

UNIVERSIDADE FEDERAL DE UBERLÂNDIA  
FACULDADE DE COMPUTAÇÃO  
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO



**ANÁLISE SOB RESTRIÇÕES BASEADA EM UM  
RACIOCÍNIO ENERGÉTICO PARA O PROBLEMA DE  
ESCALONAMENTO TEMPO REAL DE SISTEMAS DE  
GERENCIAMENTO DE WORKFLOW**

FLÁVIO FÉLIX MEDEIROS

Uberlândia - Minas Gerais

2009

# **Livros Grátis**

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.



UNIVERSIDADE FEDERAL DE UBERLÂNDIA  
FACULDADE DE COMPUTAÇÃO  
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO



FLÁVIO FÉLIX MEDEIROS

**ANÁLISE SOB RESTRIÇÕES BASEADA EM UM  
RACIOCÍNIO ENERGÉTICO PARA O PROBLEMA DE  
ESCALONAMENTO TEMPO REAL DE SISTEMAS DE  
GERENCIAMENTO DE WORKFLOW**

Dissertação de Mestrado apresentada à Faculdade de Computação da Universidade Federal de Uberlândia, Minas Gerais, como parte dos requisitos exigidos para obtenção do título de Mestre em Ciência da Computação.

Área de concentração: Engenharia de Software.

Orientador:

Prof. Dr. Stéphane Julia

Uberlândia, Minas Gerais

2009



UNIVERSIDADE FEDERAL DE UBERLÂNDIA  
FACULDADE DE COMPUTAÇÃO  
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Os abaixo assinados, por meio deste, certificam que leram e recomendam para a Faculdade de Computação a aceitação da dissertação intitulada “**Análise sob restrições baseada em um raciocínio energético para o problema de escalonamento tempo real de sistemas de gerenciamento de workflow**” por **Flávio Félix Medeiros** como parte dos requisitos exigidos para a obtenção do título de **Mestre em Ciência da Computação**.

Uberlândia, 3 de setembro de 2009

Orientador:

---

Prof. Dr. Stéphane Julia  
Universidade Federal de Uberlândia

Banca Examinadora:

---

Prof. Dr. Orides Morandin Junior  
Universidade Federal de São Carlos

---

Prof. Dr. Carlos Roberto Lopez  
Universidade Federal de Uberlândia



UNIVERSIDADE FEDERAL DE UBERLÂNDIA  
FACULDADE DE COMPUTAÇÃO  
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Data: setembro de 2009

Autor: **Flávio Félix Medeiros**  
Título: **Análise sob restrições baseada em um raciocínio energético para o problema de escalonamento tempo real de sistemas de gerenciamento de workflow**  
Faculdade: **Faculdade de Computação**  
Grau: **Mestrado**

Fica garantido à Universidade Federal de Uberlândia o direito de circulação e impressão de cópias deste documento para propósitos exclusivamente acadêmicos, desde que o autor seja devidamente informado.

---

Autor

O AUTOR RESERVA PARA SI QUALQUER OUTRO DIREITO DE PUBLICAÇÃO DESTE DOCUMENTO, NÃO PODENDO O MESMO SER IMPRESSO OU REPRODUZIDO, SEJA NA TOTALIDADE OU EM PARTES, SEM A PERMISSÃO ESCRITA DO AUTOR.





# Dedicatória

*Aos meus pais Valterson e Ilza, a minha irmã Régia, aos meus sobrinhos Laura  
Aparecida e João Paulo.*



# Agradecimentos

Agradeço...

A Deus, por minha vida.

A meus pais Valterson Gomes de Medeiros e Ilza Aparecida Félix de Medeiros pela dedicação, confiança, apoio, carinho e amor incondicional em todos os momentos.

A minha irmã Régia Félix Medeiros Fratari pelos conselhos, incentivos, carinho e amor durante toda esta etapa da minha vida.

A meus amigos que se mostraram companheiros e, diretamente ou indiretamente, contribuíram para a realização deste trabalho,

Principalmente ao professor Stéphane Julia pelo profissionalismo, apoio, paciência, amizade e orientação em todos os momentos da realização deste trabalho.



*“Há os que se queixam do vento. Os que esperam que ele mude. E os que procuram  
ajustar as velas.”  
(William G. Ward )*



# Resumo

Este trabalho propõem um método de análise sob restrições aplicado ao problema de escalonamento tempo real em sistemas de gerenciamento de *workflow*. O objetivo é filtrar as datas de início das atividades a fim de eliminar datas inconsistentes com soluções de escalonamento, bem como identificar escalonamentos inadmissíveis devido à incapacidade energética dos recursos envolvidos. O modelo adotado é uma rede de Petri p-temporal com mecanismo de alocação de recursos híbridos. O método proposto utiliza inicialmente um mecanismo de propagação de restrições temporais baseado nas durações dos casos e nos diferentes tipos de roteiros dos processos de *workflow*. Os diferentes tipos de recursos (discretos e contínuos) são incorporados ao modelo e uma abordagem baseada em um raciocínio energético é então aplicada. Posteriormente, é aplicado ao conjunto de restrições modificadas um mecanismo de inferência especializado, denominado jogador de redes de Petri, que busca obter em tempo real um cenário admissível correspondente a uma sequência específica de atividades que respeitem o conjunto das restrições temporais.

**Palavras chave:** sistemas de gerenciamento de workflow, redes de petri, análise sob restrição, raciocínio energético.





# Abstract

This work propose a method of analysis under constraints applied to the problem of real time scheduling in workflow management systems. The goal is to filter the start of activities in order to eliminate inconsistent dates with scheduling solutions and to identify inadmissible schedules due to the inability of energy resources involved. The adopted model is a p-time Petri net with hybrid resource allocation mechanism. The proposed method uses originally mechanism of time constraint propagation based on cases duration and different types of routings for workflow processes. The different types of resources (discrete and continuous) are incorporated into the model and an approach based on energetic reasoning is applied. On the set of modified constraints is then applied a specialized inference mechanism, called token player, seeking to obtain in real time an admissible scenario corresponding to a specific sequence of activities which respects all the time constraints.

**Keywords:** workflow management system, petri nets, constraint programming, energetic reasoning.



# Sumário

<b>Lista de Figuras</b>	<b>xix</b>
<b>1 Introdução</b>	<b>21</b>
<b>2 Sistemas de Gerenciamento de <i>Workflow</i></b>	<b>25</b>
2.1 Conceitos de Sistemas de Gerenciamento de <i>Workflow</i> . . . . .	25
2.2 Modelagem de processos de <i>Workflow</i> . . . . .	28
2.2.1 Workflow nets . . . . .	30
2.2.2 Redes de Petri p-temporais com mecanismo de alocação de recursos híbridos . . . . .	32
<b>3 Problema de Escalonamento</b>	<b>41</b>
3.1 Modelagem para o problema de escalonamento . . . . .	43
3.1.1 Grafos . . . . .	43
3.1.2 Redes de Petri temporais . . . . .	44
3.2 Abordagens para tratar o problema de escalonamento . . . . .	45
3.2.1 Algoritmos Branch and Bound . . . . .	45
3.2.2 Algoritmos Genéticos . . . . .	45
3.2.3 Análise Baseada em Restrições . . . . .	46
3.2.4 Jogadores de redes de Petri . . . . .	57
3.3 Problema de escalonamento no contexto dos sistemas de gerenciamento de <i>workflow</i> . . . . .	59
<b>4 Problema de escalonamento dos sistemas de gerenciamento de <i>workflow</i></b>	<b>61</b>
4.1 Mecanismos de propagação de restrições nos roteiros . . . . .	62
4.2 Análise sob restrições baseada em um Raciocínio Energético aplicado aos sistemas de gerenciamento de <i>workflow</i> . . . . .	78
4.2.1 Restrições do problema . . . . .	79
4.2.2 Intervalos de análise . . . . .	79
4.2.3 Aplicação do Raciocínio Energético . . . . .	80

---

<b>5</b>	<b>Estudo de Caso</b>	<b>101</b>
5.1	Apresentação . . . . .	101
5.2	Modelagem baseada em redes de Petri p-temporais . . . . .	101
5.2.1	Definição de Roteiros . . . . .	101
5.2.2	Alocação de Recursos . . . . .	102
5.2.3	Modelo Temporal . . . . .	103
5.3	Mecanismo de propagação de restrições . . . . .	104
5.4	Abordagem baseada no Raciocínio Energético . . . . .	106
5.5	Considerações sobre conflito no contexto das redes de Petri p-temporal e do jogador de redes de Petri . . . . .	129
<b>6</b>	<b>Conclusão</b>	<b>133</b>
	<b>Referências Bibliográficas</b>	<b>135</b>

# Lista de Figuras

2.1	Itens da representação gráfica do modelo Casati . . . . .	28
2.2	Tipos de Encadeamento de Atividades do modelo Casati . . . . .	29
2.3	Itens da Representação Gráfica do Diagrama de Atividades UML . . . . .	29
2.4	<i>Workflow net</i> ou Redes de <i>workflow</i> . . . . .	31
2.5	Serviço de reclamações . . . . .	34
2.6	Intervalos estáticos . . . . .	35
2.7	Modelo p-temporal com intervalos estáticos associado às atividades do <i>Serviço de reclamações</i> . . . . .	35
2.8	Intervalo de visibilidade . . . . .	36
2.9	Mecanismo de alocação de recurso contínuo . . . . .	37
2.10	Alocação de recurso discreto . . . . .	38
2.11	Alocação de recurso contínuo . . . . .	38
2.12	Modelo p-temporal com alocação de recursos híbridos para o processo do <i>Serviço de Reclamações</i> . . . . .	39
3.1	Exemplo da dinâmica das energias dentro de um intervalo $[r_i, d_i]$ . . . . .	49
3.2	Intervalo $\Delta=[3, 10]$ e restrições (datas de início ao mais cedo e final ao mais tarde) das atividades do conjunto $E$ . . . . .	51
3.3	Intervalos de execução da Atividade (a) dentro e fora do $\Delta$ . . . . .	51
3.4	Intervalo $\Delta=[0, 11]$ para o cálculo de consistência global considerando o conjunto $X$ . . . . .	53
3.5	Intervalo $\Delta=[20, 45]$ para o cálculo de consistência local considerando o início ao mais cedo da atividade ( $f$ ) do conjunto $Y$ . . . . .	55
3.6	Intervalo $\Delta=[17, 37]$ para o cálculo de consistência local considerando o início ao mais tarde da atividade ( $a$ ) do conjunto $Y$ . . . . .	57
3.7	Jogador de redes de Petri p-temporal . . . . .	58
4.1	Cálculo das bordas mínimas de um roteiro sequencial . . . . .	63
4.2	Cálculo das bordas máximas de um roteiro sequencial . . . . .	63
4.3	Cálculo das bordas mínimas de um roteiro paralelo . . . . .	67
4.4	Cálculo das bordas máximas de um roteiro paralelo . . . . .	67

4.5	Cálculo das bordas mínimas de uma sincronização de dois roteiros . . . . .	71
4.6	Cálculo das bordas máximas de uma sincronização de dois roteiros . . . . .	71
4.7	Cálculo das bordas mínimas de um roteiro alternativo . . . . .	75
4.8	Cálculo das bordas máximas de um roteiro alternativo . . . . .	75
4.9	Rede de Petri p-temporal que modela um fragmento da rotina de trabalho de um <i>Analista de Service Desk</i> . . . . .	81
4.10	Atualização da data de início ao mais cedo da atividade <i>A6</i> do <i>Analista de Service Desk</i> . . . . .	90
4.11	Atualização da data de início ao mais tarde das atividades <i>A4</i> e <i>A5</i> do <i>Analista de Service Desk</i> . . . . .	98
5.1	Modelo de rede de Petri para o <i>Serviço de Reclamações</i> . . . . .	102
5.2	Modelo de rede de Petri com alocação de recursos híbridos para o <i>Serviço de Reclamações</i> . . . . .	103
5.3	Modelo de rede de Petri p-temporal com alocação de recursos híbridos para o <i>Serviço de Reclamações</i> . . . . .	104
5.4	Rede com as datas mínimas e máximas para o início e término dos casos .	105
5.5	Propagação das restrições: <i>Serviço de Reclamações</i> . . . . .	106
5.6	Fragmento da rede de Petri apresentando as atividades envolvidas com o recurso <i>Funcionário - reclamações</i> . . . . .	120
5.7	Deslocamento da data de início ao mais cedo da atividade <i>A3</i> relacionada ao caso 4 . . . . .	127
5.8	Fragmento da rede de Petri após a atualização da data de início ao mais cedo da atividade <i>A3</i> relacionada ao caso 4 . . . . .	127
5.9	Aplicação da abordagem baseada no Raciocínio Energético sobre o processo do <i>Serviço de reclamações</i> - Antes da aplicação (Figura (a)) e depois da aplicação da abordagem (Figura (b)) . . . . .	128
5.10	Fragmento de rede de Petri p-temporal envolvido em conflito . . . . .	130
5.11	Disparo de $t_{14}$ na data $\delta=45$ . . . . .	130

# Capítulo 1

## Introdução

Nos sistemas baseados em processos, é comum a ocorrência de problemas relacionados com a má distribuição de atividades e a falta de planejamento e controle do trabalho. Alguns desses problemas são: dificuldade de acompanhamento e de coordenação das atividades dos processos além de atrasos na entrega de produtos ou serviços.

Para que as organizações tenham um melhor controle sobre o seu fluxo de trabalho, os Sistemas de Gerenciamento de *Workflow* (*WfMS* - *Workflow Management System*) foram introduzidos. Eles ajudam as organizações a atingirem seus objetivos de coordenação, comunicação e cooperação entre seus membros, uma vez que gerenciam a sequência das atividades do trabalho e chamam os recursos humanos ou eletrônicos apropriados para a execução das atividades.

Os sistemas de gerenciamento de *workflow* são sistemas de tecnologia da informação, portanto é necessário que suas funcionalidades sejam implementadas em um *software*. Como alguns problemas advindos do processo de desenvolvimento (não cumprimento dos prazos, desperdício de tempo e dinheiro) podem causar sérios impactos em uma organização, é preciso aplicar aos sistemas de gerenciamento de *workflow* os fundamentos da Engenharia de *Software*, para evitar a ocorrência desses problemas. Os fundamentos da Engenharia de *Software* têm como principal objetivo aplicar técnicas de engenharia (modelar antes de realizar; estimar diversos fatores antes de avançar; medir antes, durante e depois do produto realizado; analisar fatores de risco) para o desenvolvimento de *software*. Em particular, no contexto de desenvolvimento de *software*, uma das técnicas fundamentais é a modelagem. Com ela, é possível construir modelos que representem a estrutura e o comportamento desejado para o sistema.

Entre todas as notações usadas para a modelagem de processos de *workflow*, as redes de Petri são uma das mais adaptadas [Aalst & Hee, 2002], porque representam formalmente os roteiros básicos presentes nesses processos. Além disso, elas podem ser usadas para especificar características de tempo real e mecanismos de alocação de recursos em sistemas de gerenciamento de *workflow*. De fato, entregas atrasadas em uma organização, muitas vezes, devem-se à indisponibilidade de recursos necessários para a execução das tarefas;



por isso o modelo usado no sistema de gerenciamento de *workflow* deve considerar um mecanismo de alocação de recursos.

Este trabalho está focado no problema de escalonamento em tempo real dos sistemas de gerenciamento de *workflow*. Mais especificamente, ele tem como um dos seus objetivos pré-identificar escalonamentos inadmissíveis por causa da indisponibilidade dos recursos alocados.

A principal diferença entre o tradicional problema de escalonamento dos sistemas de produção e do problema de escalonamento dos sistemas de gerenciamento de *workflow* é a natureza dos recursos usados para tratar as atividades. Nos sistemas de produção, os recursos representam equipamentos físicos, retratados por simples fichas nos lugares de uma rede de Petri, por serem recursos do tipo discreto. Já no caso dos sistemas de gerenciamento de *workflow*, os recursos podem representar tanto equipamentos físicos quanto funcionários humanos. A maioria dos recursos humanos pode tratar de diversas atividades simultaneamente. Dessa forma, não podem ser representados por simples fichas nos lugares de uma rede de Petri; nesse caso, eles são representados por um número real que indica sua disponibilidade.

No trabalho de Oliveira [de Oliveira, 2005], foi apresentada uma proposta para o problema de escalonamento tempo real dos sistemas de gerenciamento de *workflow*. O principal objetivo foi o de propor uma solução para o problema baseada numa rede de Petri p-temporal híbrida. Mais especificamente, o objetivo era o de se obter uma sequência de atividades que respeitasse as restrições temporais e as restrições de alocação dos recursos compartilhados. As restrições temporais foram propagadas ao longo dos roteiros dos processos de *workflow* por meio de um mecanismo de propagação de restrição baseado em técnicas de propagação de restrição classicamente utilizadas em problemas de escalonamento baseados em grafos sem circuitos [Esquirol & Lopez, 1999]. Assim cada ficha, em um lugar de espera da rede de Petri p-temporal, ficava associada a um intervalo de data que fixava o início de cada atividade.

O principal objetivo desta dissertação é aplicar a técnica de Raciocínio Energético, derivada da disciplina de Inteligência Artificial, sobre os recursos e sobre as atividades dos processos de *workflow*, a fim de buscar uma atualização dos intervalos de datas de início das atividades que melhorem os resultados apresentados no trabalho de Oliveira [de Oliveira, 2005]. O Raciocínio Energético deverá em particular identificar datas de início de execução das atividades inconsistentes, as quais deverão ser excluídas a fim de melhorar a base de conhecimento que define o conjunto de restrições que um jogador de redes de Petri deverá respeitar durante a elaboração de uma solução de escalonamento em tempo real.

O capítulo 2 será dedicado aos principais fundamentos teóricos para os sistemas de gerenciamento de *workflow*, tais como: conceitos envolvidos, tipos de *workflow* e modelagem de *workflow*.

O capítulo 3 mostrará o problema clássico de escalonamento bem como algumas mo-

delagens e abordagens utilizadas para tratar o problema. Mais especificamente o foco será a abordagem baseada no Raciocínio Energético e a abordagem baseada no jogador de redes de Petri p-temporal.

O capítulo 4 abordará o problema de escalonamento dos sistemas de gerenciamento de *workflow* apresentando o mecanismo de propagação de restrições proposto por Oliveira [de Oliveira, 2005] bem como a abordagem baseada no Raciocínio Energético aplicada no contexto dos sistemas de gerenciamento de *workflow*.

Finalmente, no capítulo 5, aplicar-se-á a abordagem aqui descrita num exemplo de processo de *workflow* denominado *Serviço de Reclamações*.

Na conclusão deste trabalho, serão expostas as contribuições e os resultados obtidos.



## Capítulo 2

# Sistemas de Gerenciamento de *Workflow*

Atualmente, em virtude das grandes e rápidas mudanças, os sistemas de informação precisam apoiar os processos de negócio de forma mais integrada, para agregar mais qualidade e valor aos serviços prestados pelas organizações. Os processos de negócio consistem em conjuntos de atividades relacionadas que, de forma coletiva, visam a atingir um objetivo bem definido dentro do contexto de uma estrutura organizacional [Hollingsworth, 1985]. Dessa forma, faz-se necessário controlar, monitorar e apoiar os aspectos lógicos dos processos de negócio e, principalmente, gerenciar o fluxo de trabalho por meio da organização. Nesse contexto, as organizações se veem na necessidade de adquirir um novo modelo de trabalho, que é acompanhado pela adoção de novos sistemas de informação como, por exemplo, os sistemas de gerenciamento de *workflow*.

O conceito de *Workflow* corresponde à automação de um processo de negócio, no todo ou parcial, em que documentos, informações ou tarefas são passadas de um participante para o outro de acordo com um conjunto de regras procedimentais [Plesums, 2002], tendo como principal objetivo minimizar o problema da coordenação do trabalho.

### 2.1 Conceitos de Sistemas de Gerenciamento de *Workflow*

Os sistemas de gerenciamento de *workflow* tornaram-se um componente padrão para sistemas de informação empresariais. De acordo com [Aalst & Hee, 2002], [Alonso *et al.*, 1997], os sistemas de gerenciamento de *workflow* são pacotes de software genéricos para o gerenciamento de processos de negócio. Tais sistemas são responsáveis por prover a gestão desses processos, efetuar a alocação de recursos necessários para a execução de diversas atividades e, além disso, fazer com que a informação certa chegue até o recurso correto no momento certo. Portanto, os sistemas de gerenciamento de *workflow* não são

responsáveis por executar as tarefas dos processos, mas, sim gerenciá-las.

Vários conceitos são atribuídos a *workflow*, porém, em geral, todos contemplam a ideia de que *workflow* é um processo de negócio operacional. É importante ressaltar que, apesar de a definição de *workflow* ser, tradicionalmente, ligada a processos de negócios, seus princípios e ferramentas podem ser aplicados a diversas atividades nas quais a coordenação do trabalho seja exigida [Plesums, 2002]. A distribuição das atividades pode ser feita automaticamente para pessoas, equipamentos ou sistemas computacionais. Sistemas que manipulam *workflow* são ferramentas poderosas para a organização de tarefas complexas, podendo ser usados para realizar desde o sequenciamento de DNA [Meidanis *et al.*, 1996] até pesquisas científicas para comprovação dos princípios da teoria da relatividade, por exemplo, a detecção de ondas gravitacionais [Gil *et al.*, 2004].

Os sistemas de gerenciamento de *workflow* são baseados em alguns conceitos fundamentais como [Aalst & Hee, 2002]:

- **caso:** o principal objetivo de um sistema de gerenciamento de *workflow* é lidar com casos. Um paciente em um hospital, um pedido, ou uma reclamação são exemplos de casos. Cada caso tem uma única identidade, um tempo de vida limitado e envolve a execução de um processo;
- **processo:** um processo consiste em um número de tarefas que precisam ser executadas e um conjunto de condições que determinam a ordem das tarefas;
- **tarefa:** uma tarefa é uma unidade lógica de trabalho executada de uma só vez por um recurso, ou seja uma tarefa é indivisível e não pode ser interrompida;
- **recurso:** um recurso é um nome genérico para uma pessoa, uma máquina ou um grupo de pessoas ou máquinas que podem executar tarefas específicas;
- **atividade:** uma atividade é a execução de uma tarefa realizada por um recurso; ela pode ser manual ou automatizada;
- **evento:** algo que ocorre em um determinado instante de tempo específico, por exemplo, falha em um servidor de banco de dados, estouro de pilha de um compilador;
- **gatilho:** um processo só pode ser iniciado se o estado do caso em questão permitir, mas a efetiva execução do processo, frequentemente, precisa ser disparada por um gatilho que pode ser um recurso, a ocorrência de um evento externo, ou a chegada de um exato momento;
- **roteiro:** um processo pode seguir um roteiro sequencial (quando duas ou mais tarefas são executadas em sequência), pode seguir um roteiro paralelo (quando duas ou mais tarefas são executadas simultaneamente), ou pode seguir um roteiro seletivo (quando uma escolha precisa ser feita entre duas ou mais tarefas). Alguns processos também podem incluir iteração ou repetição.

Para cada organização, processos de negócios representam características próprias. Nesse sentido, um modelo de *workflow* deve representar, de forma realística, as informações qualitativas e quantitativas da entidade em estudo. A identificação dos tipos de processos de *workflow* permitirá que seja facilitada a seleção de um modelo representativo para o *workflow*. Alguns dos vários tipos de *workflow* são apresentados a seguir [Nicolao, 1998]:

- **Ad-hoc**<sup>1</sup>: são processos que possuem um fluxo de trabalho pouco estruturado. As tarefas e o fluxo de interação entre elas são normalmente imprevisíveis ou desconhecidas até o momento da execução. Não há um padrão pré-determinado de movimentação das atividades entre pessoas. A ordenação e a coordenação de tarefas em *workflow ad-hoc* são controladas por humanos. Estes sistemas estão voltados para grupos dinâmicos que executam processos únicos e altamente individualizados. Tipicamente, envolvem pequenos grupos de profissionais que têm a intenção de apoiar pequenas atividades que requerem uma solução rápida. Exemplos desses tipos de processos são aqueles que envolvem a produção de conhecimento como a elaboração de um relatório, revisão de um livro ou artigo, ou a construção de um sistema.
- **Administrativos**: gerenciam processos repetitivos que possuem maior previsibilidade na interação de suas tarefas e processos com maior grau de estruturação. Sua ordenação e coordenação podem ser facilmente automatizadas. Geralmente apoiam processos administrativos que podem mudar bastante de uma organização para outra, e são concebidos para o roteamento inteligente de formulários por meio da organização (praticamente todos os indivíduos podem ser seus usuários). Um exemplo desse tipo de *workflow* pode ser um pedido de compra de materias que, geralmente, é solicitado por uma pessoa uma vez por semana e circula por diversas áreas que vão, gradativamente, preenchendo as informações no documento eletrônico até a finalização do processo.
- **Produção**: são *workflows* que seguem uma estruturação rígida e bem definida. Nesse tipo de *workflow*, as regras de interação do processo devem ser previamente levantadas pela análise do processo. Estes processos são previsíveis e repetem-se com alta frequência. Sendo assim, possuem uma forte dependência com outros sistemas da organização, por exemplo, banco de dados. *Workflow* de produção difere do administrativo por um processamento de informações complexas envolvendo múltiplos sistemas. Parte das tarefas é executada por agentes de *software* em vez de atores humanos. Os Processos de atendimento a clientes em um *Call Center* são um exemplos típico desse tipo de *workflow*.

---

<sup>1</sup>Ad-Hoc é uma expressão em latim que significa *para isto*

De acordo com [Hollingsworth, 1985], os sistemas de gerenciamento de *workflow* precisam suportar funções referentes à definição e, possivelmente, modelagem dos processos de negócio e das atividades que constituem esses processos, funções de controle em tempo de execução referentes ao gerenciamento dos processos de *workflow* em um ambiente operacional, bem como o sequenciamento das várias atividades de cada processo, e interações em tempo de execução entre usuários humanos e ferramentas de Tecnologia da Informação, para processar as diversas atividades.

Os sistemas de gerenciamento de *workflow* implicam que os projetos relativos a sua adoção e utilização apresentem uma coordenação perfeita entre os processos de negócios e a tecnologia utilizada. No entanto, para que o projeto seja bem sucedido, torna-se obrigatório que toda a metodologia de implementação seja conduzida, cuidadosamente, prestando atenção em diferentes aspectos de ordem institucional. Em particular, as organizações devem apostar não só no desenvolvimento e implementação mas também na modelagem dos processos de negócios.

## 2.2 Modelagem de processos de *Workflow*

Para que as organizações possam ter um controle total dos seus processos por meio dos sistemas de gerenciamento de *workflow*, é necessário que, primeiramente, se desenvolva uma modelagem bem estruturada desses processos. Técnicas de modelagem de processos de *workflow* devem minimizar os problemas de coordenação do trabalho nos processos de negócios, tendo em vista o comportamento dinâmico dos processos e oferecer os recursos necessários para representar o fluxo de trabalho no processo modelado.

Muitos trabalhos foram relacionados à modelagem de processos de *workflow*. Cada um utilizava sua própria técnica de modelagem, fazendo-se perceber que não há um modelo conceitual comum, aceito e usado por todos na área de *workflow*.

Um modelo de *workflow* que responde as questões de quando, como, onde e por quem cada atividade é executada, é uma representação gráfica ou textual de um conjunto de atividades e o relacionamento existente entre elas.

Um dos modelos usados para representação gráfica de *workflow* é o modelo *Casati* [Casati *et al.*, 1998]. A figura 2.1 ilustra alguns itens desse modelo. O modelo Casati possibilita a representação das atividades e o encadeamento entre elas. Os tipos de encadeamento são mostrados na figura 2.2.



Figura 2.1: Itens da representação gráfica do modelo Casati

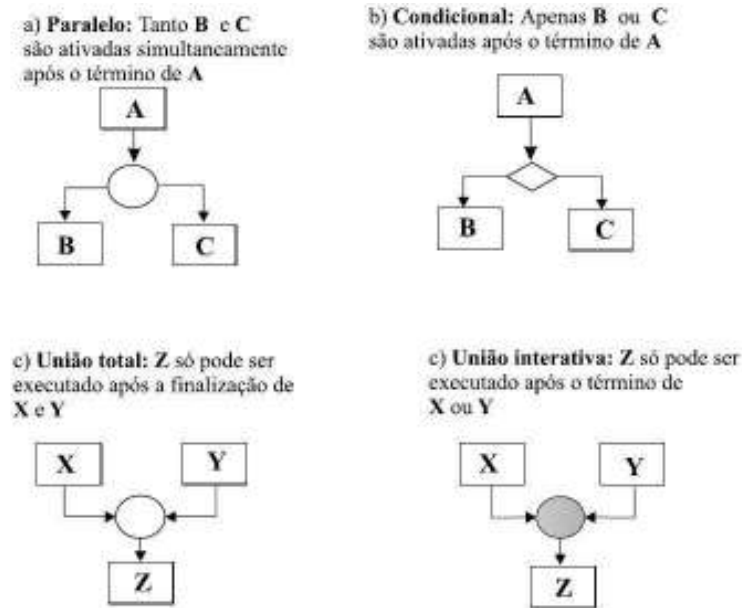


Figura 2.2: Tipos de Encadeamento de Atividades do modelo Casati

Os diagramas de atividades da UML, [Dumas & Hofstede, 2001], também podem ser usados para modelagem de processos de *workflow*. A figura 2.3 ilustra alguns itens do diagrama de atividades da UML. Tanto o modelo de Casati quanto diagramas de atividades mostram os principais roteiros existentes em um processo de *workflow*. Apesar disso, esses modelos não permitem representar, de maneira formal, os mecanismos de alocação de recursos e restrições temporais, quando são utilizados para a especificação de características de tempo real dos sistemas de gerenciamento de *workflow*.

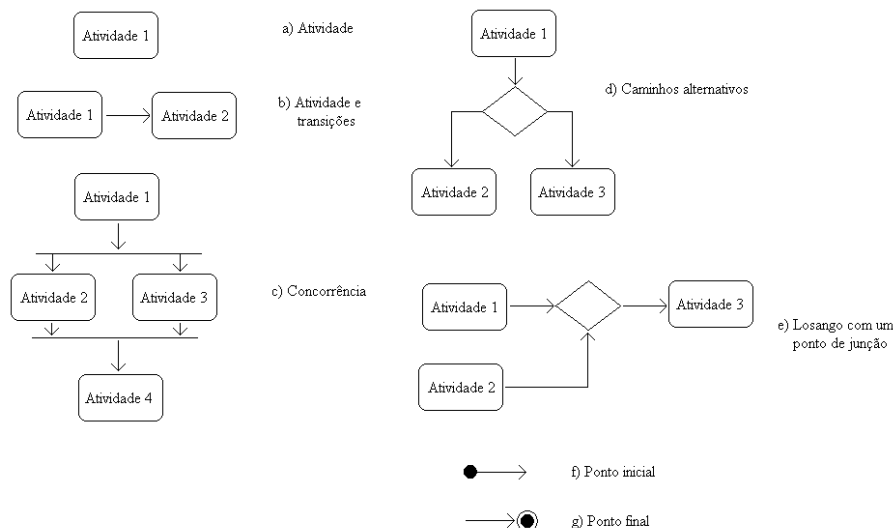


Figura 2.3: Itens da Representação Gráfica do Diagrama de Atividades UML

Alguns autores como [Eshuis, 2002], [Hruby, 1998] utilizaram as notações da UML para especificar processos de *workflow*. Contudo, como apresentado anteriormente, as notações da UML possuem suas limitações, quando são usadas para especificar as características



de tempo real dos sistemas de gerenciamento de *workflow*. Por exemplo, os diagramas de atividades não representam as restrições de tempo real de uma maneira formal, além do mais, não mostram, de maneira explícita, os mecanismos de alocação de recursos. Como consequência disso, ocorrem entregas atrasadas nas organizações, uma vez que a maioria dos problemas de atrasos de entrega ocorre por causa da sobrecarga dos recursos. A modelagem de processos de *workflow* deve considerar os mecanismos de alocação de recursos, em particular, o gerenciamento do tempo dos processos de *workflow* é crucial para melhorar a eficiência dos processos de negócio dentro de uma organização.

### 2.2.1 Workflow nets

De acordo com [Aalst & Hee, 2002], o uso das redes de Petri na modelagem de processos de *workflow* tem uma série de vantagens importantes, como o fato de forçar uma definição precisa dos processos em contraste com a maioria das técnicas de diagramação informais, por exemplo, o uso de diagramas de atividades da UML [Dumas & Hofstede, 2001].

Em especial, as redes de Petri são bem adaptadas para definição e análise de sistemas complexos, sendo apropriadas para modelagem de processos e sistemas de gerenciamento de *workflow*.

Alguns modelos baseados em redes de Petri foram definidos, exclusivamente, para a representação de *workflow*, como é o caso das *workflow nets* [Aalst & Hee, 2002].

As *Workflow nets* ou redes de *workflow*, [Aalst & Hee, 2002], [Li *et al.*, 2003], possuem apenas um lugar de início ( $\varepsilon$ ) e um lugar de término ( $\theta$ ), indicando o início e o fim do processo de negócio modelado. Uma ficha, no lugar de início, representa um caso que precisa ser tratado e uma ficha, no lugar de término, representa um caso que foi tratado. Toda tarefa é associada a uma transição, e toda condição é associada a um lugar, de modo que toda tarefa e toda condição devem estar em um caminho entre o lugar de início e o lugar de término. Tais redes são adequadas para representação, validação e verificação de processos de *workflow*.

A figura 2.4 apresenta uma *workflow net* que modela o processo de um *Serviço de Reclamações* de acordo com o modelo proposto por Aalst [Aalst & Hee, 2002]. Nesse processo, primeiro uma reclamação é registrada; em seguida, são contactados o cliente que fez a reclamação e o departamento afetado pela reclamação. O cliente é contactado com o objetivo de se obter mais informações. O departamento é informado sobre a reclamação e é questionado sobre uma reação inicial. Essas duas atividades devem ser executadas em paralelo. Depois disso, as informações são coletadas e, em sequência, uma decisão é tomada. Dependendo da decisão tomada, ou um pagamento é feito, ou uma carta é enviada ao cliente. Por fim, a reclamação é arquivada.

Na figura 2.4, a ficha no lugar *início*, representa um caso a ser tratado. Assim que a transição *registrar* for disparada, irá existir uma ficha no lugar E1 e uma outra ficha no

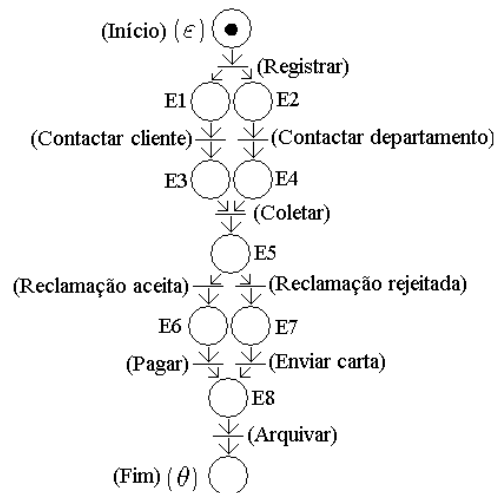


Figura 2.4: *Workflow net* ou Redes de *workflow*

lugar E2, representando o mesmo caso. Quando um caso está sendo tratado, a quantidade de fichas referente a ele pode variar; porém, quando apenas uma ficha se encontra no lugar *fim*, significa que o caso representado foi completado. Em princípio, cada processo deve cumprir dois requisitos:

- Em qualquer lugar da rede deve ser possível alcançar o lugar de término, onde haverá apenas uma ficha neste lugar.
- Quando houver uma ficha no lugar de término, todas as demais fichas referentes a esse caso não deverão mais existir.

Esses dois requisitos garantem que todos os casos iniciados no lugar de início da rede serão devidamente concluídos.

O principal critério de corretude definido para as *workflow nets* é o *Soundness* [Aalst *et al.*, 1994], [Passos, 2009]. Uma *workflow net* é *sound* se, e somente se, os três requisitos abaixo forem satisfeitos:

- Para cada ficha colocada no lugar de início, uma, e apenas uma, ficha aparecerá no lugar de término.
- Quando uma ficha aparecer no lugar de término, todos os outros lugares estarão vazios, considerando o caso em questão.
- Considerando uma tarefa associada a uma transição, é possível evoluir da marcação inicial até uma marcação que sensibiliza tal transição, ou seja, não deve haver nenhuma transição *morta* na *workflow net*.

Em [Aalst *et al.*, 1994], são apresentados três métodos para provar a propriedade *soundness*: o primeiro método é baseado na construção de grafos de alcançabilidade,

o segundo é baseado na análise das propriedades *liveness* e *boundness* para uma rede *short-circuited*; e o terceiro é baseado na substituição de blocos bem formados.

Embora, muitos trabalhos utilizem as redes de Petri para a modelagem de processos e sistemas de gerenciamento de *workflow*, grande parte deles possuem limitações no que se refere ao problema de alocação de recursos. Em [Aalst & Hee, 2002], por exemplo, a modelagem de alocação de recursos é feita apenas de modo informal.

### 2.2.2 Redes de Petri p-temporais com mecanismo de alocação de recursos híbridos

As redes de Petri são adequadas para modelar processos de *workflow* já que as mesmas são bem adaptadas para definição e análise de sistemas complexos [Aalst *et al.*, 1994]. Elas permitem uma boa representação de situações de conflito, compartilhamento de recursos, restrições de precedência, comunicações síncronas e assíncronas e explícitas restrições temporais no caso das redes de Petri temporais.

As *workflow nets* apresentadas anteriormente podem ser estendidas com informações de tempo para representar as restrições temporais, dando origem às *workflow nets* com restrições temporais [Li *et al.*, 2003].

Alguns trabalhos [Aalst *et al.*, 1994], [Aalst & Hee, 2002], [Li *et al.*, 2003], [del Foyo & Silva, 2003] sobre modelagem de sistemas de gerenciamento de *workflow* utilizaram as redes de Petri; contudo a maioria deles considera mais a dimensão dos processos relacionados ao sequenciamento das atividades. Esses trabalhos, no entanto, possuem algumas limitações no que se refere ao problema de alocação de recursos.

Em [de Oliveira, 2005], foi proposto um modelo de rede de Petri p-temporal com um mecanismo de alocação de recursos híbridos como solução para o problema de escalonamento dos sistemas de gerenciamento de *workflow*. A abordagem utilizou o diagrama de atividades da UML para mostrar as principais atividades e os diferentes tipos de roteiros dos processos. Baseado no diagrama de atividades obtido, o modelo de rede de Petri p-temporal correspondente foi produzido, atribuindo-se um intervalo de tempo a cada atividade dos processos. Os mecanismos de alocação de recursos híbridos (discretos e contínuos) foram modelados por uma rede de Petri híbrida com transições discretas a fim de se representar, de forma mais realística, os diferentes tipos de recursos presentes em sistemas de gerenciamento de *workflow* (equipamentos e recursos humanos).

Em [Jeske, 2006], foi proposto um modelo de rede de Petri p-temporal, com um mecanismo de alocação de recursos híbridos *fuzzy*, como solução para o problema de alocação de recursos humanos em sistemas de gerenciamento de *workflow*. Inicialmente, uma rede de Petri ordinária foi utilizada para mostrar as principais atividades do sistema e os diferentes roteiros do processo de *workflow* estudado. Mecanismos de alocação de recursos híbridos foram modelados por redes de Petri híbridas *fuzzy* com transições discretas em

que recursos discretos representavam equipamentos, e recursos contínuos representavam a disponibilidade de um funcionário. Para expressar, de forma mais realista, o mecanismo de alocação de recursos em que o comportamento humano é considerado, Jeske associou conjuntos *fuzzy* delimitados pela distribuição de possibilidade na forma triangular com as marcações dos lugares que representam a disponibilidade humana. Dessa forma, novas regras de disparo e evolução das marcações *fuzzy* do novo modelo foram definidas.

