de t_{51} é $(\theta_{t_{51}})_f = 1 + 1 + 2 + 1 = 5$. $(\theta_{t_{51}})_f \in [4; 8]_V$ que é o intervalo de visibilidade da ficha que se encontra dentro do lugar o_{32} . O disparo “ao mais cedo” de t_{41} é solução do problema. Analisando, teremos uma margem de 3 para o disparo de t_{42} .

Se t_{42} é a primeira transição a ser disparada, então a data de fim de disparo de t_{52} é $(\theta_{t_{52}})_f = 4 + 1 + 0 + 1 = 6$. $(\theta_{t_{52}})_f \in [1; 6]_V$ que é o intervalo de visibilidade da ficha que se encontra dentro do lugar o_{31} . O disparo “ao mais cedo” de t_{42} é solução do problema. Analisando, teremos uma margem de 0 para o disparo de t_{41} .

A primeira solução (disparo de t_{41} antes do disparo de t_{42}) é mais robusta porque fornece uma margem maior para o disparo da outra transição em conflito. A escolhemos como solução, porém não devemos descartar a outra solução. Pois, esta é uma solução aceitável que pode ser aproveitada se a sequência $t_{41}|t_{42}$ for causa de uma violação de restrição que pode aparecer mais tarde dentro da execução do “jogador”. Porém, devemos considerar que o fato de termos um grande número de restrições simultâneas a serem consideradas, será de todo modo difícil de definir um critério de escolha bem claro, e neste caso, mesmo se uma solução deve ser escolhida, as outras não deverão ser eliminadas, mas sim guardadas para eventualmente serem utilizadas posteriormente.

O mecanismo acima empenha-se em calcular uma sequência que maximize as margens temporais, uma vez que escolhe-se o disparo da transição que acarretará uma maior margem temporal para o disparo da outra transição do conflito.

Nesta dissertação, queremos obter um “jogador” de rede de Petri que faça um escalonamento previsional, mas que se adapte ao máximo ao escalonamento em tempo real. Uma vez que em tempo real, não podemos ter retrocessos e devemos ter o menor número de opções de escolha de sequências (isto é, quanto menor o número de soluções a escolher mais rápido será o sistema de decisão e execução do “jogador”), daremos prioridade de disparo para a transição sensibilizada. Isto é, verifica-se se ela pode ser disparada imediatamente, caso não seja possível, aciona-se um mecanismo de retrocesso que tentará torná-lo possível (o mecanismo de retrocesso será discutido posteriormente). Se depois

de acionado o retrocesso ainda não for possível disparar a transição sensibilizada, então ignora-se o evento de dispará-la imediatamente e prossegue-se de acordo com o calendário, os eventos seguintes. Com isto, posteriormente haverá uma reincidência de disparo desta transição e novamente o sistema de decisão será acionado.

4.4 "Jogador" de RdP p-temporal t-temporizado

Inicialmente vamos apresentar o algoritmo do "jogador" de rede de Petri p-temporal t-temporizado que é utilizado para a busca de uma sequência de operações admissíveis obtendo o conjunto das restrições a serem respeitadas.

O algoritmo possui um mecanismo de decisão que deverá ser utilizado cada vez que um conflito aparece, isto é, que um intervalo de tempo de conflito não nulo seja associado à uma dupla de transições.

Ele possui ainda um mecanismo de retrocesso ("backtrack"). Este mecanismo será utilizado ao nível de um escalonamento previewal e não para a supervisão e a pilotagem em tempo real.

Enfim, o algoritmo usa um calendário que contém uma sequência de eventos ordenados no tempo. Estes eventos são por um lado as bordas máximas e mínimas dos intervalos de visibilidade das fichas não reservadas e por outro lado as datas de fim de disparo das transições associadas às fichas reservadas.

O estado presente é caracterizado pela data presente, pela lista das fichas que estão no estado indisponível, disponível e reservado.

4.4.1 Algoritmo

Este algoritmo tem como base o algoritmo apresentado na Tese de Doutorado (Julia 1997a), utilizado para sistemas flexíveis de manufatura.

Etapa 1:

- 1.1 Disparar a transição de sincronização t_S .
- 1.2 Para a origem dos tempos $\theta \leftarrow 0$, calcular os intervalos de visibilidade de cada ficha não reservada.
- 1.3 Inserir no calendário as bordas mínimas e máximas destes intervalos.
- 1.4 Pôr todas as fichas no estado indisponível.
- 1.5 Memorizar o estado inicial da rede numa lista.

Etapa 2:

- 2.1 Considerar o primeiro evento do calendário.
- 2.2 Calcular a diferença δ entre o instante presente e o instante precedente.
- 2.3 Calcular a data presente $\theta \leftarrow \theta + \delta$.

Etapa 3: Se o primeiro evento do calendário corresponde à data de fim de disparo de uma transição:

- 3.1 Disparar a transição.
- 3.2 Verificar a caracterização de fim de ciclo. Se for verificada, então parar o algoritmo.
- 3.3 Calcular os novos intervalos de visibilidade das fichas utilizadas no disparo da transição.
- 3.4 Inserir no calendário as novas bordas mínimas e máximas destes intervalos.
- 3.5 Pôr todas as fichas utilizadas no disparo da transição no estado indisponível.
- 3.6 Memorizar o estado da rede na lista.
- 3.7 Ir à etapa 2.

Etapa 4: Se o primeiro evento do calendário corresponde à borda mínima de um intervalo de visibilidade:

- 4.1 Se a receita foi completada, ignorar a borda mínima.
 - 4.1.1 Ir à etapa 2.
- 4.2 Senão, pôr a ficha à qual é associado este intervalo no estado disponível.
- 4.3 Memorizar o estado da rede na lista.

4.4 Se uma transição diferente da transição de sincronização t_s é sensibilizada.

para as fichas disponíveis:

4.4.1 Se a transição sensibilizada não está em conflito estrutural:

4.4.1.1 Pôr as fichas que a sensibilizam no estado reservado.

4.4.1.2 Tirar do calendário as bordas máximas dos intervalos de visibilidade associados a estas fichas.

4.4.1.3 Inserir no calendário a data de fim de disparo da transição.

4.4.1.4 Memorizar o estado da rede na lista.

4.4.2 Se a transição sensibilizada está em conflito estrutural:

4.4.2.1 Chamar o sistema de decisão de conflito.

4.4.2.2 Indicar a solução escolhida para o sistema de resolução de conflito.

4.4.2.3 Se a transição sensibilizada é solução do sistema de decisão:

4.4.2.3.1 Pôr as fichas que a sensibilizam no estado reservado.

4.4.2.3.2 Tirar do calendário as bordas máximas dos intervalos de visibilidade associados a estas fichas.

4.4.2.3.3 Inserir no calendário a data de fim de disparo da transição.

4.4.2.3.4 Memorizar o estado da rede na lista.

4.4.2.4 Ir à etapa 4.4.

4.5 Ir à etapa 2.

Etapa 5: Se o primeiro evento do calendário corresponder à borda máxima de um intervalo de visibilidade, então acionar o mecanismo de retrocesso ("back-track").

O mecanismo de resolução de conflito é tal que se duas soluções são admissíveis: ou habilita-se a transição sensibilizada imediatamente, ou não habilita-se esta transição e mantém-se as fichas que a sensibiliza no estado disponível (os recursos necessários ao disparo desta transição ficam no estado disponível para, posteriormente, disparar uma outra transição).

Vale ressaltar que a caracterização de fim de ciclo será verificada no jogador toda vez que a rede atingir o seu estado inicial.

4.4.2 Mecanismo de Retrocesso

Este mecanismo é acionado em duas situações: quando uma transição sensibilizada estiver em conflito com outra transição e cujo disparo implica em violação de restrição e quando ocorrer no calendário o evento "borda máxima de um intervalo de visibilidade".

1ª situação: Quando o sistema de decisão detecta que a transição sensibilizada, ao ser disparada, imediatamente ocasiona o aparecimento de violação de restrição, verifica-se que este fato ocorre devido à chegada cedo demais da ficha no lugar de entrada da transição sensibilizada. Com isto, tenta-se atrasar a chegada da mesma de modo a resolver o problema. Este atraso é feito retrocedendo-se na receita até tornar possível o atraso necessário. Caso o retrocesso seja feito até a marcação inicial referente à receita analisada, então não será possível disparar a transição sensibilizada em questão. É importante ressaltar que os atrasos estão relacionados com o fato de podermos aumentar as bordas mínimas dos intervalos estáticos dos lugares, sendo este aumento limitado pela flexibilidade do sistema real. Na figura 4.8 temos um exemplo de retrocesso em que é possível fazer o atraso necessário para resolver um conflito.

Na figura 4.8, os intervalos de visibilidade são: $[2, 7]_V$ para o lugar O_{31} , $[0, +\infty[_V$ para o lugar r e $[6, 8]_V$ para o lugar O_{32} . Este último intervalo de visibilidade é obtido a partir da estimação da data de chegada da ficha em O_{32} . O princípio de estimação da data de chegada da ficha no lugar de entrada de uma transição em conflito estrutural consiste em:

- Considerar para a receita que contém a transição em conflito estrutural os lugares de entrada desta transição que possuem pelo menos uma ficha na data presente;
- Simular a evolução da ficha na receita de modo que para cada disparo de transição (disparo "o mais cedo possível"), obtém-se um intervalo de visibilidade que corresponde à possíveis datas de chegadas naquele lugar;

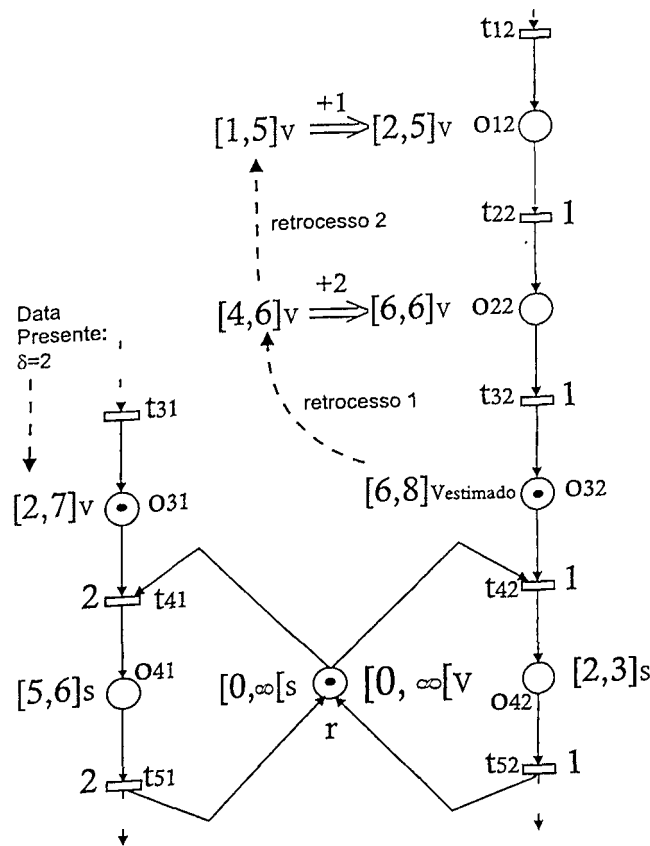


Figura 4.8: Exemplo de Uso do Mecanismo de Retrocesso

- Prosseguir a evolução até a ficha estar no lugar de entrada da transição em conflito estrutural;