No contexto da teoria das redes de Petri, já foram apresentados vários modelos que consideram restrições de tempo explícitas.

Nos modelos de redes de Petri p-temporizadas, o tempo é representado por durações associadas aos lugares do modelo [Sifakis, 1977], [Tazza, 1987], enquanto nas redes de Petri t-temporizadas, o tempo é representado por durações (números racionais positivos ou nulos) associados às transições [Ramchandani, 1974]. Nas redes de Petri t-temporais [Merlin, 1974], [Menasche, 1982], o tempo é representado por um intervalo  $[\theta_{min}, \theta_{max}]$  associado a cada transição, sendo que a duração de sensibilização deve ser maior que  $\theta_{min}$  e menor que  $\theta_{max}$ , ou seja, a transição só pode ser disparada dentro do intervalo de tempo  $[\theta_{min}, \theta_{max}]$ . Já nas redes de Petri p-temporais, o tempo é representado por um intervalo de tempo  $[\theta_{min}, \theta_{max}]$  associado ao lugar  $p$ , sendo que  $\theta_{min}$  representa a data mínima depois da qual uma ficha que está no lugar  $p$  torna-se disponível para o disparo de uma transição, e  $\theta_{max}$  representa a data máxima antes da qual a ficha em  $p$  deve ser utilizada para o disparo de uma transição. Depois de  $\theta_{max}$  a ficha torna-se *morta*, caracterizando uma violação de restrição e um funcionamento anormal do sistema. A principal vantagem das redes de Petri p-temporais é a capacidade de expressar durações de atividades não completamente determinísticas.

### Redes de Petri p-temporais para modelagem de sistemas de gerenciamento de *workflow*

Nas redes de Petri p-temporais, ao se representar atividades sequenciais (roteiro sequencial), faz-se necessário acrescentar lugares de espera entre elas. Isso ocorre porque, ao final de uma atividade, a próxima só poderá ser iniciada se o recurso correspondente estiver disponível. Atividades alternativas (roteiro alternativo) tomadas com base em alguma expressão booleana e atividades concorrente (roteiro paralelo) são igualmente comuns em *workflow*.

O *Serviço de Reclamações*, apresentado em [Aalst & Hee, 2002], será utilizado neste trabalho para ilustrar a modelagem de processos de *workflow* usando redes de Petri p-temporais.

A figura 2.5 ilustra os diversos tipos de roteiros do processo do *Serviço de Reclamações*. Os lugares  $E_j$  com  $j=0$  até 10 representam lugares de espera entre atividades. As atividades são associadas aos lugares  $A_i$  com  $i=1$  até 8. Os lugares  $E_0$  e  $E_{10}$  representam o início e o fim do processo, e a ficha em  $E_0$  representa um caso a ser tratado.

O tempo real necessário para a execução de uma atividade em um sistema de gerenciamento de *workflow* é não-determinístico. Assim, um intervalo de tempo pode ser atribuído a cada atividade. De acordo com [Julia & Valette, 2000], restrições de tempo explícitas podem ser especificadas formalmente usando um modelo de rede de Petri p-temporal.

A definição estática de uma rede de Petri p-temporal [Khansa, 1997] é baseada em intervalos estáticos que representam o intervalo de permanência das fichas nos lugares, do ponto de vista das durações das atividades.

**Definição 2.1. (Redes de Petri p-temporal)** Uma rede de Petri p-temporal [Khansa, 1997] é dada pela dupla

$\langle R; IE \rangle$  onde:

- R é uma rede de Petri marcada
- $IE: \rightarrow (Q^+ \cup 0) \times (Q^+ \cup \infty)$   
 $p_i \rightarrow IE_i = [a_i, b_i]$  com  $0 \leq a_i \leq b_i$   
 $IE_i$  representa o intervalo estático de tempo de permanência da ficha no lugar  $p_i$   
 $Q^+$  representa o conjunto dos números racionais positivos.

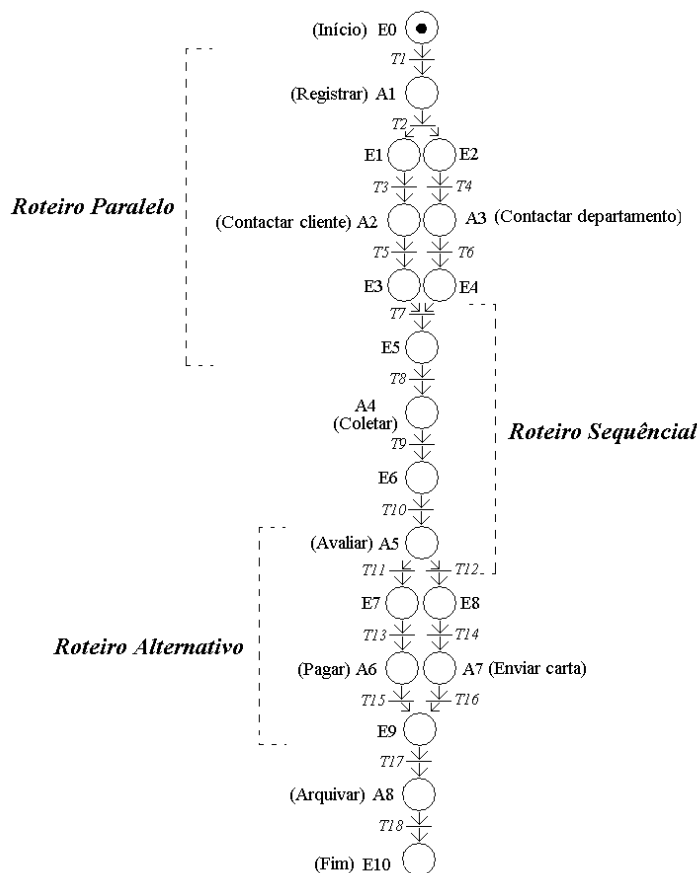


Figura 2.5: Serviço de reclamações

Na rede da figura 2.6, intervalos estáticos são associados aos lugares que representam as atividades. Por exemplo, o intervalo estático  $[10, 20]e$  associado ao lugar que representa a atividade A1 significa que o recurso que executará essa atividade precisará de, no mínimo, 10 unidades de tempo e, no máximo, 20 unidades de tempo.

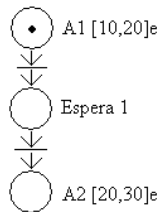


Figura 2.6: Intervalos estáticos

A figura 2.7 ilustra os intervalos estáticos vinculados às atividades do processo *Serviço de Reclamações*. Para cada tarefa  $A_i$  aparece um intervalo estático que especifica a sua duração mínima e máxima. Em especial, os intervalos estáticos associados às tarefas *Coletar Informações* em A4 e *Arquivar* em A8 são iguais a  $[0, 0]$ , porque as durações de tais tarefas são desprezíveis se comparadas com as outras tarefas desse processo.

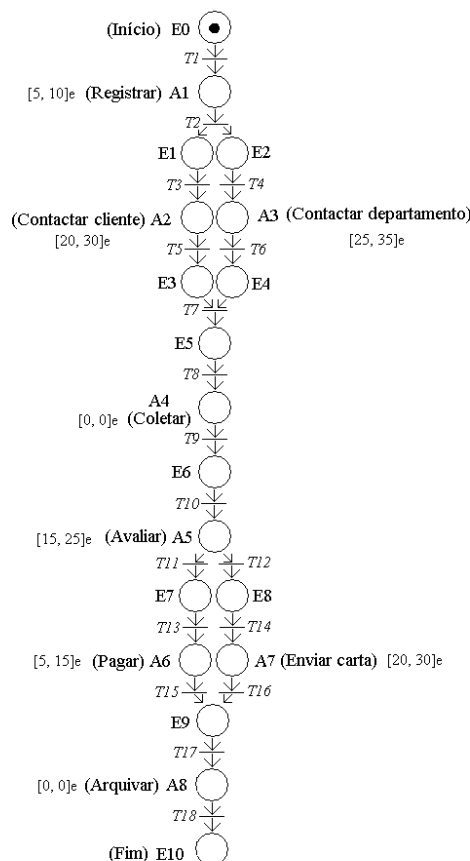


Figura 2.7: Modelo p-temporal com intervalos estáticos associado às atividades do *Serviço de reclamações*

Uma ficha pertencente a um lugar  $p_i$  participará da sensibilização de uma transição de saída de  $p_i$ , se ela tiver permanecido lá, pelo menos, durante a borda mínima e, no

máximo, durante a borda máxima do intervalo associado a  $p_i$ . Enquanto a ficha não alcançar o tempo mínimo de permanência, ela ficará indisponível para o disparo de outras transições. Dentro do limite  $[a, b]$ , a ficha ficará disponível; após o limite máximo de permanência, a ficha tornar-se-á *morta* e não poderá disparar nenhuma transição. De acordo com [de Oliveira, 2005], a morte de uma ficha indica a violação de uma restrição temporal.

O comportamento dinâmico de uma rede de Petri p-temporal depende da marcação da rede e da situação temporal das fichas que é dada pelo intervalo de visibilidade.

**Definição 2.2. (Intervalo de visibilidade [de Oliveira, 2005])** Um intervalo de visibilidade  $[(\delta_p)_{mim}, (\delta_p)_{max}]$  associado a uma ficha em um lugar  $p$  de uma rede de Petri p-temporal define a data mínima  $((\delta_p)_{mim})$  em que a ficha se torna disponível em  $p$  para o disparo de uma transição de saída de  $p$  (data de início ao mais cedo de uma atividade), bem como a data máxima  $((\delta_p)_{max})$  após a qual a ficha se torna indisponível (morta) e não pode ser utilizada para o disparo de nenhuma transição (data de início ao mais tarde de uma atividade).

Por exemplo, na figura 2.8, se a data de disparo da transição de entrada do lugar  $O_{11}$  é  $\delta=3$ , sabendo que o intervalo estático desse lugar é  $[5, 9]_e$ , então, o intervalo de visibilidade dessa ficha é  $[5 + 3, 9 + 3]_v = [8, 12]_v$ .

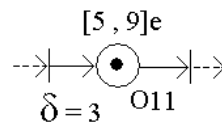


Figura 2.8: Intervalo de visibilidade

### Mecanismos de alocação de recursos híbridos

Para que as atividades de um processo de *workflow* sejam executadas, faz-se necessária a representação de alocação de recursos. Existem diferentes tipos de recursos usados para realizar as atividades em sistemas de gerenciamento de *workflow*. Alguns dos recursos usados podem ser considerados do tipo discreto, podendo ser representados por simples fichas como, por exemplo, uma impressora usada para tratar documentos. Porém alguns recursos, como a maioria dos recursos do tipo humano, não podem ser representados por simples fichas. Por exemplo, em uma agência de seguros, um funcionário pode tratar vários casos durante sua jornada diária de trabalho, não necessariamente de maneira puramente sequencial. Tais recursos são do tipo contínuo.

Em [de Oliveira, 2005], os recursos do tipo contínuo são representados por um número real que representa a disponibilidade de um funcionário humano. Por exemplo, 100% representa a disponibilidade máxima quando o recurso se encontra no estado completamente disponível. Da mesma forma, aos pesos dos arcos de saídas e entradas dos recursos, são

atribuídos valores correspondentes à previsão da disponibilidade exigida para a realização da atividade relacionada.

A figura 2.9 ilustra um exemplo de alocação de um recurso do tipo contínuo. Considere-se uma enfermeira em um hospital, que precisa cuidar de quatro pacientes e auxiliar em uma cirurgia. A enfermeira está representada pelo número 100, que é equivalente a sua disponibilidade, sendo que 25% deste valor está alocado para cuidar de cada paciente e 100% está alocado para auxiliar em uma cirurgia.

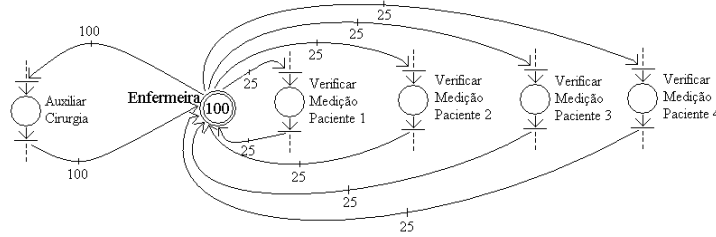


Figura 2.9: Mecanismo de alocação de recurso contínuo

Em [Jeske, 2006], os mecanismos de alocação de recursos discretos e contínuos foram formalizados. A seguir, é apresentada a definição de alocação de recurso discreto e contínuo.

**Definição 2.3. (Mecanismo de alocação de recurso discreto)** O mecanismo de alocação de recurso discreto pode ser definido formalmente por um modelo de rede de Petri marcada  $C_{DR} = \langle A_{DR}, T_{DR}, Pre_{DR}, Pos_{DR}, M_{DR} \rangle$  com:

- $A_{DR} = \bigcup_{\alpha=1}^{N_{DR}} A_{(DR)\alpha} \cup \{R_D\}$  onde  $R_D$  representa o lugar do recurso discreto,  $A_{(DR)\alpha}$  o lugar da atividade  $\alpha$  e  $N_{DR}$  o número das atividades que estão conectadas com o lugar do recurso discreto  $R_D$ .
- $T_{DR} = \bigcup_{\alpha=1}^{N_{DR}} T_{in\alpha} \cup \bigcup_{\alpha=1}^{N_{DR}} T_{out\alpha}$  onde  $T_{in\alpha}$  representa a transição de entrada da atividade  $A_{(DR)\alpha}$  e  $T_{out\alpha}$  representa a transição de saída da atividade  $A_{(DR)\alpha}$ .
- $Pre_{DR}: A_{DR} \times T_{DR} \rightarrow \{0, 1\}$  é a aplicação de incidência anterior tal que:  
 $Pre_{DR}(R_D, T_{in\alpha}) = 1$  e  $Pre_{DR}(A_{(DR)\alpha}, T_{out\alpha}) = 1$   
(outras combinações de lugar/transição são iguais a zero).
- $Pos_{DR}: A_{DR} \times T_{DR} \rightarrow \{0, 1\}$  é a aplicação de incidência posterior tal que:  
 $Pos_{DR}(R_D, T_{out\alpha}) = 1$  e  $Pos_{DR}(A_{(DR)\alpha}, T_{in\alpha}) = 1$   
(outras combinações de lugar/transição são iguais a zero).
- $M_{DR}: A_{DR} \rightarrow \mathbb{N}$  é a marcação inicial tal que:  
 $M_{DR}(R_D) = m_D$  e  $M_{DR}(A_{(DR)\alpha}) = 0$  para  $\alpha = 1$  até  $N_{DR}$  representa o número de recursos discretos.

Supondo-se que, no processo do *Serviço de Reclamações*, um único funcionário trata as atividades *contactar cliente*, *contactar departamento* e *enviar carta*. Então o mecanismo de alocação de recurso discreto é dado pela figura 2.10



**Definição 2.4. (Mecanismo de alocação de recurso contínuo)**

O mecanismo de alocação de recurso contínuo pode ser definido formalmente por um modelo de rede de Petri híbrida marcada com transições discretas

$C_{CR} = \langle A_{CR}, T_{CR}, Pre_{CR}, Pos_{CR}, M_{CR} \rangle$  com:

- $A_{CR} = \bigcup_{\alpha=1}^{N_{CR}} A_{(CR)\alpha} \cup \{R_C\}$  onde  $R_C$  representa o lugar do recurso contínuo,  $A_{(CR)\alpha}$  o lugar da atividade  $\alpha$  e  $N_{CR}$  o número das atividades que estão conectadas com o lugar do recurso discreto  $R_c$ .
- $T_{CR} = \bigcup_{\alpha=1}^{N_{CR}} T_{in\alpha} \cup \bigcup_{\alpha=1}^{N_{CR}} T_{out\alpha}$  onde  $T_{in\alpha}$  representa a transição de entrada discreta da atividade  $A_{(CR)\alpha}$  e  $T_{out\alpha}$  representa a transição de saída discreta da atividade  $A_{(CR)\alpha}$ .
- $Pre_{CR}: A_{CR} \times T_{CR} \rightarrow \mathbb{R}^+$  é a aplicação de incidência anterior tal que:  
 $Pre_{CR}(R_C, T_{in\alpha}) = X_\alpha$  com  $X_\alpha \in \mathbb{R}^+$  e  $Pre_{CR}(A_{(CR)\alpha}, T_{out\alpha}) = 1$   
(outras combinações de lugar/transição são iguais a zero).
- $Pos_{CR}: A_{CR} \times T_{CR} \rightarrow \mathbb{R}^+$  é a aplicação de incidência posterior tal que:  
 $Pos_{CR}(R_C, T_{out\alpha}) = X_\alpha$  e  $Pos_{CR}(A_{(CR)\alpha}, T_{in\alpha}) = 1$   
(outras combinações de lugar/transição são iguais a zero).
- $M_{CR}: A_{CR} \rightarrow \mathbb{R}^+$  é a marcação inicial tal que:  
 $M_{CR}(R_C) = m_C$  e  $M_{CR}(A_{(CR)\alpha}) = 0$  para  $\alpha=1$  até  $N_{CR}$  representa a disponibilidade (em porcentagem) do recurso contínuo.

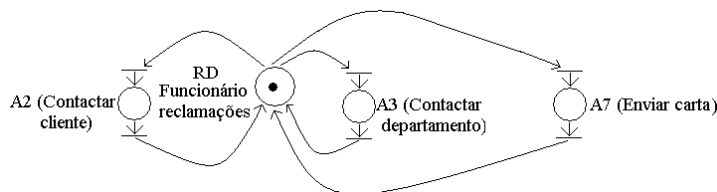


Figura 2.10: Alocação de recurso discreto

Supondo-se que, no processo do *Serviço de Reclamações*, um único funcionário trata as atividades *contactar cliente*, *contactar departamento* e *enviar carta*, então, o mecanismo de alocação de recurso contínuo é dado pela figura 2.11.

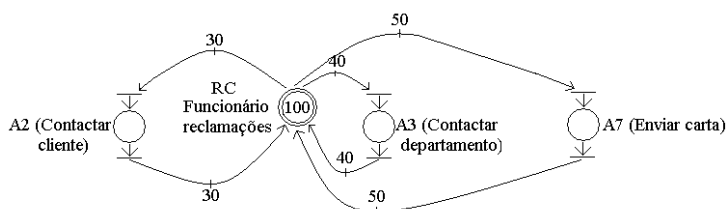


Figura 2.11: Alocação de recurso contínuo

Nessa figura, fica especificado que somente 30% da disponibilidade do funcionário é necessária para realizar a atividade *contactar cliente*, 40% para realizar a atividade *contactar departamento* e 50% para realizar a atividade *enviar carta*. Dessa forma, é possível que o funcionário trate, simultaneamente, mais de uma atividade. Por exemplo, depois do início da atividade *contactar cliente*, se o cliente não estiver disponível para responder às questões do funcionário, este último poderá usar sua disponibilidade de tempo (esperando a resposta do cliente) para iniciar outra atividade, como por exemplo, enviar uma carta. Uma vez iniciada a atividade *contactar cliente*, a qual consome 30% da disponibilidade do funcionário, ainda restam 70% de sua disponibilidade para realizar a atividade *enviar carta*, a qual consome 50% da disponibilidade do funcionário.

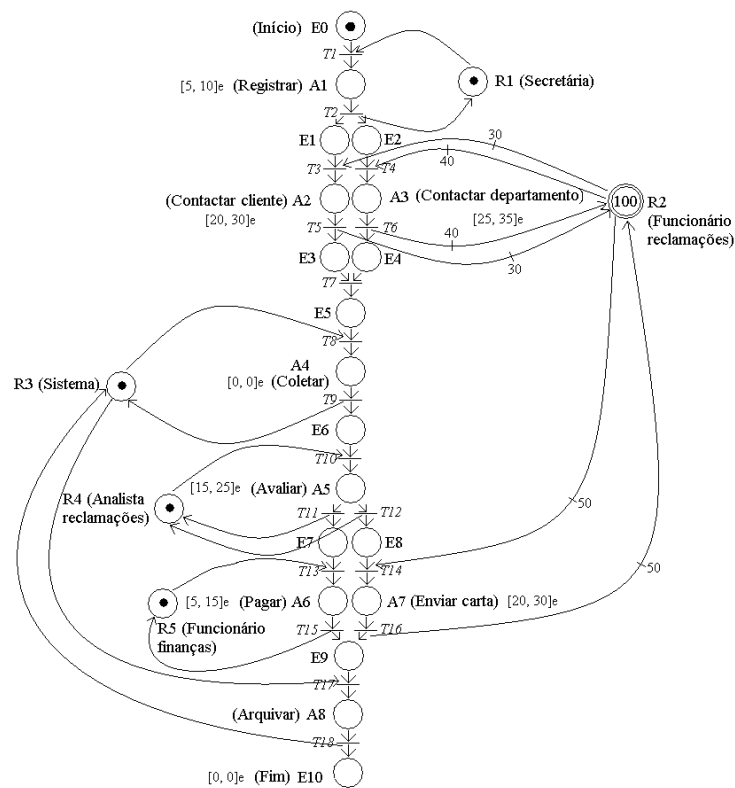


Figura 2.12: Modelo p-temporal com alocação de recursos híbridos para o processo do *Serviço de Reclamações*

Os mecanismos de alocação de recursos para o processo do *Serviço de Reclamações* estão representados na figura 2.12. A *secretária*, o *funcionário finanças* e o *sistema* são representados como recursos discretos. Estes recursos tratarão as atividades em uma ordem puramente sequencial. A *secretária*, por exemplo, representada por uma ficha em R1 pode somente registrar os dados de um cliente de uma maneira não preemptiva (sem interrupção e em seqüência). Ao contrário, o funcionário do Departamento de Reclamações é representado pelo número 100 em R2. Esse recurso contínuo pode ser usado para tratar as atividades *contactar cliente*, *contactar departamento* e *enviar carta*. A atividade *contactar cliente* irá precisar de 30% da disponibilidade desse funcionário, a atividade *contactar*

*departamento* irá precisar de 40% e, finalmente, a atividade *enviar carta* irá precisar de 50% de sua disponibilidade. Isso significa que o funcionário representado no lugar R2 poderá processar mais de uma atividade em um dado intervalo de tempo. Ele poderá escrever uma carta e, ao mesmo tempo, tentar ligar para um cliente, por exemplo.

O modelo proposto nesse trabalho é o mesmo que aquele proposto em [de Oliveira, 2005] e [Jeske, 2006]. Trata-se de uma rede de Petri p-temporal com alocação de recursos híbridos. Neste modelo, a estrutura de controle segue a mesma que aquelas das *workflow net* de Aalst [Aalst & Hee, 2002], com os lugares de início, fim e os principais roteiros existentes nos processos de *workflow*. Porém, em vez de associar as tarefas às transições, as tarefas são associadas a lugares específicos. Assim, os mecanismos de alocação de recursos podem ser formalmente visualizados durante as execuções das tarefas

## Capítulo 3

# Problema de Escalonamento

Um dos problemas clássicos da Pesquisa Operacional é o problema de escalonamento. O termo Pesquisa Operacional desenvolveu-se durante a Segunda Guerra Mundial, onde um grupo de cientistas foi convocado na Inglaterra para estudar problemas de estratégia e de tática associados com a defesa do país. O objetivo era decidir sobre a utilização mais eficaz de recursos militares limitados. A convocação desse grupo marcou a primeira atividade formal de pesquisa operacional. Porém os primeiros textos da Pesquisa Operacional só surgiram na década de 1950, quando a matéria foi considerada digna de figurar nos currículos universitários. Contudo, de acordo com [Ackoff & Sasieni, 1977], a Pesquisa Operacional parece ter sido empregada pela primeira vez em 1939.

Após o fim da guerra, a utilização de técnicas de Pesquisa Operacional atraiu o interesse de diversas áreas. A natureza dos problemas encontrados é bastante abrangente e complexa, exigindo, portanto, uma abordagem que permita conhecer os múltiplos aspectos envolvidos. Uma característica importante da Pesquisa Operacional e que facilita o processo de análise e de decisão é a utilização de modelos. Os modelos permitem, em particular, a experimentação da solução proposta. Isso significa que uma decisão pode ser mais bem avaliada e testada antes de ser efetivamente implementada. A economia obtida e a experiência adquirida pela experimentação justificam a utilização da Pesquisa Operacional.

No problema clássico de escalonamento, chamado *problema central* [Gondran & Minox, 1984], as únicas restrições são aquelas do sequenciamento no tempo: não se pode, por exemplo, iniciar a tarefa  $j$  antes de ser completada a tarefa  $i$ . Nesse problema, o objetivo a ser alcançado pode ser decomposto em porções elementares de trabalho, ou tarefas, em que cada tarefa  $i$  é caracterizada por sua duração  $p_i$  e pelas restrições que as ligam umas às outras (tarefas sucessoras ou precedentes).

O problema de escalonamento apresentado em [Esquirol *et al.*, 1995] consiste em organizar no tempo a realização de tarefas considerando restrições temporais (intervalos de tempo) e restrições de utilização dos recursos compartilhados necessários à execução das tarefas.

Um escalonamento constitui então uma das soluções do problema do escalonamento. Ele é definido pelo planejamento de execução das tarefas, isto é, ordem das tarefas e datas de início e fim das tarefas, além de alocação dos recursos envolvidos.

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

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

As restrições representam os conjuntos dos valores que podem ter as variáveis de decisão do problema. As restrições podem ser distinguidas em restrições temporais e restrições de recursos:

- Restrições temporais: As restrições temporais referem-se às 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. Restrições temporais podem se referir também às restrições de coerência tecnológica ou restrições de roteiros, que descrevem as relações de ordem relativa entre as diversas tarefas.
- Restrições de recursos: Já as restrições de recursos referem-se às restrições de utilização de recursos que exprimem a natureza e a quantidade dos meios utilizados para as tarefas, assim como as características de utilização destes meios. Restrições de recursos também podem se referir às restrições de disponibilidade dos recursos.

Na resolução de um problema de escalonamento, pode-se optar por dois grandes tipos de estratégias: otimização da solução ou admissibilidade da solução.

As abordagens por otimização supõem 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. Os critérios vão depender do sistema a ser tratado. Por exemplo, no contexto dos sistemas de manufatura e dos sistemas de gerenciamento de *workflow*, os critérios podem ser relacionados ao tempo. Por exemplo, o tempo total de execução de um conjunto de tarefas.

Nas abordagens por admissibilidade de solução, busca-se uma solução que seja admissível para o problema do sistema em questão, ou seja, uma solução admissível é aquela que respeita um conjunto de restrições pré-estabelecidas.

O problema de escalonamento pode ser estático ou dinâmico [Esquirol & Lopez, 1999]. O escalonamento estático é aquele cujo o conjunto das informações necessárias à sua resolução é fixo (conjunto de tarefas, de recursos, e suas características). A solução é um plano previsional em que o conjunto dos intervalos de execução das tarefas caracteriza o horizonte temporal no plano. A solução possui uma duração de validade limitada devido

ao aparecimento de novas informações que podem ocorrer na execução do plano como, por exemplo, o aparecimento de novas tarefas, panes, atrasos, e outros. É necessário um controle de execução constituído de uma função de decisão em tempo real que realizará os ajustes e tentará satisfazer os objetivos globais do plano.

O monitoramento em tempo real da execução de um plano previsional ou da solução de um problema de escalonamento estático pode ser considerado como um problema de escalonamento dinâmico onde as situações de conflito serão resolvidas por uma função de decisão em tempo real que poderá realizar localmente ajustes em virtude de alguns acontecimentos (eventos) dificilmente previsíveis durante a elaboração do plano previsional.

## 3.1 Modelagem para o problema de escalonamento

Originalmente, a modelagem do problema clássico de escalonamento de sistemas foi realizada a partir dos grafos [Malcom *et al.*, 1959], [Roy, 1960], [Gondran & Minox, 1984].

### 3.1.1 Grafos

A representação de um problema de escalonamento por grafos possibilita uma boa apreciação global do problema [Gondran & Minox, 1984]. Especificamente no caso do problema de escalonamento central (*problema central*) [Gondran & Minox, 1984], o estudo de um grafo facilita a identificação de tarefas prioritárias e a detecção de atrasos em tempo hábil para se tomar uma ação necessária de correção.

Algumas representações são: os grafos potenciais-tarefas e os grafos PERT.

- Grafos potenciais-tarefas:

Este tipo de grafo foi apresentado em [Roy, 1960]. Em um dado sistema, para cada tarefa  $i$  (de 1 até  $n$ ), associa-se um vertice  $i$  do grafo. Se a tarefa  $i$  preceder a tarefa  $j$ , desenha-se um arco de comprimento  $p_i$  (representando a duração da tarefa) entre  $i$  e  $j$ .

O grafo então definido não pode ter circuitos, senão algumas operações poderiam preceder-se a si mesmas. Este grafo possui um certo número de vértices sem precursor e um certo número de vértices sem sucessor.

- Grafos PERT:

PERT é um acrônimo em inglês para *project evaluation and review technique*. O grafo PERT foi apresentado em [Malcom *et al.*, 1959]. Nesta representação as tarefas são associadas aos arcos do grafo. O comprimento de cada arco representa a duração  $p_i$  da tarefa correspondente. O início e o fim da tarefa são estágios do projeto.

Define-se o estágio inicial e final do projeto e um certo número de estágios intermediários. Cada estágio é definido por um conjunto de tarefas que já foram executadas.

Se uma tarefa  $j$  precisa suceder uma tarefa  $i$ , então o ponto inicial do arco  $u_j$  (correspondente à tarefa  $j$ ) deve ser o ponto terminal do arco  $u_i$  (correspondente à tarefa  $i$ ). Da mesma forma que o grafo potencial-tarefa, o grafo PERT não pode ter circuitos.

- Grafos de eventos:

Os grafos de eventos são um tipo de rede de Petri. Uma rede de Petri é denominada um grafo de eventos se cada lugar tiver exatamente uma transição de chegada e uma transição de partida [Baccelli *et al.*, 1992]. Os grafos de eventos podem ser temporizados. Nesse caso, existe um tempo de espera associado a cada lugar ou a cada transição do grafo. Tais grafos geralmente têm circuitos e são utilizados para representar sistemas de gerenciamento repetitivos.

No trabalho de Luders [Luders, 2001], os grafos de eventos temporizados (com o tempo de espera associado aos lugares) foram utilizados a fim de propor um controle multivariável de sistemas a eventos discretos em Dióides. Sistemas a eventos discretos ocorrem em processos de manufatura e em aplicações de tráfego e compartilhamento de recursos, entre outros. Em particular, os grafos de eventos foram utilizados para modelar a sincronização de eventos. No caso do problema de escalonamento para sistemas de manufatura, faz-se, geralmente, referência a uma sequência particular de operações. Sendo assim, os grafos de eventos [Gondran & Minox, 1984] constituem um modelo suficiente para descrever uma solução do problema de escalonamento, particularmente, no caso de escalonamentos cíclicos [Lacomme *et al.*, 2003]. No contexto dos sistemas de manufatura, o escalonamento cíclico é referente a uma política em que se tem o controle de quantas peças serão tratadas em cada linha de produção durante um mesmo período.

### 3.1.2 Redes de Petri temporais

As redes de Petri  $t$ -temporizadas [Lee & DiCesare, 1994] são particularmente bem adaptadas à modelagem de sistemas flexíveis de manufatura. De fato, é natural associar as transições da rede às operações ou tarefas, bem como representar os recursos compartilhados por marcas que se acham nos lugares.

No trabalho de Moraes [Moraes, 2000], um modelo de rede de Petri  $p$ -temporal  $t$ -temporizado foi utilizado para tratar o problema de escalonamento dos sistemas de produção por lotes, com política de produção cíclica. Mais especificamente, o modelo foi aplicado para representação formal das restrições do problema. De acordo com Moraes [Moraes, 2000], 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.

## 3.2 Abordagens para tratar o problema de escalonamento

Dentre as diversas abordagens para se tratar o problema de escalonamento, destacam-se as seguintes:

### 3.2.1 Algoritmos Branch and Bound

Devido ao fato de existir uma chance mínima de os problemas NP-completos, como é o caso do problema de escalonamento, serem polinomiais faz com que os pesquisadores busquem abordagens alternativas de solução destes problemas. Tais abordagens podem ser métodos de enumeração e algoritmos de aproximação. As técnicas de *branch and bound* são baseadas nos princípios de enumeração explícitas. Tais técnicas consistem em descrever o conjunto de soluções sistematicamente na forma de arborecência, e percorrê-las sem omitir nenhum vértice. O conceito de avaliação por limite inferior permite que, ao encontrar uma boa solução admissível, seja possível passar implicitamente por todas as *branches* (ramos) da arborecência, sem ter de examinar todos os vértices.

Em [Lee & DiCesare, 1994] um modelo de rede de Petri t-temporizada serviu de representação para o problema do escalonamento flexível de um sistema de manufatura, usando um algoritmo do tipo *branch and bound* para minimizar o tempo global de execução do sistema.

Em [Esquirol & Lopez, 1999], foi utilizada uma abordagem para solucionar o problema de escalonamento dos sistemas representados por grafos com restrições disjuntivas (mecanismos de alocação dos recursos no caso não preemptivo) a qual combina um mecanismo de propagação de restrições que permite o cálculo de intervalos de datas admissíveis e um algoritmo *branch and bound* cuja meta é encontrar uma sequência ótima das tarefas que respeite as restrições de tempo (intervalos de datas).

### 3.2.2 Algoritmos Genéticos

Alguns trabalhos ([Takahashi *et al.*, 1996], [Reyes *et al.*, 2001], [Deriz, 2007]) utilizaram abordagens de Inteligência Artificial, tais como métodos de busca usando algoritmos genéticos para resolução do problema de escalonamento.



No trabalho de Deriz [Deriz, 2007], foi proposto um método de escalonamento de tarefas em sistemas de manufatura com recursos compartilhados. Foi utilizado um método de busca usando Algoritmo Genético em que um cromossomo é codificado de maneira a indicar o escalonamento no contexto da manufatura, isto é, uma solução completa para o problema. Tal solução encontrada é então *traduzida* numa solução de escalonamento por meio da leitura dos genes do cromossomo.

O uso do algoritmo genético permite geralmente encontrar soluções sub-ótimas em um tempo de execução rápido o suficiente para aplicar a técnica ao contexto de escalonamento em tempo real.

### 3.2.3 Análise Baseada em Restrições

Alguns trabalhos [Fargier, 1994], [Lopez & Esquirol, 1996], utilizaram um método de Inteligência Artificial conhecido como método de satisfação de restrições (CSP - *Constraint Satisfaction Problem*) para resolver o problema de escalonamento em sistemas de produção.

O método de satisfação de restrições ou análise por restrições é proveniente dos problemas de satisfação de restrições que, geralmente, consiste em encontrar um conjunto de variáveis as quais sejam compatíveis com um conjunto de restrições [Fargier, 1994].

#### Restrições baseadas nos roteiros

No trabalho de Esquirol [Esquirol, 1987], as restrições temporais do problema de escalonamento de tarefas da manufatura foram propagadas a cada tarefa por meio de técnicas de propagação de restrição aplicadas a grafos não conjuntivos, onde os nós representam as datas de início e fim das tarefas e os arcos representam desigualdades potenciais. As características não conjuntivas, as quais permitem introduzir alternância, foram usadas para modelar restrições de alocação de recursos e restrições temporais.

O mecanismo de propagação de restrições temporais nos roteiros de um processo de *workflow* modelado por meio de redes de Petri temporais pode ser feito por meio de técnicas clássicas de propagação de restrições baseadas em grafos sem circuitos [Gondran & Minox, 1984] como apresentado em [de Oliveira, 2005]. Fixando-se a data de início de um caso bem como a sua duração máxima para ser completado, calculam-se os intervalos de visibilidade associados aos diversos lugares  $E_j$  de espera entre as diversas atividades dos roteiros existentes.

#### Restrições baseadas em um Raciocínio Energético sobre os mecanismos de alocação de recursos

A abordagem baseada no Raciocínio Energético, cuja finalidade é evitar a enumeração explícita das combinações possíveis de solução de conflito entre tarefas em competição por

recursos limitados, foi apresentada em [Erschler & Lopez, 1990], [Erschler *et al.*, 1991]. Os mecanismos básicos são parecidos com outras técnicas já desenvolvidas no campo do escalonamento da manufatura, tais como: *Seleções Imediatas* [Carlier & Pinson, 1994] ou *Procurar as Fronteiras* [Applegate & Cook, 1991].

A seguir, é apresentada a definição do problema de escalonamento da manufatura, no contexto do raciocínio energético:

- um conjunto de  $n$  tarefas deve ser executado, sendo que as tarefas não podem ser particionadas;
- cada tarefa  $i$  é caracterizada por sua duração  $p_i$  e deve ser executada dentro de um janela de tempo  $[r_i, d_i]$ , as quais resultam de restrições temporais iniciais do processo e que são propagadas a cada tarefa por meio de restrições de precedência;
- cada tarefa  $i$  necessita de uma intensidade  $a_i$  de um dado recurso que disponibiliza uma intensidade global igual a  $A$ .
- O conjunto  $S$  denota variáveis de restrições que representam as datas de início consistentes com as janelas de tempo  $[r_i, d_i]$  e restrições de utilização dos recursos.

As propagações de restrições nesse caso têm como objetivo remover valores inconsistentes das datas de início das tarefas, a fim de reduzir a combinação das possíveis soluções.

A idéia de *energia*, que é o produto de uma duração por um determinado consumo, é uma maneira de considerar, simultaneamente, restrições de tempo e de recursos em um único raciocínio [Erschler & Lopez, 1990].

Nos primeiros estudos do Raciocínio Energético aplicado ao problema de escalonamento na manufatura utilizou-se o conceito de *time-resource interval* (TRI), na sigla em inglês [Erschler & Lopez, 1990].

Um TRI  $I$  é caracterizado por:

- Um intervalo temporal  $S_i$  (data de início) e  $F_i$  (data de fim);
- Funções de intensidade de recurso  $Q_a^I(t)$  com  $t \in [S_i, F_i]$  e  $a \in A_I$  onde  $A_I$  é o recurso associado com  $I$ .

Um TRI pode ser definido por suas características instantâneas  $(S_I, F_I, \{Q_a^I(t)\})$  e/ou por suas integrais, por exemplo:

- Sua duração:  $P_I = \int_{S_I}^{F_I} dt = F_I - S_I$
- Sua energia do recurso  $A$ :  $W_A^I = \int_{S_I}^{F_I} Q_a^I(t).dt$

Se  $\forall a, Q_a^I(t)$  for uma constante,  $I$  é dita *uniforme* (neste caso, a energia do recurso  $A$  durante o intervalo  $i$  pode ser escrita como:  $W_A^I = (F_I - S_I \cdot Q_a^I)$ ). Em [Erschler & Lopez, 1990] apenas este caso foi estudado.

As características de um TRI pode ser conhecidas, desconhecidas, independentes ou relacionadas. Por exemplo, a duração pode ser expressada como uma função de intensidade de um recurso.

Dois tipos de TRIs podem ser considerados:

- Intervalos fornecedores nos quais são alocados tempo e recurso.
- Intervalos consumidores nos quais tempo e recurso são requeridos.