Os intervalos estáticos associados aos lugares O_{41} e O_{42} são respectivamente: $[5, 6]_s$ e $[2, 3]_s$. As temporizações associadas às transições são:

$$d_{t_{41}} = 2, d_{t_{51}} = 2, d_{t_{22}} = d_{t_{32}} = d_{t_{42}} = d_{t_{52}} = 1$$

O intervalo de sensibilização de t_{41} vale:

$$[2, 7]_v \cap [0, +\infty[_v = [2, 7]_{t_{41}}$$

Isto quer dizer que a marca de O_{31} pode ser reservada para o disparo de t_{41} "ao mais cedo" à data $\delta = 2$ (borda mínima do intervalo de sensibilização) e ao mais tarde à data $\delta = 7$ (borda máxima do intervalo de sensibilização). O intervalo de sensibilização de t_{42}

vale:

$$[6, 8]_V \cap [0, +\infty[_V = [6, 8]_{t_{42}}$$

Isto quer dizer que a marca de O_{32} pode ser reservada para o disparo de t_{42} "ao mais cedo" à data $\delta = 6$ e "ao mais tarde" à data $\delta = 8$. O intervalo de conflito associado à dupla t_{41} e t_{42} vale:

$$[2, 7]_{t_{41}} \cap [6, 8]_{t_{42}} = [6, 7]_C$$

Como este intervalo não é vazio, temos um conflito efetivo no sentido de uma rede de Petri p-temporal t-temporizada. Se t_{41} é a primeira transição a ser disparada, então a data na qual o recurso r vai estar disponível de novo é $(\theta_{t_{51}})_f$ (data de fim de disparo de t_{51}). No melhor dos casos (considerando-se a borda mínima do intervalo estático associado a O_{41}), a expressão desta data é:

$$(\theta_{t_{51}})_f = 2 + 2 + 5 + 2 = 11 \notin [6, 8]_V$$

Como a transição sensibilizada não pode ser disparada imediatamente, faz-se um retrocesso. No caso de uma violação decorrente de um conflito não resolvido de uma rede de Petri p-temporal t-temporizada, a causa da violação será a chegada cedo demais de uma das duas marcas, que se encontram nos lugares de entrada das transições em conflito, em relação a outra. Por exemplo, na rede da figura 4.8, para poder disparar a transição t_{41} na data $\delta = 2$ (borda mínima do intervalo de visibilidade da marca de O_{31}), será preciso aumentar a borda máxima do intervalo de visibilidade da marca O_{32} . Para isto é preciso atrasar a data de chegada da marca de O_{32} de pelo menos 3 unidades de tempo, o que permitiria ter como novo intervalo de visibilidade da marca de O_{32} :

$$[6 + 3, 8 + 3]_V = [9, 11]_V$$

Assim teríamos como data de fim de disparo de t_{51} :

$$(\theta_{t_{51}})_f = 2 + 2 + 5 + 2 = 11 \in [9, 11]_V$$

com $[9, 11]_V$ sendo o novo intervalo de visibilidade da marca de O_{32} depois do atraso da chegada da marca de O_{32} de 3 unidades de tempo. Suponhamos que as transições t_{32} e t_{22} tenham sido disparadas "ao mais cedo" possível, durante a evolução do jogador, e que os intervalos de visibilidade da marca quando ela passou pelos lugares O_{12} e O_{22} eram $[1, 5]_V$ e $[4, 6]_V$, respectivamente. Para poder atrasar de 3 unidades, no mínimo, a chegada da marca em O_{32} , podemos aumentar, por exemplo, de 2 unidades a borda mínima do intervalo de visibilidade da marca quando ela se acha em O_{22} (o intervalo de visibilidade $[4, 6]_V$ torna-se $[4 + 2, 6]_V = [6, 6]_V$), e de 1 unidade a borda mínima do intervalo de visibilidade da marca quando ela se acha em O_{12} (o intervalo de visibilidade $[1, 5]_V$ torna-se $[1 + 1, 5]_V = [2, 5]_V$). Depois destas modificações, considerando o estado global da rede na data $\delta = 1$, o jogador prossegue na busca de uma solução considerando os novos valores das bordas mínimas dos intervalos de visibilidade. De fato, a data $\delta = 1$ representa o momento em que foi tomada a primeira decisão (disparo da transição t_{22} na data $\delta = 1$, quando o intervalo de visibilidade da ficha em O_{12} valia $[1, 5]_V$) diretamente relacionada com a violação de restrição que envolve o recurso r (morte das marcas em O_{31} ou O_{32}).

Vale lembrar que no exemplo da figura 4.8, as bordas mínimas podem se aumentadas até o valor das respectivas bordas máximas. Na prática, para poder levar em conta os possíveis atrasos, não diretamente previstos no modelo, durante o escalonamento do sistema, será de grande interesse sempre deixar uma margem mínima entre as bordas mínimas e máximas dos intervalos de visibilidade.

Desta forma observamos que utilizou-se um retrocesso "inteligente" que não opera de forma sistemática (como o de Prolog, por exemplo) de modo a buscar uma sequência admissível.

2ª situação: Quando o calendário acusar o aparecimento do evento "borda máxima de um intervalo de visibilidade", utiliza-se o mesmo mecanismo de retrocesso "inteligente" mencionado anteriormente.

O retrocesso "inteligente" retorna mais rápido às possíveis causas da violação evitando assim o problema de explosão combinatória. Princípios de mecanismos de retrocesso "inteligente" já foram apresentados em (Dechter 1986) e (Verfaillie and Schiex 1994). A idéia básica é tirar proveito das condições que conduziram a uma violação de restrição, para direcionar de modo inteligente à escolha dos pontos de retrocesso, ou para não explorar as partes do espaço de busca que vão conduzir de maneira quase certa a outras novas violações de restrições. O método tem por objetivo, achar uma variável de decisão bem apropriada para realizar o retrocesso apoiando-se na estrutura do modelo de representação do espaço de busca (que é, no caso mais comum, um grafo).

Quando um evento "borda máxima de um intervalo de visibilidade" aparece, o mecanismo de retrocesso retrocede até a última decisão diretamente relacionada com o conflito atual e busca-se uma nova solução para ele usando o mecanismo de resolução de conflito. Vale ressaltar que a cada aparecimento de um novo conflito, memoriza-se numa lista as transições envolvidas nele com seu respectivo estado de já ter sido ou não solução de um conflito anterior. Assim, podemos saber a qualquer momento quais conflitos ocorreram e quais foram as suas soluções. Quando um conflito não possui solução, ou seja, quando todas as possibilidades de disparo desta transição foram calculadas e nenhuma está livre da ocorrência de violação de restrição, então o mecanismo indica ao usuário que deve-se mudar as condições iniciais da rede para que seja solucionada a violação de restrição, e então a evolução da rede é reiniciada.

Na figura 4.9, a data presente é $\delta = 2$. Inicialmente, verifica-se o aparecimento de conflito efetivo entre as transições t_{41} e t_{42} . Portanto, memoriza-se as duas transições no estado inativo, ou seja, nenhuma ainda foi escolhida como solução de disparo. Ao ser acionado, o mecanismo de decisão verifica que não é possível disparar a transição t_{41} sem que haja violação de restrição. Então o mecanismo de retrocesso é acionado para tentar atrasar o disparo da transição t_{42} de modo a eliminar a violação de restrição prevista no disparo imediato da transição t_{41} . Verifica-se que não é possível fazer o atraso necessário, ou seja, a flexibilidade das bordas mínimas dos intervalos estáticos dos lugares O_{12} e O_{22}

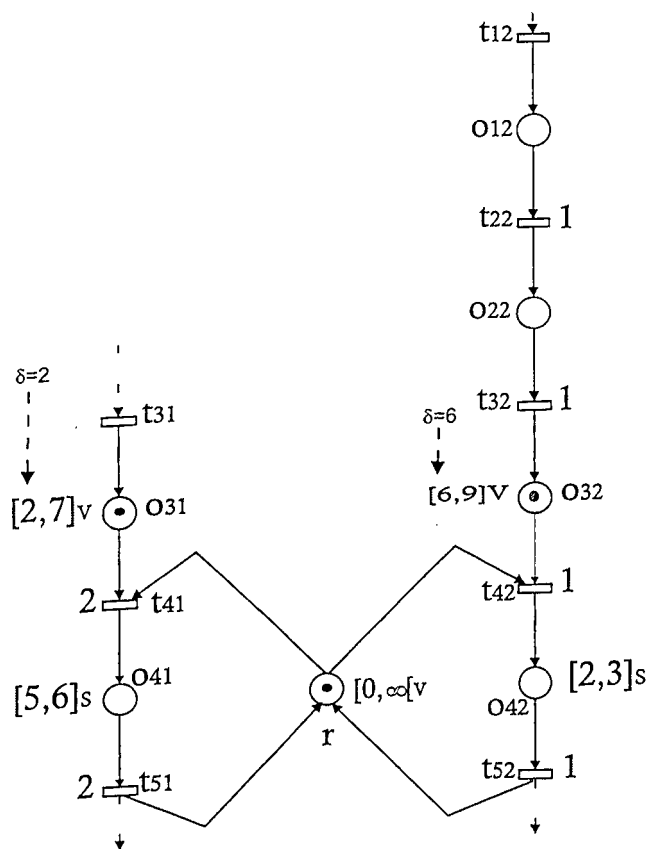


Figura 4.9: Exemplo de Mudança da Marcação Inicial

não é suficientemente grande para que a data de disparo ao mais cedo da transição t_{51} pertença ao intervalo de visibilidade do lugar O_{32} . Então, ignora-se o disparo de t_{41} na data $\delta = 2$ e prossegue-se na evolução da rede. Na data $\delta = 6$ (data de chegada da ficha no lugar O_{32}), o sistema de decisão verifica que o disparo imediato da transição t_{42} também causa violação de restrição, ou seja, a data de disparo ao mais cedo da transição t_{52} não pertence ao intervalo de visibilidade do lugar O_{31} . Nesta situação, tanto a transição t_{41} quanto a transição t_{42} não podem ser escolhidas como solução de primeiro disparo. Finalmente, o "jogador" informa ao usuário a situação apresentada, informando que as condições iniciais da rede diretamente relacionadas com a violação detectada devem ser mudadas (novas marcações iniciais). Além disso, o jogador informa de que forma esta mudança deve ser feita. Ao fazê-lo, o usuário reinicia a evolução da rede com as novas condições iniciais.

4.5 Princípio Geral da Pilotagem em Tempo Real

Para poder monitorar e pilotar um sistema de produção por lotes, deve ser possível comparar, a qualquer instante, o estado do sistema com o estado do modelo que o representa. O modelo deve considerar toda complexidade do sistema. Conseqüentemente, uma rede de alto nível mais geral que uma rede lugares/transição deve, geralmente, ser utilizada. Por exemplo, podem ser utilizadas as redes predicados/transições (Bako 1990). (Atabakhche 1987) define uma rede de Petri à decisão, particularmente bem adaptada à pilotagem em tempo real. Este modelo é derivado das redes de Petri a objetos (Sibertin-Blanc 1994). A implementação do sistema é feita através de um "jogador" de rede de Petri que é um mecanismo de inferência particular aplicado às redes de Petri (Atabakhche 1987);(Atabakhche *et al.* 1987);(Silva and Valette 1989). Este algoritmo resolve, em tempo real, todos os conflitos existentes, executando, para tanto, o escalonamento explícito ou implícito que é definido na etapa do escalonamento.

As redes de Petri servem também para vigiar e detectar as possíveis falhas do sistema. Com efeito, a rede interpretada descreve, em tempo real, o comportamento dinâmico do sistema. Ela pode ser utilizada como modelo do processo comandado no quadro do princípio clássico da *supervisão/vigilância* descrito na figura 4.10 (Combacau 1991); (Cépadueès 1991). A cada mudança significativa do estado do processo (essencialmente os finais de operações), o módulo de detecção verifica se o estado do modelo é coerente com o estado do sistema e atualiza o estado do modelo. O módulo de decisão deduz, então, quais ações iniciar, definindo, essencialmente, os inícios de operações.

Se o modelo do sistema é um grafo de eventos, o módulo de decisão não poderá mudar a ordem de execução das operações nos recursos. Se o modelo é uma rede com os conflitos não resolvidos, então é o módulo de decisão que terá dificuldades para trabalhar, de maneira coerente, com um escalonamento definido a partir de um grafo de eventos.

Um aspecto a ser levado em conta na pilotagem em tempo real é que no modelo de rede de Petri utilizado pode-se associar às transições, operações de fabricação que são indivisíveis e ininterruptíveis.

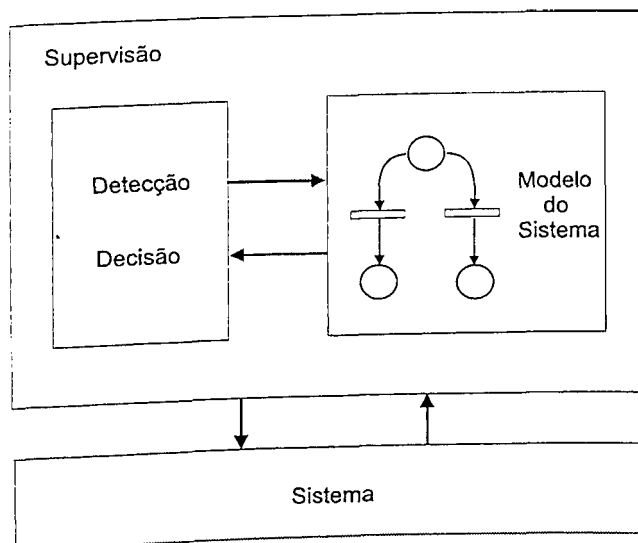


Figura 4.10: Princípio Supervisão/Pilotagem

Tais transições são do mesmo tipo mostrado na figura 4.11 que mostra a transição $t_{i,j}$. A transição $t_{i,j}$ é abreviada na sequência elementar formada de uma transição de entrada de operação $(t_{i,j})_c$, de um lugar $t_{i,j}$ descrevendo a operação em curso e de uma transição de fim de operação $(t_{i,j})_f$. Este lugar suplementar $t_{i,j}$ é um lugar substituível (segundo a regra de redução, não há modificação das propriedades da rede subjacente) (Cardoso and Valette 1997). Os eventos associados às transições $(t_{i,j})_c$ e $(t_{i,j})_f$ são de duração nula e o modelo não é mais agora p-temporal t-temporizado e sim p-temporal. Para representar a duração da operação $d_{t_{i,j}}$ que está associada à transição do modelo p-temporal t-temporizado, podemos associar o novo lugar $t_{i,j}$ ao intervalo estático

$$[d_{t_{i,j}}; d_{t_{i,j}}]$$

Como a pilotagem concerne um sistema de produção por lotes real, não conhecemos com exatidão as durações de tratamento, portanto será mais realístico escolhermos o intervalo

$$[d_{t_{i,j}} - \delta; d_{t_{i,j}} + \delta]$$

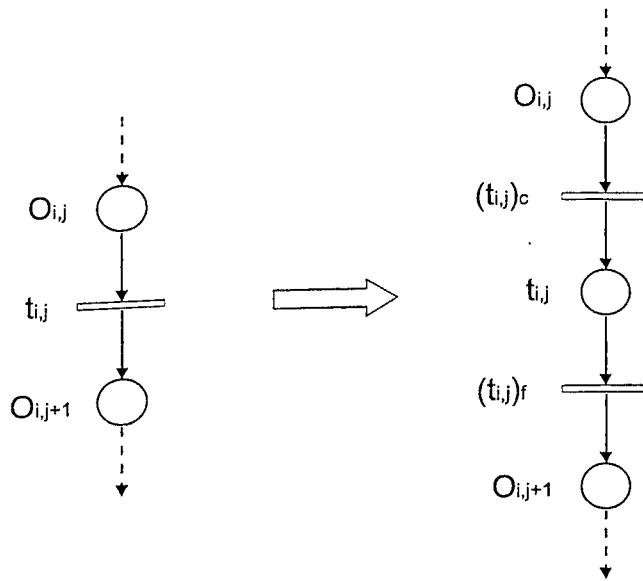


Figura 4.11: Associação de uma Operação a uma Transição

onde, o valor de δ corresponde a uma margem existente entre a duração prevista da operação e a duração real.

Como não usamos mais temporizações, as fichas se encontram ou no estado indisponível ou no estado disponível conforme a definição das redes p-temporais dada em (Khansa *et al.* 1996).

Vale ressaltar que quanto aos lugares intermediários, preserva-se os intervalos estáticos usados no nível do escalonamento previsional da rede p-temporal t-temporizada.

Na supervisão devemos obter informações do mundo exterior, isto é as mensagens enviadas aos atuadores do sistema que são associados às transições de entrada da operação, e as mensagens recebidas dos captadores do sistema que são associadas às transições de fim de operação.

Como a pilotagem do sistema é realizada em tempo real, o supervisor deve poder utilizar a medida de tempo fixada para o relógio em tempo real e deve realizar os cálculos da sequência afim de estabelecer as condições e os tratamentos ou ações que serão associadas às transições e aos lugares da rede. Em particular, o supervisor deve verificar a sequência de restrições temporais (margens temporais e intervalos de visibilidade das fichas) com o

envio de um sinal de alarme ao operador no caso de uma violação de restrição. Temos assim os envios de mensagens aos atuadores para acionar as operações do sistema e as recepções de mensagens dos captadores para indicar os fins de operação ao modelo do sistema.

Em (Julia 1997a) temos em detalhe o algoritmo do jogador em tempo real com todos os seus procedimentos, cuja abordagem foi feita em células flexíveis mas que pode ser usada muito bem em sistemas de produção por lotes (ambos sistemas possuem características de flexibilidade).

Capítulo 5

Exemplo

5.1 Modelagem

Como já foi dito, utilizou-se para esta dissertação um modelo de rede de Petri p-temporal t-temporizado para modelar o sistema de produção por lotes. A figura 5.1 mostra o sistema de produção por lotes utilizado como exemplo na implementação feita nesta dissertação.

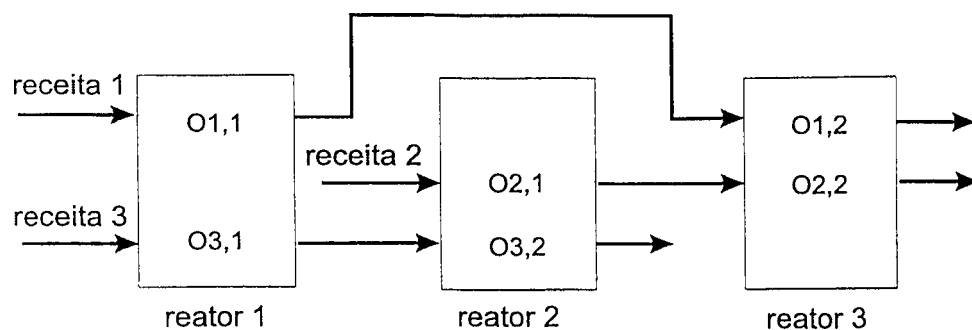


Figura 5.1: Exemplo de Sistema de Produção por Lotes

Para maior facilidade de aplicação, consideramos neste exemplo que os reatores podem ser solicitados no máximo por duas receitas e que nenhuma delas possui estoques intermediários, ou seja, após uma operação de tratamento de um lote num reator, o lote não será estocado para espera de outro tratamento.

A seguir veremos as restrições de sincronização do nosso modelo e posteriormente, as

5.1. MODELAGEM

respectivas restrições temporais.

5.1.1 Restrições de Sincronização

Receitas:

As sub-redes de Petri que representam as três receitas do nosso exemplo são dadas na figura 5.2:

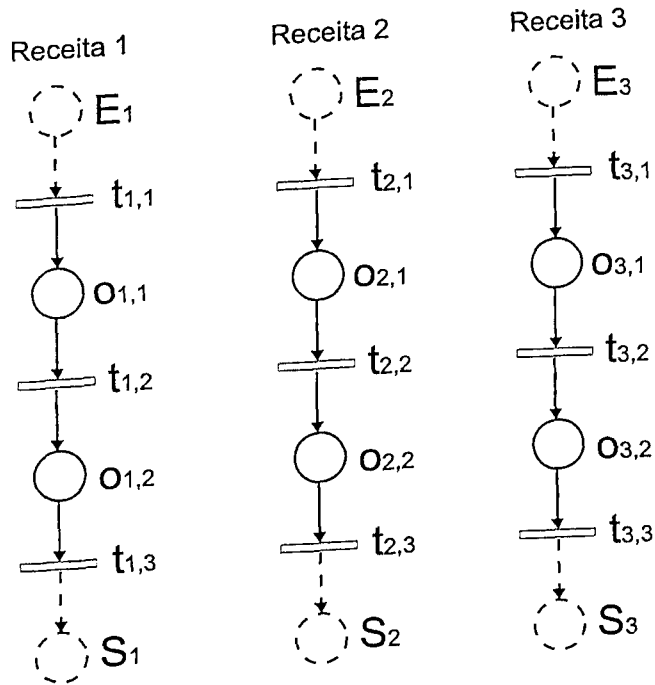


Figura 5.2: Modelagem das receitas Utilizando Rede de Petri

Observe que a modelagem das receitas é dada por sequências de transições e lugares de modo que o primeiro lugar da sequência representa sempre o estoque de entrada e o último lugar da sequência representa sempre o estoque de saída.