Um intervalo consumidor é associado às atividades. Considerando-se uma atividade  $i=(S_i, F_i, \{q_a^i\})$ , sua duração  $p_i$  (a qual é uma constante) é conhecida. Já, os intervalos fornecedores são associado às janelas temporais conhecidas e iguais a constantes.

Erschler & Lopez [Erschler & Lopez, 1990] consideraram  $\Delta=(S_\Delta, F_\Delta, \{Q_A^\Delta\})$  um intervalo fornecedor e  $i=(S_i, F_i, \{q_a^i\})$  uma atividade (relembrando que  $Q_A^\Delta$  e  $q_a^i$  são constantes) com duração  $p_i$ ; a consumação mínima de energia do recurso  $A$  por  $i$  durante o  $\Delta$  foi representada por  $W_a^{i,\Delta}$  sendo dada por:

$$W_a^{i,\Delta} = \max\{0, \min[(r_i+p_i-S_\Delta), (F_\Delta-(d_i-p_i)), F_\Delta-S_\Delta, p_i]\} \cdot q_a^i$$

com  $S^{i,\Delta} \cdot q_a^i \geq 0$ .

Considerando-se uma atividade  $i=(S_i, F_i, \{q_a^i\})$ , com duração  $p_i$  e um intervalo fornecedor  $\delta_S(t)=(r_i, t, \{Q_A^{\delta_S(t)}\})$ , Erschler & Lopez definiram as seguintes variáveis:

- $W'(t)=(t-r_i) \cdot Q_A^{\delta_S(t)} =$  Energia do recurso  $A$  provida por  $\delta_S(t)$ ;
- $W''(t)=\sum_{j \neq i} W_A^{j,\delta_S(t)} =$  Energia mínima consumida do recurso  $A$  por todas as atividades diferentes de  $i$  em  $\delta_S(t)$ ;
- $W_1(t)=W'(t) - W''(t) =$  Energia disponível do recurso  $A$  em  $\delta_S(t)$  para a execução de  $i$ ;
- $W_2(t, S_i)=\min(t-S_i, p_i) \cdot q_a^i =$  Energia do recurso  $A$  usada por  $i$  em  $\delta_S(t)$  começando em  $S_i$ .

A figura 3.1 ilustra um exemplo da dinâmica dessas variáveis para  $t$  variando de  $r_i$  até  $d_i$ . Os instantes  $\varphi_1, \varphi_2, \varphi_3$  e  $\varphi_4$  foram associados com os *pontos de quebra* da curva  $W_1(t)$  (Considerando apenas TRIs uniformes,  $W'(t)$  é linear e  $W''(t)$  é contínua e seccionalmente linear, dessa forma, pode-se observar que os *pontos de quebra* são os mesmos para  $W_1(t)$  e  $W''(t)$ ). Esses instantes (*pontos de quebra*) são chamados *instantes notáveis* do intervalo  $\delta_S(d_i)$ . Estudando a curva  $W_1(t)$ , Erschler & Lopez, conseguiram extrair várias informações sobre a distribuição da energia disponível e seus efeitos na localização da atividade  $i$  (localização ao mais cedo nesse caso). De fato,  $\forall t W_1(t) \geq W_2(t, S_i)$  é uma condição necessária para iniciar a execução da atividade  $i$  no instante  $S_i$ . Essa condição pode envolver um deslocamento da curva  $W_2(t, S_i)$  para a direita, como mostrado na figura 3.1; dessa forma a tarefa  $i$  não pode começar em  $S_i$  mas somente em  $S'_i$ . Isso envolveu a atualização da data de início ao mais cedo da atividade  $i$ .

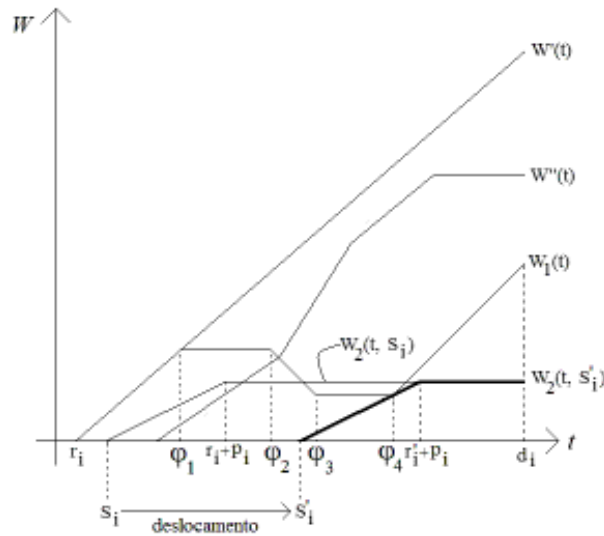


Figura 3.1: Exemplo da dinâmica das energias dentro de um intervalo  $[r_i, d_i]$

Erschler & Lopez afirmam ainda que considerando-se um intervalo fornecedor  $\delta_F(t) = (t, r_i, \{Q_A^{\delta_F(t)}\})$  é possível extrair conclusões simétricas. A condição  $W_1(t) \geq W_2(F_i, t)$ , para  $t$  variando de  $d_i$  até  $r_i$ , pode envolver uma atualização da data de início ao mais tarde ( $\bar{S}_i$ ) da atividade  $i$ . Porém Erschler & Lopez [Erschler & Lopez, 1990] consideraram apenas os intervalos fornecedores tais como  $\delta_S(t)$ .

O objetivo da análise é gerar as datas limites associadas com uma atividade  $i$  o mais exatamente possível, considerando restrições de recursos por meio de análises baseada na energia. Como resultado a meta era maximizar a atualização da data  $r_i$ . Com isso em mente, a seguinte abordagem foi proposta:

- Procurar por todos os *instantes notáveis*  $\varphi_\rho$  do intervalo fornecedor  $\delta_{d_i}(t)$ ;
- Procurar por um  $\varphi_*$  o qual seja o maior *instante notável* tal que  $W_1(t) \leq W_2(t, r_i)$ , por exemplo, começar a atividade  $i$  na data  $r_i$  não é permitido;
- Se  $\varphi_*$  existir, então  $r_i$  passa a ser atualizado para  $S'_i$ .

Esses pontos estão detalhados em [Erschler & Lopez, 1990].

### Consumo energético de uma atividade

De acordo com [Lopez & Esquirol, 1996], a energia máxima disponível que um recurso pode prover em uma dada janela de tempo  $\Delta = [s_\Delta, f_\Delta]$  é definida como:

$$W^\Delta = A \cdot (f_\Delta - s_\Delta) \quad (1).$$

A constante  $A$  é o valor da intensidade energética que o recurso em questão disponibiliza para execução das atividades que o utilizam.

Considerando-se uma atividade  $i$ , começando no instante  $s_i$  (terminando em  $f_i = s_i + p_i$ ); considerando-se também a intensidade energética  $a_i$  que a atividade necessita do recurso

e uma dada janela de tempo  $\Delta$ ; duas situações podem ocorrer para se definir a energia requerida por  $i$  em  $\Delta$ :

- Se  $([s_i, f_i] \cap \Delta) \neq \emptyset$  então  $W_i^\Delta = a_i \cdot [\min(f_i, f_\Delta) - \max(s_i, s_\Delta)]$ .
- Se  $([s_i, f_i] \cap \Delta) = \emptyset$  então  $W_i^\Delta = 0$ .

Para ambas situações  $W_i^\Delta$  é a energia requerida por  $i$  em  $\Delta$ . Mais genericamente, pode-se dizer que:

$$W_i^\Delta = a_i \cdot \max[0, \min(f_i, f_\Delta) - \max(s_i, s_\Delta)] \quad (2).$$

Uma vez que  $s_i$  (data de início da atividade  $i$ ) é uma variável,  $W_i^\Delta$  também é uma variável, podendo ser derivados valores mínimos e máximos, levando em consideração o intervalo temporal  $[r_i, d_i]$ .

A energia mínima requerida por uma atividade  $i$  ao longo da janela de tempo  $\Delta$ , chamada  $\underline{W}_i^\Delta$ , é obtida pela posição da atividade  $i$  que sobrepõe o menos possível a janela de tempo  $\Delta$ :

$$\underline{W}_i^\Delta = a_i \cdot \max[0, \min(p_i, f_\Delta - s_\Delta, r_i + p_i - s_\Delta, f_\Delta - d_i + p_i)] \quad (3).$$

Simetricamente, a energia máxima requerida por uma atividade  $i$  ao longo da janela de tempo  $\Delta$ , chamada  $\overline{W}_i^\Delta$ , é obtida pelas posições da atividade  $i$  que sobrepõem  $\Delta$  o máximo possível:

$$\overline{W}_i^\Delta = a_i \cdot \max[0, \min(p_i, f_\Delta - s_\Delta, d_i - s_\Delta, f_\Delta - r_i)] \quad (4).$$

Por exemplo, considerando-se um conjunto de atividades  $E = \{a, b, c, d, e, f, g\}$  com as características da tabela 3.1, o consumo mínimo de energia sobre  $\Delta = [3, 10]$  é:

$$\sum_{i \in E} \underline{W}_i^\Delta = 3 \times 4 + 3 \times 3 + 7 \times 2 + 2 \times 1 + 3 \times 1 + 2 \times 3 + 0 \times 2 = 46.$$

O consumo máximo é igual a:

$$\sum_{i \in E} \overline{W}_i^\Delta = 4 \times 4 + 5 \times 3 + 7 \times 2 + 2 \times 1 + 6 \times 1 + 2 \times 3 + 1 \times 2 = 61.$$

$i$	a	b	c	d	e	f	g
$r_i$	0	5	1	4	0	8	8
$p_i$	6	5	10	2	6	4	1
$d_i$	7	12	13	8	11	12	12
$a_i$	4	3	2	1	1	3	2

Tabela 3.1: Conjunto  $E$  de atividades com suas respectivas restrições (datas de início ao mais cedo e final ao mais tarde), durações e intensidades energéticas

A figura 3.2 apresenta o intervalo  $\Delta = [3, 10]$  do exemplo anterior numa linha temporal bem como as datas de início ao mais cedo e final ao mais tarde das atividades do conjunto  $E$ .

Para o cálculo da energia mínima ( $\underline{W}_i^\Delta$ ) que uma atividade utiliza de um dado recurso  $A$  em um intervalo  $\Delta$ , considera-se que a atividade seja executada o máximo possível fora do intervalo  $\Delta$  (sem que a atividade seja interrompida) respeitando sua data de início ao mais cedo ( $r_i$ ) e de final ao mais tarde ( $d_i$ ). A figura 3.3(A) apresenta a parte da

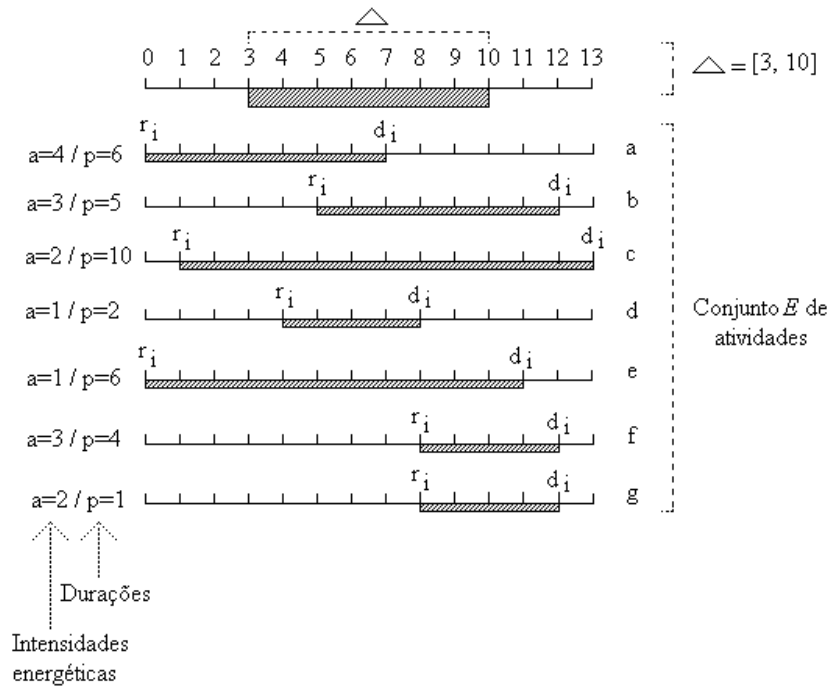
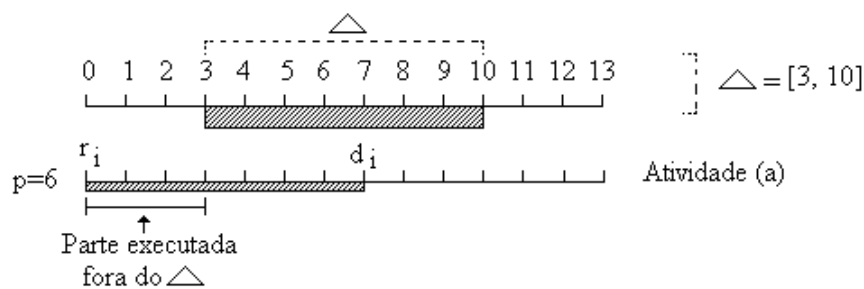
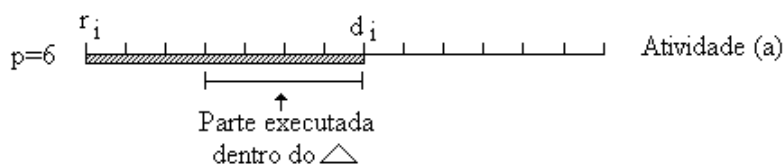


Figura 3.2: Intervalo  $\Delta=[3, 10]$  e restrições (datas de início ao mais cedo e final ao mais tarde) das atividades do conjunto  $E$

atividade (a) do conjunto  $E$  que deverá ser executada fora do  $\Delta=[3, 10]$  para o cálculo da energia mínima. Já para o cálculo da energia máxima ( $\overline{W}_i^\Delta$ ), considera-se que a atividade seja executada o máximo possível dentro do intervalo  $\Delta$  (novamente sem interrupção) respeitando da mesma forma sua data de início ao mais cedo e de final ao mais tarde. A figura 3.3(B) apresenta a parte da atividade (a) que deverá ser executada dentro do  $\Delta [3, 10]$  para o cálculo da energia máxima.



**Figura A**



**Figura B**

Figura 3.3: Intervalos de execução da Atividade (a) dentro e fora do  $\Delta$

### Condição de Consistência Global para existência de uma solução de escalonamento

Considerando-se  $W^\Delta$ , a energia provida por um recurso em uma dada janela de tempo  $\Delta$ , e  $\sum_{i=1}^n W_i^\Delta$ , as energias requeridas pelas atividades no mesmo  $\Delta$ , pode-se derivar uma condição de consistência global necessária a ser respeitada por qualquer escalonamento em qualquer intervalo  $\Delta$ :

$$\forall \Delta, \sum_{i=1}^n W_i^\Delta \leq W^\Delta \quad (5)$$

A fórmula (5) significa que a soma das energias gastas pelas diversas atividades que utilizam um mesmo recurso em um intervalo  $\Delta$  não são maiores que a energia total produzida pelo recurso utilizado nesse  $\Delta$ . A partir da condição (5), pode-se extrair a seguinte proposição:

$$\text{Se } \exists \Delta \text{ tal que } \sum_{i=1}^n W_i^\Delta > W^\Delta \quad (6)$$

então não existe um escalonamento admissível.

Considerando-se, por exemplo, um conjunto  $X$  de atividades com as características da tabela 3.2, onde cada umas das atividades (a, b, c, d, e) deste conjunto são executadas por um mesmo recurso que fornece a intensidade energética  $A=5$ . Verificando-se a condição de consistência global (fórmula (6)) do conjunto  $X$  levando em consideração um  $\Delta$  particular igual a  $[0, 11]$ , tem-se:

- $W^\Delta = (11 - 0) \times 5 = 55$
- $W_a^\Delta = 4 \times 6 = 24$
- $W_b^\Delta = 3 \times 4 = 12$
- $W_c^\Delta = 2 \times 8 = 16$
- $W_d^\Delta = 1 \times 2 = 2$
- $W_e^\Delta = 1 \times 6 = 6$
- $\sum_{i=1}^5 W_i^\Delta = 60$

Uma vez que  $W^\Delta = 55$  é menor que  $\sum_{i=1}^5 W_i^\Delta = 60$ , tem-se uma inconsistência global, o que caracteriza a não existência de um escalonamento admissível, pois o recurso utilizado para execução das atividades não consegue fornecer energia suficiente para completa execução das referidas atividades durante o  $\Delta$  considerado.

O escalonamento das atividades só será possível, quando a condição de inconsistência global (6) não for satisfeita para qualquer  $\Delta$  ao se considerar um determinado recurso. Caso contrário, não será possível realizar o escalonamento das atividades devido à indisponibilidade energética do recurso em questão, não sendo, portanto necessário aplicar um algoritmo de escalonamento.

A figura 3.4 ilustra o intervalo  $\Delta$  estudado nesse exemplo.

$i$	a	b	c	d	e
$r_i$	0	5	1	4	0
$p_i$	6	5	10	2	6
$d_i$	7	12	13	8	11
$a_i$	4	3	2	1	1

Tabela 3.2: Conjunto  $X$  de atividades com suas respectivas restrições (datas de início ao mais cedo e final ao mais tarde), durações e intensidades energéticas

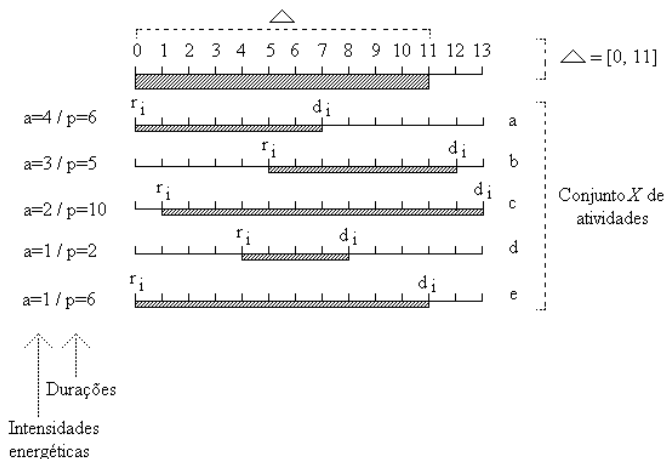


Figura 3.4: Intervalo  $\Delta = [0, 11]$  para o cálculo de consistência global considerando o conjunto  $X$

### Condição de Consistência Local

As regras de consistência global não permitem atualizar diretamente as variáveis do conjunto  $S$  (conjunto que contém as datas de início das atividades) do problema. Para isso, Lopez e Esquirol [Lopez & Esquirol, 1996] utilizaram regras de consistência local.

Ao considerar qualquer atividade  $i$ , a condição da consistência global (5) pode ser reescrita da seguinte maneira:

$$\forall i, \forall \Delta, W_i^\Delta \leq W^\Delta - \sum_{j=1, j \neq i}^n W_j^\Delta \quad (7).$$

O termo  $A_i^\Delta = W^\Delta - \sum_{j=1, j \neq i}^n W_j^\Delta$  é chamado de energia máxima disponível deixada pelas demais atividades no intervalo  $\Delta$  para a execução da atividade  $i$ . Qualquer data de início  $s_i$  da atividade  $i$  que leva a um consumo de energia maior do que a energia máxima disponível para a atividade  $i$  deve então ser removida.

De acordo com [Hennet, 2002], uma maneira de verificar localmente se o recurso utilizado por uma determinada atividade fornece a energia suficiente para sua completa execução levando em consideração  $A_i^\Delta$  (energia máxima disponível para atividade  $i$ ) é efetuar o cálculo da duração máxima equivalente  $\bar{p}_i^\Delta$ . Tal duração representa a duração máxima de disponibilidade do recurso para execução da atividade  $i$  respeitando os gastos energéticos mínimos das outras atividades que utilizam o mesmo recurso no intervalo  $\Delta$ . Dessa forma  $\bar{p}_i^\Delta$  é dada pela razão entre  $A_i^\Delta$  (energia máxima disponível no  $\Delta$ ) e  $a_i$  (intensidade energética que  $i$  necessita de um recurso):  $\bar{p}_i^\Delta = \frac{A_i^\Delta}{a_i}$ .



Quando a duração equivalente  $\bar{p}_i^\Delta$  de uma atividade  $i$  for menor do que a sua duração  $p_i$ , significa que durante o  $\Delta$  estudado não há energia suficiente para a atividade ser executada por completo, caracterizando uma inconsistência local. Dessa forma uma atualização das datas de início da atividade  $i$  deve ser buscada:

$$\left\{ \begin{array}{l} \text{Se } \exists \Delta, \exists i \text{ tal que } p_i > \bar{p}_i^\Delta \\ \text{então } s_i \notin ]s_\Delta + \bar{p}_i^\Delta - p_i, f_\Delta - \bar{p}_i^\Delta[ \end{array} \right.$$

Considere, por exemplo, um conjunto  $Y$  de atividades com as características da tabela 3.3, onde cada umas das atividades (a, b, c, d, e, f) deste conjunto são executadas por um mesmo recurso com intensidade energética  $A=100$ .

Uma vez que a condição de consistência global seja satisfeita, é possível verificar se elas podem efetivamente ser iniciadas ao mais cedo nas datas  $r_i$  ou ao mais tarde nas datas  $d_i$  através da verificação de consistência local.

$i$	a	b	c	d	e	f
$r_i$	2	7	2	7	7	20
$\bar{s}_i$	17	22	12	17	17	30
$p_i$	20	20	25	25	25	25
$d_i$	47	52	47	52	52	65
$a_i$	30	30	40	40	40	40

Tabela 3.3: Conjunto  $Y$  de atividades com suas respectivas restrições (datas de início ao mais cedo e final ao mais tarde), datas de início ao mais tarde ( $\bar{s}_i$ ), durações e intensidades energéticas

Por exemplo, para a atividade  $f$ , considerando um  $\Delta=[20,45]$  e se  $s_i$  (data de início da execução da atividade) for igual a  $r_i$  (data de início ao mais cedo da atividade) tem-se:

- Verificação de consistência global sobre a atividade  $f$

$$\begin{aligned} - W^\Delta &= (45 - 20) \times 100 = 2500 \\ - \underline{W}_a^\Delta &= 30 \times 2 = 60 \\ - \underline{W}_b^\Delta &= 30 \times 7 = 210 \\ - \underline{W}_c^\Delta &= 40 \times 7 = 280 \\ - \underline{W}_d^\Delta &= 40 \times 12 = 480 \\ - \underline{W}_e^\Delta &= 40 \times 12 = 480 \\ - \underline{W}_f^\Delta &= 40 \times 5 = 200 \\ - \sum_{i=1}^6 \underline{W}_i^\Delta &= 1710 \\ - W^\Delta &= 2500 \geq \sum_{i=1}^6 \underline{W}_i^\Delta = 1710 \end{aligned}$$

Satisfeita a condição de consistência global ( $W^\Delta \geq \sum_{i=1}^6 \underline{W}_i^\Delta$ ) é possível, por exemplo, verificar se na data de início ao mais cedo ( $r_i$ ) da atividade em questão haverá

energia disponível o suficiente para sua completa execução. Tal verificação pode ser feita por meio do cálculo da consistência local.

- Verificação de consistência local sobre a atividade  $f$  (considerando a data de início ao mais cedo)

- $W^\Delta = (45 - 20) \times 100 = 2500$
- $W_a^\Delta = 30 \times 2 = 60$
- $W_b^\Delta = 30 \times 7 = 210$
- $W_c^\Delta = 40 \times 7 = 280$
- $W_d^\Delta = 40 \times 12 = 480$
- $W_e^\Delta = 40 \times 12 = 480$
- $\sum_{j=1, j \neq i}^n W_j^\Delta = 1510$
- $A_f^\Delta = 2500 - 1510 = 990$
- $\bar{p}_f^\Delta = \frac{990}{40} = 24.75 < p_f = 25$

Como a duração máxima equivalente sobre o  $\Delta$  da atividade  $f$  ( $\bar{p}_f^\Delta$ ) é menor do que a sua duração ( $p_f$ ), tal atividade não poderá ser iniciada ao mais cedo na data prevista devido a indisponibilidade energética do recurso em questão.

A figura 3.5 ilustra o intervalo  $\Delta$  estudado para o cálculo da consistência local da atividade ( $f$ ) do conjunto  $Y$ .

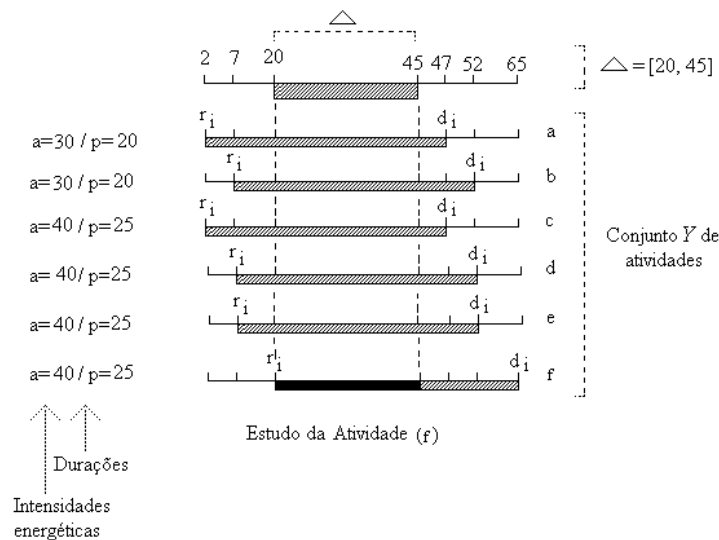


Figura 3.5: Intervalo  $\Delta = [20, 45]$  para o cálculo de consistência local considerando o início ao mais cedo da atividade ( $f$ ) do conjunto  $Y$

Considerando agora, por exemplo, a atividade  $a$  do conjunto  $Y$ . Levando-se em consideração um  $\Delta = [17, 37]$  e  $s_i$  (data de início da execução da atividade) igual a  $\bar{s}_i$  (data de início ao mais tarde) tem-se:

- Verificação de consistência global sobre a atividade  $a$

$$- W^\Delta = (37 - 17) \times 100 = 2000$$

$$- \underline{W}_a^\Delta = 30 \times 5 = 150$$

$$- \underline{W}_b^\Delta = 30 \times 5 = 150$$

$$- \underline{W}_c^\Delta = 40 \times 10 = 400$$

$$- \underline{W}_d^\Delta = 40 \times 10 = 400$$

$$- \underline{W}_e^\Delta = 40 \times 10 = 400$$

$$- \underline{W}_f^\Delta = 40 \times 0 = 0$$

$$- \sum_{i=1}^6 \underline{W}_i^\Delta = 1500$$

$$- W^\Delta = 2000 \geq \sum_{i=1}^6 \underline{W}_i^\Delta = 1500$$

Satisfeita a condição de consistência global ( $W^\Delta \geq \sum_{i=1}^6 \underline{W}_i^\Delta$ ) é possível, por exemplo, verificar se na data de início ao mais tarde ( $d_i$ ) da atividade em questão haverá energia disponível o suficiente para sua completa execução. Tal verificação pode ser feita por meio do cálculo da consistência local.

- Verificação de consistência local sobre a atividade  $a$  (considerando a data de início ao mais tarde)

$$- W^\Delta = (37 - 17) \times 100 = 2000$$

$$- \underline{W}_b^\Delta = 30 \times 5 = 150$$

$$- \underline{W}_c^\Delta = 40 \times 10 = 400$$

$$- \underline{W}_d^\Delta = 40 \times 10 = 400$$

$$- \underline{W}_e^\Delta = 40 \times 10 = 400$$

$$- \underline{W}_f^\Delta = 40 \times 0 = 0$$

$$- \sum_{j=1, j \neq i}^n \underline{W}_j^\Delta = 1350$$

$$- A_a^\Delta = 2000 - 1350 = 650$$

$$- \bar{p}_a^\Delta = \frac{650}{30} = 21.6 > p_a = 20$$

Como a duração máxima equivalente sobre o  $\Delta$  da atividade  $a$  ( $\bar{p}_a^\Delta$ ) é maior do que a sua duração ( $p_a$ ), ela poderá ser iniciada ao mais tarde na data prevista, visto que haverá energia suficiente para sua completa execução.

A figura 3.6 ilustra o intervalo  $\Delta$  estudado para o cálculo da consistência local da atividade  $a$  do conjunto  $Y$ .

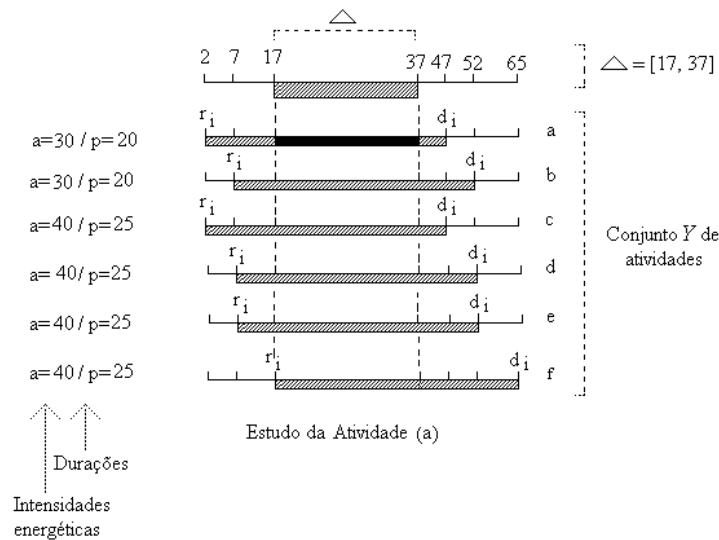


Figura 3.6: Intervalo  $\Delta=[17, 37]$  para o cálculo de consistência local considerando o início ao mais tarde da atividade ( $a$ ) do conjunto  $Y$

### 3.2.4 Jogadores de redes de Petri

Os jogadores de redes de Petri são mecanismos de inferência especializados aplicados em tempo real à modelos de redes de Petri. O funcionamento normal de um jogador de redes de Petri é o disparo das transições assim que elas se tornam habilitadas. Essa estratégia permite o escalonamento das atividades de um sistema em tempo real e dá a necessária flexibilidade para se levar em conta possíveis perturbações (como atrasos, por exemplo).

No trabalho de [Moraes, 2000], menciona-se que nesse tipo de abordagem, as transições são geralmente disparadas o mais cedo possível, o que pode conduzir a um escalonamento que não seja ótimo, conforme mostrado num exemplo particular em [Silva & Vallete, 1989].

Em [Julia & Valette, 2000] foi apresentado um jogador de redes de Petri em que as transições não são disparadas necessariamente o mais cedo graças à construção de um grafo das classes que estuda localmente situações de conflito.

Em [Julia & dos Santos Soares, 2003] foi apresentado um jogador de redes de Petri no qual as situações de conflito são estudadas por meio da lógica linear. As situações de conflito são resolvidas em tempo real (sem a construção do grafo das classes) por meio de fórmulas com durações simbólicas.

Nos trabalhos de [de Oliveira, 2005] e [Jeske, 2006] foi utilizado um jogador de redes de Petri p-temporal como solução para o problema de escalonamento tempo real dos sistemas de gerenciamento de *workflow*. O algoritmo de tal jogador foi implementado

por Ramos [Júlia *et al.*, 2007], sendo desenvolvido com o objetivo de apresentar uma abordagem baseada nos diagramas dinâmicos da notação UML e nas redes de Petri p-temporais para a verificação de cenários de sistemas de processamento em tempo real. A idéia principal consiste em traduzir os diagramas dinâmicos de UML que representam as especificações iniciais do sistema numa única rede de Petri p-temporal representando o comportamento global do sistema. Um mecanismo de inferência especializado (o jogador de redes de Petri p-temporais), foi então implementado e aplicado ao modelo de rede de Petri p-temporal obtido a fim de realizar a atividade conhecida em engenharia de software como atividade de verificação de cenário. Tal atividade, quando se trata de sistemas de processamento em tempo real, é equivalente à resolução de um problema de escalonamento. O jogador de rede de Petri p-temporal implementado em linguagem de programação Java pode então ser aplicado a diversos exemplos de sistemas de processamento em tempo real tais como sistemas de gerenciamento de *workflow* com mecanismos de resolução de conflito em tempo real (sem mecanismos de retrocesso).

Quando o modelo é baseado no modelo de rede de Petri p-temporal, o algoritmo do jogador deve resolver as situações de conflito em tempo real de forma a não permitir a possibilidade de violação de restrição temporal que pode ser causada pela *morte* de uma ficha [Julia & Valette, 2000].

O diagrama de atividades da figura 3.7 mostra o funcionamento básico do algoritmo do jogador de redes de Petri p-temporal.

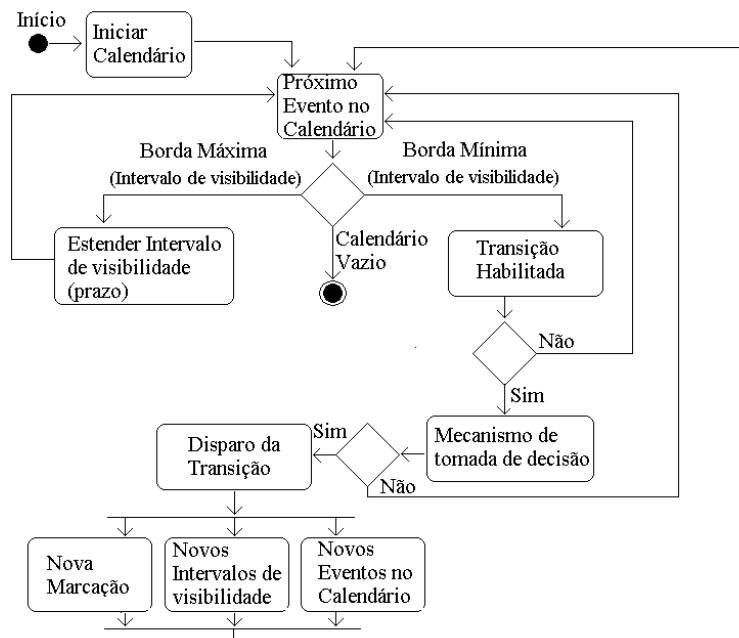


Figura 3.7: Jogador de redes de Petri p-temporal

O jogador possui um calendário dos eventos (bordas mínimas e máximas de todos os intervalos de visibilidade) escalonados no tempo. A cada vez que uma borda mínima é alcançada, uma ficha torna-se disponível. Se essa ficha sensibilizar uma transição e se não

houver conflito efetivo, a transição é disparada; caso contrário, o conflito é isolado e o mecanismo de tomada de decisões é ativado para verificar se a transição pode ser disparada na data mínima. Se a decisão for o disparo, uma nova marcação é gerada e novos intervalos de visibilidade são definidos; caso contrário, é tratado o próximo evento do calendário. O mecanismo de tomada de decisão é baseado na geração de um grafo das classes e foi apresentado em [de Oliveira, 2005]. Se um limite máximo (borda máxima de um intervalo de visibilidade) for atingido, então a ficha *morre*, representando um comportamento não permitido. Nesse caso, não haverá garantia de que os prazos de entrega serão respeitados.

### 3.3 Problema de escalonamento no contexto dos sistemas de gerenciamento de *workflow*

No contexto dos sistemas de gerenciamento de *workflow*, ainda não existem muitas propostas para o problema de escalonamento. Alguns trabalhos como o de Aalst [Aalst & Hee, 2002], por exemplo, apontam a habilidade limitada dos atuais sistemas de gerenciamento de *workflow* no que se refere à alocação dos recursos e à decisão com relação à ordem em que devem ser executadas as tarefas que usam os mesmos recursos. Especificamente, para os sistemas de gerenciamento de *workflow*, o problema de escalonamento é bastante complexo. De acordo com [Tramontina *et al.*, 2004], a literatura, não há resultados sobre o escalonamento que cubram uma proporção significativa das características destes sistemas. Uma solução para o problema de escalonamento de sistemas de gerenciamento de *workflow* deverá considerar os diferentes tipos de recursos utilizados, o compartilhamento de recursos bem como restrições temporais relativas aos prazos de entrega para os casos.

As abordagens clássicas para solucionar o problema de escalonamento dos sistemas de manufatura representados por grafos apresentam limitações no caso de sistemas de gerenciamento de *workflow*, pois as soluções são dadas, geralmente, por uma sequência perfeita de atividades que será difícil de seguir em tempo real para sistemas que consideram o comportamento humano (nos sistemas de gerenciamento de *workflow* muitos recursos são do tipo humano e as durações das atividades não são bem definidas).

Em [Senkul & Toroslu, 2005], foi apresentada uma arquitetura para modelagem e escalonamento de sistemas de gerenciamento de *workflow* com restrições de tempo e alocação de recursos. A abordagem procura por soluções de escalonamento admissíveis. Em especial, o trabalho preocupou-se com os custos de execução das tarefas. A programação por restrição [Marriott & Stuckey, 1998] foi utilizada usando a linguagem Oz. A principal contribuição do trabalho foi a arquitetura proposta. Esta arquitetura provê uma linguagem de especificação que consegue modelar informações dos recursos e restrições de alocação de recursos, fornecendo ainda um modelo para o escalonamento o qual incorpora as restrições do problema a fim de encontrar o recurso apropriado para a execução de uma

determinada tarefa. Os autores sugerem, como trabalhos futuros, estender a arquitetura a fim de encontrar soluções de escalonamento mais eficientes.

No trabalho de Oliveira [de Oliveira, 2005], o objetivo foi o de propor um modelo de rede de Petri híbrida p-temporal como solução para o problema de escalonamento dos sistemas de gerenciamento de *workflow*, em que foi aplicado um jogador de redes de Petri ao modelo, a fim de se obter um cenário admissível correspondente à uma sequência específica das atividades que respeitassem as restrições temporais.

Em Jeske [Jeske, 2006], o objetivo foi o de propor um modelo de rede de Petri p-temporal com recursos híbridos *fuzzy* como solução para o problema de alocação de recursos humanos em sistemas de gerenciamento de *workflow*, em que um jogador de redes de Petri também foi aplicado ao modelo com o mesmo objetivo do trabalho de Oliveira.

A abordagem proposta neste trabalho é baseada na combinação de um mecanismo de propagação de restrições [de Oliveira, 2005] (as quais são refinadas por meio de um segundo mecanismo baseado em um *Raciocínio Energético*) e um jogador de redes de Petri que usa um mecanismo de resolução de conflitos cujo objetivo é calcular uma sequência de atividades que respeitem as restrições disjuntivas (mecanismo de alocação de recursos) e as restrições temporais (intervalos de datas).

## Capítulo 4

# Problema de escalonamento dos sistemas de gerenciamento de *workflow*

No contexto dos sistemas de gerenciamento de *workflow*, um escalonamento é definido pelo planejamento de execução das atividades, isto é, ordem de execução das atividades bem como datas de início e fim das atividades além de alocação dos recursos envolvidos. Uma solução para o problema de escalonamento de sistemas de gerenciamento de *workflow* precisa considerar características importantes referentes a recursos (tipos de recursos e compartilhamento de recursos por exemplo). Nesses sistemas, vários casos podem ser executados simultaneamente, o que pode desencadear situações de conflito. Estes conflitos devem ser tratados em tempo real, ou seja, sem um mecanismo de retrocesso. Tal solução deve considerar também as restrições temporais relativas aos prazos de entrega dos casos, além dos diferentes roteiros que cada caso pode seguir.

Uma situação de conflito pode ocorrer sempre que um mesmo recurso não preemptivo (que não pode ser interrompido) for chamado ao mesmo tempo para a execução de atividades que pertencem a diferentes cenários (diferentes casos).

Esquirol e Lopez [Esquirol & Lopez, 1999] apresentaram uma abordagem para solucionar o problema de escalonamento de sistemas representados por grafos com restrições não-conjuntivas (mecanismos de alocação de recursos em caso não preemptivo) que utiliza um mecanismo de propagação de restrição que permite o cálculo de intervalos de datas admissíveis para o início das atividades. Um algoritmo do tipo *branch and bound* cuja meta é encontrar um sequência ótima das atividades que respeitasse as restrições temporais (intervalo de datas) complementa a abordagem proposta. Contudo, para os sistemas de gerenciamento de *workflow*, tal abordagem apresenta uma limitação uma vez que a solução, gerada por um algoritmo de otimização, geralmente é dada por uma sequência perfeita de atividades que dificilmente poderá ser seguida em tempo real para sistemas que levam em consideração o comportamento humano.

Silva e Valette [Silva & Vallete, 1989] apresentaram uma abordagem baseada em jogador de redes de Petri na qual foi mostrado que disparar as transições assim que elas se



tornam disponíveis nem sempre é a melhor estratégia. Dessa forma, um jogador de redes de Petri tradicional dificilmente respeitará as restrições referentes aos prazos de entrega dos casos de um processo no caso de sistemas de gerenciamento de *workflow*.

Este trabalho propõe uma abordagem baseada na combinação de três mecanismos: um mecanismo de propagação de restrições aplicados aos roteiros dos processos [de Oliveira, 2005], um mecanismo baseado em um Raciocínio Energético sobre o mecanismo de alocação de recursos e finalmente um jogador de redes de Petri [Júlia *et al.*, 2007] que usa um mecanismo de resolução de conflitos, tendo como objetivo calcular uma sequência de atividades que respeite as restrições disjuntivas (mecanismos de alocação de recursos) e as restrições temporais (intervalos de datas) sem mecanismo de retrocesso.

## 4.1 Mecanismos de propagação de restrições nos roteiros

A dinâmica de uma rede de Petri p-temporal depende da marcação da rede e da situação temporal das fichas que é dada pelo intervalo de visibilidade, como foi apresentado no capítulo 2 (Definição 2.4, intervalo de visibilidade).

Nos sistemas de gerenciamento de *workflow*, os tempos de espera existentes entre as atividades foram representados em [de Oliveira, 2005] por um intervalo de visibilidade cujas bordas mínimas e máximas dependem das datas ao mais cedo e ao mais tarde de entrega para os casos considerados. Dessa forma, conhecendo a data de início e a máxima duração de um caso, é possível calcular os intervalos de visibilidade associados às fichas nos lugares de espera usando técnicas de propagação de restrição utilizadas nos clássicos problemas de escalonamento baseados em grafos sem circuitos [Esquirol & Lopez, 1999].

De acordo com os roteiros dos sistemas de gerenciamento de *workflow*, são calculados os intervalos de visibilidade associados às fichas nos lugares de espera. Em [de Oliveira, 2005], foi ilustrada a técnica de propagação de restrições no modelo de redes de petri p-temporal com exemplos que mostram como calcular as borda mínimas e máximas dos intervalos de visibilidade para cada tipo de roteiro. As notações seguintes foram utilizadas nos exemplos:

- $\text{minEY}$  = borda mínima do intervalo de visibilidade associada a um caso no lugar de espera EY.
- $\text{maxEY}$  = borda máxima do intervalo de visibilidade associada a um caso no lugar de espera EY.
- $\text{minAY}$  = borda mínima do intervalo estático associada ao lugar que representa a atividade AY.
- $\text{maxAY}$  = borda máxima do intervalo estático associada ao lugar que representa a atividade AY.

Tomando-se como exemplo  $\min E2 = \min E1 + \min A1$ , tem-se que a borda mínima do intervalo de visibilidade associada a um caso específico no lugar E2 é igual à borda mínima do intervalo de visibilidade associada ao mesmo caso no lugar E1 mais a borda mínima do intervalo estático associada ao lugar que representa a atividade A1. No trabalho de Oliveira, foi ressaltado ainda que os intervalos de visibilidade representam intervalos de datas, enquanto os intervalos estáticos representam intervalos de duração.

As figuras 4.1 e 4.2 mostram como calcular as bordas mínimas e máximas, respectivamente, em um roteiro sequencial. O caso considerado é representado por uma ficha no lugar E1, onde ele pode ser iniciado na data 30 (borda mínima de intervalo de visibilidade associado ao caso em E1). A máxima duração do caso foi estipulada em 45 unidades de tempo e, conseqüentemente, a borda máxima do intervalo de visibilidade associada ao caso em E3 é de 75 (borda mínima do intervalo de visibilidade associada ao caso em E1 mais a máxima duração do caso). A fim de se calcular as demais bordas dos intervalos de visibilidade associadas aos lugares de espera, consideram-se as bordas mínimas e máximas associadas aos lugares que representam as atividades, como mostra as fórmulas associadas às figuras.

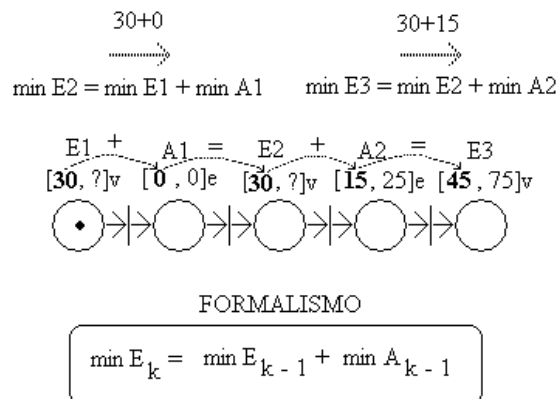


Figura 4.1: Cálculo das bordas mínimas de um roteiro sequencial

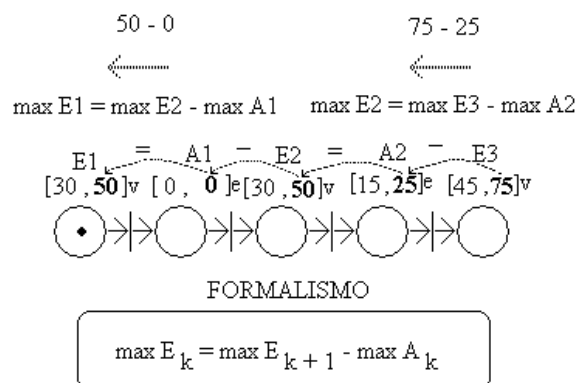


Figura 4.2: Cálculo das bordas máximas de um roteiro sequencial

A linguagem de programação Visual Prolog foi adotada neste trabalho, para implemen-

tação de um protótipo da abordagem apresentada. O Prolog é uma linguagem declarativa na qual se especifica o que se sabe e o que deve ser feito. Por ser uma linguagem mais direcionada ao conhecimento e menos direcionada aos algoritmos, ela é adequada para trabalhar com programação por restrição. O Prolog não possui estruturas de controle *do-while*, *repeat-until*, *if-then-else*, *for*, *case* ou *switch* conforme são encontradas em outras linguagens. Em Prolog, utilizam-se métodos lógicos para declarar como o programa atinge seu objetivo.

O Prolog se destaca das demais linguagens de programação devido a sua capacidade de Busca e Casamento de Padrões.

O Turbo Prolog é uma linguagem que foi criada pela Borland e hoje é desenvolvida e comercializada pelo *Prolog Development Center*, na Dinamarca. Turbo Prolog é diferente do Prolog padrão assim como o Turbo Pascal é diferente do Pascal e o Turbo C é diferente do C ANSI. O Visual Prolog, adotado neste trabalho, é uma implementação para Windows do Turbo Prolog também comercializado pelo *Prolog Development Center*.

Um programa escrito em Visual Prolog é dividido em várias partes, cada uma com um propósito específico. Algumas dessas partes são:

- **predicates:** nessa parte são declarados os predicados que poderão ser usados no programa, indicando o domínio de seus argumentos. A definição de cada predicado pode ser comparada com a declaração do protótipo da função que é feita em linguagem C. Um predicado deve ser definido para cada fato ou regra a ser utilizado no programa.
- **clauses:** nessa parte são declarados todos os fatos e todas as regras do programa.
- **goal:** é a chamada meta, ou seja, o objetivo do programa. É uma questão que vai dar início ao programa.

O código fonte, a seguir apresenta o programa em linguagem Visual Prolog do cálculo das bordas mínimas do roteiro sequencial apresentado na figura 4.1.

```

01. predicates
02. calc_borda_minima (real, string) % definição do predicado
    % para o cálculo da borda mínima dos intervalos de
    % visibilidade.

03. clauses
04. calc_borda_minima (X, "borda_minima_E1"):- % início do fato
    % de que a data de início ao mais cedo associada ao caso
    % considerado será no instante igual a 30.

05. X = 30.0,!. % final do fato de que a data de início ao mais

```

```
% cedo associada ao caso considerado será no instante igual
% a 30.
06. calc_borda_minima (X, "borda_minima_A1"):- % início do fato
% de que a duração mínima da atividade A1 é igual a 0.
07. X = 0.0,! . % final do fato de que a duração mínima
% da atividade A1 é igual a 0.
08. calc_borda_minima (X, "borda_minima_A2"):- % início do fato
% de que a duração mínima da atividade A2 é igual a 15.
09. X = 15.0,! . % final do fato de que a duração mínima
% da atividade A2 é igual a 15.
10. calc_borda_minima (X, "borda_minima_E2"):- % início da regra
% para calcular a borda mínima do intervalo de visibilidade
% associada ao caso no lugar de espera E2.
11. calc_borda_minima (Y, "borda_minima_E1"),
12. A = Y,
13. calc_borda_minima (Z, "borda_minima_A1"),
14. B = Z,
15. X = A + B,! . % final da regra para calcular a borda mínima do
% intervalo de visibilidade associada ao caso no lugar de
% espera E2.
16. calc_borda_minima (X, "borda_minima_E3"):- % início da regra
% para calcular a borda mínima do intervalo de visibilidade
% associada ao caso no lugar de espera E3.
17. calc_borda_minima (Y, "borda_minima_E2"),
18. A = Y,
19. calc_borda_minima (Z, "borda_minima_A2"),
20. B = Z,
21. X = A + B,! . % final da regra para calcular a borda mínima do
% intervalo de visibilidade associada ao caso no lugar de
% espera E3.

22. goal
23. calc_borda_minima(MinE2, "borda_minima_E2"), % chamada para
% calcular a borda mínima do intervalo de visibilidade
% associada ao caso no lugar de espera E2.
24. calc_borda_minima(MinE3, "borda_minima_E3"). % chamada para
% calcular a borda mínima do intervalo de visibilidade
% associada ao caso no lugar de espera E3.
```

O código fonte a seguir apresenta o programa em linguagem Visual Prolog do cálculo das bordas máximas do roteiro sequencial apresentado na figura 4.2.

```
01. predicates
02. calc_borda_maxima (real, string) % definição do predicado para
    % o cálculo da borda máxima dos intervalos de visibilidade.

03. clauses
04. calc_borda_maxima (X, "borda_maxima_E3"):- % início do fato de
    % que a data de término ao mais tarde associada ao caso
    % considerado será no instante igual a 75.
05. X = 75.0,! . % final do fato de que a data de término ao mais
    % tarde associada ao caso considerado será no instante igual
    % a 75.
06. calc_borda_maxima (X, "borda_maxima_A1"):- % início do fato de
    % que a duração máxima da atividade A1 é igual a 0.
07. X = 0.0,! . % final do fato de que a duração máxima
    % da atividade A1 é igual a 0.
08. calc_borda_maxima (X, "borda_maxima_A2"):- % início do fato de
    % que a duração máxima da atividade A2 é igual a 25.
09. X = 25.0,! . % final do fato de que a duração máxima
    % da atividade A2 é igual a 25.
10. calc_borda_maxima (X, "borda_maxima_E2"):- % início da regra
    % para calcular a borda máxima do intervalo de visibilidade
    % associada ao caso considerado no lugar de espera E2.
11. calc_borda_maxima (Y, "borda_maxima_E3"),
12. A = Y,
13. calc_borda_maxima (Z, "borda_maxima_A2"),
14. B = Z,
15. X = A - B,! . % final da regra para calcular a borda máxima do
    % intervalo de visibilidade associada ao caso considerado
    % no lugar de espera E2.
16. calc_borda_maxima (X, "borda_maxima_E1"):- % início da regra para
    % calcular a borda máxima do intervalo de visibilidade associada ao
    % caso considerado no lugar de espera E1.
17. calc_borda_maxima (Y, "borda_maxima_E2"),
18. A = Y,
19. calc_borda_maxima (Z, "borda_maxima_A1"),
20. B = Z,
```

- 21. `X = A - B, !. % final da regra para calcular a borda máxima do  
% intervalo de visibilidade máximo associada ao caso considerado  
% no lugar de espera E1.`
- 22. `goal`
- 23. `calc_borda_maxima(MaxE2, "borda_maxima_E2"), % chamada para calcular  
% a borda máxima do intervalo de visibilidade associada ao caso  
% considerado no lugar de espera E2.`
- 24. `calc_borda_maxima(MaxE3, "borda_maxima_E1"). % chamada para calcular  
% a borda máxima do intervalo de visibilidade associada ao caso  
% considerado no lugar de espera E1.`

As figuras 4.3 e 4.4 ilustram, respectivamente, a maneira de se calcular as bordas mínimas e máximas de um roteiro paralelo.

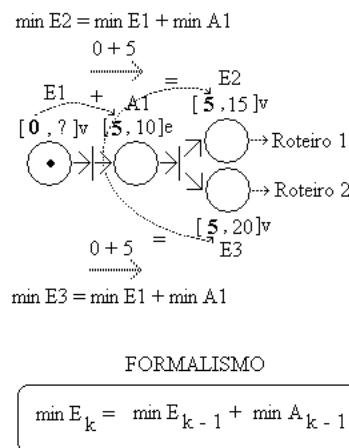


Figura 4.3: Cálculo das bordas mínimas de um roteiro paralelo

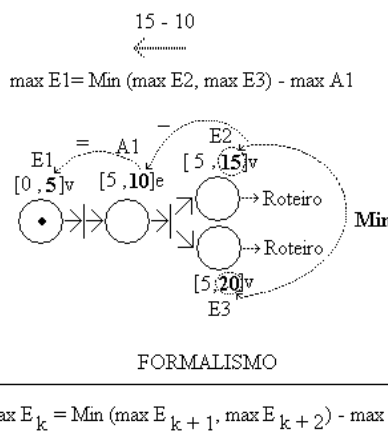


Figura 4.4: Cálculo das bordas máximas de um roteiro paralelo

O caso considerado para os roteiros paralelos, nas figuras 4.3 e 4.4, é representado por uma ficha no lugar E1 e pode ser iniciado na data 0. Supondo-se que a duração máxima do caso seja tal que as bordas máximas calculadas para os lugares E2 e E3 sejam 15 e 20, respectivamente. Uma vez que estes lugares pertencem a diferentes roteiros (roteiro 1 e roteiro 2) suas bordas não são necessariamente iguais. As bordas mínimas associadas ao caso nos lugares E2 e E3 são calculadas considerando-se a borda mínima do intervalo de visibilidade associada ao caso no lugar E1 e a borda mínima do intervalo estático associado ao lugar A1. A borda máxima do intervalo de visibilidade associada ao caso no lugar E1 é calculada levando em consideração o menor valor entre as bordas máximas dos intervalos de visibilidade associadas ao caso nos lugares E2 e E3, isto é  $(\text{Min}(\text{maxE2}, \text{maxE3}))$  como apresenta a figura 4.4. Calculando-se a borda máxima do intervalo de visibilidade em E1 desta maneira, fica claro que a ficha neste lugar irá chegar em E2 e E3, no máximo na data 15 ( $\text{maxE1} + \text{maxA1}$ ). Dessa forma, não irá ocorrer nenhuma violação de restrição temporal, ao se considerar os intervalos de visibilidade previstos associados ao caso em E2 e E3.

O código fonte a seguir apresenta o programa em linguagem Visual Prolog do cálculo das bordas mínimas do roteiro paralelo apresentado na figura 4.3.

```

predicates
01.   calc_borda_minima (real, string) % definição do predicado para
      % o cálculo da borda mínima dos intervalos de visibilidade.
02.   clauses

03.   calc_borda_minima (X, "borda_minima_E1"):- % início do fato de
      % que a data de início ao mais cedo associada ao caso considerado
      % será no instante igual a 0.
04.   X = 0.0,! . % final do fato de que a data de início ao mais cedo
      % associada ao caso considerado será no instante igual a 0.
05.   calc_borda_minima (X, "borda_minima_A1"):- % início do fato de que
      % a duração mínima da atividade A1 é igual a 5.
06.   X = 5.0,! . % final do fato de que a duração mínima
      % da atividade A1 é igual a 5.
07.   calc_borda_minima (X, "borda_minima_E2"):- % início da regra para
      % calcular a borda mínima do intervalo de visibilidade associada ao
      % caso no lugar de espera E2.
08.   calc_borda_minima (Y, "borda_minima_E1"),
09.   A = Y,
10.   calc_borda_minima (Z, "borda_minima_A1"),
11.   B = Z,

```

```

12.  X = A + B,! . % final da regra para calcular a borda mínima do
      % intervalo de visibilidade associada ao caso no lugar de espera E2.
13.  calc_borda_minima (X, "borda_minima_E3"):- % início da regra para
      % calcular a borda mínima do intervalo de visibilidade associada ao
      % caso no lugar de espera E3.
14.  calc_borda_minima (Y, "borda_minima_E1"),
15.  A = Y,
16.  calc_borda_minima (Z, "borda_minima_A1"),
17.  B = Z,
18.  X = A + B,! . % final da regra para calcular a borda mínima do
      % intervalo de visibilidade associada ao caso no lugar de
      % espera E3.

19.  goal
20.  calc_borda_minima(MinE2, "borda_minima_E2"), % chamada para
      %calcular a borda mínima do intervalo de visibilidade associada
      % ao caso no lugar de espera E2.
21.  calc_borda_minima(MinE3, "borda_minima_E3"). % chamada para
      % calcular a borda mínima do intervalo de visibilidade associada
      % ao caso no lugar de espera E3.

```

O código fonte a seguir apresenta o programa em linguagem Visual Prolog do cálculo das bordas máximas do roteiro paralelo apresentado na figura 4.4.

```

01.  predicates
02.  calc_borda_maxima (real, string) % definição do predicado para
      % o cálculo da borda máxima dos intervalos de visibilidade.
03.  minimo(real, real, real) % definição do predicado para
      % o cálculo do valor mínimo entre duas bordas.

04.  clauses
05.  minimo(X, Min, Max):- % início da regra para calcular o
      % valor mínimo entre duas bordas (quando o valor da variável
      % Min for menor que o valor da variável Max).
06.  Min < Max,
07.  X = Min,! . % final da regra para calcular o valor mínimo
      % entre duas bordas (quando o valor da variável Min
      % for menor que o valor da variável Max).

```



```
08. minimo(X, Min,Max):- % início da regra para calcular o valor
    % mínimo entre duas bordas (quando o valor da variável Min
    % for maior que o valor da variável Max).
09. Min > Max,! ,
10. X = Max,! . % final da regra para calcular o valor mínimo
    % entre duas bordas (quando o valor da variável Min
    % for maior que o valor da variável Max).

11. calc_borda_maxima (X, "borda_maxima_E2"):- % início do
    % fato de que a data de término ao mais tarde associada
    % ao caso considerado, levando em consideração o lugar
    % E2, será no instante igual a 15.
12. X = 15.0,! . % final do fato de que a data de término ao
    % mais tarde associada ao caso considerado, levando em
    % consideração o lugar E2, será no instante igual a 15.
13. calc_borda_maxima (X, "borda_maxima_E3"):- % início do
    % fato de que a data de término ao mais tarde associada
    % ao caso considerado, levando em consideração o lugar
    % E3, será no instante igual a 20.
14. X = 20.0,! . % final do fato de que a data de término ao
    % mais tarde associada ao caso considerado, levando em
    % consideração o lugar E3, será no instante igual a 20.
15. calc_borda_maxima (X, "borda_maxima_A1"):- % início do
    % fato de que a duração máxima da atividade A1 é igual
    % a 10.
16. X = 10.0,! . % final do fato de que a duração máxima
    % da atividade A1 é igual a 10.
17. calc_borda_maxima (X, "borda_maxima_E1"):- % início da
    % regra para calcular a borda máxima do intervalo de
    % visibilidade associada ao caso no lugar de espera E1.
18. calc_borda_maxima (Y, "borda_maxima_E2"),
19. A = Y,
20. calc_borda_maxima (Z, "borda_maxima_E3"),
21. B = Z,
22. minimo(Min, A, B),
23. calc_borda_maxima (W, "borda_maxima_A1"),
24. X = Min - W,! . % final da regra para calcular a borda
    % máxima do intervalo de visibilidade associada ao caso no
```

```

% lugar de espera E1.

25. goal.
26. calc_borda_maxima(MaxE1, "borda_maxima_E1"). % chamada para
% calcular a borda máxima do intervalo de visibilidade
% associada ao caso no lugar de espera E1.

```

As figuras 4.5 e 4.6 ilustram, respectivamente, o cálculo das bordas mínimas e máximas de uma sincronização entre dois roteiros.

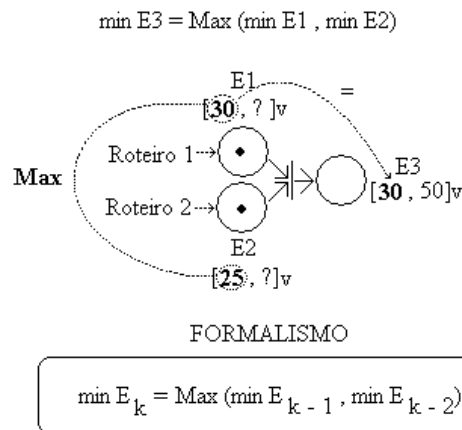


Figura 4.5: Cálculo das bordas mínimas de uma sincronização de dois roteiros

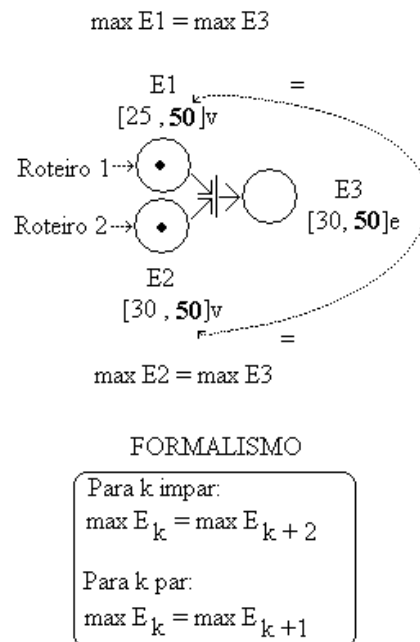


Figura 4.6: Cálculo das bordas máximas de uma sincronização de dois roteiros

Supondo-se que a data mínima de início de um caso seja tal que as bordas mínimas calculadas para os lugares E2 e E3, sejam respectivamente, 25 e 30, e a máxima duração do

caso seja tal que a borda máxima calculada para o lugar E3 seja 50. Pertencendo a roteiros diferentes (roteiro 1 e roteiro 2), as bordas dos lugares E1 e E2 não serão necessariamente iguais. A borda mínima associada ao caso no lugar E3 é calculada considerando o maior valor entre as bordas mínimas dos intervalos de visibilidade associadas ao caso nos lugares E1 e E2 ( $\text{Max}(\text{minE1}, \text{min E2})$ ). As bordas máximas associadas ao caso nos lugares E1 e E2 são iguais à máxima borda do intervalo de visibilidade associada ao caso em E3.

O código fonte a seguir apresenta o programa em linguagem Visual Prolog do cálculo das bordas mínimas da sincronização de dois roteiros como apresentado na figura 4.5.

```

01. predicates
02. calc_borda_minima (real, string) % definição do predicado para
    % o cálculo da borda mínima dos intervalos de visibilidade.
03. maximo(real, real, real) % definição do predicado para
    % o cálculo do valor máximo entre duas bordas.

04. clauses
05. maximo(Y, Min,Max):- % início da regra para calcular o
    % valor máximo entre duas bordas (quando o valor da variável
    % Min for menor que o valor da variável Max).
06. Min < Max,
07. Y = Max,! % final da regra para calcular o valor máximo
    % entre duas bordas (quando o valor da variável Min
    % for menor que o valor da variável Max).

08. maximo(Y, Min,Max):- % início da regra para calcular o valor
    % máximo entre duas bordas (quando o valor da variável Min
    % for maior que o valor da variável Max).
09. Min > Max,! ,
10. Y = Min,! % final da regra para calcular o valor máximo
    % entre duas bordas (quando o valor da variável Min
    % for maior que o valor da variável Max).

11. calc_borda_minima (X, "borda_minima_E1"):- % início do
    % fato de que a data de início ao mais cedo do caso
    % considerado, levando em consideração o lugar E1, será
    % no instante igual a 30.
12. X = 30.0,! % final do fato de que a data de início ao mais cedo
    % do caso considerado, levando em consideração
    % o lugar E1, será no instante igual a 30.

```

```

13.  calc_borda_minima (X, "borda_minima_E2"):- % início do fato de
      % que a data de início ao mais cedo do caso considerado,
      % levando em consideração o lugar E2, será no instante igual a 25.
14.  X = 25.0,! . % final do fato de que a data de início ao mais cedo
      % do caso considerado, levando em consideração o lugar E2, será
      % no instante igual a 25.
15.  calc_borda_minima (X, "borda_minima_E3"):- % início da regra para
      % calcular a borda mínima do intervalo de visibilidade associada ao
      % caso no lugar de espera E3.
16.  calc_borda_minima (Y, "borda_minima_E1"),
17.  A = Y,
18.  calc_borda_minima (Z, "borda_minima_E2"),
19.  B = Z,
20.  maximo(Max, A, B),
21.  X = Max,! . % final da regra para calcular a borda mínima do
      % intervalo de visibilidade associada ao caso no lugar de
      % espera E3.

22.  goal
23.  calc_borda_minima(MinE3, "borda_minima_E3"). % chamada para calcular
      % a borda mínima do intervalo de visibilidade associada ao caso no
      % lugar de espera E3.

```

O código fonte a seguir apresenta o programa em linguagem Visual Prolog do cálculo das bordas máximas da sincronização de dois roteiros como apresentado na figura 4.6.

```

01.  predicates
02.  calc_borda_maxima (real, string) % definição do predicado para
      % o cálculo da borda máxima dos intervalos de visibilidade.

03.  clauses
04.  calc_borda_maxima (X, "borda_maxima_E3"):- % início do fato de
      % que a data de término ao mais tarde associada ao caso
      % considerado no lugar de espera E3 será no instante igual a 50.
05.  X = 50.0,! . % final do fato de que a data de término ao mais
      % tarde associada ao caso considerado no lugar de espera E3
      % será no instante igual a 50.
06.  calc_borda_maxima (X, "borda_maxima_E1"):- % início da regra para

```

```

% calcular a borda máxima do intervalo de visibilidade associada
% ao caso no lugar de espera E1.
07. calc_borda_maxima (X, "borda_maxima_E3"),!. % final da regra para
% calcular a borda máxima do intervalo de visibilidade associada
% ao caso no lugar de espera E1.

08. calc_borda_maxima (X, "borda_maxima_E2"):- % início da regra para
% calcular a borda máxima do intervalo de visibilidade associada
% ao caso no lugar de espera E2.
09. calc_borda_maxima (X, "borda_maxima_E3"),!. % final da regra para
% calcular a borda máxima do intervalo de visibilidade associada
% ao caso no lugar de espera E2.

10. goal
11. calc_borda_maxima(MaxE2, "borda_maxima_E1"), % chamada para
% calcular a borda máxima do intervalo de visibilidade associada
% ao caso no lugar de espera E2.
12. calc_borda_maxima(MaxE3, "borda_maxima_E2"). % chamada para
% calcular a borda máxima do intervalo de visibilidade associada
% ao caso no lugar de espera E3.

```

No caso de um roteiro alternativo, as figuras 4.7 e 4.8 mostram, respectivamente, como se calcular as bordas mínimas e máximas. O caso considerado é representado por uma ficha no lugar E1 e pode ser iniciado na data 30. Suponha-se que a máxima duração do caso seja tal que as bordas máximas calculadas para os lugares E2 e E3 sejam, respectivamente, 90 e 75. Como esses lugares pertencem a diferentes roteiros (roteiro 1 e roteiro 2) suas bordas não são iguais necessariamente. As bordas mínimas associadas ao caso nos lugares E2 e E3 são calculadas da mesma forma que um roteiro sequencial, isto é, considerando a borda mínima do intervalo de visibilidade associada ao caso no lugar E1 e a borda mínima do intervalo estático associada ao lugar A1. Já a borda máxima do intervalo de visibilidade associada ao caso no lugar E1 é calculada considerando o menor valor entre as bordas máximas dos intervalos de visibilidade associadas ao caso nos lugares E2 e E3 ( $\text{Min}(\text{maxE1}, \text{maxE2})$ ) como mostrado na figura 4.8. Calculando-se, dessa maneira, a borda máxima do intervalo de visibilidade em E1, é possível perceber que a ficha em E1 irá chegar em E2 ou E3 na data máxima 75 ( $\text{maxE1} + \text{maxA1}$ ), não ocorrendo nenhuma violação temporal explícita, ao considerar os intervalos de visibilidade previsíveis associados à E2 e E3.

O código fonte a seguir apresenta o programa em linguagem Visual Prolog do cálculo das bordas mínimas do roteiro alternativo apresentado na figura 4.7.

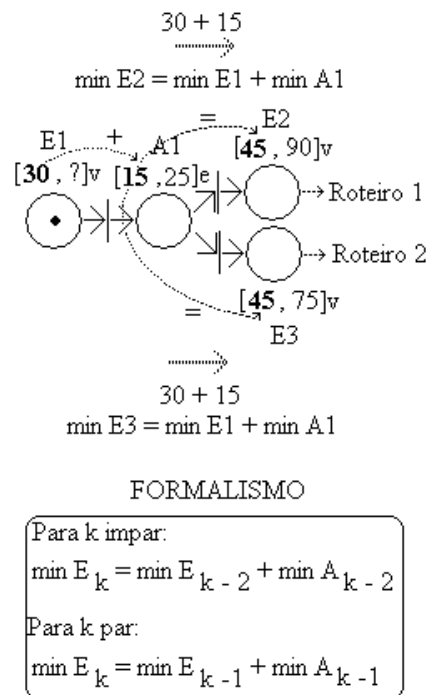


Figura 4.7: Cálculo das bordas mínimas de um roteiro alternativo

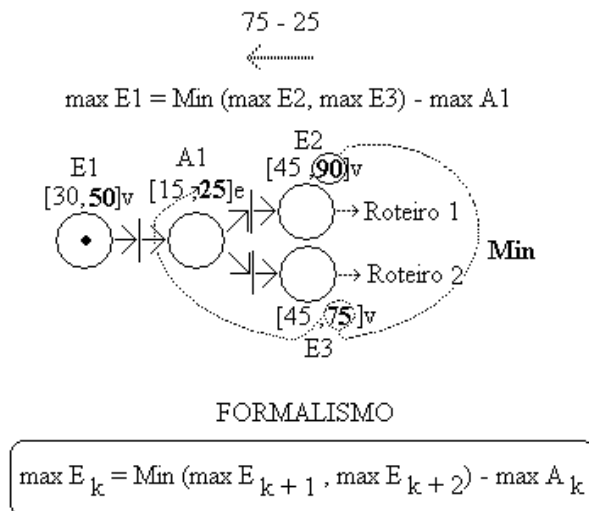


Figura 4.8: Cálculo das bordas máximas de um roteiro alternativo

01. predicates
02. calc\_borda\_minima (real, string) % definição do predicado para % o cálculo da borda mínima dos intervalos de visibilidade.
03. clauses
04. calc\_borda\_minima (X, "borda\_minima\_E1"):- % início do fato de % que a data de início ao mais cedo, do caso considerado, % levando em consideração o lugar E1, será no instante igual

```
% a 30.
05. X = 30.0,! . % final do fato de que a data de início ao mais cedo,
    % do caso considerado, levando em consideração o lugar E1, será
    % no instante igual a 30.
06. calc_borda_minima (X, "borda_minima_A1"):- % início do fato de que
    % a duração mínima da atividade A1 é igual a 15.
07. X = 15.0,! . % final do fato de que a duração mínima da atividade A1
    % é igual a 15.
08. calc_borda_minima (X, "borda_minima_E2"):- % início da regra para
    % calcular a borda mínima do intervalo de visibilidade associada ao
    % caso no lugar de espera E2.
09. calc_borda_minima (Y, "borda_minima_E1"),
10. A = Y,
11. calc_borda_minima (Z, "borda_minima_A1"),
12. B = Z,
13. X = A + B,! . % final da regra para calcular a borda mínima do
    % intervalo de visibilidade associada ao caso no lugar de
    % espera E2.
14. calc_borda_minima (X, "borda_minima_E3"):- % início da regra
    % para calcular a borda mínima do intervalo de visibilidade
    % associada ao caso no lugar de espera E3.
15. calc_borda_minima (Y, "borda_minima_E1"),
16. A = Y,
17. calc_borda_minima (Z, "borda_minima_A1"),
18. B = Z,
19. X = A + B,! . % final da regra para calcular a borda mínima do
    % intervalo de visibilidade associada ao caso no lugar de
    % espera E3.

20. goal
21. calc_borda_minima(MinE2, "borda_minima_E2"), % chamada para
    % calcular a borda mínima do intervalo de visibilidade associada
    % ao caso no lugar de espera E2.
22. calc_borda_minima(MinE3, "borda_minima_E3"). % chamada para
    % calcular a borda mínima do intervalo de visibilidade associada
    % ao caso no lugar de espera E3.
```

O código fonte a seguir apresenta o programa em linguagem Visual Prolog do cálculo das bordas máximos do roteiro alternativo apresentado na figura 4.8.

```
01. predicates
02. calc_borda_maxima (real, string) % definição do predicado para
    % o cálculo da borda máxima dos intervalos de visibilidade.
03. minimo(real, real, real) % definição do predicado para
    % o cálculo do valor mínimo entre duas bordas.

04. clauses
05. minimo(X, Min, Max):- % início da regra para calcular o valor
    % mínimo entre duas bordas (quando o valor da variável Min
    % for menor que o valor da variável Max).
06. Min < Max,
07. X = Min,! . % final da regra para calcular o valor mínimo
    % entre duas bordas (quando o valor da variável Min
    % for menor que o valor da variável Max).

08. minimo(X, Min,Max):- % início da regra para calcular o
    % valor mínimo entre duas bordas (quando o valor da variável
    % Min for maior que o valor da variável Max).
09. Min > Max,! ,
10. X = Max,! . % final da regra para calcular o valor mínimo
    % entre duas bordas (quando o valor da variável Min
    % for maior que o valor da variável Max).

11. calc_borda_maxima (X, "borda_maxima_E2"):- % início do fato de
    % que a data de término ao mais tarde associada ao caso
    % considerado no lugar de espera E2 será no instante igual a 90.
12. X = 90.0,! . % final do fato de que a data de término ao mais
    % tarde associada ao caso considerado no lugar de espera E2
    % será no instante igual a 90.
13. calc_borda_maxima (X, "borda_maxima_E3"):- % início do fato de
    % que a data de término ao mais tarde associada ao caso
    % considerado no lugar de espera E3 será no instante igual a 75.
14. X = 75.0,! . % final do fato de que a data de término ao mais
    % tarde associada ao caso considerado no lugar de espera E3
    % será no instante igual a 75.
15. calc_borda_maxima (X, "borda_maxima_A1"):- % início do fato de
    % que a duração máxima da atividade A1 é igual a 25.
16. X = 25.0,! . % final do fato de que a duração máxima
```



```
% da atividade A1 é igual a 25.

17. calc_borda_maxima (X, "borda_maxima_E1"):- % início da regra para
    % calcular a borda máxima do intervalo de visibilidade associada
    % ao caso no lugar de espera E1.
18. calc_borda_maxima (Y, "borda_maxima_E2"),
19. A = Y,
20. calc_borda_maxima (Z, "borda_maxima_E3"),
21. B = Z,
22. minimo(Min, A, B),
23. calc_borda_maxima (W, "borda_maxima_A1"),
24. X = Min - W,! % final da regra para calcular a borda máxima do
    % intervalo de visibilidade associada ao caso no lugar de
    % espera E1.

25. goal
26. calc_borda_maxima(MaxE1, "borda_maxima_E1"). % chamada para
    % calcular a borda máxima do intervalo de visibilidade associada
    % ao caso no lugar de espera E1.
```

## 4.2 Análise sob restrições baseada em um Raciocínio Energético aplicado aos sistemas de gerenciamento de *workflow*

O Raciocínio Energético é uma técnica de análise sob restrições. Como já apresentado no capítulo 3, tal técnica foi utilizada inicialmente no contexto do problema de escalonamento dos sistemas de manufatura [Lopez & Esquirol, 1996] com o objetivo de evitar enumeração explícita de possíveis utilizações de recursos compartilhados (conflitos entre tarefas em competição por recursos limitados).

Um dos objetivos da abordagem proposta neste trabalho é o de aplicar o Raciocínio Energético no contexto do problema de escalonamento tempo real dos sistemas de gerenciamento de *workflow*, a fim de melhorar o conjunto de restrições temporais (base de conhecimento) que o jogador de redes de Petri irá utilizar para tentar encontrar uma solução de escalonamento em tempo real.

### 4.2.1 Restrições do problema

A seguir, são apresentadas as variáveis (restrições) do problema de escalonamento de sistemas de gerenciamento de *workflow* proposto neste trabalho:

- em um processo de *workflow*, um conjunto de  $n$  tarefas precisa ser executado, sendo que suas execuções não podem ser interrompidas (problema de escalonamento de tarefas não-preemptivo);
- cada tarefa  $i$  possui uma data de início ao mais cedo  $r_i$ , uma data de início ao mais tarde  $\bar{s}_i$  e uma data de final ao mais tarde  $d_i$ .
- cada tarefa  $i$  é caracterizada por suas durações mínimas  $p_{min_i}$  e máximas  $p_{max_i}$ , (relativas aos intervalos estáticos do modelo de redes de Petri p-temporal) e deve ser executada dentro de uma janela de tempo  $[r_i, d_i]$ , a qual resulta do intervalo de visibilidade associado à execução do caso tratado.
- cada tarefa  $i$  necessita de uma intensidade constante  $a_i$  de um dado recurso que disponibiliza uma intensidade global  $A$  para as diversas tarefas que o utilizam.
- o conjunto  $S$  denota variáveis de restrições as quais representam as datas de início das tarefas consistentes com as janelas de tempo  $[r_i, d_i]$  e com as restrições de utilização dos recursos.

### 4.2.2 Intervalos de análise

Para se aplicar a técnica do Raciocínio Energético, precisa-se, primeiramente definir um intervalo de análise ou intervalo  $\Delta$ . Existe uma infinidade de deltas que podem ser utilizados para se aplicar a técnica do Raciocínio Energético. Nos trabalhos de Lopez *et al* [Lopez & Esquirol, 1996], [Erschler & Lopez, 1990] e no trabalho de Henet [Henet, 2002] não é especificado como pode ser feita a escolha desses intervalos; contudo precisa-se especificar os intervalos deltas a partir dos quais o Raciocínio Energético poderá ser aplicado. A seguir, são apresentados os intervalos deltas que foram considerados neste trabalho.