Lugares Kanbans:

As sub-redes de Petri que representam os três lugares Kanban do nosso exemplo são dadas na figura 5.3:

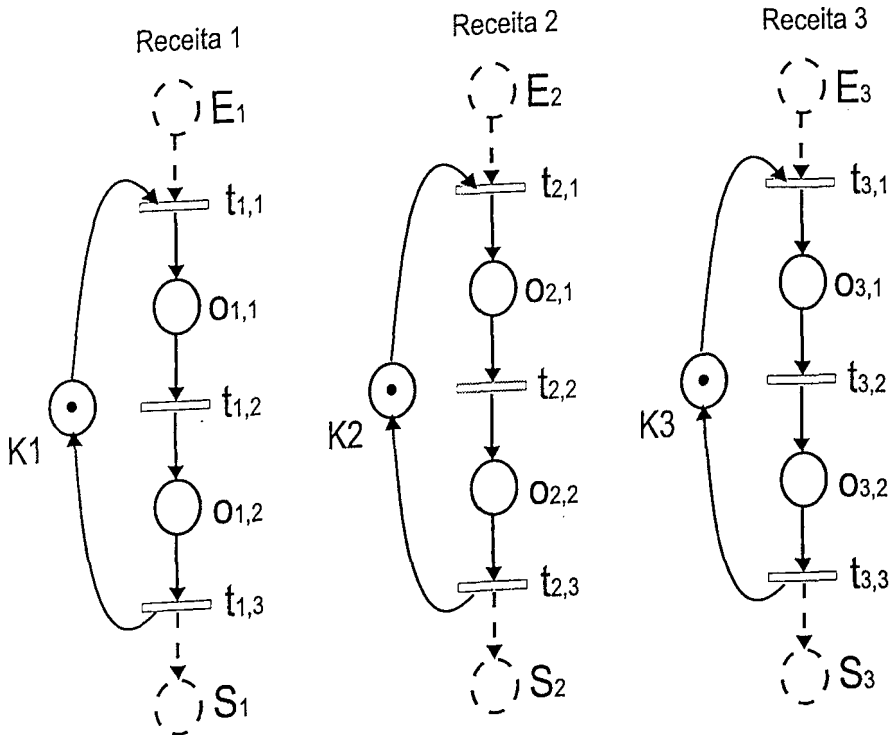


Figura 5.3: Modelagem dos lugares Kanban Utilizando Rede de Petri

Observe que cada lugar Kanban do nosso exemplo possui uma ficha, ou seja, no nosso exemplo para cada ciclo completo da rede, um único lote será tratado em cada receita.

Reatores:

As sub-redes de Petri que representam os três reatores do nosso exemplo são dadas na figura 5.4.

Podemos observar que no nosso exemplo, cada reator tratará apenas dois lotes de receitas distintas. Além disso, cada reator só poderá ser solicitado para tratar um lote por vez de cada receita, daí o fato de cada reator possuir uma única ficha.

Lugares Cíclicos:

As sub-redes de Petri que representam os seis lugares cíclicos do nosso exemplo são dadas na figura 5.5.

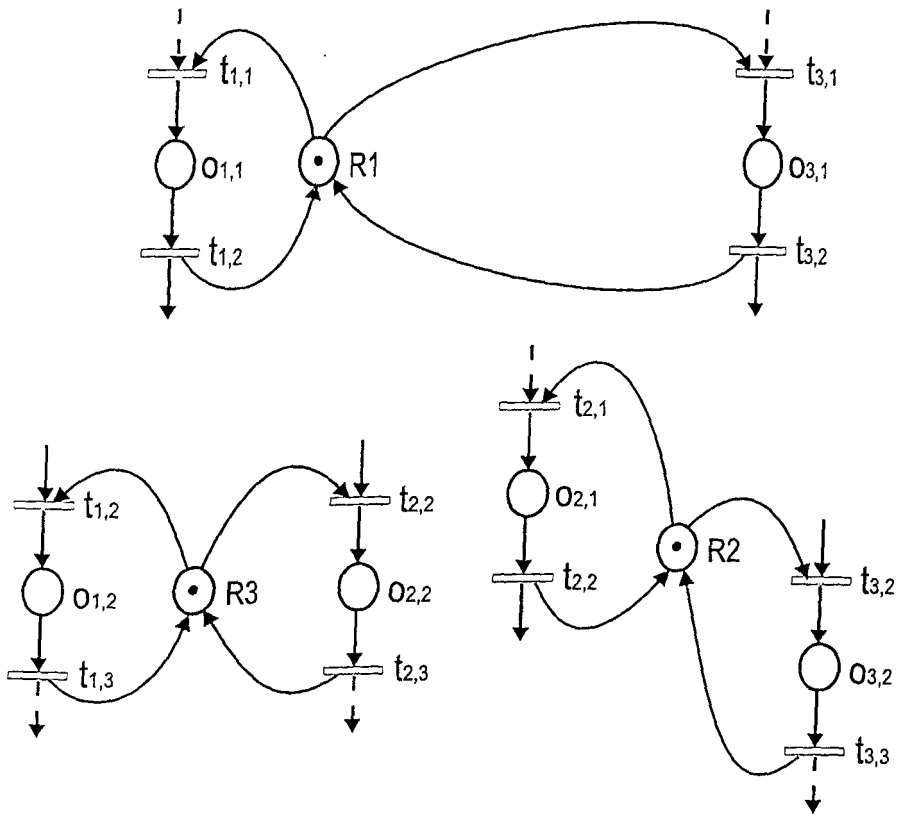


Figura 5.4: Modelagem dos reatores R_1 , R_2 e R_3 Utilizando Rede de Petri

Observe que no nosso exemplo, cada receita possui um lugar cíclico de entrada e um lugar cíclico de saída totalizando seis lugares cíclicos. Inicialmente, as fichas se encontram nos lugares cíclicos de saída para permitir o disparo da transição de sincronização t_s . Consideraremos ainda que cada arco da restrição cíclica possuirá peso igual a 1, ou seja, nunca nenhum lugar cíclico do nosso exemplo possuirá mais de uma ficha. Desta forma, para que haja um novo ciclo é necessário que cada uma das três receitas tenha sido completada uma única vez.

A seguir veremos as restrições temporais do nosso exemplo.

5.1.2 Restrições Temporais do Exemplo

Intervalos Estáticos dos lugares pertencentes às receitas:

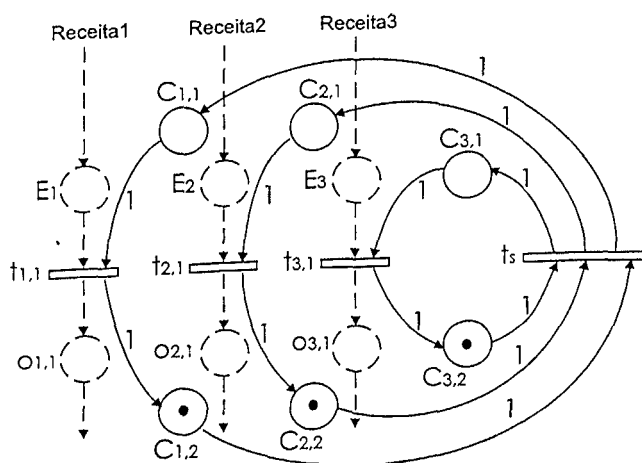


Figura 5.5: Modelagem dos lugares cíclicos Utilizando Rede de Petri

Estes intervalos de tempos são durações $[min, max]_s$ de permanência de um lote num determinado lugar de uma receita do sistema. Esta permanência é que define o tempo que o lote gastará para ser tratado por um determinado reator. Por exemplo, dos valores que o lote gastará para ser tratado no lugar $O_{1,1}$ da figura 5.2, possui um tempo de tratamento mínimo de 2 segundos e um tempo de tratamento máximo de 6 segundos gasto no reator R_1 , e assim por diante.

$$\text{Receita 1: } O_{1,1} = [2; 6]_s \quad O_{1,2} = [2; 5]_s$$

$$\text{Receita 2: } O_{2,1} = [4; 8]_s \quad O_{2,2} = [2; 3]_s$$

$$\text{Receita 3: } O_{3,1} = [2; 6]_s \quad O_{3,2} = [3; 4]_s$$

Intervalos Estáticos dos lugares Kanban:

Estes intervalos de tempos são durações de espera gastas para que um lote possa ser tratado por uma receita. Por exemplo, na *receita1* o lugar Kanban K_1 possui uma duração mínima de tempo de 1 segundo que equivale a dizer que o lote precisa esperar no mínimo 1 segundo para ser tratado pela *receita1* e uma duração máxima de tempo de 9 segundos que equivale a dizer que o lote pode esperar no máximo 9 segundos para ser tratado pela *receita1*. Os intervalos estáticos dos lugares Kanban do nosso exemplo foram obtidos segundo uma heurística. Escolheu-se neste caso que em princípio os lotes

podem ser tratados pelas receitas o mais rápido possível (bordas mínimas pequenas). Para a escolha das bordas máximas dos intervalos Kanban, com base nos conhecimentos do usuário sobre o sistema, define-se seus valores. A figura 5.6 ilustra um exemplo de como escolher os valores das bordas máximas dos lugares Kanbans.