- **1° intervalo  $\Delta$** : Utilizado para o cálculo da consistência global:  $S_{\Delta}=r_i; F_{\Delta}=d_i$ ;
- **2° intervalo  $\Delta$** : Utilizado para o cálculo da consistência local a fim de verificar a consistência da data de início ao mais cedo:  $S_{\Delta}=r_i; F_{\Delta}=r_i+p_{min_i}$ ;
- **3° intervalo  $\Delta$** : Utilizado para o cálculo da consistência local a fim de verificar a consistência da data de início ao mais tarde:  $S_{\Delta}=\bar{s}_i; F_{\Delta}=\bar{s}_i+p_{min_i}$ ;

### 4.2.3 Aplicação do Raciocínio Energético

A aplicação do Raciocínio Energético deve ser feita sobre cada uma das atividades de um determinado processo de *workflow*. A seguir é apresentada passo a passo a metodologia genérica para aplicação do Raciocínio Energético:

- **1° Passo:** Verifica-se a consistência global sobre cada uma das atividades do processo levando em consideração o recurso utilizado pela atividade em questão. A consistência global é calculada considerando o 1° intervalo  $\Delta$ . No caso em que a energia disponível do recurso em questão ( $W^\Delta$ ) for menor do que a soma das energias mínimas necessárias para execução das atividades ( $\underline{W}_i^\Delta$ ), o escalonamento é pré-identificado como inadmissível, por causa da incapacidade energética do recurso em questão. Caso contrário ( $\sum_{i=1}^n \underline{W}_i^\Delta \leq W^\Delta$ ) passa-se para o 2° passo. Da mesma forma que no problema de escalonamento da manufatura [Lopez & Esquirol, 1996], no problema de escalonamento dos sistemas de gerenciamento de *workflow*, o escalonamento das atividades só será possível quando a condição de consistência global for satisfeita por todas as atividades envolvidas com um determinado recurso.
- **2° Passo:** Verifica-se a consistência local sobre cada uma das atividades do processo a fim de verificar a consistência da data de início ao mais cedo da atividade em questão. Esta consistência local é calculada considerando o 2° intervalo  $\Delta$ . No caso em que a duração máxima equivalente da atividade em questão ( $\bar{p}_i^\Delta$ ) for menor que a sua duração mínima ( $pmin_i$ ), um novo intervalo  $\Delta$  pode ser definido até se encontrar uma situação onde  $\bar{p}_i^\Delta \geq pmin_i$ . O novo intervalo  $\Delta$  será, na verdade, um deslocamento à direita do antigo intervalo, respeitando a restrição temporal  $d_i$  (final ao mais tarde da atividade). O valor desse deslocamento é calculado subtraindo-se o valor da duração máxima equivalente ( $\bar{p}_i^\Delta$ ) do valor da duração mínima ( $pmin_i$ ) da atividade em questão.
- **3° Passo:** Verifica-se a consistência local sobre cada uma das atividades do processo a fim de verificar a consistência da data de início ao mais tarde da atividade em questão. Esta consistência local é calculada considerando o 3° intervalo  $\Delta$ . Da mesma forma que no 2° passo, quando acontecer de a duração máxima equivalente da atividade em questão ( $\bar{p}_i^\Delta$ ) ser menor que a sua duração mínima ( $pmin_i$ ), um novo intervalo  $\Delta$  pode ser definido até se encontrar uma situação onde  $\bar{p}_i^\Delta \geq pmin_i$ . O novo intervalo  $\Delta$  será, na verdade, um deslocamento à esquerda do antigo intervalo, respeitando a restrição temporal  $r_i$  (início ao mais cedo da atividade). Como no 2° passo, o valor desse deslocamento é calculado subtraindo-se o valor da duração máxima equivalente ( $\bar{p}_i^\Delta$ ) do valor da duração mínima ( $pmin_i$ ) da atividade em questão.

A fim de ilustrar a aplicação do Raciocínio Energético, o exemplo a seguir mostra um fragmento da rotina de trabalho de um *Analista de Service Desk*. A figura 4.9 ilustra tal exemplo modelado por meio de rede de Petri p-temporal.

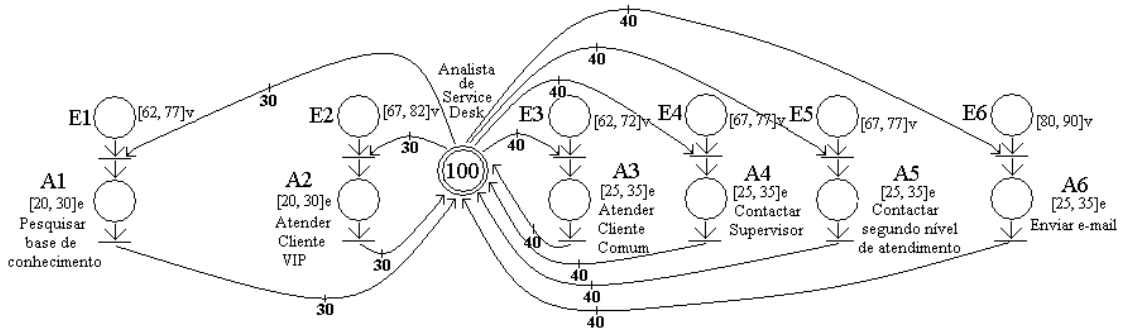


Figura 4.9: Rede de Petri p-temporal que modela um fragmento da rotina de trabalho de um *Analista de Service Desk*

$i$	A1	A2	A3	A4	A5	A6
$r_i$	62	67	62	67	67	80
$\bar{s}_i$	77	82	72	77	77	90
$p_{min_i}$	20	20	25	25	25	25
$p_{max_i}$	30	30	35	35	35	35
$d_i$	107	112	107	112	112	125
$a_i$	30	30	40	40	40	40

Tabela 4.1: Variáveis (restrições) das atividades de um *Analista de Service Desk*

Durante o dia de trabalho de um *Analista de Service Desk*, várias atividades são executadas, tais como: enviar e-mail, atender clientes comuns, atender clientes *VIPs*, entrar em contato com seu supervisor, entrar em contato com a equipe de suporte de segundo nível de atendimento e pesquisar na base de conhecimento se existe uma solução para os pedidos do cliente. O *Analista de Service Desk*, ao atender um cliente comum, precisa consultar a base de conhecimento para atender o pedido do cliente (atender cliente e consultar base de conhecimento são atividades que devem ser executadas em paralelo); já, ao atender um cliente *VIP*, ele precisa direcionar o cliente ao segundo nível de atendimento e comunicar o seu supervisor sobre o pedido do cliente (atender cliente *VIP*, contactar segundo nível de atendimento e contactar supervisor também são atividades que precisam ser executadas em paralelo). Eventualmente, o analista precisa enviar um e-mail para um cliente comunicando sobre o status da resolução do pedido. A tabela 4.1 ilustra algumas possíveis restrições relacionadas a esse exemplo em particular.

Primeiramente, a técnica do Raciocínio Energético precisa verificar se o escalonamento das tarefas não é admissível devido a indisponibilidade energética dos recursos envolvidos. O cálculo da consistência global consegue identificar escalonamentos inadmissíveis

devido a este aspecto (escassez de energia) apontando uma condição de inconsistência global. Contudo uma condição de consistência global não garante a admissibilidade de um escalonamento; neste caso, deve-se aplicar o cálculo da consistência local.

$i \rightarrow$	A1	A2
$\Delta$	[62,107]	[67,112]
$W^\Delta$	$(107 - 62) \times 100 = 4500$	$(112 - 67) \times 100 = 4500$
$\underline{W}_{A1}^\Delta$	$30 \times 20 = 600$	$30 \times 15 = 450$
$\underline{W}_{A2}^\Delta$	$30 \times 15 = 450$	$30 \times 20 = 600$
$\underline{W}_{A3}^\Delta$	$40 \times 25 = 1000$	$40 \times 20 = 800$
$\underline{W}_{A4}^\Delta$	$40 \times 20 = 800$	$40 \times 25 = 1000$
$\underline{W}_{A5}^\Delta$	$40 \times 20 = 800$	$40 \times 25 = 1000$
$\underline{W}_{A6}^\Delta$	$40 \times 7 = 280$	$40 \times 12 = 480$
$\sum_{i=1}^6 \underline{W}_i^\Delta$	3930	4330
$\sum_{i=1}^6 \underline{W}_i^\Delta \leq W^\Delta$	OK	OK

$i \rightarrow$	A3	A4
$\Delta$	[62,107]	[67,112]
$W^\Delta$	$(107 - 62) \times 100 = 4500$	$(112 - 67) \times 100 = 4500$
$\underline{W}_{A1}^\Delta$	$30 \times 20 = 600$	$30 \times 15 = 450$
$\underline{W}_{A2}^\Delta$	$30 \times 15 = 450$	$30 \times 20 = 600$
$\underline{W}_{A3}^\Delta$	$40 \times 25 = 1000$	$40 \times 20 = 800$
$\underline{W}_{A4}^\Delta$	$40 \times 20 = 800$	$40 \times 25 = 1000$
$\underline{W}_{A5}^\Delta$	$40 \times 20 = 800$	$40 \times 25 = 1000$
$\underline{W}_{A6}^\Delta$	$40 \times 7 = 280$	$40 \times 12 = 480$
$\sum_{i=1}^6 \underline{W}_i^\Delta$	3930	4330
$\sum_{i=1}^6 \underline{W}_i^\Delta \leq W^\Delta$	OK	OK

$i \rightarrow$	A5	A6
$\Delta$	[67,112]	[80,125]
$W^\Delta$	$(112 - 67) \times 100 = 4500$	$(125 - 80) \times 100 = 4500$
$\underline{W}_{A1}^\Delta$	$30 \times 15 = 450$	$30 \times 2 = 60$
$\underline{W}_{A2}^\Delta$	$30 \times 20 = 600$	$30 \times 7 = 210$
$\underline{W}_{A3}^\Delta$	$40 \times 20 = 800$	$40 \times 7 = 280$
$\underline{W}_{A4}^\Delta$	$40 \times 25 = 1000$	$40 \times 12 = 480$
$\underline{W}_{A5}^\Delta$	$40 \times 25 = 1000$	$40 \times 12 = 480$
$\underline{W}_{A6}^\Delta$	$40 \times 12 = 480$	$40 \times 25 = 1000$
$\sum_{i=1}^6 \underline{W}_i^\Delta$	4330	2510
$\sum_{i=1}^6 \underline{W}_i^\Delta \leq W^\Delta$	OK	OK

Tabela 4.2: cálculos para verificar a consistência global sobre as atividades do *Analista de Service Desk*

Voltando ao exemplo do *Analista de Service Desk*, é possível verificar se existe consistência global sobre todas as atividades ( $A1$ ,  $A2$ ,  $A3$ ,  $A4$ ,  $A5$ ,  $A6$ ). O recurso do tipo humano, *Analista de Service Desk*, possui inicialmente 100% de sua disponibilidade energética para execução dessas tarefas (figura 4.9).

A tabela 4.2 apresenta os cálculos da verificação de consistência global do exemplo do *Analista de service desk* considerando-se o critério de escolhas dos deltas adotados neste trabalho.

Como  $W^\Delta$  (para cada  $\Delta$  das atividades relacionadas com o recurso em questão) é maior que cada  $\sum_{i=1}^6 W_i^\Delta$  (relacionado ao  $\Delta$  da atividade estudada), a condição de consistência global ( $\sum_{i=1}^n W_i^\Delta \leq W^\Delta$ ) é satisfeita.

O código fonte a seguir apresenta o programa em linguagem Visual Prolog do cálculo da consistência global da atividade *A1* para o exemplo do *Analista de Service Desk*:

```
01. predicates

02. calc_W_delta(real, real, real, real) % definição do predicado para
    % o cálculo da energia fornecida por um recurso em um dado
    % intervalo delta.

03. minimo(real, real, real) % definição do predicado para
    % o cálculo da duração mínima de uma atividade dentro
    % do intervalo delta.

04. maximo(real, real, real) % definição do predicado para
    % o cálculo da interseção entre o intervalo de execução da
    % atividade e o intervalo delta.

05. calc_Wi_delta_mimima(real, real, real, real, real, real, real)
    % definição do predicado para o cálculo da energia mínima consumida
    % por uma atividade dentro de um intervalo delta.

06. calc_somatorio_energia_minima(real, real, real, real, real, real, real)
    % definição do predicado para o cálculo do somatório das energias
    % mínimas consumidas pelas atividades dentro de um intervalo delta.

07. verifica_consistencia_global(real, real) % definição do predicado para
    % a verificação de consistência global.

08. clauses

09. minimo(X, Min, Max):- % início da regra para calcular a duração
    % mínima de uma atividade dentro de um intervalo delta (quando o
    % valor da variável Min for menor que o valor da variável Max).

10. Min < Max,

11. X = Min,! % final da regra para calcular a duração mínima de uma
    % atividade dentro de um intervalo delta (quando o valor da variável
    % Min for menor que o valor da variável Max).
```

12. minimo(X, Min,Max):- % início da regra para calcular a duração  
% mínima de uma atividade dentro de um intervalo delta (quando o  
% valor da variável Min for maior que o valor da variável Max).
13. Min > Max,! ,
14. X = Max,! . % final da regra para calcular a duração mínima de uma  
% atividade dentro de um intervalo delta (quando o valor da variável  
% Min for maior que o valor da variável Max).
  
15. minimo(\_\_,\_\_,\_\_). % regra para calcular a duração mínima de uma  
% atividade dentro do um intervalo delta (quando o valor da variável  
% Min for igual ao valor da variável Max).
  
16. maximo(Y, Min,Max):- % início da regra para calcular a interseção  
% entre o intervalo de execução da atividade e o intervalo delta  
% (quando o valor da variável Min for menor que o valor da variável  
% Max).
17. Min < Max,
18. Y = Max,! . % final da regra para calcular a interseção entre o  
% intervalo de execução da atividade e o intervalo delta (quando o  
% valor da variável Min for menor que o valor da variável Max).
  
19. maximo(Y, Min,Max):- % início da regra para calcular a interseção  
% entre o intervalo de execução da atividade e o intervalo delta  
% (quando o valor da variável Min for maior que o valor da variável  
% Max).
20. Min > Max,! ,
21. Y = Min,! . % final da regra para calcular a interseção entre o  
% intervalo de execução da atividade e o intervalo delta (quando o  
% valor da variável Min for maior que o valor da variável Max).
  
22. maximo(\_\_,\_\_,\_\_). % regra para calcular a interseção entre o  
% intervalo de execução da atividade e o intervalo delta (quando  
% o valor da variável Min for igual ao valor da variável Max).
  
23. calc\_W\_delta(W\_delta, A, D2, D1):- % início da regra para  
% calcular a energia fornecida por um recurso em um dado  
% intervalo delta.
24. W\_delta = A\*(D2 - D1),! . % final da regra para calcular a

```
% energia fornecida por um recurso em um dado intervalo
% delta.

25. calc_Wi_delta_mimima(Wi_delta_minima, Ai, P, D2, D1, T1, T3):-
    % início da regra para calcular a energia mínima consumida
    % por uma atividade dentro de um intervalo delta.
26. Parcela_1 = D2 - D1,
27. Parcela_2 = T1 + P - D1,
28. Parcela_3 = D2 - T3 + P,
29. minimo(A, P, Parcela_1),
30. minimo(B, A, Parcela_2),
31. minimo(C, B, Parcela_3),
32. maximo(Result_parcial, 0, C),
33. Wi_delta_minima = Ai * Result_parcial,! % final da regra
    % para calcular a energia mínima consumida por uma
    % atividade dentro de um intervalo delta.

34. calc_somatorio_energia_minima(Somatorio, A_1, A_2, A_3, A_4, A_5, A_6):-
    % início da regra para calcular o somatório das energias mínimas
    % consumidas pelas atividades dentro de um intervalo delta.
35. Somatorio = A_1 + A_2 + A_3 + A_4 + A_5 + A_6,! % final da regra
    % para calcular o somatório das energias mínimas consumidas
    % pelas atividades dentro de um intervalo delta.

36. verifica_consistência_global(W, Somatorio_energias_minimas):-
    % início da regra para verificar a consistência global (quando a
    % energia disponível para execução das atividades for maior que o
    % somatório das energias mínimas das atividades dentro do intervalo
    % delta).
37. W > Somatorio_energias_minimas,
38. write("Não há inconsistência global"),! % final da regra para
    % verificar a consistência global (quando a energia disponível
    % para execução das atividades for maior que o somatório das
    % energias mínimas das atividades dentro do intervalo delta).

39. verifica_consistência_global(W, Somatorio_energias_minimas):-
    % início da regra para verificar a consistência global (quando a
    % energia disponível para execução das atividades for igual ao
    % somatório das energias mínimas das atividades dentro do intervalo
```



```
% delta).
40. W = Somatorio_energias_minimas,
41. write("Não há inconsistência global"),!. % final da regra para
    % verificar a consistência global (quando a energia disponível
    % para execução das atividades for igual ao somatório das energias
    % mínimas das atividades dentro do intervalo delta).

42. verifica_consistência_global(W, Somatorio_energias_minimas):-
    % início da regra para verificar a consistência global (quando
    % a energia disponível para execução das atividades for menor
    % que o somatório das energias mínimas das atividades dentro do
    % intervalo delta).
43. W < Somatorio_energias_minimas,
44. write("Há inconsistência global, o escalonamento é
45. inadmissível"),!. % final da regra para verificar a consistência
    % global (quando a energia disponível para execução das atividades
    % for menor que o somatório das energias mínimas das atividades
    % dentro do intervalo delta).

46. goal

47. calc_W_delta(W,100,107,62), % chamada para calcular a energia
    % disponível no intervalo delta para execução das atividades.

48. calc_Wi_delta_mimima(A1,30,20,107,62,62,107), % chamada para
    % calcular a energia mínima necessária para execução da
    % atividade A1 dentro do intervalo delta.

49. calc_Wi_delta_mimima(A2,30,20,107,62,67,112), % chamada para
    % calcular a energia mínima necessária para execução da
    % atividade A2 dentro do intervalo delta.

50. calc_Wi_delta_mimima(A3,40,25,107,62,62,107), % chamada para
    % calcular a energia mínima necessária para execução da
    % atividade A3 dentro do intervalo delta.

51. calc_Wi_delta_mimima(A4,40,25,107,62,67,112), % chamada para
    % calcular a energia mínima necessária para execução da
```

```
% atividade A4 dentro do intervalo delta.

52. calc_Wi_delta_mimima(A5,40,25,107,62,67,112), % chamada para
    % calcular a energia mínima necessária para execução da
    % atividade A5 dentro do intervalo delta.

53. calc_Wi_delta_mimima(A6,40,25,107,62,80,125), % chamada para
    % calcular a energia mínima necessária para execução da
    % atividade A6 dentro do intervalo delta.

54. calc_somatorio_energia_minima(Somatorio_energias_minimas, A1,
55. A2, A3, A4, A5, A6), % chamada para calcular o somatório das
    % energias mínimas necessárias para execução das atividades A1,
    % A2, A3, A4, A5 e A6 dentro do intervalo delta.

56. verifica_consistencia_global(W, Somatorio_energias_minimas).
    % chamada para verificar a consistência global.
```

Para o cálculo de consistência global das demais atividades, basta trocar os valores D2 e D1 dos parâmetros dos procedimentos *calc\_Wi\_delta\_minima* (linhas 48 até 53) e *calc\_W\_delta* (linha 47) pelo valor do  $\Delta$  correspondente à atividade em questão.

Satisfeita a condição de consistência global por todas as atividades envolvidas com o recuso em questão, o próximo passo é verificar se as atividades podem, efetivamente, serem iniciadas ao mais cedo nas datas  $r_i$  e ao mais tarde nas datas  $\bar{s}_i$ , por meio da verificação da condição de consistência local. Se for observada uma condição de inconsistência local, um deslocamento do  $\Delta$  à direita ou à esquerda poderá ser feito a fim de encontrar uma condição de consistência local.

A tabela 4.3 apresenta os cálculos da verificação de consistência local (levando em consideração o critério de escolha dos deltas) sobre a data de início ao mais cedo ( $r_i$ ) das atividades A1, A2, A3 e A4 do *Analista de Service Desk*. A tabela 4.4 apresenta os cálculos da verificação de consistência local (levando em consideração o critério de escolha dos deltas) sobre a data de início ao mais cedo ( $r_i$ ) das atividades A5 e A6 do *analista de Service Desk*.

Para a atividade A6, pode-se observar que  $\bar{p}_i^\Delta=24.75$  é menor do que sua duração mínima  $p_{\min_i}=25$ ; logo existe uma inconsistência local sobre esta atividade. Em particular, o fato de a duração equivalente no  $\Delta$  estudado ( $\bar{p}_i^\Delta$ ) ser menor que a duração mínima da atividade em questão ( $p_{\min_i}$ ) significa que o recurso *Analista de Service Desk* consegue

fornecer energia para a atividade  $A6$  durante 24.75 unidades de tempo, sendo que a duração mínima desta atividade corresponde a 25 unidades de tempo, caracterizando, assim uma inconsistência local, já que a duração de disponibilidade do recurso é inferior à duração mínima necessária para a realização desta atividade.

$i \rightarrow$	A1	A2
$\Delta$	[62,82]	[67,87]
$W^\Delta$	$(82 - 62) \times 100 = 2000$	$(87 - 67) \times 100 = 2000$
$\underline{W}_{A1}^\Delta$	Não se aplica	$30 \times 0 = 0$
$\underline{W}_{A2}^\Delta$	$30 \times 0 = 0$	Não se aplica
$\underline{W}_{A3}^\Delta$	$40 \times 0 = 0$	$40 \times 5 = 200$
$\underline{W}_{A4}^\Delta$	$40 \times 0 = 0$	$40 \times 0 = 0$
$\underline{W}_{A5}^\Delta$	$40 \times 0 = 0$	$40 \times 0 = 0$
$\underline{W}_{A6}^\Delta$	$40 \times 0 = 0$	$40 \times 0 = 0$
$\sum_{j=1, j \neq i}^n \underline{W}_j^\Delta$	0	200
$A_i^\Delta$	$2000 - 0 = 2000$	$2000 - 200 = 1800$
$\bar{p}_i^\Delta$	$\frac{2000}{30} = 66.7$	$\frac{1800}{30} = 60$
$P_{min_i}$	20	20
$\bar{p}_i^\Delta \geq P_{min_i}$	OK	OK

$i \rightarrow$	A3	A4
$\Delta$	[62,87]	[67,92]
$W^\Delta$	$(87 - 62) \times 100 = 2500$	$(92 - 67) \times 100 = 2500$
$\underline{W}_{A1}^\Delta$	$30 \times 0 = 0$	$30 \times 5 = 150$
$\underline{W}_{A2}^\Delta$	$30 \times 0 = 0$	$30 \times 0 = 0$
$\underline{W}_{A3}^\Delta$	Não se aplica	$40 \times 10 = 400$
$\underline{W}_{A4}^\Delta$	$40 \times 0 = 0$	Não se aplica
$\underline{W}_{A5}^\Delta$	$40 \times 0 = 0$	$40 \times 5 = 200$
$\underline{W}_{A6}^\Delta$	$40 \times 0 = 0$	$40 \times 0 = 0$
$\sum_{j=1, j \neq i}^n \underline{W}_j^\Delta$	0	750
$A_i^\Delta$	$2500 - 0 = 2500$	$2500 - 750 = 1750$
$\bar{p}_i^\Delta$	$\frac{2500}{40} = 62.5$	$\frac{1750}{40} = 43.8$
$P_{min_i}$	25	25
$\bar{p}_i^\Delta \geq P_{min_i}$	OK	OK

Tabela 4.3: cálculos para verificar a consistência local sobre as datas de início ao mais cedo das atividades A1, A2, A3 e A4 do *Analista de Service Desk*

$i \rightarrow$	A5	A6
$\Delta$	[67,92]	[80,105]
$W^\Delta$	$(92 - 67) \times 100 = 2500$	$(105 - 80) \times 100 = 2500$
$\underline{W}_{A1}^\Delta$	$30 \times 5 = 150$	$30 \times 2 = 60$
$\underline{W}_{A2}^\Delta$	$30 \times 0 = 0$	$30 \times 7 = 210$
$\underline{W}_{A3}^\Delta$	$40 \times 10 = 400$	$40 \times 7 = 280$
$\underline{W}_{A4}^\Delta$	$40 \times 5 = 200$	$40 \times 12 = 480$
$\underline{W}_{A5}^\Delta$	Não se aplica	$40 \times 12 = 480$
$\underline{W}_{A6}^\Delta$	$40 \times 0 = 0$	Não se aplica
$\sum_{j=1, j \neq i}^n \underline{W}_j^\Delta$	750	1510
$A_i^\Delta$	$2500 - 750 = 1750$	$2500 - 1510 = 990$
$\bar{p}_i^\Delta$	$\frac{1750}{40} = 43.8$	$\frac{990}{40} = 24.75$
$p_{min_i}$	25	25
$\bar{p}_i^\Delta \geq p_{min_i}$	OK	Inconsistência Local

Tabela 4.4: cálculos para verificar a consistência local sobre as datas de início ao mais cedo das atividades A5 e A6 do *Analista de Service Desk*

Como o critério de escolha dos deltas, adotado neste trabalho, para o estudo da consistência local, cobre exatamente a duração mínima de cada uma das atividades começando na data  $r_i$  (no caso do  $\Delta$  para a atualização da data de início ao mais cedo), é possível tentar um deslocamento à direita do  $\Delta$  onde foi diagnosticada a inconsistência local. O valor do deslocamento é calculado subtraindo-se o valor da duração máxima equivalente ( $\bar{p}_i^\Delta$ ) do valor da duração mínima ( $p_{min_i}$ ) da atividade em questão. Dessa forma, para a atividade A6 tem-se:

- Deslocamento do  $\Delta = 25 - 24.75 = 0.25$
- Novo  $\Delta = [80.25, 105.25]$
- $W^\Delta = (105.25 - 80.25) \times 100 = 2500$
- $\underline{W}_{A1}^\Delta = 30 \times 1.75 = 52.5$
- $\underline{W}_{A2}^\Delta = 30 \times 6.75 = 202.5$
- $\underline{W}_{A3}^\Delta = 40 \times 6.75 = 270$
- $\underline{W}_{A4}^\Delta = 40 \times 11.75 = 470$
- $\underline{W}_{A5}^\Delta = 40 \times 11.75 = 470$
- $\sum_{j=1, j \neq i}^n \underline{W}_j^\Delta = 1465$
- $A_i^\Delta = 2500 - 1465 = 1035$
- $\bar{p}_i^\Delta = \frac{1035}{40} = 25.9$
- $p_{min_i} = 25$

Sendo a  $\bar{p}_i^\Delta = 25.9$  maior que a duração mínima  $p_{min_i} = 25$  da atividade A6, então, a inconsistência local deixa de existir e a data de início ao mais cedo desta atividade passa

a ser igual a 80.25 (valor de  $S_{\Delta}$  do novo  $\Delta$ ). De fato, começando a execução da atividade  $A6$  nesta data, é possível que ela seja executada sem interrupção por parte do recurso envolvido (analista de *service desk*). A figura 4.10 ilustra o deslocamento à direita da data de início ao mais cedo da atividade  $A6$  do exemplo considerado.

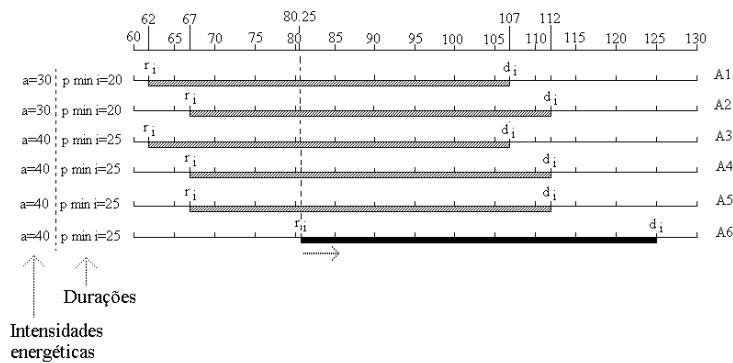
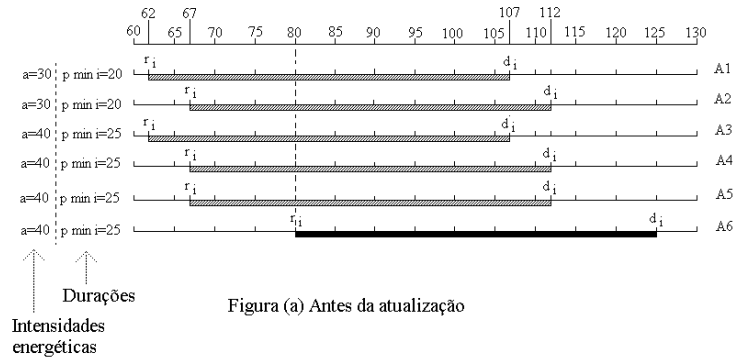


Figura 4.10: Atualização da data de início ao mais cedo da atividade  $A6$  do *Analista de Service Desk*

O código fonte a seguir apresenta o programa em linguagem Visual Prolog do cálculo da verificação de inconsistência local da atividade  $A6$  do exemplo do *Analista de Service Desk*:

01. predicates

02. calc\_W\_delta(real, real, real, real) % definição do predicado para  
% o cálculo da energia fornecida por um recurso em um dado  
% intervalo delta.

03. minimo(real, real, real) % definição do predicado para  
% o cálculo da duração mínima de uma atividade dentro  
% do intervalo delta.

04. maximo(real, real, real) % definição do predicado para  
% o cálculo da interseção entre o intervalo de execução da  
% atividade e o intervalo delta.

05. `calc_Wi_delta_mimima(real, real, real, real, real, real, real)`  
% definição do predicado para o cálculo da energia mínima consumida  
% por uma atividade dentro de um intervalo delta.
06. `calc_somatorio_energia_minima(real, real, real, real, real, real, real)`  
% definição do predicado para o cálculo do somatório das energias  
% mínimas consumidas pelas atividades (diferentes da atividade em  
% questão) dentro de um intervalo delta.
07. `calc_energia_maxima_disponivel(real, real, real)` % definição do  
% predicado para o cálculo da energia máxima disponível no  
% intervalo delta (após a execução das demais atividades) para  
% execução da atividade em questão.
08. `calc_duracao_maxima_equivalente(real, real, real)` % definição do  
% predicado para o cálculo da duração máxima equivalente.
09. `verifica_consistencia_local(real, real)` % definição do predicado  
% para a verificação de consistência local.
  
10. `clauses`
  
11. `minimo(X, Min, Max):-` % início da regra para calcular a duração  
% mínima de uma atividade dentro de um intervalo delta (quando o  
% valor da variável Min for menor que o valor da variável Max).
12. `Min < Max,`
13. `X = Min,!.` % final da regra para calcular a duração mínima de  
% uma atividade dentro de um intervalo delta (quando o valor  
% da variável Min for menor que o valor da variável Max).
  
14. `minimo(X, Min,Max):-` % início da regra para calcular a duração  
% mínima de uma atividade dentro de um intervalo delta (quando o  
% valor da variável Min for maior que o valor da variável Max).
15. `Min > Max,!,`
16. `X = Max,!.` % final da regra para calcular a duração mínima de  
% uma atividade dentro de um intervalo delta (quando o valor  
% da variável Min for maior que o valor da variável Max).
  
17. `minimo(__, __, __).` % regra para calcular a duração mínima de uma  
% atividade dentro do um intervalo delta (quando o valor da  
% variável Min for igual ao valor da variável Max).
  
18. `maximo(Y, Min,Max):-` % início da regra para calcular a interseção

```
% entre o intervalo de execução da atividade e o intervalo delta
% (quando o valor da variável Min for menor que o valor da variável
% Max).
19. Min < Max,
20. Y = Max,! . % final da regra para calcular a interseção entre o
% intervalo de execução da atividade e o intervalo delta (quando o
% valor da variável Min for menor que o valor da variável Max).

21. maximo(Y, Min,Max):- % início da regra para calcular a interseção
% entre o intervalo de execução da atividade e o intervalo delta
% (quando o valor da variável Min for maior que o valor da variável
% Max).
22. Min > Max,! ,
23. Y = Min,! . % final da regra para calcular a interseção entre o
% intervalo de execução da atividade e o intervalo delta (quando o
% valor da variável Min for maior que o valor da variável Max).

24. maximo(__,__,__). % regra para calcular a interseção entre o
% intervalo de execução da atividade e o intervalo delta (quando
% o valor da variável Min for igual ao valor da variável Max).

25. calc_W_delta(W_delta, A, D2, D1):- % início da regra para
% calcular a energia fornecida por um recurso em um dado
% intervalo delta.
26. W_delta = A*(D2 - D1),! . % final da regra para calcular a
% energia fornecida por um recurso em um dado intervalo
% delta.

27. calc_Wi_delta_minima(Wi_delta_minima, Ai, P, D2, D1, T1, T3):-
% início da regra para calcular a energia mínima consumida
% por uma atividade dentro de um intervalo delta.
28. Parcela_1 = D2 - D1,
29. Parcela_2 = T1 + P - D1,
30. Parcela_3 = D2 - T3 + P,
31. minimo(A, P, Parcela_1),
32. minimo(B, A, Parcela_2),
33. minimo(C, B, Parcela_3),
34. maximo(Result_parcial, 0, C),
35. Wi_delta_minima = Ai * Result_parcial,! . % final da regra
```

```
% para calcular a energia mínima consumida por uma
% atividade dentro de um intervalo delta.

36. calc_somatorio_energia_minima(Somatorio, A_1, A_2, A_3, A_4, A_5):-
    % início da regra para calcular o somatório das energias mínimas
    % consumidas pelas atividades (diferentes da atividade em questão)
    % dentro de um intervalo delta.
37. Somatorio = A_1 + A_2 + A_3 + A_4 + A_5,! % final da regra
    % para calcular o somatório das energias mínimas consumidas
    % pelas atividades (diferentes da atividade em questão) dentro
    % de um intervalo delta.

38. calc_energia_maxima_disponivel(A_delta_i, W_delta, Somatorio):-
    % início da regra para calcular a energia máxima disponível no
    % intervalo delta (após a execução das demais atividades) para
    % execução da atividade em questão.
39. A_delta_i = W_delta - Somatorio,! % final da regra para
    % calcular a energia máxima disponível no intervalo delta
    % (após a execução das demais atividades) para execução da
    % atividade em questão.

40. calc_duracao_maxima_equivalente(Pe, A_delta_i, Ai):- % início
    % da regra para calcular a duração máxima equivalente de uma
    % atividade.
41. Pe = A_delta_i / Ai,! % final da regra para calcular a duração
    % máxima equivalente de uma atividade.

42. verifica_consistencia_local(Duracao, Duracao_maxima_equivalente):-
    % início da regra para verificar a consistência local sobre a
    % a data de início de uma atividade (quando a duração mínima
    % da atividade for maior que a sua duração máxima equivalente).
43. Duracao > Duracao_maxima_equivalente,
44. write("Há inconsistência local, tente deslocamento do delta à direita"),!
    % final da regra para verificar a consistência local sobre a
    % a data de início de uma atividade (quando a duração mínima
    % da atividade for maior que a sua duração máxima equivalente).

45. verifica_consistencia_local(Duracao, Duracao_maxima_equivalente):-
    % início da regra para verificar a consistência local sobre a
```



```
% a data de início de uma atividade (quando a duração mínima
% da atividade for menor que a sua duração máxima equivalente).
46. Duracao < Duracao_maxima_equivalente,
47. write("Consistência local, a atividade pode iniciar ao mais cedo na data
48. original"),!.
% final da regra para verificar a consistência local sobre a
% a data de início de uma atividade (quando a duração mínima
% da atividade for menor que a sua duração máxima equivalente).

49. verifica_consistencia_local(Duracao, Duracao_maxima_equivalente):-
% início da regra para verificar a consistência local sobre a
% a data de início de uma atividade (quando a duração mínima
% da atividade for igual a sua duração máxima equivalente).
50. Duracao = Duracao_maxima_equivalente,
51. write("Consistência local, a atividade pode iniciar ao mais cedo na data
52. original"),!. % final da regra para verificar a consistência local
% sobre a data de início de uma atividade (quando a duração mínima
% da atividade for igual a sua duração máxima equivalente).

53. goal

54. calc_W_delta(W,100,105,80), % chamada para calcular a energia
% disponível no intervalo delta para execução das atividades.

55. calc_Wi_delta_minima(A1,30,20,105,80,62,107), % chamada para
% calcular a energia mínima necessária para execução da
% atidade A1 dentro do intervalo delta.

56. calc_Wi_delta_minima(A2,30,20,105,80,67,112), % chamada para
% calcular a energia mínima necessária para execução da
% atidade A2 dentro do intervalo delta.

57. calc_Wi_delta_minima(A3,40,25,105,80,62,107), % chamada para
% calcular a energia mínima necessária para execução da
% atidade A3 dentro do intervalo delta.

58. calc_Wi_delta_minima(A4,40,25,105,80,67,112), % chamada para
% calcular a energia mínima necessária para execução da
% atidade A4 dentro do intervalo delta.
```

```
59. calc_Wi_delta_minima(A5,40,25,105,80,67,112), % chamada para
    % calcular a energia mínima necessária para execução da
    % atividade A5 dentro do intervalo delta.

60. calc_somatorio_energia_minima(Somatorio_energias_minimas, A1, A2, A3,
61. A4, A5), % chamada para calcular o somatório das energias
    % mínimas necessárias para execução das atividades A1, A2,
    % A3, A4, A5 dentro do intervalo delta.

62. calc_energia_maxima_disponivel(Energia_maxima_disponivel, W,
63. Somatorio_energias_minimas), % chamada para calcular a
    % energia máxima disponível no intervalo delta (após
    % a execução das demais atividades) para execução da
    % atividade em questão.

64. calc_duracao_maxima_equivalente(Duracao_maxima_equivalente,
65. Energia_maxima_disponivel, 40), % chamada para calcular a
    % a duração máxima equivalente de uma atividade.

66. verifica_consistencia_local(25, Duracao_maxima_equivalente).
    % chamada para verificar a consistência local sobre a data de
    % início de uma atividade
```

Para a verificação de inconsistência local das demais atividades, trocam-se os valores  $D2$  e  $D1$  dos procedimentos *calc\_Wi\_delta\_minima* (linhas 55 até 59) e *calc\_W\_delta* (linha 54) pelo valor do  $\Delta$  correspondente à atividade em questão. Trocam-se também: o valor  $A_i$  do procedimento *calc\_duracao\_maxima\_equivalente* (linhas 64 e 65) pelo valor da intensidade energética correspondente da atividade em questão; e o valor *Duração* do procedimento *verifica\_consistencia\_local* (linha 66) pelo valor da duração mínima da atividade em questão.

A tabela 4.5 apresenta os cálculos da verificação de consistência local (levando em consideração o critério de escolha dos deltas) sobre a data de início ao mais tarde ( $\bar{s}_i$ ) das atividades A1, A2, A3 e A4 do *Analista de Service Desk*. A tabela 4.6 apresenta os cálculos da verificação de consistência local (levando em consideração o critério de escolha

dos deltas) sobre a data de início ao mais tarde ( $\bar{s}_i$ ) das atividades A5 e A6 do *Analista de Service Desk*.

$i \rightarrow$	A1	A2
$\Delta$	[77,97]	[82,102]
$W^\Delta$	$(97 - 77) \times 100 = 2000$	$(102 - 82) \times 100 = 2000$
$\underline{W}_{A1}^\Delta$	Não se aplica	$30 \times 0 = 0$
$\underline{W}_{A2}^\Delta$	$30 \times 5 = 150$	Não se aplica
$\underline{W}_{A3}^\Delta$	$40 \times 10 = 400$	$40 \times 5 = 200$
$\underline{W}_{A4}^\Delta$	$40 \times 10 = 400$	$40 \times 10 = 400$
$\underline{W}_{A5}^\Delta$	$40 \times 10 = 400$	$40 \times 10 = 400$
$\underline{W}_{A6}^\Delta$	$40 \times 0 = 0$	$40 \times 2 = 80$
$\sum_{j=1, j \neq i}^n \underline{W}_j^\Delta$	1350	1080
$A_i^\Delta$	$2000 - 1350 = 650$	$2000 - 1080 = 920$
$\bar{p}_i^\Delta$	$\frac{650}{30} = 21.6$	$\frac{920}{30} = 30.7$
$p_{min_i}$	20	20
$\bar{p}_i^\Delta \geq p_{min_i}$	OK	OK

$i \rightarrow$	A3	A4
$\Delta$	[72,97]	[77,102]
$W^\Delta$	$(97 - 72) \times 100 = 2500$	$(102 - 77) \times 100 = 2500$
$\underline{W}_{A1}^\Delta$	$30 \times 10 = 300$	$30 \times 5 = 150$
$\underline{W}_{A2}^\Delta$	$30 \times 5 = 150$	$30 \times 10 = 300$
$\underline{W}_{A3}^\Delta$	Não se aplica	$40 \times 10 = 400$
$\underline{W}_{A4}^\Delta$	$40 \times 10 = 400$	Não se aplica
$\underline{W}_{A5}^\Delta$	$40 \times 10 = 400$	$40 \times 15 = 600$
$\underline{W}_{A6}^\Delta$	$40 \times 0 = 0$	$40 \times 2 = 80$
$\sum_{j=1, j \neq i}^n \underline{W}_j^\Delta$	1250	1530
$A_i^\Delta$	$2500 - 1250 = 1250$	$2500 - 1530 = 970$
$\bar{p}_i^\Delta$	$\frac{1250}{40} = 31.25$	$\frac{970}{40} = 24.25$
$p_{min_i}$	25	25
$\bar{p}_i^\Delta \geq p_{min_i}$	OK	Inconsistência Local

Tabela 4.5: cálculos para verificar a consistência local sobre as datas de início ao mais tarde das atividades A1, A2, A3 e A4 do *Analista de Service Desk*

$i \rightarrow$	A5	A6
$\Delta$	[77,102]	[90,115]
$W^\Delta$	$(102 - 77) \times 100 = 2500$	$(115 - 90) \times 100 = 2500$
$\underline{W}_{A1}^\Delta$	$30 \times 5 = 150$	$30 \times 0 = 0$
$\underline{W}_{A2}^\Delta$	$30 \times 10 = 300$	$30 \times 0 = 0$
$\underline{W}_{A3}^\Delta$	$40 \times 10 = 400$	$40 \times 0 = 0$
$\underline{W}_{A4}^\Delta$	$40 \times 15 = 600$	$40 \times 2 = 80$
$\underline{W}_{A5}^\Delta$	Não se aplica	$40 \times 2 = 80$
$\underline{W}_{A6}^\Delta$	$40 \times 2 = 80$	Não se aplica
$\sum_{j=1, j \neq i}^n \underline{W}_j^\Delta$	1530	160
$A_i^\Delta$	$2500 - 1530 = 970$	$2500 - 160 = 2340$
$\bar{p}_i^\Delta$	$\frac{970}{40} = 24.25$	$\frac{2340}{40} = 58.5$
$p_{min_i}$	25	25
$\bar{p}_i^\Delta \geq p_{min_i}$	Inconsistência Local	OK

Tabela 4.6: cálculos para verificar a consistência local sobre as datas de início ao mais tarde das atividades A5 e A6 do *Analista de Service Desk*

Para as atividades  $A4$  e  $A5$ , pode-se observar que suas durações equivalentes  $\bar{p}_i^\Delta$  (24.25) são menores do que suas durações mínimas  $p_{min_i}=(25)$ ; logo existe uma inconsistência local sobre essas atividades.

Como o critério de escolha dos deltas para o estudo da consistência local adotado neste trabalho cobre exatamente a duração mínima de cada uma das atividades, começando na data  $\bar{s}_i$  (no caso do  $\Delta$  verificar a consistência local da data de início ao mais tarde), é possível tentar um deslocamento dos deltas à esquerda onde foram diagnosticadas condições de inconsistência local. Da mesma forma que no deslocamento à direita, o valor do deslocamento do  $\Delta$  à esquerda é calculado subtraindo-se o valor da duração mínima da atividade ( $p_{min_i}$ ) do valor da sua duração máxima equivalente sobre o  $\Delta$  ( $\bar{p}_i^\Delta$ ).

Dessa forma, para as atividades  $A4$  e  $A5$  tem-se:

- Para a atividade  $A4$ :
  - Deslocamento do  $\Delta = 25 - 24.25 = 0.75$
  - Novo  $\Delta=[76.25, 101.25]$
  - $W^\Delta=(101.25 - 76.25) \times 100 = 2500$
  - $\underline{W}_{A1}^\Delta=30 \times 5.75 = 172.5$
  - $\underline{W}_{A2}^\Delta=30 \times 9.25 = 277.5$
  - $\underline{W}_{A3}^\Delta=40 \times 10.75 = 430$
  - $\underline{W}_{A5}^\Delta=40 \times 14.25 = 570$
  - $\underline{W}_{A6}^\Delta=40 \times 1.25 = 50$
  - $\sum_{j=1, j \neq i}^n \underline{W}_j^\Delta=1500$
  - $A_i^\Delta=2500 - 1500 = 1000$

- $\bar{p}_i^\Delta = \frac{1000}{40} = 25$
- $p_{min_i} = 25$

• Para a atividade A5:

- Deslocamento do  $\Delta = 25 - 24.25 = 0.75$
- Novo  $\Delta = [76.25, 101.25]$
- $W^\Delta = (101.25 - 76.25) \times 100 = 2500$
- $W_{A1}^\Delta = 30 \times 5.75 = 172.5$
- $W_{A2}^\Delta = 30 \times 9.25 = 277.5$
- $W_{A3}^\Delta = 40 \times 10.75 = 430$
- $W_{A4}^\Delta = 40 \times 14.25 = 570$
- $W_{A6}^\Delta = 40 \times 1.25 = 50$
- $\sum_{j=1, j \neq i}^n W_j^\Delta = 1500$
- $A_i^\Delta = 2500 - 1500 = 1000$
- $\bar{p}_i^\Delta = \frac{1000}{40} = 25$
- $p_{min_i} = 25$

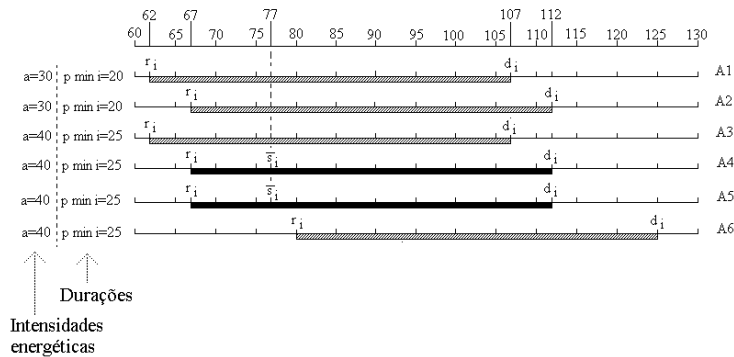


Figura (a) Antes da atualização

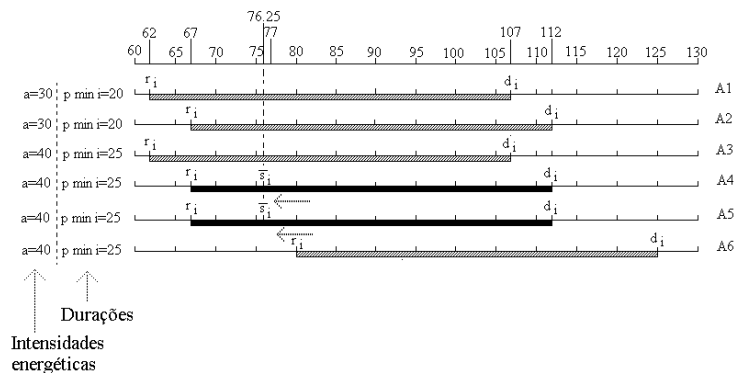


Figura (b) Depois da atualização

Figura 4.11: Atualização da data de início ao mais tarde das atividades A4 e A5 do Analista de Service Desk

Como a duração máxima equivalente sobre o  $\Delta$  ( $\bar{p}_i^\Delta=25$ ) de ambas atividades é igual às suas respectivas durações mínimas ( $p_{min_i}=25$ ), a inconsistência local deixa de existir e as datas de início ao mais tarde de ambas passam a ser igual a 76.25 (valor de  $S_\Delta$  do último  $\Delta$ ).

De fato, começando a execução das atividades  $A_4$  e  $A_5$  nessa data, é possível que elas sejam executadas sem interrupção por parte do recurso envolvido (Analista de *service desk*).

A figura 4.11 ilustra o deslocamento à esquerda das datas de início ao mais tarde das atividades  $A_4$  e  $A_5$  do exemplo considerado.

Para os recursos do tipo discreto, o Raciocínio Energético pode ser aplicado da mesma forma que nos recursos do tipo contínuo. De fato, um recurso discreto é um caso particular de recurso contínuo em que qualquer atividade precisa de 100% da intensidade energética do recurso.



# Capítulo 5

## Estudo de Caso

### 5.1 Apresentação

A abordagem proposta nesta dissertação será aplicada ao processo do *Serviço de Reclamações* apresentado primeiramente por Aalst [Aalst & Hee, 2002]. A vantagem desse processo é que ele mostra os principais roteiros existentes em sistemas de gerenciamento de *workflow*: o roteiro sequencial, o roteiro paralelo e o roteiro alternativo.

Nesse processo, primeiro uma reclamação é registrada. Depois, o cliente que fez a reclamação e o departamento afetado pela reclamação são contactados. O cliente é contactado a fim de se obter mais informações. O departamento é informado sobre a reclamação e é questionado sobre uma reação inicial. Essas duas atividades devem ser executadas em paralelo. Depois disso, as informações são coletadas e uma decisão é tomada. Dependendo da decisão, ou um pagamento é feito, ou uma carta é enviada ao cliente. Finalmente, a reclamação é arquivada.

### 5.2 Modelagem baseada em redes de Petri p-temporais

#### 5.2.1 Definição de Roteiros

O modelo de rede de Petri para o processo de *Serviço de Reclamações* é mostrado na figura 5.1. As atividades do processo são representadas pelos lugares  $A_i$  (para  $i=1$  até 8) e os tempos de espera são representados pelos lugares  $E_j$  (para  $j=0$  até 10).



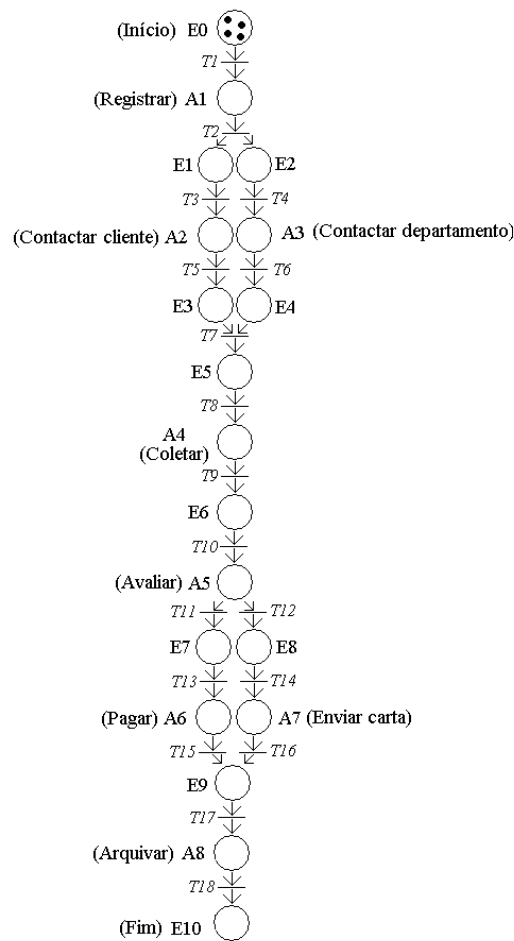


Figura 5.1: Modelo de rede de Petri para o *Serviço de Reclamações*

### 5.2.2 Alocação de Recursos

O próximo passo é a alocação dos recursos utilizados para tratar as atividades. Os mecanismos de alocação de recursos para o processo de *Serviço de Reclamações* estão representados na figura 5.2. A *Secretária*, o *Analista reclamações*, o *Funcionário finanças* e o *Sistema* são representados como recursos discretos. Estes recursos tratarão as atividades em uma ordem puramente sequencial. Por exemplo, a *Secretária*, representada por uma ficha em R1, pode somente registrar os dados de um cliente de uma maneira não preemptiva (sem interrupção). Ao contrário, o funcionário do Departamento de Reclamações é representado pelo número 100 em R2. Esse recurso contínuo pode ser usado para tratar as atividades: *contactar cliente*, *contactar departamento* e *enviar carta*. A atividade *contactar cliente* irá precisar de 30% da disponibilidade desse funcionário, a atividade *contactar departamento* irá precisar de 40% e, finalmente, a atividade *enviar carta* irá precisar de 50% de sua disponibilidade. Isso significa que, de certa maneira, o funcionário representado no lugar R2, poderá processar mais de uma atividade em um dado espaço de tempo. Por exemplo, ele poderá escrever uma carta (50% da sua disponibilidade para

a atividade *enviar carta*) e, ao mesmo tempo, tentar ligar para um departamento (40% da sua disponibilidade para a atividade *contactar departamento*).

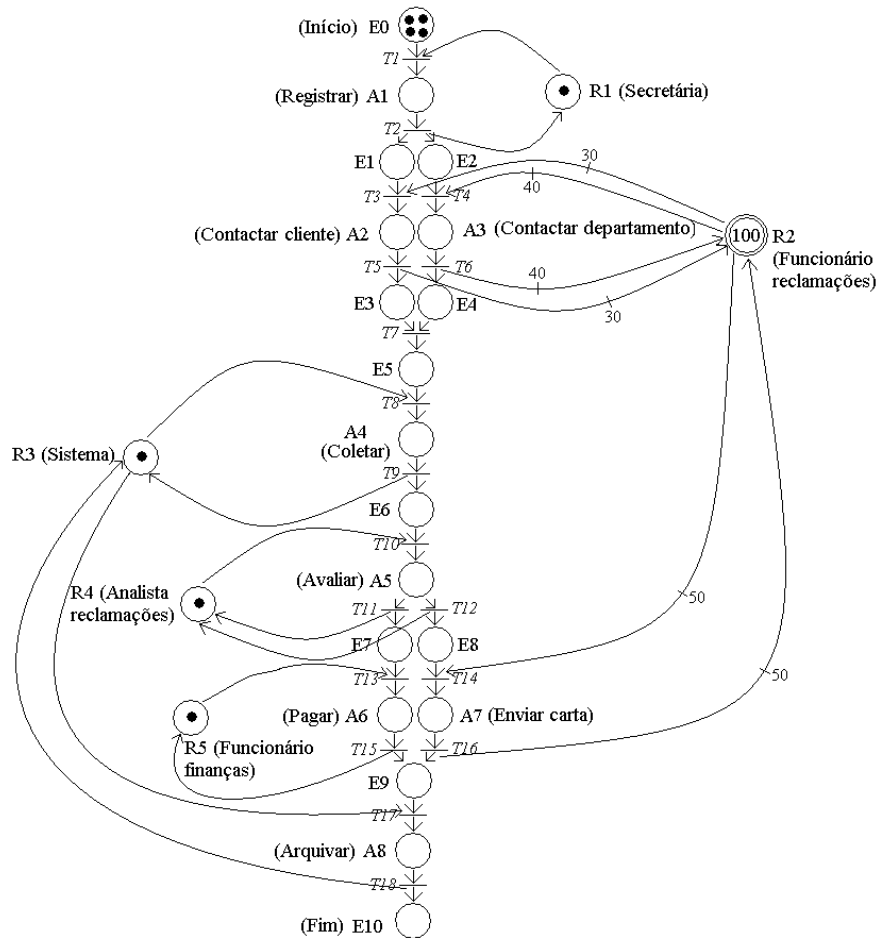


Figura 5.2: Modelo de rede de Petri com alocação de recursos híbridos para o *Serviço de Reclamações*

### 5.2.3 Modelo Temporal

A fim de representar as explícitas restrições de tempo existentes, um intervalo de tempo pode ser atribuído a cada atividade, representando um tempo mínimo e um tempo máximo para a duração das atividades.

O modelo p-temporal para o processo do *Serviço de Reclamações* é mostrado na figura 5.3. As durações foram estipuladas apenas a título de exemplificação. As durações das atividades *Coletar Informações* A4 e *Arquivar* A8 são iguais a zero porque suas durações são desprezíveis se comparada com as outras atividades do *Serviço de reclamações*.

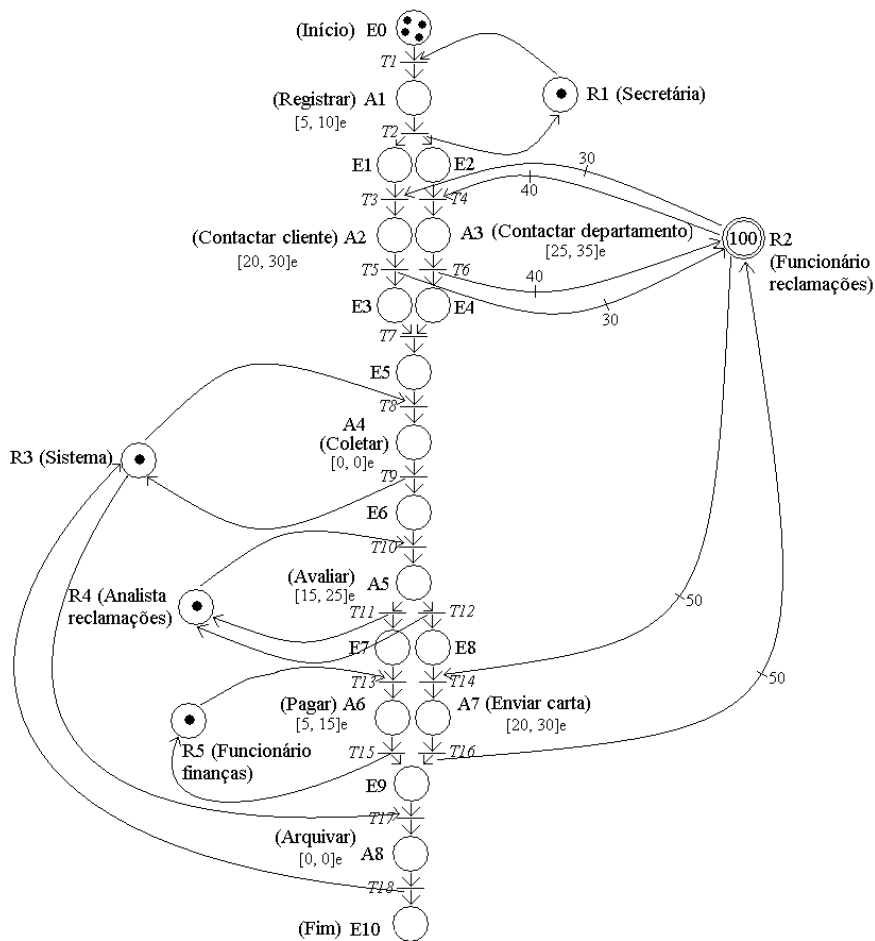


Figura 5.3: Modelo de rede de Petri p-temporal com alocação de recursos híbridos para o *Serviço de Reclamações*

### 5.3 Mecanismo de propagação de restrições

Considerando quatro casos a serem tratados pelo processo do *Serviço de reclamações*, é possível aplicar as técnicas do mecanismo de propagação de restrições nos roteiros. Para cada caso, foi considerado que a duração máxima permitida era de 105 unidades de tempo.

Na rede de Petri da figura 5.4, são considerados os quatro casos, representados por 4 fichas no lugar E0. O primeiro caso pode ser iniciado na data 0 que é representada como a borda mínima do intervalo de visibilidade V1 em E0. O segundo, o terceiro e o quarto casos podem ser iniciados, respectivamente, nas datas 13, 18 e 23 que são representadas como as bordas mínimas dos intervalos de visibilidades V2, V3 e V4, respectivamente, em E0. Como a duração máxima permitida para cada caso é de 105 unidades de tempo, a borda máxima do intervalo de visibilidade associada ao último lugar (E10) do processo será igual a 0 (data de início ao mais cedo do caso 1) + 105 (máxima duração de um caso) = 105 (máxima data para a entrega do caso 1) para o primeiro caso. Para os outros três casos, as demais datas máximas de entrega foram calculadas da mesma forma, ou seja, levando-se em consideração a data de início ao mais cedo e a máxima duração de cada

caso.

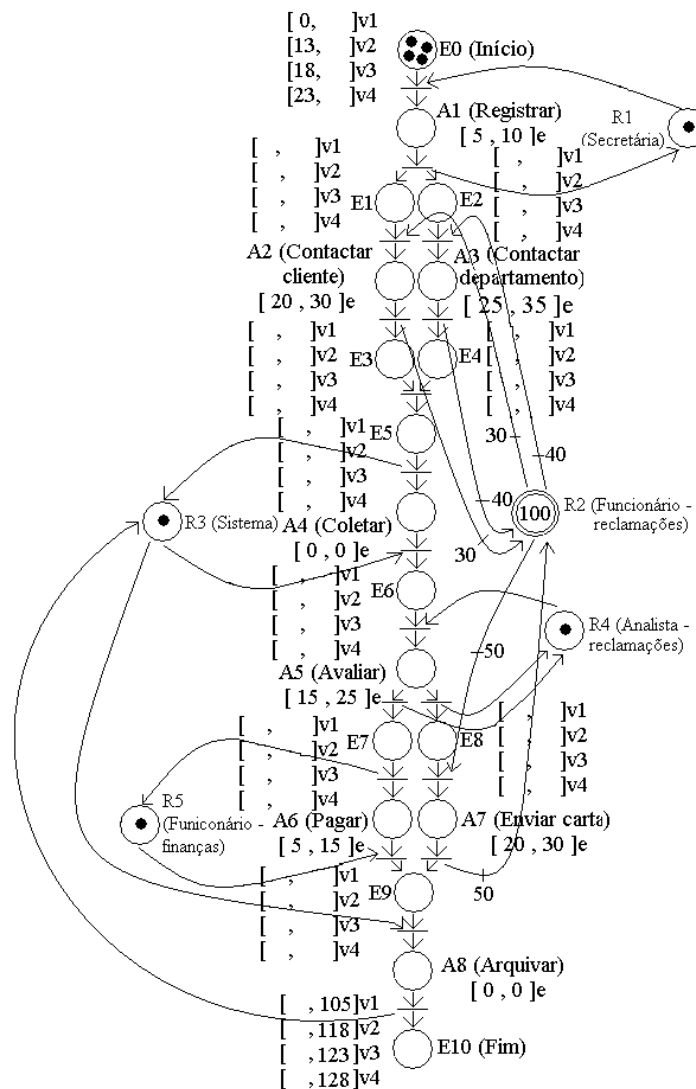


Figura 5.4: Rede com as datas mínimas e máximas para o início e término dos casos

As bordas dos outros intervalos de visibilidade associadas aos casos nos lugares de espera foram calculadas aplicando-se as técnicas de propagação de restrições apresentadas no capítulo 4 e o resultado é dado pela figura 5.5.

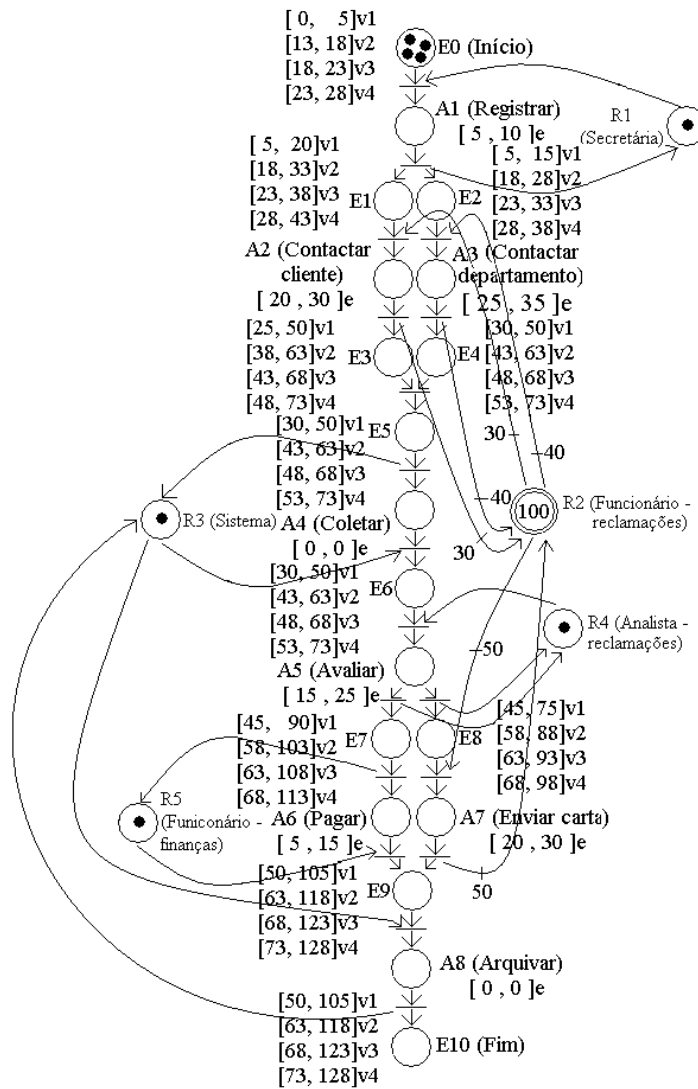


Figura 5.5: Propagação das restrições: *Serviço de Reclamações*

## 5.4 Abordagem baseada no Raciocínio Energético

A aplicação do Raciocínio Energético sobre o processo *Serviço de Reclamações* inicia-se com a verificação de consistência global sobre todas as atividades relacionadas com cada um dos seus respectivos recursos. Não sendo diagnosticada nenhuma ocorrência de inconsistência global é possível passar para o próximo passo que é a verificação de consistência local, na qual será verificada a possibilidade de atualização das datas de início ao mais cedo e ao mais tarde de cada uma das atividades. Como explanado no capítulo 4, tais atualizações ocorrem quando a duração máxima equivalente de uma atividade sobre o  $\Delta$  for menor do que a duração mínima dessa mesma atividade.

Após aplicado o mecanismo de alocação de recursos híbridos e o mecanismo de propagação das restrições nos roteiros do processo considerado, é possível construir uma tabela para cada um dos recursos envolvidos com a execução das atividades. Cada uma dessas

tabelas fornecem algumas variáveis de restrições do processo relacionadas com o recurso em questão. Conforme o que foi apresentado no capítulo 4, algumas dessas variáveis de restrições são:

- $r_i$ : início ao mais cedo de uma atividade;
- $\bar{s}_i$ : início ao mais tarde de uma atividade;
- $p_{min_i}$ : duração mínima de uma atividade;
- $p_{max_i}$ : duração máxima de uma atividade;
- $d_i$ : final ao mais tarde de uma atividade;
- $a_i$ : intensidade energética necessária para a completa execução de uma atividade.

As tabelas: 5.1, 5.2, 5.3, 5.4 e 5.5 apresentam as variáveis de restrições de cada uma das atividades relacionadas a cada um dos recursos do processo *Serviço de Reclamações*.

$i$	$r_i$	$\bar{s}_i$	$p_{min_i}$	$p_{max_i}$	$d_i$	$a_i$
A1 - Caso 1	0	5	5	10	20	100
A1 - Caso 2	13	18	5	10	33	100
A1 - Caso 3	18	23	5	10	38	100
A1 - Caso 4	23	28	5	10	43	100

Tabela 5.1: Variáveis (restrições) das atividades relacionadas ao recurso R1 (Secretária)

$i$	$r_i$	$\bar{s}_i$	$p_{min_i}$	$p_{max_i}$	$d_i$	$a_i$
A2 - Caso 1	5	20	20	30	50	30
A2 - Caso 2	18	33	20	30	63	30
A2 - Caso 3	23	38	20	30	68	30
A2 - Caso 4	28	43	20	30	73	30
A3 - Caso 1	5	15	25	35	50	40
A3 - Caso 2	18	28	25	35	63	40
A3 - Caso 3	23	33	25	35	68	40
A3 - Caso 4	28	38	25	35	73	40
A7 - Caso 1	45	75	20	30	105	50
A7 - Caso 2	58	88	20	30	118	50
A7 - Caso 3	63	93	20	30	123	50
A7 - Caso 4	68	98	20	30	128	50

Tabela 5.2: Variáveis (restrições) das atividades relacionadas ao recurso R2 (Funcionário Reclamações)

$i$	$r_i$	$\bar{s}_i$	$p_{min_i}$	$p_{max_i}$	$d_i$	$a_i$
A4 - Caso 1	30	50	0	0	50	100
A4 - Caso 2	43	63	0	0	63	100
A4 - Caso 3	48	68	0	0	68	100
A4 - Caso 4	53	73	0	0	73	100
A8 - Caso 1	50	105	0	0	105	100
A8 - Caso 2	63	118	0	0	118	100
A8 - Caso 3	68	123	0	0	123	100
A8 - Caso 4	73	128	0	0	128	100

Tabela 5.3: Variáveis (restrições) das atividades relacionadas ao recurso R3 (Sistema)

$i$	$r_i$	$\bar{s}_i$	$p_{min_i}$	$p_{max_i}$	$d_i$	$a_i$
A5 - Caso 1	30	50	15	25	90	100
A5 - Caso 2	43	63	15	25	103	100
A5 - Caso 3	48	68	15	25	108	100
A5 - Caso 4	53	73	15	25	113	100

Tabela 5.4: Variáveis (restrições) das atividades relacionadas ao recurso R4 (Analista Reclamações)

$i$	$r_i$	$\bar{s}_i$	$p_{min_i}$	$p_{max_i}$	$d_i$	$a_i$
A6 - Caso 1	45	90	5	15	105	100
A6 - Caso 2	58	103	5	15	118	100
A6 - Caso 3	63	108	5	15	123	100
A6 - Caso 4	68	113	5	15	128	100

Tabela 5.5: Variáveis (restrições) das atividades relacionadas ao recurso R5 (Funcionário Finanças)

Para os recursos discretos (R1, R3, R4 e R5) considerou-se que suas atividades necessitavam de 100% da intensidade energética de cada um desses recursos. Tal consideração pode ser feita em função da natureza desses recursos.

A verificação de consistência global <sup>1</sup> é feita da forma como foi apresentada no capítulo 4 no exemplo do *Analista de Service Desk*.

Na tabela 5.6 são apresentados os resultados dos cálculos considerados na verificação de consistência global tendo em conta as atividades envolvidas com o recurso R1 (Secretária).

<sup>1</sup>As verificações de consistência global e local considera o critério de escolha dos deltas apresentado no capítulo 4

$i \rightarrow$	A1 - Caso 1	A1 - Caso 2
$\Delta$	[0,20]	[13,33]
$W^\Delta$	$(20 - 0) \times 100 = 2000$	$(33 - 13) \times 100 = 2000$
$\underline{W}_{A1-Caso1}^\Delta$	$100 \times 5 = 500$	$100 \times 0 = 0$
$\underline{W}_{A1-Caso2}^\Delta$	$100 \times 0 = 0$	$100 \times 5 = 500$
$\underline{W}_{A1-Caso3}^\Delta$	$100 \times 0 = 0$	$100 \times 0 = 0$
$\underline{W}_{A1-Caso4}^\Delta$	$100 \times 0 = 0$	$100 \times 0 = 0$
$\sum_{i=1}^4 \underline{W}_i^\Delta$	500	500
$\sum_{i=1}^4 \underline{W}_i^\Delta \leq W^\Delta$	OK	OK

$i \rightarrow$	A1 - Caso 3	A1 - Caso 4
$\Delta$	[18,38]	[43,23]
$W^\Delta$	$(38 - 18) \times 100 = 2000$	$(43 - 23) \times 100 = 2000$
$\underline{W}_{A1-Caso1}^\Delta$	$100 \times 0 = 0$	$100 \times 0 = 0$
$\underline{W}_{A1-Caso2}^\Delta$	$100 \times 0 = 0$	$100 \times 0 = 0$
$\underline{W}_{A1-Caso3}^\Delta$	$100 \times 5 = 500$	$100 \times 0 = 0$
$\underline{W}_{A1-Caso4}^\Delta$	$100 \times 0 = 0$	$100 \times 5 = 500$
$\sum_{i=1}^4 \underline{W}_i^\Delta$	500	500
$\sum_{i=1}^4 \underline{W}_i^\Delta \leq W^\Delta$	OK	OK

Tabela 5.6: cálculos para verificar a consistência global sobre as atividades do recurso R1 (Secretária)

Nas tabelas 5.7, 5.8 e 5.9 são apresentados os resultados dos cálculos na verificação de consistência global considerando as atividades envolvidas com o recurso R2 (Funcionário Reclamações).