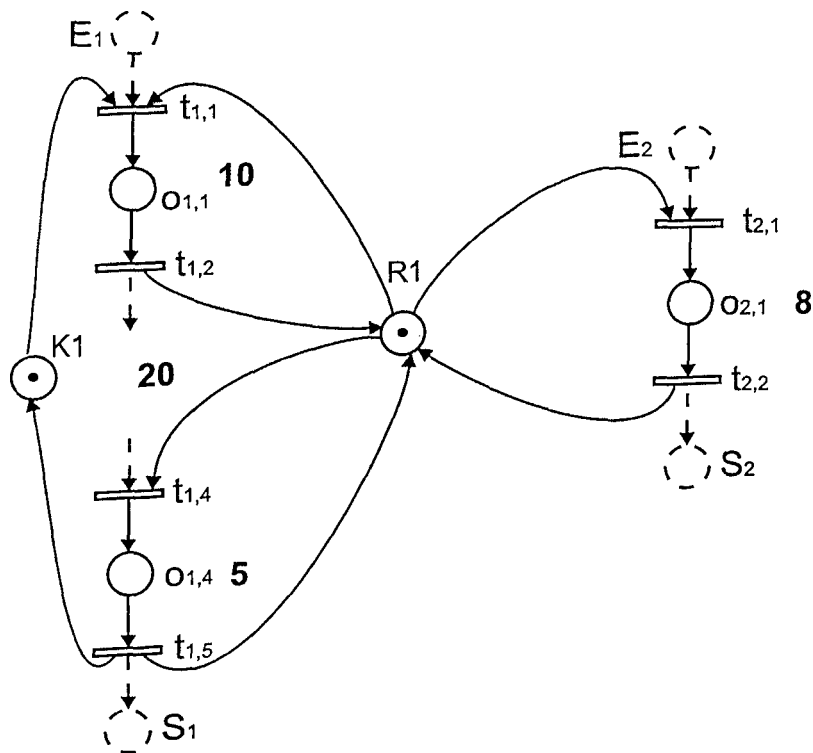


Figura 5.6: Escolha das Bordas Máx. dos Intervalos Estáticos dos Lugares Kanbans

No exemplo da figura 5.6 podemos verificar que o reator R_1 gasta 10 unidades de tempo para realizar a operação $O_{1,1}$, 8 unidades de tempo para realizar a operação $O_{2,1}$ e 5 unidades de tempo para realizar a operação $O_{1,4}$. Vamos supor que entre as transições $t_{1,2}$ e $t_{1,4}$, não existam recursos compartilhados. Para escolhermos a borda máxima do intervalo estático do lugar Kanban K_1 , iremos considerar que o reator R_1 , ao ser solicitado pela receita 1, seja utilizado ao mais tarde possível, ou seja, R_1 será utilizado primeiro pela receita 2 e depois pela receita 1. Desta forma, o máximo de tempo que a ficha dentro do lugar K_1 deverá esperar para disparar $t_{1,1}$ é de 8 unidades de tempo.

Desta forma, no nosso exemplo, as três restrições Kanbans terão intervalos estáticos

iguais a:

$$K_1 = [1; 9]_S \quad K_2 = [2; 4]_S \quad K_3 = [4; 7]_S$$

Intervalos Estáticos dos reatores:

Estes intervalos de tempo são durações de espera que um reator pode ter para ser utilizado por uma receita. No nosso exemplo, com a chegada de uma marca num reator, o mesmo é imediatamente disponibilizado para ser utilizado por uma receita (borda mínima nula), e pode esperar um tempo infinito de tempo para ser utilizado por uma receita (borda máxima infinita).

$$R_1 = [0; +\infty]_S \quad R_2 = [0; +\infty]_S \quad R_3 = [0; +\infty]_S$$

Intervalos Estáticos dos lugares cíclicos:

Estes intervalos de tempo de durações de espera, representam quanto tempo uma receita irá poder esperar para ser iniciada após a sincronização do sistema. No nosso exemplo consideramos um valor de espera mínimo nulo e um valor de espera máximo ilimitado.

$$C_{1,1} = [0; +\infty]_S \quad C_{1,2} = [0; +\infty]_S$$

$$C_{2,1} = [0; +\infty]_S \quad C_{2,2} = [0; +\infty]_S$$

$$C_{3,1} = [0; +\infty]_S \quad C_{3,2} = [0; +\infty]_S$$

Temporizações das transições pertencentes às receitas:

Estas durações de tempo representam o tempo gasto para que um determinado lote seja transferido de um reator a outro. No nosso exemplo, as transferências de lote de um reator a outro serão definidas como mostrado abaixo:

$$d_{t1,1} = 2; d_{t1,2} = 3; d_{t1,3} = 1$$

$$d_{t2,1} = d_{t2,2} = d_{t2,3} = 1$$

$$d_{t3,1} = d_{t3,2} = d_{t3,3} = 1$$

Temporização da transição de sincronização:

Esta duração de tempo representa o tempo de espera para que, após terminado o ciclo do sistema, seja iniciada uma nova sincronização do sistema e, conseqüentemente, seja iniciado um novo ciclo. No nosso exemplo, consideramos que a evolução do sistema seja ininterrupta, ou seja, uma vez terminado um ciclo, um novo ciclo será inicializado imediatamente.

$$d_{ts} = 0$$

Com relação a todos os arcos de entrada e de saída de todos os lugares da rede, nós consideramos um valor unitário para seus pesos, ou seja, apenas uma ficha sai de um determinado lugar e apenas uma ficha entra em um determinado lugar. Isto vai implicar no fato de que apenas um lote por receita será tratado em cada ciclo da rede.

Finalmente, considerando todas as sub-redes de Petri do sistema e suas respectivas restrições temporais, para chegar no modelo global do sistema mostrado na figura 5.7 fazemos a fusão de todas elas. As condições iniciais da rede mostram que: os reatores e lugares Kanban estão disponíveis, além do fato de que apenas um lote será tratado por vez em cada receita. Durante a evolução da rede todos os lugares de entrada das receitas $E1$, $E2$ e $E3$ possuem lotes disponíveis para serem tratados.

5.2 Implementação da RdP p-temporal t-temporizada

Na fase de implementação feita neste trabalho, optou-se utilizar a linguagem de programação C para criarmos a estrutura da rede de Petri bem como suas restrições temporais associadas. Além disso, implementou-se o “jogador” de rede de Petri p-temporal t-temporizado com o mecanismo de retrocesso “inteligente”. O uso da linguagem C foi devido a sua portabilidade e também a sua rapidez. Poder-se-ia pensar em usar simplesmente uma linguagem por restrições já que o “jogador” utiliza análise por restrições. Mas, no trabalho de doutorado Frédéric Boutet (Boutet 1999), verificou-se que para sistemas de complexidade considerável, o uso de linguagens lógicas tornam o programa lento e portan-

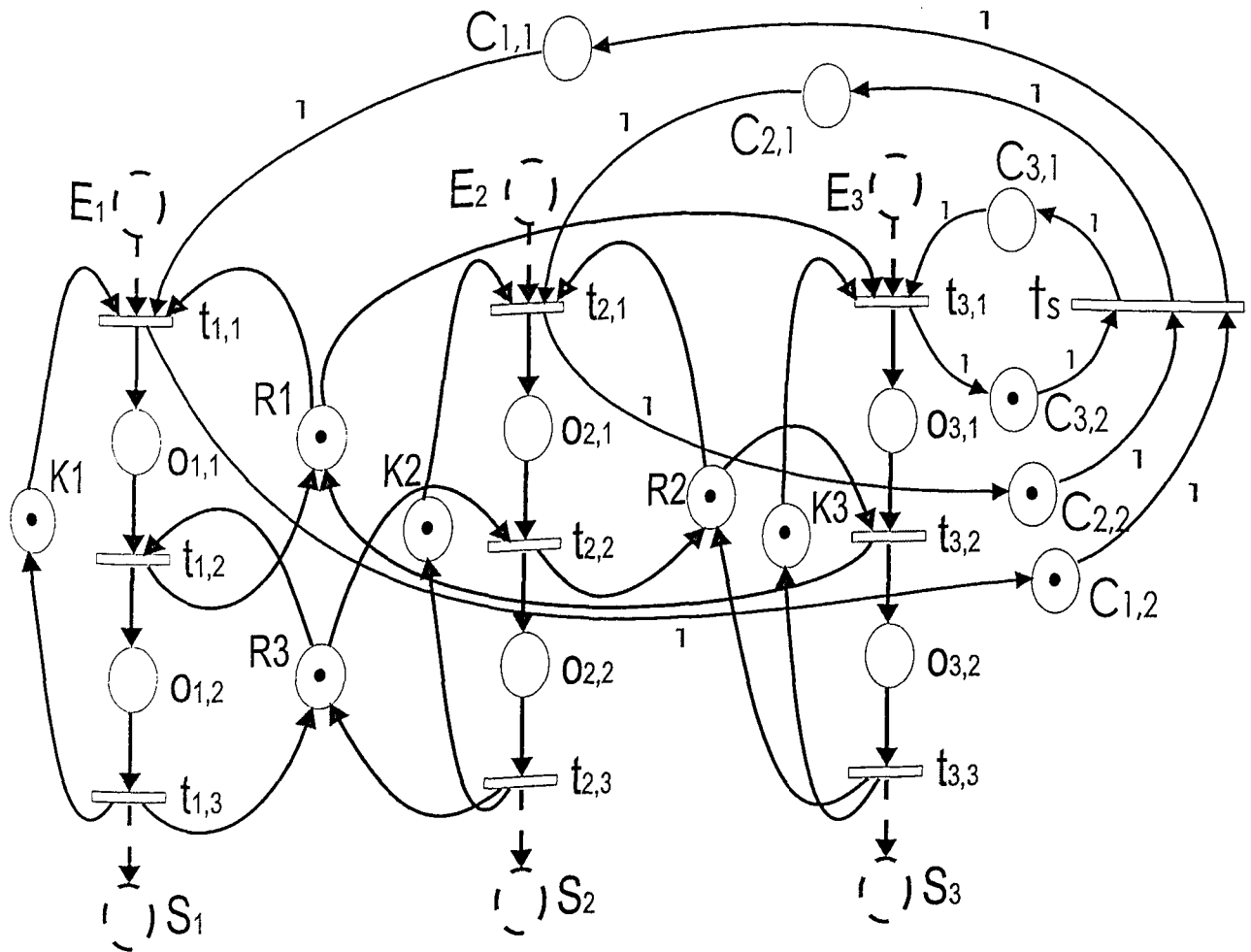


Figura 5.7: Modelagem do Sistema de Produção por Lotes Utilizando Rede de Petri

to pouco eficiente, principalmente se o objetivo for associar o escalonamento previewal ao escalonamento em tempo real.

A estrutura de dados utilizada para representar as entidades da rede (fichas, lugares, transições e arcos) é uma lista duplamente encadeada. Porque usar uma estrutura de lista? Esta estrutura possui ponteiros que são muito eficazes quando queremos manipular dados dispostos de maneira irregular. O fato de usarmos uma lista duplamente encadeada e não simplesmente encadeada é pelo fato de que durante a evolução da rede precisamos obter informações de entidades precedentes à que estamos avaliando. A figura 5.8 mostra

a representação estrutural correspondente à sequência de entidades $t_{1,2} - O_{1,2} - t_{1,3}$ da rede.

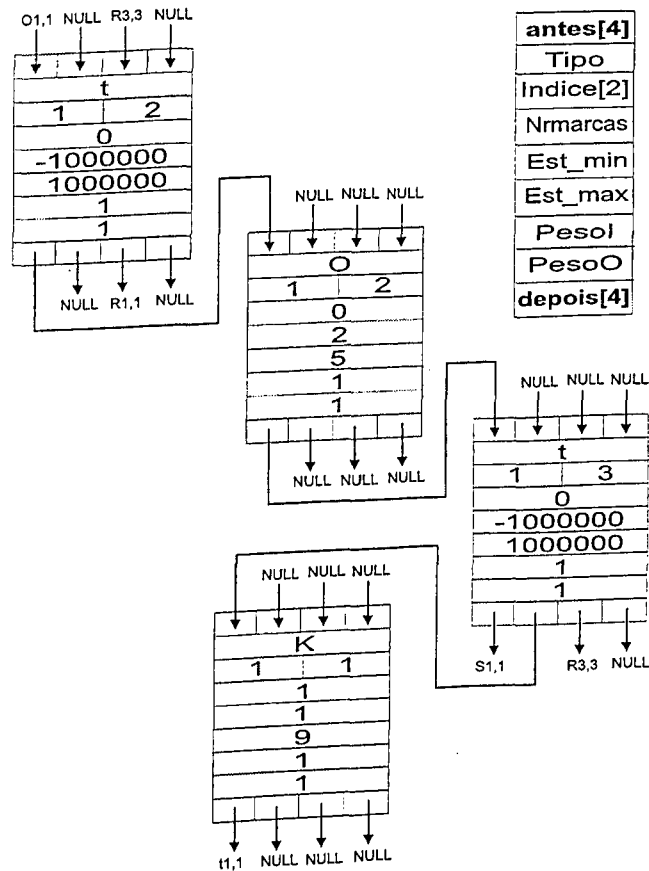


Figura 5.8: Estrutura de Dados local da rede de Petri

Cada nó da lista é uma estrutura que especifica o tipo da entidade (lugar ou transição) da rede *Tipo*, o índice que identifica qual é a identidade *Indice[2]*, o número de marcas de um lugar *Nrmarcas*, a borda mínima do intervalo estático do lugar *Est_min*, a borda máxima do intervalo estático do lugar *Est_max*, o peso do arco de entrada da entidade *PesoI*, o peso do arco de saída da entidade *PesoO*, os arcos de entrada da entidade **antes[4]* e os arcos de saída da entidade **depois[4]*.

Com base no fato de que na nossa rede uma entidade (lugar ou transição) pode ter no máximo quatro arcos de entrada e quatro arcos de saída, consideramos um vetor de ponteiros contendo quatro posições de entrada e um vetor de ponteiros contendo quatro

posições de saída. Para entidades que possuem menos de quatro arcos de entrada ou saída, os ponteiros excedentes apontarão para NULL, como mostrado na figura 5.8.

Quanto ao programa, ele permite ao usuário acompanhar tanto o calendário de eventos quanto o estado das marcações durante a evolução da rede. O calendário exibe a data atual e os eventos escalonados no tempo, que são: borda mínima de um intervalo de visibilidade, borda máxima de um intervalo de visibilidade e data de fim de disparo. A estrutura inicial do calendário no nosso exemplo é dada na tabela a seguir:

Id	δ	Tipo	Indice		Identif.			Ativo
			0	1	0	1	2	
1	0	C	1	1	b	m	i	1
2	0	R	1	1	b	m	i	1
3	0	R	3	3	b	m	i	1
4	0	C	1	2	b	m	i	1
5	0	R	2	2	b	m	i	1
6	0	C	1	3	b	m	i	1
7	1	K	1	1	b	m	i	1
8	1	K	3	3	b	m	i	1
9	2	K	2	2	b	m	i	1
10	7	K	3	3	b	m	a	1
11	8	K	2	2	b	m	a	1
12	9	K	1	1	b	m	a	1
13	$+\infty$	C	1	1	b	m	a	1
14	$+\infty$	R	1	1	b	m	a	1
15	$+\infty$	R	3	3	b	m	a	1
16	$+\infty$	C	1	2	b	m	a	1
17	$+\infty$	R	2	2	b	m	a	1
18	$+\infty$	C	1	3	b	m	a	1

Na tabela acima, Id é o número do evento, δ é a data de ocorrência do evento, $Tipo$ é a entidade (transição/lugar) da rede associada ao evento, $Indice$ é o índice correspondente à entidade, $Identif.$ é o tipo de evento e $Ativo$ indica a ocorrência do evento. Por exemplo, no primeiro evento da tabela acima, a data de ocorrência é 0, o evento ocorre no lugar C_{11} , o evento é do tipo "bmi" (borda mínima do intervalo de visibilidade do lugar C_{11}) e com estado ativo igual a 1, ou seja, ainda não ocorreu o evento (quando ocorrer o evento, o estado ativo será 0). Para facilitar na implementação os lugares $R_1, R_2, R_3, K_1, K_2, K_3$ são representados por $R_{11}, R_{22}, R_{33}, K_{11}, K_{22}, K_{33}$, respectivamente.

O estado das marcações da rede se caracteriza pelo estado das fichas que estão indisponíveis, disponíveis, reservadas e não-reservadas. As estruturas dos estados das marcações são matrizes, onde o número de linhas representa o número de memorizações que podem ocorrer durante um ciclo completo e o número de colunas representa um campo para identificar o número da memorização e 18 campos para representar o número de lugares da rede que no nosso exemplo são 18 lugares. A declaração é do tipo:

```
int mndi[45][19], mdis[45][19];
```

```
int mres[45][19], mmor[45][19];
```

Por exemplo, inicialmente todos as marcas existentes na rede se encontram no estado não disponível, portanto, a primeira linha das matrizes que representam as marcas no estado disponível ($mdis[45][19]$), as marcas no estado reservado ($mres[45][19]$) e as marcas "mortas" da rede ($mmor[45][19]$) é toda inicializada com valor nulo. Já a primeira linha da matriz que representa as marcas no estado não disponível ($mndi[45][19]$), é inicializada da seguinte forma:

	O	O	K	O	O	K	O	O	K	R	R	R	C	C	C	C	C	C
Id	11	12	11	21	22	22	31	32	33	11	22	33	11	21	31	12	22	32
1	0	0	1	0	0	1	0	0	1	1	1	1	1	1	1	0	0	0

O programa permite ainda que, durante a evolução da rede, o usuário saiba quais conflitos estruturais e efetivos aparecem, bem como os resultados provenientes do uso do sistema de decisão e do mecanismo de retrocesso. A estrutura usada para os conflitos é

dada a seguir:

```

struct Conflito{
    int Id;
    int Data;
    int Ativo;
    int Indice[2];
    int Ativo1;
    int Indice1[2];
};

```

Na estrutura mostrada acima, *Id* refere-se ao número da ocorrência do evento do calendário que originou o conflito em questão, *Data* é a data de ocorrência do conflito efetivo, *Ativo* refere-se ao fato de uma das transições envolvidas no conflito já ter sido escolhida como solução (*Ativo* igual a 1) ou não (*Ativo* igual a 0), *Indice[2]* são os índices que identificam uma das transições do conflito, *Ativo1* refere-se ao fato da outra transição envolvida no conflito ter já ter sido escolhida como solução (*Ativo1* igual a 1) ou não (*Ativo1* igual a 0) e *Indice1[2]* são os índices que identificam a outra transição do conflito. Por exemplo, a tabela a seguir indica que o evento 7 do calendário refere-se ao conflito estrutural entre as transições $t_{1,1}$ e $t_{3,1}$ na data $\delta = 1$, e que $t_{1,1}$ foi escolhida como solução de primeiro disparo:

Id	data	Ativo	Indice		Ativo1	Indice1	
			0	1		0	1
7	1	1	1	1	0	3	1

Caso uma violação de restrição não possa ser resolvida pelo "jogador", então o programa mostra um diagnóstico ao usuário de modo que ele possa alterar as condições iniciais da rede.

5.3 Simulação e Resultados

Ao simularmos o programa com as condições iniciais da rede mostradas nas seções 5.1.1 e 5.1.2, verificamos que:

- $t_{1,1}$ e $t_{3,1}$ estão em conflito efetivo pelo recurso R_1 em $[4, 7]_C$ (intervalo de conflito). O mecanismo de decisão é acionado e decide o disparo de $t_{1,1}$ primeiro na data $\delta = 3$ sem causar violação de restrição.
- $t_{2,1}$ e $t_{3,2}$ não estão em conflito efetivo pelo recurso R_2 , mas o disparo de $t_{2,1}$ primeiro causa “morte” da marca em $O_{3,1}$ na data $\delta = 16$ e o disparo de $t_{3,2}$ primeiro causa a “morte” da marca em K_3 na data $\delta = 8$. Portanto, verifica-se uma violação de restrição devido ao disparo cedo demais da transição $t_{3,1}$ em duas unidades de tempo e o mecanismo de retrocesso é acionado retrocedendo até o estado da rede na data $\delta = 1$ (uma marca em k_3). Assim o disparo de $t_{3,1}$ é atrasado no mínimo de 2 unidades de tempo ocasionando um aumento na borda mínima de k_3 de $[4, 7]_V$ para $[6, 7]_V$.
- Na data $\delta = 5$ dispara-se $t_{2,1}$.
- Na data $\delta = 8$ dispara-se $t_{1,2}$.
- Na data $\delta = 8$ ocorre morte da marca em K_3 .

Podemos observar que apesar do atraso, ocorreu morte da marca em K_3 . Neste caso, ocorreu uma violação de restrição e o “jogador” exhibe ao usuário que a marcação em K_3 no início do ciclo deve ser mudada de modo que a marca que seja colocada no lugar anterior $O_{3,2}$. Desta forma, consegue-se um atraso ainda maior que o obtido anteriormente. A figura 5.9 mostra a nova configuração inicial da rede.

Simulando-se a nova rede, temos:

- $t_{1,1}$ e $t_{3,1}$ não estão em conflito efetivo. O mecanismo de decisão é acionado e decide o disparo de $t_{1,1}$ primeiro na data $\delta = 3$ sem causar violação de restrição.
- $t_{2,1}$ e $t_{3,2}$ não estão em conflito efetivo pelo recurso R_2 . O mecanismo de decisão é acionado e verifica-se que o disparo de $t_{2,1}$ primeiro causa morte da marca em $O_{3,1}$ na data $\delta = 16$ e o disparo de $t_{3,2}$ primeiro causa morte da marca em K_3 na

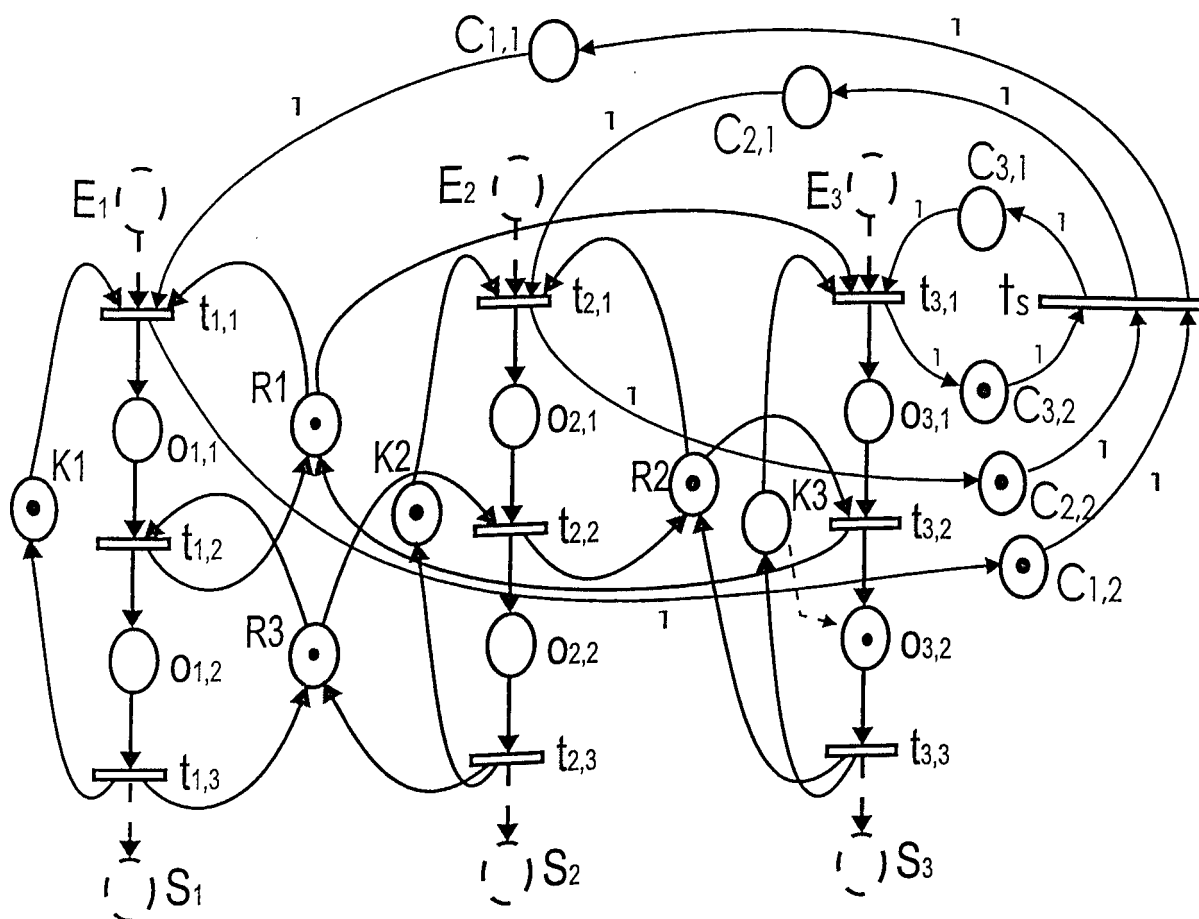


Figura 5.9: Rede de Petri com as novas condições iniciais

data $\delta = 13$. Portanto, verifica-se uma violação de restrição devido ao disparo cedo demais da transição $t_{3,1}$ em duas unidades de tempo e o mecanismo de retrocesso é acionado retrocedendo até o estado da rede na data $\delta = 4$ (uma marca em k_3). Assim o disparo de $t_{1,3}$ é atrasado no mínimo de 4 unidades de tempo ocasionando um aumento na borda mínima de k_3 de $[8, 11]_V$ para $[10, 11]_V$.

- Na data $\delta = 4$ dispara-se $t_{3,3}$.
- Na data $\delta = 5$ dispara-se $t_{2,1}$.
- Na data $\delta = 8$ dispara-se $t_{1,2}$.
- Na data $\delta = 9$ dispara-se $t_{3,1}$.

5.3. SIMULAÇÃO E RESULTADOS

- Na data $\delta = 11$ dispara-se $t_{2,2}$.
- Na data $\delta = 14$ dispara-se $t_{1,3}$.
- Na data $\delta = 16$ dispara-se $t_{3,2}$.
- Na data $\delta = 17$ a rede atinge um ciclo completo.

Com a mudança das condições iniciais sugerida pelo “jogador”, o usuário excluiu a violação de restrição ocorrida na evolução da rede e o ciclo foi completado.

Capítulo 6

Conclusão

Neste trabalho, mostramos um ensaio de como os modelos e técnicas de resolução mais tradicionalmente aplicados no quadro de sistemas flexíveis de manufatura podem ser usados para tratar a problemática dos sistemas de produção híbridos, mais especificamente, os sistemas de produção por lotes. Neste contexto, observamos que certas diferenças existem entre estes dois tipos de sistemas e que as técnicas utilizadas na manufatura precisam de certas adaptações para poderem ser aplicadas ao caso dos sistemas de produção por lotes.

Foi apresentada, ainda, uma metodologia formal para a resolução do problema de escalonamento cíclico dos sistemas de produção por lotes. Definimos um “jogador” de rede de Petri p -temporal t -temporalizada que utiliza um mecanismo de retrocesso inteligente, que por sua vez busca resolver violações que possam aparecer quando um recurso compartilhado é solicitado. O retrocesso é feito segundo um diagnóstico realizado para cada violação presente e não de maneira sistemática, dando-nos assim uma sequência admissível. A resolução é feita num horizonte temporal mínimo (resolução local) de modo que não irá garantir a ausência de futuras violações durante a evolução da rede.

O “jogador” leva em conta, também, possíveis alterações no funcionamento normal do sistema permitindo ao usuário, no momento da inicialização dos dados, fornecer parâmetros que irão auxiliar a evolução da rede, prevendo tais alterações.

Outra característica importante do “jogador” é que nos casos em que o retrocesso não é possível, o usuário poderá interagir com o programa e de acordo com os dados fornecidos, ele poderá alterar as condições iniciais da rede.

A linguagem utilizada, C, permitiu representarmos todas as restrições do modelo, bem como, acompanharmos a evolução da rede mostrando os passos do calendário e as eventuais situações de conflito com suas resoluções.

O modelo utilizado reflete bem o comportamento híbrido dos sistemas de produção por lotes, principalmente no fato de trabalharmos com tempos imprecisos como por exemplo o tempo de tratamento dos lotes associado aos lugares da rede. Como trabalho futuro, seria interessante refinarmos estes intervalos utilizando, por exemplo, intervalos “fuzzy”. Outra sugestão seria associarmos outras técnicas de inteligência artificial ao modelo apresentado de forma a reduzir o número de retrocessos feitos no escalonamento previsual (técnicas de filtragem utilizadas no CSP “Constraint Satisfaction Problems”). Isto é particularmente interessante quando tratamos do escalonamento em tempo real, onde retrocessos não são permitidos. No nível de implementação do “jogador” de rede de Petri p-temporal t-temporizada, seria interessante associarmos a linguagem C a uma linguagem por restrição (Prolog III, por exemplo) para tratar do sistema de decisão.

Bibliografia

- Alla, H. (1995). Modeling and simulation of event driven systems by petri nets. *ADE-DOPS*.
- Alla, H. and J. M. Flaus (1998). Modeling of gas storage unit using hybrid flow nets. *Proceedings of ADPM'98*.
- Alur, R., C. Courcoubetis, N. Halbwachs, T. A. Henzinger and P. H. Ho (1995). The algorithmic analysis of hybrid systems. *Theoretical computer science* **138**, 3-34.
- Andreu, D., J. C. Pascal and R. Valette (1995a). Interaction of discrete and continuous parts of a batch process control system. *ADEDOPS*.
- Andreu, D., J. C. Pascal and R. Valette (1996). Events as a key of a bath process control system. *CESA '96 IMACS Multiconference, symposium on discrete events and manufacturing systems* pp. 297-302.
- Andreu, D., J. C. Pascal, H. Pingaud and R. Valette (1995b). Batch process modelling using petri nets. *International conference on systems, Man and cybernetics, IEEE* pp. 314-319.
- Atabakhche, H. (1987). Utilisation conjointe de l'intelligence artificielle et des réseaux de Petri: application au contrôle d'exécution d'un plan de fabrication. Thèse de doctorat. Université Paul Sabatier. Toulouse-FRANCE.

- Atabakhche, H., D. Simonetti-Barbalho, R. Valette and M. Couvoisier (1987). Commande d'ateliers: un compromis est-il possible entre une approche graphique et une approche intelligence artificielle?. *APII* pp. 377-394.
- Azzopardi, D. (1994). Scheduling and Simulation of batch process plant through Petri net modelling. Thesis. University of Sussex. Brighton-UK.
- Azzopardi, D. and S. Loyd (1994a). Reduction of search space for scheduling of multi-product batch process plant through petri net modelling. *2nd IFAC/IFIP/IFURS Workshop on intelligent manufacturing systems*.
- Azzopardi, D. and S. Loyd (1994b). Reduction of search space for scheduling of multi-product batch process plant through petri net modelling. In: *x. 2nd IFAC/IFIP/IFURS Workshop on intelligent manufacturing systems*. Vienna-AUSTRIA.
- Bail, J. Le, H. Alla and R. David (1991a). Hybrid petri net. *European control conference*.
- Bail, J. Le, H. Alla and R. David (1991b). Hybrid petri net. *Proceedings of the European control conference*.
- Bako, B. (1990). Mise en oeuvre et simulation du niveau coordination de la comande des ateliers flexibles: une approche mixte réseaux de Petri et systèmes de règles. Thèse de doctorat. Université Paul Sabatier. Toulouse-FRANCE.
- Barros, T. C. and A. Perkusich (1998). Redes de petri: Um procedimento de modelagem aplicado aos sistemas de produção em lotes. *Proceedings of XII Brazilian Automatic Control Conference - XII CBA IV*, 1377-1382.
- Berthelot, G. (1987). Transformations and decompositions of nets, petri nets central models and their properties. *Lecture Notes in Computer Science* 254, 359-376.
- Berthelot, G., G. Roucairol and R. Valk (1980). Reductions of nets and parallel programs. *Lecture Notes in Computer Science* 84 pp. 277-290.

- Boutet, Frédéric (1999). Contraintes et temps absolu dans les réseaux de Petri. Application à l'ordonnement de tâches à durées incertaines dans les systèmes temps réel. Thèse de doctorat. INSA. Toulouse-FRANCE.
- Brams, G. W., Ed.) (1983). *Réseaux de Petri: Théorie et Pratique*. Masson.
- Caradec, M. and F. Prunet (1997). A new modeling tool for hybrid flexible systems. *IEEE* pp. 2246-2251.
- Cardoso. J. and Valette, R., Eds.) (1997). *Redes de Petri*. DAUFSC.
- Cassandras, C. G. and P. J. Ramadge (1990). Toward a control theory for discrete event system. *IEEE control systems magazine* pp. 66-68.
- Champagnat, R., H. Pingaud, H. Alla, C. Valentin-Roubinet and J. M. Flaus (1998a). A gas storage example as a benchmark for hybrid modeling. *APII-JESA* **32**, nro 9-10, 1233-1253.
- Champagnat, R., P. Esteban, H. Pingaud and R. Valette (1996). Petri net based modeling of hybrid systems. *ASI'96*.
- Champagnat, R., P. Esteban, H. Pingaud and R. Valette (1998b). From scheduling to supervision in bath process. *II IMACS Multiconference CESA '98* pp. 673-678.
- Champagnat, R., P. Esteban, H. Pingaud and R. Valette (1998c). Modeling and simulation of a hybrid systems through pr/tr pn-dae model. *Proceedings of ADPM'98*.
- Cohen, G., P. Moller, J. Quadrat and M. Viot (1989). Algebraic tools for the performance evaluation of discrete event systems. *IEEE* **77**(1), 39-57.
- Combacau, M. (1991). Commande et surveillance des systèmes à événements discrets complexes: application aux ateliers flexibles.. Thèse de doctorat. Université Paul Sabatier. Toulouse-FRANCE.

- Courvoisier, M. and R. Valette (1986). Commande des procédés discontinus-logique séquentielle. *Dunod Université*.
- Courvoisier, M., R. Valette, A. Sahraoui and M. Combacau (1989). Specification and implementation techniques for multilevel control of fms. *CAPEs'89* pp. 509–516.
- Cépadueès, Ed.) (1991). *Concepts et outils pour les systèmes de production*. J. C. Hennet.
- Daubas, B., A. Pagès and H. Pingaud (1994). Combined simulation of hybrid processes. *IEEE-SMC* pp. 320–325.
- David, R. and Alla, H., Eds.) (1992). *Petri nets and Grafcet*. Prentice-Hall.
- David, R. and H. Alla (1987). Continuous petri nets. *Proceedings of the eighth European workshop on application and theory of Petri nets* pp. 275–294.
- Dechter, R. (1986). Learning while searching in csp. removing redundancies in constraint network. *12th National conference on artificial intelligence (AAAI86)* pp. 105–109.
- Dunod, Ed.) (1983). *Programmation mathématique: Théorie et Algorithmes*. M. Minoux.
- Dutilleul, S. Collart and J-P. Denat (1998). P-time petri nets and the hoist scheduling problem. *IEEE* pp. 558–563.
- Elmaghraby, S. E. (1977). *Activity networks*. John Wiley and Sons.
- Erschler, J. and P. Esquirol (1986). Decision aid in job-shop scheduling: a knowledge based approach. *IEEE Int. Conf. on Robotics and Automation* pp. 1651–1656.
- Esquirol, P. and Lopez, P., Eds.) (1999). *L'ordonnancement*. Economica. Paris.
- Flaus, J. M. (1996). Hybrid flow nets for batch process modelling. *Proceedings of CESA '96 (Symposium on discrete events and manufacturing systems)* pp. 273–278.
- Florin, G. and S. Natkin (1984). Définition formelle des réseaux de petri stochastiques. Technical report. Technical Report CNAM - Paris.

- Genrich, H.J., H.M. Hanish and K. Wollhaf (1994). Verification of recipe-based control procedures by means of predicate/transition nets. *Application as theory of Petri nets*, Robert Valette pp. 278–297.
- Gondran, M. and M. Minoux (1984). Graphes et algorithmes. *Eyrolles*.
- GOTHA (1993). Les problèmes d'ordonnancement. *Recherche opérationnelle* 27, 77–150.
- Guyot, J. (1994). Mechatronic components design in the automotive industry. *Proceedings of the 2nd Japan-France congress on mechatronics*.
- Hanish, H. M. (1993). Analysis of place/transition nets with timed arcs and its application to batch process control. *International conference on theory and application of Petri nets* pp. 282–299.
- Hillion, H. and J.M. Proth (1989). Performance evaluation of job-shop systems using timed event-graphs. *IEEE Trans. on Automatic Control* 34(1), 3–9.
- Hénault, V. (1996). Méthodologie de développement des systèmes électroniques embarqués automobiles, matériels et logiciels, sûrs de fonctionnement. Thèse de doctorat. I'RESTE.
- Ho, Y. C. (1989). Scanning the issue: Dynamics of discrete events systems. *Proceedings of the IEEE* 77(1), 3–6.
- Hopcroft, J. E. and Ullman, J.D., Eds.) (1979). *Introduction to automata theory, languages and computation*. Addison-Wesley.
- Jeng, M. and X. Xie (1997). Synthesis of resource control nets using siphons. *IEEE* pp. 1192–11197.
- Julia, S. (1997a). Conception et pilotage de cellules flexibles à fonctionnement répétitif modélisées par réseaux de Petri. Thèse de doctorat. Université Paul Sabatier. Toulouse-FRANCE.

- Julia, S. (1997b). Conception et pilotage de cellules flexibles à fonctionnement répétitif modélisées par réseaux de Petri. Thèse de doctorat. Université Paul Sabatier. Toulouse-FRANCE.
- Julia, S. (1998). Da concepção ao controle em tempo real de sistemas flexíveis de manufatura usando as redes de petri. *XII Congresso Brasileiro de Automática; Mini Curso* pp. 1–35.
- Julia, S., R. Valette and M. Tazza (1995). Computing a feasible schedule under a set of cyclic constraints. *II Conference international on industrial automation* pp. 141–146.
- Julia, S., R. Valette, C. M. M. Júnior and José M. Fernandes (1998). Escalonamento de sistemas de produção híbridos usando-se um jogador de rede de petri. In: *XII Congresso Brasileiro de Automática*. Uberlândia-BRASIL.
- Khansa, W., P. Aygalinc and J.P. Denat (1996). Structural analysis of p-time petri nets. In: *Symposium on discrete events and manufacturing systems, CESA'96 IMACS Multiconference*. Lille-FRANÇA.
- Kusiak, A. (1990). Intelligent manufacturing systems. *Prentice Hall International series in industrial and systems engineering*.
- Kusiak, A. and W. He (1994). Design of components for schedulability. *European Journal of Operational Research* 76 pp. 49–59.
- Lee, D. Y. and F. DiCesare (1994a). Integrated scheduling of flexible manufacturing systems employing automated guided vehicles. *IEEE Transactions on industrial electronics* 41, 602–610.
- Lee, D. Y. and F. DiCesare (1994b). Scheduling flexible manufacturing systems using petri nets and heuristic search. *IEEE Transactions on robotics and automation* 10, 123–132.

- Menasche, M. (1996). Parâmetros temporais em redes de petri. Technical report. Technical Report 8. Pontifícia Universidade Católica do Rio de Janeiro.
- Merlin, P. (1974). A study of recoverability of computer systems. Phd thesis. University of Califórnia.
- Merlin, P. (1979). Methodology for the design and implementation of communication protocols. *IEEE Transaction on Communications*.
- Miyagi, P. E., Ed.) (1999). *Controle Programável. Fundamentos do Controle de Sistemas a Eventos Discretos*. Edgard Blücher LTDA.
- M.J.Sanders (1998). Constraint programming with object-oriented petri nets. *IEEE* pp. 289-294.
- Moncelet, G., S. Christensen, H. Demmou, M. Paludetto and J. Porras (1998). Dependability evaluation a simple mechatronic system using coloured petri nets. *Workshop on practical use of coloured Petri nets and design CPN*.
- Murata, T. (1989). Petri nets: Properties, analysis and applications. *Proceedings of the IEEE* 77(4) pp. 541-580.
- Nau, D. S., S. K. Gupta and W. C. Regli (1996). Ai planing versus manufacturing operation planning: A case study. *IJCAI-95*.
- Ol, H., E. Castelain and J.C. Gentina (1994). Synchrony theory applied to control problems in flexible manufacturing systems. *IEEE International Conference on Systems, Man and Cybernetics*.
- Pagès, A. and H. Pingaud (1995). An hybrid process model based on petri nets applied to short term scheduling of batch-semi continuous plants. *ADEDOPS workshop*.
- Peterson, J. L., Ed.) (1981). *Petri net theory and the modeling of systems*. Prentice-Hall.

- Petri, C. A. (1966). Communication with automata. Technical report. Technical Report RADCTR-65-377. Darmstadt University.
- Prins, C. (1994). Algorithmes de graphes (avec programmes en pascal). *Eyrolles*.
- Ramammoorthy, C. V. and G. S. Ho (n.d.). Performance evaluation of asynchronous concurrent systems using petri nets. *IEEE Trans. on Soft. Eng. SE6* 5, 440-449.
- Ramchandani, C. (1974). Analysis of synchronous concurrent systems by timed Petri net models. Phd thesis. MIT. Project MAC TR-120.
- Reutenauer, C., Ed.) (1990). *The mathematics of Petri nets*. Masson e Prentice-Hall.
- Rillo, M. (1988). Aplicações de redes de Petri em sistemas de manufatura. Tese de doutorado. Escola Politécnica da USP.
- Sahraouri, A. K. (1987). Contribution à la commande et à la surveillance d'ateliers flexibles. Thèse de doctorat. Université Paul Sabatier. Toulouse-FRANCE.
- Seibel, C. W. and J. M. Farines (1997). O uso de autômatos híbridos para modelagem e verificação formal de planos de voo de aeronaves não-tripuladas. *Terceiro Simpósio Brasileiro de automação inteligente*.
- Sibertin-Blanc, C. (1994). Cooperative nets. In: *Lecture Notes in Computer Science*. Vol. 815. Springer-Verlag.
- Sifakis, J. (1977). Use of petri nets for performance evaluation. *3rd International Symposium on Modeling and Performance Evaluation of Computer Systems (H. Beilner and E. Gelenbe, Eds)*.
- Silva, M. and R. Valette (1989). Petri nets and flexible manufacturing. In: *Advances in Petri nets (G. Rosenberg, Ed) Lectures Notes in Computer Science*. Springer-Verlag. pp. 374-417.

- Silva, M., E. Teruel, R. Valette and H. Pingaud (1998). Petri nets and production systems. In *Lectures on Petri nets II: applications. Lecture notes in Computer Science 1492* pp. 85-124.
- Tazza, M. (1987). Quantitative analysis of a resource allocation problem: a net theory based proposal. In: *Concurrency and nets*. Springer-Verlag. pp. 511-532.
- Valentin, C. (1994). Modelling and analysis methods for a class of hybrid dynamic systems. *Proceedings of automation of mixed process: Dynamical hybrid systems SEEIBRA-AFCET*.
- Valentin, C. and P. Ladet (1993). Flow modelling in a class of hybrid (continuous-discrete) systems. *International conference on systems, man and cybernetics IEEE-SMC*.
- Valentin-Roubinet, C. (1998). Modelling of hybrid systems: Dae supervised by petri nets the example of a gas storage. *Proceedings of ADPM'98*.
- Valette, R. (1995). Petri nets for control and monitoring: specification, verification and implementation. *ADEDOPS*.
- Verfaillie, G. and T. Schiex (1994). Solution reuse in dynamic csps. *12th National conference on artificial intelligence (AAAI94)*.
- Viswanadham, N. and Y. Narahari (1987). Coloured petri net model for automated manufacturing system. *Proceeding of the IEEE. International conference on robotic and automation* pp. 1985-1990.
- W.Reisig (1985). Petri nets: An introduction. In: *Springer-Verlag*.

FU-00014349-2