$i \rightarrow$	A2 - Caso 1	A2 - Caso 2
$\Delta$	[5,50]	[18,63]
$W^\Delta$	$(50 - 5) \times 100 = 4500$	$(63 - 18) \times 100 = 4500$
$\frac{W^\Delta}{A2-Caso1}$	$30 \times 20 = 600$	$30 \times 7 = 210$
$\frac{W^\Delta}{A2-Caso2}$	$30 \times 7 = 210$	$30 \times 20 = 600$
$\frac{W^\Delta}{A2-Caso3}$	$30 \times 2 = 60$	$30 \times 15 = 450$
$\frac{W^\Delta}{A2-Caso4}$	$30 \times 0 = 0$	$30 \times 10 = 300$
$\frac{W^\Delta}{A3-Caso1}$	$40 \times 25 = 1000$	$40 \times 12 = 480$
$\frac{W^\Delta}{A3-Caso2}$	$40 \times 12 = 480$	$40 \times 25 = 1000$
$\frac{W^\Delta}{A3-Caso3}$	$40 \times 7 = 280$	$40 \times 20 = 800$
$\frac{W^\Delta}{A3-Caso4}$	$40 \times 2 = 80$	$40 \times 15 = 600$
$\frac{W^\Delta}{A7-Caso1}$	$50 \times 0 = 0$	$50 \times 0 = 0$
$\frac{W^\Delta}{A7-Caso2}$	$50 \times 0 = 0$	$50 \times 0 = 0$
$\frac{W^\Delta}{A7-Caso3}$	$50 \times 0 = 0$	$50 \times 0 = 0$
$\frac{W^\Delta}{A7-Caso4}$	$50 \times 0 = 0$	$50 \times 0 = 0$
$\sum_{i=1}^{12} W_i^\Delta$	2710	4440
$\sum_{i=1}^{12} W_i^\Delta \leq W^\Delta$	OK	OK

$i \rightarrow$	A2 - Caso 3	A2 - Caso 4
$\Delta$	[23,68]	[28,73]
$W^\Delta$	$(68 - 23) \times 100 = 4500$	$(73 - 28) \times 100 = 4500$
$\frac{W^\Delta}{A2-Caso1}$	$30 \times 2 = 60$	$30 \times 0 = 0$
$\frac{W^\Delta}{A2-Caso2}$	$30 \times 15 = 450$	$30 \times 10 = 300$
$\frac{W^\Delta}{A2-Caso3}$	$30 \times 20 = 600$	$30 \times 15 = 450$
$\frac{W^\Delta}{A2-Caso4}$	$30 \times 15 = 450$	$30 \times 20 = 600$
$\frac{W^\Delta}{A3-Caso1}$	$40 \times 7 = 280$	$40 \times 2 = 80$
$\frac{W^\Delta}{A3-Caso2}$	$40 \times 20 = 800$	$40 \times 15 = 600$
$\frac{W^\Delta}{A3-Caso3}$	$40 \times 25 = 1000$	$40 \times 20 = 800$
$\frac{W^\Delta}{A3-Caso4}$	$40 \times 20 = 800$	$40 \times 25 = 1000$
$\frac{W^\Delta}{A7-Caso1}$	$50 \times 0 = 0$	$50 \times 0 = 0$
$\frac{W^\Delta}{A7-Caso2}$	$50 \times 0 = 0$	$50 \times 0 = 0$
$\frac{W^\Delta}{A7-Caso3}$	$50 \times 0 = 0$	$50 \times 0 = 0$
$\frac{W^\Delta}{A7-Caso4}$	$50 \times 0 = 0$	$50 \times 0 = 0$
$\sum_{i=1}^{12} W_i^\Delta$	4440	3830
$\sum_{i=1}^{12} W_i^\Delta \leq W^\Delta$	OK	OK

Tabela 5.7: cálculos para verificar a consistência global sobre as atividades *A2 - Caso 1*, *A2 - Caso 2*, *A2 - Caso 3* e *A2 - Caso 4* do recurso R2 (Funcionário Reclamações)

$i \rightarrow$	A3 - Caso 1	A3 - Caso 2
$\Delta$	[5,50]	[18,63]
$W^\Delta$	$(50 - 5) \times 100 = 4500$	$(63 - 18) \times 100 = 4500$
$\frac{W_{A2-Caso1}^\Delta}{W_{A2-Caso1}^\Delta}$	$30 \times 20 = 600$	$30 \times 7 = 210$
$\frac{W_{A2-Caso2}^\Delta}{W_{A2-Caso2}^\Delta}$	$30 \times 7 = 210$	$30 \times 20 = 600$
$\frac{W_{A2-Caso3}^\Delta}{W_{A2-Caso3}^\Delta}$	$30 \times 2 = 60$	$30 \times 15 = 450$
$\frac{W_{A2-Caso4}^\Delta}{W_{A2-Caso4}^\Delta}$	$30 \times 0 = 0$	$30 \times 10 = 300$
$\frac{W_{A3-Caso1}^\Delta}{W_{A3-Caso1}^\Delta}$	$40 \times 25 = 1000$	$40 \times 12 = 480$
$\frac{W_{A3-Caso2}^\Delta}{W_{A3-Caso2}^\Delta}$	$40 \times 12 = 480$	$40 \times 25 = 1000$
$\frac{W_{A3-Caso3}^\Delta}{W_{A3-Caso3}^\Delta}$	$40 \times 7 = 280$	$40 \times 20 = 800$
$\frac{W_{A3-Caso4}^\Delta}{W_{A3-Caso4}^\Delta}$	$40 \times 2 = 80$	$40 \times 15 = 600$
$\frac{W_{A7-Caso1}^\Delta}{W_{A7-Caso1}^\Delta}$	$50 \times 0 = 0$	$50 \times 0 = 0$
$\frac{W_{A7-Caso2}^\Delta}{W_{A7-Caso2}^\Delta}$	$50 \times 0 = 0$	$50 \times 0 = 0$
$\frac{W_{A7-Caso3}^\Delta}{W_{A7-Caso3}^\Delta}$	$50 \times 0 = 0$	$50 \times 0 = 0$
$\frac{W_{A7-Caso4}^\Delta}{W_{A7-Caso4}^\Delta}$	$50 \times 0 = 0$	$50 \times 0 = 0$
$\sum_{i=1}^{12} W_i^\Delta$	2710	4440
$\sum_{i=1}^{12} \frac{W_i^\Delta}{W_i^\Delta} \leq W^\Delta$	OK	OK

$i \rightarrow$	A3 - Caso 3	A3 - Caso 4
$\Delta$	[23,68]	[28,73]
$W^\Delta$	$(68 - 23) \times 100 = 4500$	$(73 - 28) \times 100 = 4500$
$\frac{W_{A2-Caso1}^\Delta}{W_{A2-Caso1}^\Delta}$	$30 \times 2 = 60$	$30 \times 0 = 0$
$\frac{W_{A2-Caso2}^\Delta}{W_{A2-Caso2}^\Delta}$	$30 \times 15 = 450$	$30 \times 10 = 300$
$\frac{W_{A2-Caso3}^\Delta}{W_{A2-Caso3}^\Delta}$	$30 \times 20 = 600$	$30 \times 15 = 450$
$\frac{W_{A2-Caso4}^\Delta}{W_{A2-Caso4}^\Delta}$	$30 \times 15 = 450$	$30 \times 20 = 600$
$\frac{W_{A3-Caso1}^\Delta}{W_{A3-Caso1}^\Delta}$	$40 \times 7 = 280$	$40 \times 2 = 80$
$\frac{W_{A3-Caso2}^\Delta}{W_{A3-Caso2}^\Delta}$	$40 \times 20 = 800$	$40 \times 15 = 600$
$\frac{W_{A3-Caso3}^\Delta}{W_{A3-Caso3}^\Delta}$	$40 \times 25 = 1000$	$40 \times 20 = 800$
$\frac{W_{A3-Caso4}^\Delta}{W_{A3-Caso4}^\Delta}$	$40 \times 20 = 800$	$40 \times 25 = 1000$
$\frac{W_{A7-Caso1}^\Delta}{W_{A7-Caso1}^\Delta}$	$50 \times 0 = 0$	$50 \times 0 = 0$
$\frac{W_{A7-Caso2}^\Delta}{W_{A7-Caso2}^\Delta}$	$50 \times 0 = 0$	$50 \times 0 = 0$
$\frac{W_{A7-Caso3}^\Delta}{W_{A7-Caso3}^\Delta}$	$50 \times 0 = 0$	$50 \times 0 = 0$
$\frac{W_{A7-Caso4}^\Delta}{W_{A7-Caso4}^\Delta}$	$50 \times 0 = 0$	$50 \times 0 = 0$
$\sum_{i=1}^{12} W_i^\Delta$	4440	3830
$\sum_{i=1}^{12} \frac{W_i^\Delta}{W_i^\Delta} \leq W^\Delta$	OK	OK

Tabela 5.8: cálculos para verificar a consistência global sobre as atividades *A3 - Caso 1*, *A3 - Caso 2*, *A3 - Caso 3* e *A3 - Caso 4* do recurso R2 (Funcionário Reclamações)

$i \rightarrow$	A7 - Caso 1	A7 - Caso 2
$\Delta$	[45,105]	[58,118]
$W^\Delta$	$(105 - 45) \times 100 = 6000$	$(118 - 58) \times 100 = 6000$
$\frac{W^\Delta}{A2-Caso1}$	$30 \times 0 = 0$	$30 \times 0 = 0$
$\frac{W^\Delta}{A2-Caso2}$	$30 \times 0 = 0$	$30 \times 0 = 0$
$\frac{W^\Delta}{A2-Caso3}$	$30 \times 0 = 0$	$30 \times 0 = 0$
$\frac{W^\Delta}{A2-Caso4}$	$30 \times 3 = 90$	$30 \times 0 = 0$
$\frac{W^\Delta}{A3-Caso1}$	$40 \times 0 = 0$	$40 \times 0 = 0$
$\frac{W^\Delta}{A3-Caso2}$	$40 \times 0 = 0$	$40 \times 0 = 0$
$\frac{W^\Delta}{A3-Caso3}$	$40 \times 3 = 120$	$40 \times 0 = 0$
$\frac{W^\Delta}{A3-Caso4}$	$40 \times 8 = 320$	$40 \times 0 = 0$
$\frac{W^\Delta}{A7-Caso1}$	$50 \times 20 = 1000$	$50 \times 7 = 350$
$\frac{W^\Delta}{A7-Caso2}$	$50 \times 7 = 350$	$50 \times 20 = 1000$
$\frac{W^\Delta}{A7-Caso3}$	$50 \times 2 = 100$	$50 \times 15 = 750$
$\frac{W^\Delta}{A7-Caso4}$	$50 \times 0 = 0$	$50 \times 10 = 500$
$\sum_{i=1}^{12} \frac{W^\Delta}{W_i}$	1980	2600
$\sum_{i=1}^{12} \frac{W^\Delta}{W_i} \leq W^\Delta$	OK	OK

$i \rightarrow$	A7 - Caso 3	A7 - Caso 4
$\Delta$	[63,123]	[68,128]
$W^\Delta$	$(123 - 63) \times 100 = 6000$	$(128 - 68) \times 100 = 6000$
$\frac{W^\Delta}{A2-Caso1}$	$30 \times 0 = 0$	$30 \times 0 = 0$
$\frac{W^\Delta}{A2-Caso2}$	$30 \times 0 = 0$	$30 \times 0 = 0$
$\frac{W^\Delta}{A2-Caso3}$	$30 \times 0 = 0$	$30 \times 0 = 0$
$\frac{W^\Delta}{A2-Caso4}$	$30 \times 0 = 0$	$30 \times 0 = 0$
$\frac{W^\Delta}{A3-Caso1}$	$40 \times 0 = 0$	$40 \times 0 = 0$
$\frac{W^\Delta}{A3-Caso2}$	$40 \times 0 = 0$	$40 \times 0 = 0$
$\frac{W^\Delta}{A3-Caso3}$	$40 \times 0 = 0$	$40 \times 0 = 0$
$\frac{W^\Delta}{A3-Caso4}$	$40 \times 0 = 0$	$40 \times 0 = 0$
$\frac{W^\Delta}{A7-Caso1}$	$50 \times 2 = 100$	$50 \times 0 = 0$
$\frac{W^\Delta}{A7-Caso2}$	$50 \times 15 = 750$	$50 \times 10 = 500$
$\frac{W^\Delta}{A7-Caso3}$	$50 \times 20 = 1000$	$50 \times 15 = 750$
$\frac{W^\Delta}{A7-Caso4}$	$50 \times 15 = 750$	$50 \times 20 = 1000$
$\sum_{i=1}^{12} \frac{W^\Delta}{W_i}$	2600	2250
$\sum_{i=1}^{12} \frac{W^\Delta}{W_i} \leq W^\Delta$	OK	OK

Tabela 5.9: cálculos para verificar a consistência global sobre as atividades *A7 - Caso 1*, *A7 - Caso 2*, *A7 - Caso 3* e *A7 - Caso 4* do recurso R2 (Funcionário Reclamações)

Realizando-se a verificação de consistência global considerando as atividades envolvidas com o recurso R3 (Sistema) <sup>2</sup>, tem-se  $\sum_{i=1}^8 \frac{W_i^\Delta}{W_i} = 0$  para cada uma das 8 atividades (*A4 - Caso 1*, *A4 - Caso 2*, *A4 - Caso 3*, *A4 - Caso 4*, *A8 - Caso 1*, *A8 - Caso 2*, *A8 - Caso 3* e *A8 - Caso 4*) envolvidas com esse recurso.

Na tabela 5.10 são apresentados os resultados dos cálculos considerados na verificação

<sup>2</sup>As durações das atividades relacionadas ao recurso R3 são igualadas a zero, já que são desprezíveis levando em conta as durações das demais atividades

de consistência global tendo em conta as atividades envolvidas com o recurso R4 (Analista Reclamações).

$i \rightarrow$	A5 - Caso 1	A5 - Caso 2
$\Delta$	[30,90]	[43,103]
$W^\Delta$	$(90 - 30) \times 100 = 6000$	$(103 - 43) \times 100 = 6000$
$\frac{W_{A5-Caso1}^\Delta}{W_{A5-Caso1}^\Delta}$	$100 \times 15 = 1500$	$100 \times 2 = 200$
$\frac{W_{A5-Caso2}^\Delta}{W_{A5-Caso2}^\Delta}$	$100 \times 2 = 200$	$100 \times 15 = 1500$
$\frac{W_{A5-Caso3}^\Delta}{W_{A5-Caso3}^\Delta}$	$100 \times 0 = 0$	$100 \times 10 = 1000$
$\frac{W_{A5-Caso4}^\Delta}{W_{A5-Caso4}^\Delta}$	$100 \times 0 = 0$	$100 \times 5 = 500$
$\sum_{i=1}^4 W_i^\Delta$	1700	3200
$\sum_{i=1}^4 \frac{W_i^\Delta}{W_i^\Delta} \leq W^\Delta$	OK	OK

$i \rightarrow$	A5 - Caso 3	A5 - Caso 4
$\Delta$	[48,108]	[53,113]
$W^\Delta$	$(108 - 48) \times 100 = 6000$	$(113 - 53) \times 100 = 6000$
$\frac{W_{A5-Caso1}^\Delta}{W_{A5-Caso1}^\Delta}$	$100 \times 0 = 0$	$100 \times 0 = 0$
$\frac{W_{A5-Caso2}^\Delta}{W_{A5-Caso2}^\Delta}$	$100 \times 10 = 1000$	$100 \times 5 = 500$
$\frac{W_{A5-Caso3}^\Delta}{W_{A5-Caso3}^\Delta}$	$100 \times 15 = 1500$	$100 \times 10 = 1000$
$\frac{W_{A5-Caso4}^\Delta}{W_{A5-Caso4}^\Delta}$	$100 \times 10 = 1000$	$100 \times 15 = 1500$
$\sum_{i=1}^4 W_i^\Delta$	3500	3000
$\sum_{i=1}^4 \frac{W_i^\Delta}{W_i^\Delta} \leq W^\Delta$	OK	OK

Tabela 5.10: cálculos para verificar a consistência global sobre as atividades do recurso R4 (Analista Reclamações)

Na tabela 5.11 são apresentados os resultados dos cálculos considerados na verificação de consistência global levando em conta as atividades envolvidas com o recurso R5 (Funcionário Finanças).

$i \rightarrow$	A6 - Caso 1	A6 - Caso 2
$\Delta$	[45,105]	[58,118]
$W^\Delta$	$(105 - 45) \times 100 = 6000$	$(118 - 58) \times 100 = 6000$
$\frac{W^\Delta}{A6-Caso1}$	$100 \times 5 = 500$	$100 \times 0 = 0$
$\frac{W^\Delta}{A6-Caso2}$	$100 \times 0 = 0$	$100 \times 5 = 500$
$\frac{W^\Delta}{A6-Caso3}$	$100 \times 0 = 0$	$100 \times 0 = 0$
$\frac{W^\Delta}{A6-Caso4}$	$100 \times 0 = 0$	$100 \times 0 = 0$
$\sum_{i=1}^4 \frac{W_i^\Delta}{W_i^\Delta}$	500	500
$\sum_{i=1}^4 \frac{W_i^\Delta}{W_i^\Delta} \leq W^\Delta$	OK	OK

$i \rightarrow$	A6 - Caso 3	A6 - Caso 4
$\Delta$	[63,123]	[68,128]
$W^\Delta$	$(123 - 63) \times 100 = 6000$	$(128 - 68) \times 100 = 6000$
$\frac{W^\Delta}{A6-Caso1}$	$100 \times 0 = 0$	$100 \times 0 = 0$
$\frac{W^\Delta}{A6-Caso2}$	$100 \times 0 = 0$	$100 \times 0 = 0$
$\frac{W^\Delta}{A6-Caso3}$	$100 \times 5 = 500$	$100 \times 0 = 0$
$\frac{W^\Delta}{A6-Caso4}$	$100 \times 0 = 0$	$100 \times 5 = 500$
$\sum_{i=1}^4 \frac{W_i^\Delta}{W_i^\Delta}$	500	500
$\sum_{i=1}^4 \frac{W_i^\Delta}{W_i^\Delta} \leq W^\Delta$	OK	OK

Tabela 5.11: cálculos para verificar a consistência global sobre as atividades do recurso R5 (Funcionário Finanças)

Satisfeita a condição de consistência global ( $W^\Delta \geq \sum_{i=1}^n \frac{W_i^\Delta}{W_i^\Delta}$ ) por todas as atividades envolvidas com os recusos do processo *Serviço de Reclamações*, o próximo passo é verificar se as atividades podem, efetivamente, serem iniciadas ao mais cedo nas datas  $r_i$  e ao mais tarde nas datas  $\bar{s}_i$  por meio da verificação da condição de consistência local.

Verificando-se a condição de consistência local ( $\bar{p}_i^\Delta \geq p_{mini}$ ) e considerando a data de início ao mais cedo ( $r_i$ ) das atividades envolvidas com os recursos *R1*, *R3*, *R4* e *R5*, é possível diagnosticar que não ocorre nenhuma condição de inconsistência local. Isso significa que, caso tais atividades iniciem suas execuções nas suas respectivas datas de início ao mais cedo, haverá energia suficiente do recurso em questão para completa execução das mesmas.

As tabelas 5.12, 5.13, 5.14, 5.15, 5.16, 5.17 apresentam os cálculos da verificação de consistência local sobre as datas de início ao mais cedo ( $r_i$ ) de cada uma das atividades envolvidas com o recurso R2 (*Funcionário Reclamações*).

$i \rightarrow$	A2 - Caso 1	A2 - Caso 2
$\Delta$	[5,25]	[18,38]
$W^\Delta$	$(25 - 5) \times 100 = 2000$	$(38 - 18) \times 100 = 2000$
$\overline{W}_{A2-Caso1}^\Delta$	Não se aplica	$30 \times 7 = 210$
$\overline{W}_{A2-Caso2}^\Delta$	$30 \times 0 = 0$	Não se aplica
$\overline{W}_{A2-Caso3}^\Delta$	$30 \times 0 = 0$	$30 \times 0 = 0$
$\overline{W}_{A2-Caso4}^\Delta$	$30 \times 0 = 0$	$30 \times 0 = 0$
$\overline{W}_{A3-Caso1}^\Delta$	$40 \times 0 = 0$	$40 \times 12 = 480$
$\overline{W}_{A3-Caso2}^\Delta$	$40 \times 0 = 0$	$40 \times 0 = 0$
$\overline{W}_{A3-Caso3}^\Delta$	$40 \times 0 = 0$	$40 \times 0 = 0$
$\overline{W}_{A3-Caso4}^\Delta$	$40 \times 0 = 0$	$40 \times 0 = 0$
$\overline{W}_{A7-Caso1}^\Delta$	$50 \times 0 = 0$	$50 \times 0 = 0$
$\overline{W}_{A7-Caso2}^\Delta$	$50 \times 0 = 0$	$50 \times 0 = 0$
$\overline{W}_{A7-Caso3}^\Delta$	$50 \times 0 = 0$	$50 \times 0 = 0$
$\overline{W}_{A7-Caso4}^\Delta$	$50 \times 0 = 0$	$50 \times 0 = 0$
$\sum_{j=1, j \neq i}^n \overline{W}_j^\Delta$	0	690
$A_i^\Delta$	$2000 - 0 = 2000$	$2000 - 690 = 1310$
$\overline{p}_i^\Delta$	$\frac{2000}{30} = 66.7$	$\frac{1310}{30} = 43.7$
$p_{min_i}$	20	20
$\overline{p}_i^\Delta \geq p_{min_i}$	OK	OK

Tabela 5.12: Cálculos para verificar a consistência local sobre as datas de início ao mais cedo das atividades *A2 - Caso 1*, *A2 - Caso 2* do processo *Serviço de Reclamações*

$i \rightarrow$	A2 - Caso 3	A2 - Caso 4
$\Delta$	[23,43]	[28,48]
$W^\Delta$	$(43 - 23) \times 100 = 2000$	$(48 - 28) \times 100 = 2000$
$\underline{W}_{A2-Caso1}^\Delta$	$30 \times 2 = 60$	$30 \times 2 = 60$
$\underline{W}_{A2-Caso2}^\Delta$	$30 \times 0 = 0$	$30 \times 5 = 150$
$\underline{W}_{A2-Caso3}^\Delta$	Não se aplica	$30 \times 0 = 0$
$\underline{W}_{A2-Caso4}^\Delta$	$30 \times 0 = 0$	Não se aplica
$\underline{W}_{A3-Caso1}^\Delta$	$40 \times 7 = 280$	$40 \times 2 = 80$
$\underline{W}_{A3-Caso2}^\Delta$	$40 \times 5 = 200$	$40 \times 10 = 400$
$\underline{W}_{A3-Caso3}^\Delta$	$40 \times 0 = 0$	$40 \times 5 = 200$
$\underline{W}_{A3-Caso4}^\Delta$	$40 \times 0 = 0$	$40 \times 0 = 0$
$\underline{W}_{A7-Caso1}^\Delta$	$50 \times 0 = 0$	$50 \times 0 = 0$
$\underline{W}_{A7-Caso2}^\Delta$	$50 \times 0 = 0$	$50 \times 0 = 0$
$\underline{W}_{A7-Caso3}^\Delta$	$50 \times 0 = 0$	$50 \times 0 = 0$
$\underline{W}_{A7-Caso4}^\Delta$	$50 \times 0 = 0$	$50 \times 0 = 0$
$\sum_{j=1, j \neq i}^n \underline{W}_j^\Delta$	540	890
$A_i^\Delta$	$2000 - 540 = 1460$	$2000 - 890 = 1110$
$\bar{p}_i^\Delta$	$\frac{1460}{30} = 48.7$	$\frac{1110}{30} = 37$
$p_{min_i}$	20	20
$\bar{p}_i^\Delta \geq p_{min_i}$	OK	OK

Tabela 5.13: Cálculos para verificar a consistência local sobre as datas de início ao mais cedo das atividades *A2 - Caso 3* e *A2 - Caso 4* do processo *Serviço de Reclamações*

$i \rightarrow$	A3 - Caso 1	A3 - Caso 2
$\Delta$	[5,30]	[18,43]
$W^\Delta$	$(30 - 5) \times 100 = 2500$	$(43 - 18) \times 100 = 2500$
$\overline{W}_{A2-Caso1}^\Delta$	$30 \times 0 = 0$	$30 \times 7 = 210$
$\overline{W}_{A2-Caso2}^\Delta$	$30 \times 0 = 0$	$30 \times 0 = 0$
$\overline{W}_{A2-Caso3}^\Delta$	$30 \times 0 = 0$	$30 \times 0 = 0$
$\overline{W}_{A2-Caso4}^\Delta$	$30 \times 0 = 0$	$30 \times 0 = 0$
$\overline{W}_{A3-Caso1}^\Delta$	Não se aplica	$40 \times 12 = 480$
$\overline{W}_{A3-Caso2}^\Delta$	$40 \times 0 = 0$	Não se aplica
$\overline{W}_{A3-Caso3}^\Delta$	$40 \times 0 = 0$	$40 \times 0 = 0$
$\overline{W}_{A3-Caso4}^\Delta$	$40 \times 0 = 0$	$40 \times 0 = 0$
$\overline{W}_{A7-Caso1}^\Delta$	$50 \times 0 = 0$	$50 \times 0 = 0$
$\overline{W}_{A7-Caso2}^\Delta$	$50 \times 0 = 0$	$50 \times 0 = 0$
$\overline{W}_{A7-Caso3}^\Delta$	$50 \times 0 = 0$	$50 \times 0 = 0$
$\overline{W}_{A7-Caso4}^\Delta$	$50 \times 0 = 0$	$50 \times 0 = 0$
$\sum_{j=1, j \neq i}^n \overline{W}_j^\Delta$	0	690
$A_i^\Delta$	$2500 - 0 = 2500$	$2500 - 690 = 1810$
$\overline{p}_i^\Delta$	$\frac{2500}{40} = 62.5$	$\frac{1810}{40} = 45.2$
$p_{min_i}$	25	25
$\overline{p}_i^\Delta \geq p_{min_i}$	OK	OK

Tabela 5.14: Cálculos para verificar a consistência local sobre as datas de início ao mais cedo das atividades *A3 - Caso 1* e *A3 - Caso 2* do processo *Serviço de Reclamações*



$i \rightarrow$	A3 - Caso 3	A3 - Caso 4
$\Delta$	[23,48]	[28,53]
$W^\Delta$	$(48 - 23) \times 100 = 2500$	$(53 - 28) \times 100 = 2500$
$\underline{W}_{A2-Caso1}^\Delta$	$30 \times 2 = 60$	$30 \times 0 = 0$
$\underline{W}_{A2-Caso2}^\Delta$	$30 \times 5 = 150$	$30 \times 10 = 300$
$\underline{W}_{A2-Caso3}^\Delta$	$30 \times 0 = 0$	$30 \times 5 = 150$
$\underline{W}_{A2-Caso4}^\Delta$	$30 \times 0 = 0$	$30 \times 0 = 0$
$\underline{W}_{A3-Caso1}^\Delta$	$40 \times 7 = 280$	$40 \times 2 = 80$
$\underline{W}_{A3-Caso2}^\Delta$	$40 \times 10 = 400$	$40 \times 15 = 600$
$\underline{W}_{A3-Caso3}^\Delta$	Não se aplica	$40 \times 10 = 400$
$\underline{W}_{A3-Caso4}^\Delta$	$40 \times 0 = 0$	Não se aplica
$\underline{W}_{A7-Caso1}^\Delta$	$50 \times 0 = 0$	$50 \times 0 = 0$
$\underline{W}_{A7-Caso2}^\Delta$	$50 \times 0 = 0$	$50 \times 0 = 0$
$\underline{W}_{A7-Caso3}^\Delta$	$50 \times 0 = 0$	$50 \times 0 = 0$
$\underline{W}_{A7-Caso4}^\Delta$	$50 \times 0 = 0$	$50 \times 0 = 0$
$\sum_{j=1, j \neq i}^n \underline{W}_j^\Delta$	890	1530
$A_i^\Delta$	$2500 - 890 = 1610$	$2500 - 1530 = 970$
$\bar{p}_i^\Delta$	$\frac{1610}{40} = 40.2$	$\frac{970}{40} = 24.2$
$p_{min_i}$	25	25
$\bar{p}_i^\Delta \geq p_{min_i}$	OK	Inconsistência Local

Tabela 5.15: Cálculos para verificar a consistência local sobre as datas de início ao mais cedo das atividades *A3 - Caso 3* e *A3 - Caso 4* do processo *Serviço de Reclamações*

$i \rightarrow$	A7 - Caso 1	A7 - Caso 2
$\Delta$	[45,65]	[58,78]
$W^\Delta$	$(65 - 45) \times 100 = 2000$	$(78 - 58) \times 100 = 2000$
$\underline{W}_{A2-Caso1}^\Delta$	$30 \times 0 = 0$	$30 \times 0 = 0$
$\underline{W}_{A2-Caso2}^\Delta$	$30 \times 0 = 0$	$30 \times 0 = 0$
$\underline{W}_{A2-Caso3}^\Delta$	$30 \times 0 = 0$	$30 \times 0 = 0$
$\underline{W}_{A2-Caso4}^\Delta$	$30 \times 0 = 0$	$30 \times 0 = 0$
$\underline{W}_{A3-Caso1}^\Delta$	$40 \times 0 = 0$	$40 \times 0 = 0$
$\underline{W}_{A3-Caso2}^\Delta$	$40 \times 0 = 0$	$40 \times 0 = 0$
$\underline{W}_{A3-Caso3}^\Delta$	$40 \times 0 = 0$	$40 \times 0 = 0$
$\underline{W}_{A3-Caso4}^\Delta$	$40 \times 0 = 0$	$40 \times 0 = 0$
$\underline{W}_{A7-Caso1}^\Delta$	Não se aplica	$50 \times 0 = 0$
$\underline{W}_{A7-Caso2}^\Delta$	$50 \times 0 = 0$	Não se aplica
$\underline{W}_{A7-Caso3}^\Delta$	$50 \times 0 = 0$	$50 \times 0 = 0$
$\underline{W}_{A7-Caso4}^\Delta$	$50 \times 0 = 0$	$50 \times 0 = 0$
$\sum_{j=1, j \neq i}^n \underline{W}_j^\Delta$	0	0
$A_i^\Delta$	$2000 - 0 = 2000$	$2000 - 0 = 2000$
$\bar{p}_i^\Delta$	$\frac{2000}{50} = 40$	$\frac{2000}{50} = 40$
$p_{min_i}$	25	25
$\bar{p}_i^\Delta \geq p_{min_i}$	OK	OK

Tabela 5.16: Cálculos para verificar a consistência local sobre as datas de início ao mais cedo das atividades *A7 - Caso 1* e *A7 - Caso 2* do processo *Serviço de Reclamações*

$i \rightarrow$	A7 - Caso 3	A7 - Caso 4
$\Delta$	[63,83]	[68,88]
$W^\Delta$	$(83 - 63) \times 100 = 2000$	$(88 - 68) \times 100 = 2000$
$\underline{W}_{A2-Caso1}^\Delta$	$30 \times 0 = 0$	$30 \times 0 = 0$
$\underline{W}_{A2-Caso2}^\Delta$	$30 \times 0 = 0$	$30 \times 0 = 0$
$\underline{W}_{A2-Caso3}^\Delta$	$30 \times 0 = 0$	$30 \times 0 = 0$
$\underline{W}_{A2-Caso4}^\Delta$	$30 \times 0 = 0$	$30 \times 0 = 0$
$\underline{W}_{A3-Caso1}^\Delta$	$40 \times 0 = 0$	$40 \times 0 = 0$
$\underline{W}_{A3-Caso2}^\Delta$	$40 \times 0 = 0$	$40 \times 0 = 0$
$\underline{W}_{A3-Caso3}^\Delta$	$40 \times 0 = 0$	$40 \times 0 = 0$
$\underline{W}_{A3-Caso4}^\Delta$	$40 \times 0 = 0$	$40 \times 0 = 0$
$\underline{W}_{A7-Caso1}^\Delta$	$50 \times 0 = 0$	$50 \times 0 = 0$
$\underline{W}_{A7-Caso2}^\Delta$	$50 \times 0 = 0$	$50 \times 0 = 0$
$\underline{W}_{A7-Caso3}^\Delta$	Não se aplica	$50 \times 0 = 0$
$\underline{W}_{A7-Caso4}^\Delta$	$50 \times 0 = 0$	Não se aplica
$\sum_{j=1, j \neq i}^n \underline{W}_j^\Delta$	0	0
$A_i^\Delta$	$2000 - 0 = 2000$	$2000 - 0 = 2000$
$\bar{p}_i^\Delta$	$\frac{2000}{50} = 40$	$\frac{2000}{50} = 40$
$p_{min_i}$	25	25
$\bar{p}_i^\Delta \geq p_{min_i}$	OK	OK

Tabela 5.17: Cálculos para verificar a consistência local sobre as datas de início ao mais cedo das atividades *A7 - Caso 3* e *A7 - Caso 4* do processo *Serviço de Reclamações*

A figura 5.6 ilustra o fragmento da rede de Petri com as atividades relacionadas com o recurso *R2* (Funcionário reclamações).

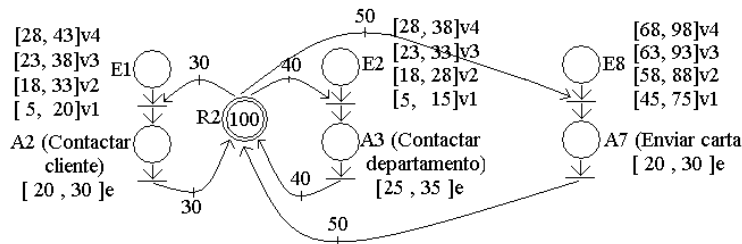


Figura 5.6: Fragmento da rede de Petri apresentando as atividades envolvidas com o recurso *Funcionário - reclamações*

Para a atividade *A3 - Caso 4*, pode-se observar que  $\bar{p}_i^\Delta = 24.2$  é menor do que sua duração mínima  $p_{min_i} = 25$ ; logo, pela desigualdade  $\bar{p}_i^\Delta < p_{min_i}$ , existe uma inconsistência local sobre a data de início ao mais cedo dessa atividade. Dessa forma, é necessário tentar um deslocamento à direita do  $\Delta$  ( $\Delta = [28, 53]$ ) a fim de encontrar uma nova data de início ao mais cedo para essa atividade.

Contudo é importante notar que o valor de  $F_\Delta$  (neste caso particular tem-se  $F_\Delta = 53$ ) não poderá superar o valor de  $d_i$  correspondente à atividade em questão (neste caso particular tem-se  $d_i = 73$ ). Caso contrário, não será possível garantir que seja cumprido o prazo de entrega para o caso considerado.

Como foi apresentado no capítulo 4, o valor do deslocamento do  $\Delta$  é calculado subtraindo-se o valor da duração mínima da atividade ( $p_{min_i}$ ) do valor da sua duração máxima equivalente sobre o  $\Delta$  ( $\bar{p}_i^\Delta$ ). Assim, para a atividade  $A3 - Caso 4$ , tem-se:

- 1° Deslocamento do  $\Delta = 25 - 24.2 = 0.8$
- Novo  $\Delta = [28.8, 53.8]$
- $W^\Delta = (53.8 - 28.8) \times 100 = 2500$
- $\underline{W}_{A2-caso1}^\Delta = 30 \times 0 = 0$
- $\underline{W}_{A2-caso2}^\Delta = 30 \times 9.2 = 276$
- $\underline{W}_{A2-caso3}^\Delta = 30 \times 5.8 = 174$
- $\underline{W}_{A2-caso4}^\Delta = 30 \times 0.8 = 24$
- $\underline{W}_{A3-caso1}^\Delta = 40 \times 1.2 = 48$
- $\underline{W}_{A3-caso3}^\Delta = 40 \times 14.2 = 568$
- $\underline{W}_{A3-caso3}^\Delta = 40 \times 10.8 = 432$
- $\underline{W}_{A7-caso1}^\Delta = 50 \times 0 = 0$
- $\underline{W}_{A7-caso2}^\Delta = 50 \times 0 = 0$
- $\underline{W}_{A7-caso3}^\Delta = 50 \times 0 = 0$
- $\underline{W}_{A7-caso4}^\Delta = 50 \times 0 = 0$
- $\sum_{j=1, j \neq i}^n \underline{W}_j^\Delta = 1522$
- $A_i^\Delta = 2500 - 1522 = 978$
- $\bar{p}_i^\Delta = \frac{978}{40} = 24.4$
- $p_{min_i} = 25$

Dessa forma, pode-se observar que  $\bar{p}_i^\Delta = 24.4$  ainda é menor do que a duração mínima  $p_{min_i} = 25$  da atividade  $A3 - Caso 4$ ; logo, pela desigualdade  $\bar{p}_i^\Delta < p_{min_i}$ , ainda existe uma inconsistência local sobre a data de início ao mais cedo dessa atividade. Assim faz-se necessário continuar tentando um deslocamento à direita do  $\Delta$  até que seja encontrada uma condição de consistência local.

A seguir, são apresentados os deslocamentos à direita do  $\Delta$  até onde foi encontrada a condição de consistência local:

- 2° Deslocamento do  $\Delta = 25 - 24.4 = 0.6$ 
  - Novo  $\Delta = [29.4, 54.4]$
  - $W^\Delta = (54.4 - 29.4) \times 100 = 2500$
  - $\underline{W}_{A2-caso1}^\Delta = 30 \times 0 = 0$

- $\underline{W}_{A2-caso2}^{\Delta} = 30 \times 8.6 = 258$
- $\underline{W}_{A2-caso3}^{\Delta} = 30 \times 6.4 = 192$
- $\underline{W}_{A2-caso4}^{\Delta} = 30 \times 1.4 = 42$
- $\underline{W}_{A3-caso1}^{\Delta} = 40 \times 0.6 = 24$
- $\underline{W}_{A3-caso3}^{\Delta} = 40 \times 13.6 = 544$
- $\underline{W}_{A3-caso3}^{\Delta} = 40 \times 11.4 = 456$
- $\underline{W}_{A7-caso1}^{\Delta} = 50 \times 0 = 0$
- $\underline{W}_{A7-caso2}^{\Delta} = 50 \times 0 = 0$
- $\underline{W}_{A7-caso3}^{\Delta} = 50 \times 0 = 0$
- $\underline{W}_{A7-caso4}^{\Delta} = 50 \times 0 = 0$
- $\sum_{j=1, j \neq i}^n \underline{W}_j^{\Delta} = 1516$
- $A_i^{\Delta} = 2500 - 1516 = 984$
- $\bar{p}_i^{\Delta} = \frac{984}{40} = 24.6$
- $p_{min_i} = 25$

Sendo  $\bar{p}_i^{\Delta} = 24.6$  menor que  $p_{min_i} = 25$ , a atividade *A3 - Caso 4* não pode ser iniciada ao mais cedo na data  $r_i = 29.4$  ( $r_i$  agora é igual ao novo  $S_{\Delta}$ ). Como o valor do novo  $F_{\Delta} = 54.4$  não superou o valor de  $d_i = 73$ , é necessário tentar outro deslocamento do  $\Delta$  à direita até que seja encontrada uma condição de consistência local:

- 3º Deslocamento do  $\Delta = 25 - 24.6 = 0.4$

- Novo  $\Delta = [29.8, 54.8]$
- $W^{\Delta} = (54.8 - 29.8) \times 100 = 2500$
- $\underline{W}_{A2-caso1}^{\Delta} = 30 \times 0 = 0$
- $\underline{W}_{A2-caso2}^{\Delta} = 30 \times 8.2 = 246$
- $\underline{W}_{A2-caso3}^{\Delta} = 30 \times 6.8 = 204$
- $\underline{W}_{A2-caso4}^{\Delta} = 30 \times 1.8 = 54$
- $\underline{W}_{A3-caso1}^{\Delta} = 40 \times 0.2 = 8$
- $\underline{W}_{A3-caso3}^{\Delta} = 40 \times 13.2 = 528$
- $\underline{W}_{A3-caso3}^{\Delta} = 40 \times 11.8 = 472$
- $\underline{W}_{A7-caso1}^{\Delta} = 50 \times 0 = 0$
- $\underline{W}_{A7-caso2}^{\Delta} = 50 \times 0 = 0$
- $\underline{W}_{A7-caso3}^{\Delta} = 50 \times 0 = 0$
- $\underline{W}_{A7-caso4}^{\Delta} = 50 \times 0 = 0$
- $\sum_{j=1, j \neq i}^n \underline{W}_j^{\Delta} = 1512$
- $A_i^{\Delta} = 2500 - 1512 = 988$

- $\bar{p}_i^\Delta = \frac{988}{40} = 24.7$
- $p_{min_i} = 25$

Com  $\bar{p}_i^\Delta = 24.7$  menor que  $p_{min_i} = 25$ , a atividade *A3 - Caso 4* não pode ser iniciada ao mais cedo na data  $r_i = 29.8$  ( $r_i$  agora é igual ao novo  $S_\Delta$ ). Como o valor do novo  $F_\Delta = 54.8$  não superou o valor de  $d_i = 73$ , é necessário tentar outro deslocamento do  $\Delta$  à direita até que seja encontrada uma condição de consistência local:

- 4° Deslocamento do  $\Delta = 25 - 24.7 = 0.3$

- Novo  $\Delta = [30.1, 55.1]$
- $W^\Delta = (55.1 - 30.1) \times 100 = 2500$
- $\underline{W}_{A2-caso1}^\Delta = 30 \times 0 = 0$
- $\underline{W}_{A2-caso2}^\Delta = 30 \times 7.9 = 237$
- $\underline{W}_{A2-caso3}^\Delta = 30 \times 7.1 = 213$
- $\underline{W}_{A2-caso4}^\Delta = 30 \times 2.1 = 63$
- $\underline{W}_{A3-caso1}^\Delta = 40 \times 0 = 0$
- $\underline{W}_{A3-caso3}^\Delta = 40 \times 12.9 = 516$
- $\underline{W}_{A3-caso3}^\Delta = 40 \times 12.1 = 484$
- $\underline{W}_{A7-caso1}^\Delta = 50 \times 0 = 0$
- $\underline{W}_{A7-caso2}^\Delta = 50 \times 0 = 0$
- $\underline{W}_{A7-caso3}^\Delta = 50 \times 0 = 0$
- $\underline{W}_{A7-caso4}^\Delta = 50 \times 0 = 0$
- $\sum_{j=1, j \neq i}^n \underline{W}_j^\Delta = 1513$
- $A_i^\Delta = 2500 - 1513 = 987$
- $\bar{p}_i^\Delta = \frac{987}{40} = 24.7$
- $p_{min_i} = 25$

Com  $\bar{p}_i^\Delta = 24.7$  menor que  $p_{min_i} = 25$ , a atividade *A3 - Caso 4* não pode ser iniciada ao mais cedo na data  $r_i = 30.1$  ( $r_i$  agora é igual ao novo  $S_\Delta$ ). Como o valor do novo  $F_\Delta = 55.1$  não superou o valor de  $d_i = 73$ , é necessário tentar outro deslocamento do  $\Delta$  à direita até que seja encontrada uma condição de consistência local:

- 5° Deslocamento do  $\Delta = 25 - 24.7 = 0.3$

- Novo  $\Delta = [30.4, 55.4]$
- $W^\Delta = (55.4 - 30.4) \times 100 = 2500$
- $\underline{W}_{A2-caso1}^\Delta = 30 \times 0 = 0$
- $\underline{W}_{A2-caso2}^\Delta = 30 \times 7.6 = 228$

- $\underline{W}_{A2-caso3}^{\Delta} = 30 \times 7.4 = 222$
- $\underline{W}_{A2-caso4}^{\Delta} = 30 \times 2.4 = 72$
- $\underline{W}_{A3-caso1}^{\Delta} = 40 \times 0 = 0$
- $\underline{W}_{A3-caso3}^{\Delta} = 40 \times 12.6 = 504$
- $\underline{W}_{A3-caso3}^{\Delta} = 40 \times 12.4 = 496$
- $\underline{W}_{A7-caso1}^{\Delta} = 50 \times 0 = 0$
- $\underline{W}_{A7-caso2}^{\Delta} = 50 \times 0 = 0$
- $\underline{W}_{A7-caso3}^{\Delta} = 50 \times 0 = 0$
- $\underline{W}_{A7-caso4}^{\Delta} = 50 \times 0 = 0$
- $\sum_{j=1, j \neq i}^n \underline{W}_j^{\Delta} = 1522$
- $A_i^{\Delta} = 2500 - 1522 = 978$
- $\bar{p}_i^{\Delta} = \frac{978}{40} = 24.4$
- $p_{min_i} = 25$

Com  $\bar{p}_i^{\Delta} = 24.4$  menor que  $p_{min_i} = 25$ , a atividade *A3 - Caso 4* não pode ser iniciada ao mais cedo na data  $r_i = 30.4$  ( $r_i$  agora é igual ao novo  $S_{\Delta}$ ). Como o valor do novo  $F_{\Delta} = 55.4$  não superou o valor de  $d_i = 73$ , é necessário tentar outro deslocamento do  $\Delta$  à direita até que seja encontrada uma condição de consistência local:

- 6° Deslocamento do  $\Delta = 25 - 24.4 = 0.6$

- Novo  $\Delta = [31, 56]$
- $W^{\Delta} = (56 - 31) \times 100 = 2500$
- $\underline{W}_{A2-caso1}^{\Delta} = 30 \times 0 = 0$
- $\underline{W}_{A2-caso2}^{\Delta} = 30 \times 7 = 210$
- $\underline{W}_{A2-caso3}^{\Delta} = 30 \times 8 = 240$
- $\underline{W}_{A2-caso4}^{\Delta} = 30 \times 3 = 90$
- $\underline{W}_{A3-caso1}^{\Delta} = 40 \times 0 = 0$
- $\underline{W}_{A3-caso3}^{\Delta} = 40 \times 12 = 480$
- $\underline{W}_{A3-caso3}^{\Delta} = 40 \times 13 = 520$
- $\underline{W}_{A7-caso1}^{\Delta} = 50 \times 0 = 0$
- $\underline{W}_{A7-caso2}^{\Delta} = 50 \times 0 = 0$
- $\underline{W}_{A7-caso3}^{\Delta} = 50 \times 0 = 0$
- $\underline{W}_{A7-caso4}^{\Delta} = 50 \times 0 = 0$
- $\sum_{j=1, j \neq i}^n \underline{W}_j^{\Delta} = 1540$
- $A_i^{\Delta} = 2500 - 1540 = 960$
- $\bar{p}_i^{\Delta} = \frac{960}{40} = 24$

$$- p_{min_i} = 25$$

Com  $\bar{p}_i^\Delta = 24$  menor que  $p_{min_i} = 25$ , a atividade *A3 - Caso 4* não pode ser iniciada ao mais cedo na data  $r_i = 31$  ( $r_i$  agora é igual ao novo  $S_\Delta$ ). Como o valor do novo  $F_\Delta = 56$  não superou o valor de  $d_i = 73$ , é necessário tentar outro deslocamento do  $\Delta$  à direita até que seja encontrada uma condição de consistência local:

- 7° Deslocamento do  $\Delta = 25 - 24 = 1$

$$- Novo \Delta = [32, 57]$$

$$- W^\Delta = (57 - 32) \times 100 = 2500$$

$$- \underline{W}_{A2-caso1}^\Delta = 30 \times 0 = 0$$

$$- \underline{W}_{A2-caso2}^\Delta = 30 \times 6 = 180$$

$$- \underline{W}_{A2-caso3}^\Delta = 30 \times 9 = 270$$

$$- \underline{W}_{A2-caso4}^\Delta = 30 \times 4 = 120$$

$$- \underline{W}_{A3-caso1}^\Delta = 40 \times 0 = 0$$

$$- \underline{W}_{A3-caso3}^\Delta = 40 \times 11 = 440$$

$$- \underline{W}_{A3-caso3}^\Delta = 40 \times 14 = 560$$

$$- \underline{W}_{A7-caso1}^\Delta = 50 \times 0 = 0$$

$$- \underline{W}_{A7-caso2}^\Delta = 50 \times 0 = 0$$

$$- \underline{W}_{A7-caso3}^\Delta = 50 \times 0 = 0$$

$$- \underline{W}_{A7-caso4}^\Delta = 50 \times 0 = 0$$

$$- \sum_{j=1, j \neq i}^n \underline{W}_j^\Delta = 1570$$

$$- A_i^\Delta = 2500 - 1570 = 930$$

$$- \bar{p}_i^\Delta = \frac{930}{40} = 23.2$$

$$- p_{min_i} = 25$$

Com  $\bar{p}_i^\Delta = 23.2$  menor que  $p_{min_i} = 25$ , a atividade *A3 - Caso 4* não pode ser iniciada ao mais cedo na data  $r_i = 32$  ( $r_i$  agora é igual ao novo  $S_\Delta$ ). Como o valor do novo  $F_\Delta = 57$  não superou o valor de  $d_i = 73$ , é necessário tentar outro deslocamento do  $\Delta$  à direita até que seja encontrada uma condição de consistência local:

- 8° Deslocamento do  $\Delta = 25 - 23.2 = 1.8$

$$- Novo \Delta = [33.8, 58.8]$$

$$- W^\Delta = (58.8 - 33.8) \times 100 = 2500$$

$$- \underline{W}_{A2-caso1}^\Delta = 30 \times 0 = 0$$

$$- \underline{W}_{A2-caso2}^\Delta = 30 \times 4.2 = 126$$

$$- \underline{W}_{A2-caso3}^\Delta = 30 \times 9.2 = 276$$



- $\underline{W}_{A2-caso4}^{\Delta} = 30 \times 8.7 = 174$
- $\underline{W}_{A3-caso1}^{\Delta} = 40 \times 0 = 0$
- $\underline{W}_{A3-caso3}^{\Delta} = 40 \times 9.2 = 368$
- $\underline{W}_{A3-caso3}^{\Delta} = 40 \times 14.2 = 568$
- $\underline{W}_{A7-caso1}^{\Delta} = 50 \times 0 = 0$
- $\underline{W}_{A7-caso2}^{\Delta} = 50 \times 0 = 0$
- $\underline{W}_{A7-caso3}^{\Delta} = 50 \times 0 = 0$
- $\underline{W}_{A7-caso4}^{\Delta} = 50 \times 0 = 0$
- $\sum_{j=1, j \neq i}^n \underline{W}_j^{\Delta} = 1512$
- $A_i^{\Delta} = 2500 - 1512 = 988$
- $\bar{p}_i^{\Delta} = \frac{988}{40} = 24.7$
- $p_{min_i} = 25$

Com  $\bar{p}_i^{\Delta} = 24.7$  menor que  $p_{min_i} = 25$ , a atividade *A3 - Caso 4* não pode ser iniciada ao mais cedo na data  $r_i = 33.8$  ( $r_i$  agora é igual ao novo  $S_{\Delta}$ ). Como o valor do novo  $F_{\Delta} = 58.8$  não superou o valor de  $d_i = 73$ , é necessário tentar outro deslocamento do  $\Delta$  à direita até que seja encontrada uma condição de consistência local:

- 9º Deslocamento do  $\Delta = 25 - 24.7 = 0.3$

- Novo  $\Delta = [34.1, 59.1]$
- $W^{\Delta} = (59.1 - 34.1) \times 100 = 2500$
- $\underline{W}_{A2-caso1}^{\Delta} = 30 \times 0 = 0$
- $\underline{W}_{A2-caso2}^{\Delta} = 30 \times 3.9 = 117$
- $\underline{W}_{A2-caso3}^{\Delta} = 30 \times 8.9 = 267$
- $\underline{W}_{A2-caso4}^{\Delta} = 30 \times 6.1 = 183$
- $\underline{W}_{A3-caso1}^{\Delta} = 40 \times 0 = 0$
- $\underline{W}_{A3-caso3}^{\Delta} = 40 \times 8.9 = 356$
- $\underline{W}_{A3-caso3}^{\Delta} = 40 \times 13.9 = 556$
- $\underline{W}_{A7-caso1}^{\Delta} = 50 \times 0 = 0$
- $\underline{W}_{A7-caso2}^{\Delta} = 50 \times 0 = 0$
- $\underline{W}_{A7-caso3}^{\Delta} = 50 \times 0 = 0$
- $\underline{W}_{A7-caso4}^{\Delta} = 50 \times 0 = 0$
- $\sum_{j=1, j \neq i}^n \underline{W}_j^{\Delta} = 1479$
- $A_i^{\Delta} = 2500 - 1479 = 1021$
- $\bar{p}_i^{\Delta} = \frac{1021}{40} = 25.5$
- $p_{min_i} = 25$

Como  $\bar{p}_i^A = 25.5$  é maior a duração mínima  $p_{\min_i} = 25$  da atividade  $A3$  - *Caso 4* então a inconsistência local deixa de existir e a data de início ao mais cedo dessa atividade passa a ser igual a 34.1 ou, aproximadamente, 34. A figura 5.7 ilustra o deslocamento à direita da data de início ao mais cedo dessa atividade.

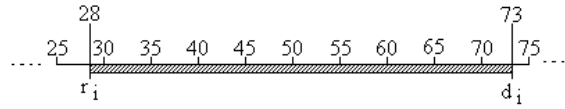


Figura (a) - Antes do deslocamento

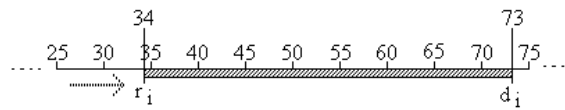


Figura (b) - Depois do deslocamento

Figura 5.7: Deslocamento da data de início ao mais cedo da atividade  $A3$  relacionada ao caso 4

A figura 5.8 ilustra o fragmento da rede de Petri com as atividades relacionadas com o recurso  $R2$  (Funcionário Reclamações) após a atualização da data de início ao mais cedo da atividade  $A3$  - *Caso 4*.

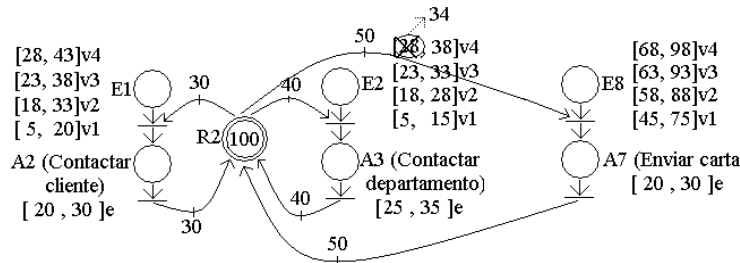


Figura 5.8: Fragmento da rede de Petri após a atualização da data de início ao mais cedo da atividade  $A3$  relacionada ao caso 4

Verificando-se a condição de consistência local ( $\bar{p}_i^A \geq p_{\min_i}$ ) e considerando a data de início ao mais tarde ( $\bar{s}_i$ ) das atividades envolvidas com todos os recursos  $R1$ ,  $R2$ ,  $R3$ ,  $R4$  e  $R5$  do processo *Serviço de Reclamações*, é possível diagnosticar que não ocorre nenhuma condição de inconsistência local. Isso significa que, caso essas atividades iniciem suas execuções nas suas respectivas datas de início ao mais tarde, haverá energia suficiente do recurso em questão para a completa execução das mesmas.

Após a atualização do intervalo de visibilidade  $V4$  em  $E2$  relacionado à atividade  $A3$  do caso 4 (figura 5.8), é necessário aplicar novamente o mecanismo de propagação das

restrições temporais ao longo dos roteiros do processo completo do *Serviço de reclamações*. Desse modo, acontecem atualizações nos intervalos de visibilidades associados aos lugares  $E_4$ ,  $E_5$ ,  $E_6$ ,  $E_7$ ,  $E_8$  e  $E_9$  referentes ao caso 4 conforme ilustrado pela figura 5.9(B). Tais atualizações correspondem, respectivamente, às datas de início ao mais cedo das atividades  $A_4$ ,  $A_5$ ,  $A_6$ ,  $A_7$  e  $A_8$  relacionadas com o caso 4. A atualização da borda mínima do intervalo de visibilidade associado a  $E_{10}$  relacionada ao caso 4 corresponde à data de término ao mais cedo desse caso.

A figura 5.9 apresenta o processo do *Serviço de Reclamações* antes (Figura (a)) e após (Figura (b)) a aplicação da abordagem baseada no Raciocínio Energético.

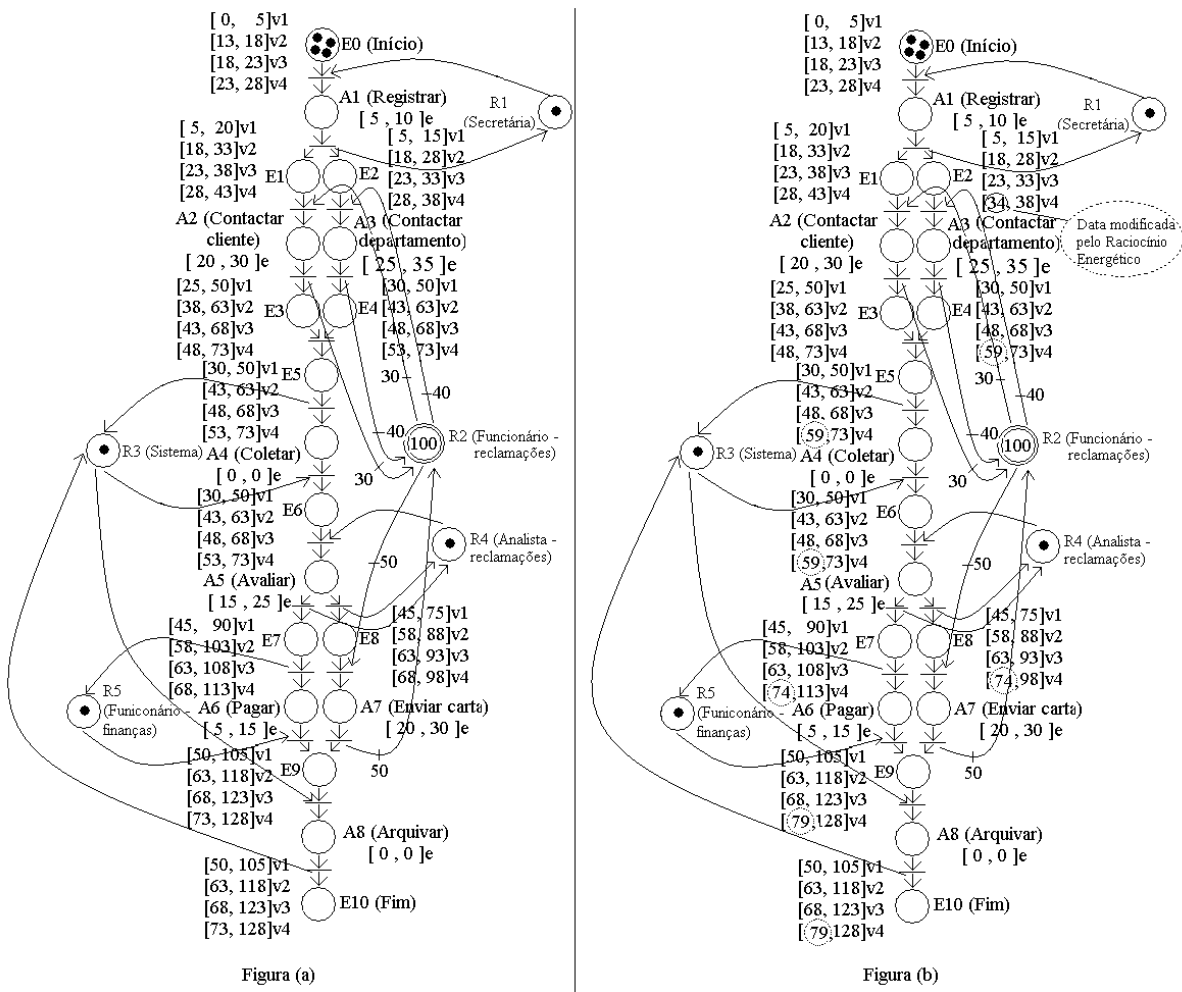


Figura 5.9: Aplicação da abordagem baseada no Raciocínio Energético sobre o processo do *Serviço de reclamações* - Antes da aplicação (Figura (a)) e depois da aplicação da abordagem (Figura (b))

## 5.5 Considerações sobre conflito no contexto das redes de Petri p-temporal e do jogador de redes de Petri

Um dos propósitos da abordagem proposta é fornecer melhores intervalos de visibilidade ao jogador de redes de Petri p-temporal utilizado no trabalho de Oliveira [de Oliveira, 2005], [Júlia *et al.*, 2007], ou seja, intervalos de visibilidades que gerem menos conflitos futuros.

Primeiramente, é preciso definir o que é um conflito para uma rede de Petri p-temporal. De acordo com Oliveira [de Oliveira, 2005], para uma rede de Petri ordinária, somente informações qualitativas do tempo são representadas; por exemplo, duas atividades executadas ao mesmo tempo (paralelismo) ou duas atividades em exclusão mútua (transições em conflito para uma dada marcação). No caso das redes de Petri p-temporais em que a representação de tempo no modelo é explícita, o problema é muito mais complexo. Nas redes de Petri p-temporais, os conflitos são visíveis durante um intervalo de tempo e não somente num dado instante. Assim, é possível gerar mais sequências de disparos que podem levar a melhores soluções.

A seguir, são apresentadas as definições necessárias para entender a noção de conflito de uma rede de Petri p-temporal:

- **intervalo de sensibilização de uma transição:** o intervalo de sensibilização de uma transição  $t$  é o intervalo resultante da interseção dos intervalos de visibilidade das fichas que participam do disparo de  $t$ ;
- **intervalo de conflito:** sendo  $t_1$  e  $t_2$  duas transições que estão em conflito efetivo, o intervalo de conflito é definido pela interseção dos intervalos de sensibilização de  $t_1$  e  $t_2$ .
- **conflito estrutural:** duas transições  $t_1$  e  $t_2$  estão em conflito estrutural se ambas possuem, pelo menos, um lugar de entrada em comum;
- **conflito efetivo:** duas transições estão em conflito efetivo, no contexto de uma rede de Petri p-temporal, quando ambas estão em conflito estrutural e quando existe um intervalo de conflito (quando a interseção dos seus intervalos de sensibilização é diferente de vazio).

Considerando-se, por exemplo, apenas dois casos a serem tratados pelo processo do *Serviço de Reclamações* apresentado no estudo de caso; considerando-se também que o caso 1 é iniciado na data 0 e o caso 2 é iniciado na data 40; a figura 5.10 ilustra um fragmento do processo do *Serviço de Reclamações* considerado neste exemplo em particular.

No fragmento de rede de Petri p-temporal da figura 5.10, o recurso compartilhado R2 está envolvido em conflito. O intervalo de conflito associado aos pares de transições

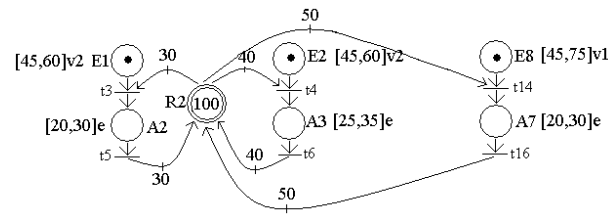


Figura 5.10: Fragmento de rede de Petri p-temporal envolvido em conflito

$(t3, t14)$  e  $(t4, t14)$  são dados pelas interseções dos intervalos de visibilidade  $[45,60]v2 \cap [45,75]v1 = [45,60]_{(t3, t14)}$  e  $[45,60]v2 \cap [45,75]v1 = [45,60]_{(t4, t14)}$ . Isso significa que um conflito efetivo entre  $t3$  e  $t14$  e entre  $t4$  e  $t14$  pode ocorrer durante o intervalo de datas  $[45,60]$ .

Quando um conflito estrutural é detectado entre duas ou mais transições, um mecanismo de tomada de decisão deve verificar, em um nível local, as consequências de uma decisão em particular. Por exemplo, considerando-se o fragmento de rede de Petri da figura 5.10, parece normal disparar  $t14$  tão logo seja possível, já que os mecanismos de simulação geralmente tentam disparar as transições assim que elas se tornam habilitadas. Se é tomada a decisão de disparar  $t14$  na data  $\delta=45$ , então, uma ficha aparece em  $A7$  com o intervalo de visibilidade  $[20+45, 30+45]=[65,75]v1$  e o número real (intensidade energética do recurso  $R2$ ) torna-se igual a  $100-50=50$ , como pode ser visto na figura 5.11

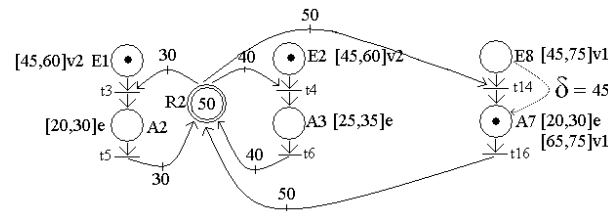


Figura 5.11: Disparo de  $t14$  na data  $\delta=45$

A partir dessa nova situação, é fácil perceber que uma violação de restrição temporal irá ocorrer em um dos intervalos de visibilidade associados às fichas em  $E1$  e  $E2$ . De fato, após o disparo de  $t14$ , se  $t3$  for disparada antes de  $t4$ , uma ficha será produzida no lugar  $A2$  com o intervalo de visibilidade  $[45+20, 45+30]=[65,75]v2$  e o número real (intensidade energética do recurso  $R2$ ) tornar-se-á  $50-30=20$ . Nesse caso, a transição  $t4$  não poderá ser disparada antes da data 60, que é a borda máxima do intervalo de visibilidade em  $E2$ , porque o número real (intensidade energético do recurso  $R2$ ) ficará igual a 20 (que é menos que o peso necessário para disparar  $t4$ ) até a data 65 (bordas mínimas do intervalo de visibilidade em  $A2$  e  $A7$ ) causando uma violação de restrição temporal. É possível mostrar que, após o disparo de  $t14$  na data  $\delta=45$ , se  $t4$  for disparada antes de  $t3$ , uma violação de restrição temporal ocorrerá da mesma maneira com o intervalo de visibilidade  $[45,60]v2$  em  $E1$ . Pode-se concluir, então, que o disparo de  $t14$  na data  $\delta=45$ , assim que a ficha em  $E8$  se tornar disponível, não deverá ser permitido pelo mecanismo de tomada de decisão.

A ferramenta que permite representar todas as evoluções possíveis de uma rede de Petri temporal (p-temporal ou t-temporal) é o grafo das classes [Khansa, 1997], [Menasche, 1982]. No trabalho de Oliveira [de Oliveira, 2005], é apresentado o princípio básico para gerar o grafo das classes, quando considerado as redes de Petri p-temporais.

Como foi apresentado no capítulo 4, um dos objetivos da abordagem proposta neste trabalho é o de aplicar o Raciocínio Energético no contexto do problema de escalonamento tempo real dos sistemas de gerenciamento de *workflow*, a fim de evitar combinações de conflitos que poderão ser resolvidos pelo jogador de redes de Petri p-temporal. Contudo, em alguns casos, o Raciocínio Energético pode não ser eficaz nesse sentido. Tomando-se, como exemplo particular, o conflito apresentado na figura 5.10, é possível verificar que não há atualização das datas de início de nenhum dos dois casos envolvidos no conflito utilizando a técnica do Raciocínio Energético. Isso pode ser explicado pelo fato de se trabalhar com uma técnica de Inteligência Artificial em que a quantidade de restrições envolvidas no problema pode influenciar nos resultados. Em particular, no exemplo da figura 5.10, apenas dois casos relacionados a três atividades estão sendo tratados pelo processo de *workflow*. Aumentando-se a quantidade de casos e atividades, aumenta-se também a quantidade de restrições envolvidas no problema, sendo possível, dessa maneira, conseguir melhores resultados com a técnica do Raciocínio Energético.

Uma das técnicas que permite a execução dinâmica das redes de Petri p-temporais é o algoritmo do jogador de rede de Petri p-temporal [de Oliveira, 2005], [Jeske, 2006], [Júlia *et al.*, 2007], que pode ser usado para a resolução do problema de escalonamento de sistemas de gerenciamento de *workflow* em tempo real. Como foi apresentado no capítulo 3, o jogador de rede de Petri p-temporal trata-se de um mecanismo de inferência especializado que permite a simulação de uma rede de Petri p-temporal. Quando o modelo é baseado no modelo de rede de Petri p-temporal, o algoritmo do jogador deve resolver as situações de conflito em tempo real de forma a não permitir a possibilidade de violação de restrição temporal que pode ser causada pela *morte* de uma ficha.

O jogador possui um calendário dos eventos (bordas mínimas e máximas de todos os intervalos de visibilidade) escalonados no tempo. A cada vez que uma borda mínima é alcançada, uma ficha torna-se disponível. Se essa ficha sensibilizar uma transição e se não houver conflito efetivo, a transição será disparada; caso contrário, o conflito será isolado e o mecanismo de tomada de decisões será ativado para verificar se a transição pode ser disparada na data mínima. Se a decisão for o disparo, uma nova marcação será gerada e novos intervalos de visibilidade serão definidos; caso contrário, será tratado o próximo evento do calendário. Se um limite máximo (borda máxima de um intervalo de visibilidade) for atingido, então a ficha *morrerá*, representando um comportamento não permitido. Nesse caso, não haverá garantia de que os prazos de entrega serão respeitados.

As restrições temporais (intervalos de visibilidade) calculadas por meio do mecanismo de propagação dos roteiros e refinadas por meio do Raciocínio Energético devem ser repassadas ao jogador de redes de Petri p-temporal. Essas restrições serão respeitadas pelo jogador de redes de Petri durante a elaboração de uma solução de escalonamento em tempo real.

# Capítulo 6

## Conclusão

Neste trabalho, foi apresentada uma proposta para solução do problema de escalonamento tempo real dos sistemas de gerenciamento de *workflow*. Mais especificamente a abordagem proposta consegue estabelecer restrições temporais de início de execução das atividades levando em consideração os prazos de entrega dos casos, as definições dos roteiros existentes e a disponibilidade dos recursos envolvidos. Posteriormente, uma abordagem baseada em um jogador de redes de Petri p-temporal utiliza as restrições calculadas por meio do mecanismo de propagação de restrições nos roteiros e refinadas pelo Raciocínio Energético, a fim de encontrar uma sequência de execução das atividades que respeitem as restrições disjuntivas (mecanismos de alocação de recursos) e as restrições temporais (intervalos de datas).

Baseou-se a abordagem proposta considerando outras duas abordagens: uma relacionada à propagação de restrições temporais ao longo dos roteiros dos processos de *workflow* e a outra baseada num tipo de Raciocínio Energético que leva em consideração a disponibilidade dos recursos para execução das atividades.

A maior contribuição deste trabalho foi a melhoria da base de conhecimento que define o conjunto de restrições que um jogador de redes de Petri p-temporal deve respeitar durante a elaboração de uma solução de escalonamento em tempo real. Em particular, o Raciocínio Energético filtrou a base de conhecimento (calculada inicialmente por mecanismos de propagação de restrições baseados em grafos sem circuitos, que não considerava restrições do tipo disjuntivas) e conseqüentemente, melhorando os intervalos de espera (intervalos de visibilidade) entre as atividades dos processos de *workflow*.

Outro ponto que merece destaque foi que o Raciocínio Energético conseguiu pre-identificar escalonamentos inadmissíveis no contexto dos sistemas de gerenciamento de *workflow* devido à indisponibilidade dos recursos envolvidos. Isso sem necessitar de algoritmos de natureza combinatorial que possuem normalmente uma complexidade do tipo exponencial já que o problema de escalonamento é um problema NP-completo.

Forma resultados deste trabalho não só a escrita desta dissertação, mas também a escrita do artigo *Análise sob restrições baseada em um raciocínio energético para o problema*



de escalonamento tempo real de sistemas de gerenciamento de workflow, aceito pelo IX SBAI 2009 - *Simpósio Brasileiro de Automação Inteligente* que acontecerá em setembro de 2009 em Brasília.

Este trabalho utilizou também os recursos de programação por restrições da linguagem *Prolog* que permitiu a implementação de um protótipo da abordagem proposta. Os métodos de propagação de restrições proposto por Oliveira [de Oliveira, 2005] foram formalizados para serem diretamente expressados na forma de restrições, sendo assim passíveis de implementação. Durante o desenvolvimento do protótipo, observou-se que o compilador *Prolog* não consegue trabalhar com um grande número de variáveis. Dependendo do tamanho do processo de *workflow* a ser analisado e do número de casos a serem tratados, provavelmente, será necessário reescrever os algoritmos de propagação de restrições em uma linguagem mais eficiente como C++, por exemplo.

Como trabalho futuro é possível integrar o mecanismo de propagação de restrições ao jogador de redes de Petri p-temporal implementado por Ramos [Júlia *et al.*, 2007] e utilizá-lo para a gestão de projeto envolvendo produção de *software*. O *software* proposto poderia, por exemplo, servir para monitorar em tempo real o ciclo de execução de um processo de desenvolvimento do tipo RUP (*Rational Unified Process*).

# Referências Bibliográficas

- [Aalst & Hee, 2002] Aalst, W. V. D., & Hee, K. V. 2002. *Workflow Management: Models, Methods and Systems*. The MIT Pres.
- [Aalst *et al.*, 1994] Aalst, Wil Van Der, Hee, Kees Van, & Houben, Geert-Jan. 1994. Modelling workflow management systems with high-level Petri nets. *Second Workshop on Computer-Supported Cooperative Work, Petri nets and related formalisms. Proceedings*, 31–50.
- [Ackoff & Sasieni, 1977] Ackoff, Russel L., & Sasieni, Maurice W. 1977. *Pesquisa Operacional*. Livros Tec. e Científicos.
- [Alonso *et al.*, 1997] Alonso, G., Agrawal, D., Abbadi, A., Abbadi, A. El, & Mohan, C. 1997. Functionality and limitations of current Workflow Management Systems. *Submitted to IEEE Expert*.
- [Applegate & Cook, 1991] Applegate, D., & Cook, W. 1991. A computational study of job shop scheduling. *Research report CMU-CS*, 90–145.
- [Bacelli *et al.*, 1992] Bacelli, L. F., Cohen, G., Olsder, G. J., & Quadrat, J. P. 1992. *Synchronization and Linearity: an algebra for discrete event systems*.
- [Carlier & Pinson, 1994] Carlier, J., & Pinson, E. 1994. Adjustment of heads and tails for the job shop scheduling problem. *European Journal of Operational Research*, **78**, 146–161.
- [Casati *et al.*, 1998] Casati, F., Ceri, S., Pernici, B., & Pozzi, G. 1998. Workflow Evolution. *Special issue on ER'96*, **24**, 211–238.
- [de Oliveira, 2005] de Oliveira, Fernanda Francielle. 2005. *Problema do Escalonamento em Tempo Real dos Sistemas de Gerenciamento de Workflow baseado em um modelo de Rede de Petri híbrida p-temporal*. Dissertação de Mestrado - Faculdade de Computação. Universidade Federal de Uberlândia.
- [del Foyo & Silva, 2003] del Foyo, P. G., & Silva, J. R. 2003. Towards a unified view of Petri nets and object oriented modeling. *17 th International Congress of Mechanical Engineering. Proceedings of COBEM*.
- [Deriz, 2007] Deriz, A. Claudia. 2007. *Um método de busca usando algoritmo genético para programação reativa da produção de sistemas de manufatura com recursos compartilhados*. Dissertação de Mestrado - Universidade Federal de São Carlos.

- [Dumas & Hofstede, 2001] Dumas, Marlon, & Hofstede, Arthur H. M. Ter. 2001. UML Activity Diagrams as a Workflow Specification Language. *Proceedings fo the 4th International Conference on The Unified Modeling Language, Modeling Languages, Concepts, and Tools*, 76–90.
- [Erschler & Lopez, 1990] Erschler, J., & Lopez, P. 1990. Energy based approach for task scheduling under time and resource constraints. *Second International Workshop on Project Management and Scheduling*, 115–121.
- [Erschler *et al.*, 1991] Erschler, J., Lopez, P., & Thuriot, C. 1991. Raisonnement temporel sous contraintes de ressources et problèmes d’ordonnancement. *Revue d’Intelligence Artificielle*, **5**(3), 7–36.
- [Eshuis, 2002] Eshuis, H. 2002. *Semantics and Verification of UML Activity Diagramas for Workflow Modelling*. Tese de Doutorado - University of Twente.
- [Esquirol, 1987] Esquirol, P. 1987. *Règles et processus d’inférence pour l’aide à l’ordannancement de tâches en présence de contraintes*. Tese de Doutorado - Université Paul Sabatier.
- [Esquirol & Lopez, 1999] Esquirol, P., & Lopez, P. 1999. *L’ordonnancement*. Paris.
- [Esquirol *et al.*, 1995] Esquirol, P., Huguet, M. J., & Lopez, P. 1995. Modeling and managing disjunctions in scheduling problems. *Journal of Intelligent Manufacturing* **6**, 133–144.
- [Fargier, 1994] Fargier, H. 1994. *Problèmes de satisfaction de contraintes flexibles applications à l’ordannancement de production*. Tese de Doutorado - I’Universtié Paul Sabatier de Toulouse.
- [Gil *et al.*, 2004] Gil, Y., Deelman, E., Blythe, J., Kesselman, C., & Tangmunarunkit, H. 2004. Artificial Intelligence and Grids: Workflow Planning and Beyond. *IEEE Intelligent Systems*, **19**(1), 26–33.
- [Gondran & Minox, 1984] Gondran, M., & Minox, M. 1984. *Graphes et algorithmes*.
- [Hennet, 2002] Hennet, Jean-Claude. 2002. *Concepts et outils pour les systèmes de production*.
- [Hollingsworth, 1985] Hollingsworth, D. 1985. *The Workflow Reference Model*. Tech. rept. Workflow Management Coalition - TC00-1003.
- [Hruby, 1998] Hruby, P. 1998. Specification of Workflow Management Systems with UML. *Workshop on Implementation and Application of Object-oriented Workflow Management Systems, Proccedings of the 1998 OOPSLA*.
- [Jeske, 2006] Jeske, Joslaine Cristina. 2006. *Mecanismo de alocação de recurso fuzzy para sistemas de gerenciamento de workflow*. Dissertação de Mestrado - Faculdade de Computação. Universidade Federal de Uberlândia.
- [Júlia *et al.*, 2007] Júlia, S., Valette, R., Junior, S. B., dos S. Soares, M., Jeske, J. C., de Oliveira, F. F., & Ramos, A. L. B. 2007. *Uma abordagem UML/Redes de Petri para a verificação de cenários de Sistemas Tempo Real*. Tech. rept. Fundação de Amparo à Pesquisa do Estado de Minas Gerais - Auxílio financeiro.

- [Julia & dos Santos Soares, 2003] Julia, S., & dos Santos Soares, M. 2003. Verification of real time UML specifications through a specialized inference mechanism based on a token player algorithm and the sequent calculus of linear logic. *Proceedings of the 15th European Simulation Symposium*, 65–70.
- [Julia & Valette, 2000] Julia, S., & Valette, R. 2000. Real time scheduling of batch Systems. *Simulation Practice and Theory. Elsevier Science*, 307–319.
- [Khansa, 1997] Khansa, W. 1997. *Réseaux de Petri p-temporels Contribution a l'étude des systemes a événements discrets*. Tese de Doutorado - Université de Savoie, França.
- [Lacomme *et al.*, 2003] Lacomme, P., Prins, C., & Sevaux, M. 2003. *Algorithmes de graphes*.
- [Lee & DiCesare, 1994] Lee, D. Y., & DiCesare, F. 1994. Scheduling flexible manufacturing systems using Petri nets and heuristic search. *IEEE Transactions on Robotics and Automation*, 123–132.
- [Li *et al.*, 2003] Li, J., Fran, Y., & Zhou, M. 2003. Timing Constraint Workflow Nets for Workflow Analysis. *IEEE Transactions on Systems, Man and Cybernetics. Part A: Systems and Humans*, **33**(2).
- [Lopez & Esquirol, 1996] Lopez, Pierre, & Esquirol, Patrick. 1996. Consistency Enforcing in scheduling: A general formulation based on energetic reasoning. *5th International Workshop on Project Management and Scheduling*.
- [Luders, 2001] Luders, R. 2001. *Controle Multivariável de Sistemas a Eventos Discretos em Dióides*. Tese de Doutorado - Universidade de Campinas - UNICAMP.
- [Malcom *et al.*, 1959] Malcom, D. G., Roseboom, J. H., Clark, C. E., & Fazar, W. 1959. Application of a technique for research and development programme evaluation. *Ops Res.* 7, 646–669.
- [Marriott & Stuckey, 1998] Marriott, K., & Stuckey, Peter J. 1998. *Programming with Constraints: an Introduction*.
- [Meidanis *et al.*, 1996] Meidanis, J., Vossen, G., & Weske, M. 1996. Using Workflow Management in DNA Sequencing. *In: Proceedings: First International Conference on Cooperative Information Systems*.
- [Menasche, 1982] Menasche, M. 1982. *Analyse des Réseaux de Petri Temporisés et Application aus Systèmes Distribués*. Tese de Doutorado - Université Paul Sabatier.
- [Merlin, 1974] Merlin, P. 1974. *A Study of Recoverability of Computer Systems*. Tese de Doutorado - University of California.
- [Moraes, 2000] Moraes, C. M. 2000. *Escalonamento de sistemas de produção híbridos usando uma rede de Petri p-temporal com mecanismo de retrocesso inteligente*. Dissertação de Mestrado - Faculdade de Computação. Universidade Federal de Uberlândia.
- [Nicolao, 1998] Nicolao, M. 1998. *Modelagem de Workflow utilizando um Modelo de Dados Temporal Orientado a Objetos com Papéis*. Dissertação de Mestrado - Universidade Federal do Rio Grande do Sul.

- [Passos, 2009] Passos, Lígia Maria Soares. 2009. *Formalização de workflow nets utilizando lógica linear: Análise qualitativa e quantitativa*. Dissertação de Mestrado - Faculdade de Computação. Universidade Federal de Uberlândia.
- [Plesums, 2002] Plesums, Charles. 2002. *Workflow Handbook*.
- [Ramchandani, 1974] Ramchandani, C. 1974. *Analysis of asynchronous concurrent systems by timed Petri net Models*. Tese de Doutorado - MIT.
- [Reyes *et al.*, 2001] Reyes, A., Yu, H., & Lloyd, S. 2001. An evolutionary hybrid scheduler based in petri net structures for FMS scheduling. *IEEE International Conference on Systems Man and Cybernetics*, **4**, 2516–2521.
- [Roy, 1960] Roy, B. 1960. Contribution de la théorie des graphes à l'études des problèmes d'ordannancement. *Les problèmes d' ordonnancement, applications et methodes*, 109–125.
- [Senkul & Toroslu, 2005] Senkul, P., & Toroslu, Ismail H. 2005. An architecture for workflow scheduling under resource allocation constraints. *Elsevier - Information Systems*, **30**, 399–422.
- [Sifakis, 1977] Sifakis, J. 1977. Use of Petri nets for performance evaluation. *In: Third International Symposium on Modeling and Performance Evaluation of Computer Systems*.
- [Silva & Vallete, 1989] Silva, M., & Vallete, R. 1989. Petri nets and Flexible Manufacturing. *Advances in Petri nets (G. Rozemberg, Ed.), Lecture Notes in Computer Science. (Springer Verlag)*.
- [Takahashi *et al.*, 1996] Takahashi, K., Yamamura, M., & Kobayashi, S. 1996. A GA approach to solving reachability problems for petri nets. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, **79-A**(11), 1774–1780.
- [Tazza, 1987] Tazza, M. 1987. Quantitative Analysis of a resource allocation problem: a net theory based proposal. *Springer-Verlag*, 511–532.
- [Tramontina *et al.*, 2004] Tramontina, G. B., Wainer, J., & Ellis, C. 2004. Applying Scheduling Techniques to Minimize the Number of Late Jobs in Workflow Systems. *In: ACM Symposium on Applied Computing*.

# Livros Grátis

( <http://www.livrosgratis.com.br> )

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)  
[Baixar livros de Literatura de Cordel](#)  
[Baixar livros de Literatura Infantil](#)  
[Baixar livros de Matemática](#)  
[Baixar livros de Medicina](#)  
[Baixar livros de Medicina Veterinária](#)  
[Baixar livros de Meio Ambiente](#)  
[Baixar livros de Meteorologia](#)  
[Baixar Monografias e TCC](#)  
[Baixar livros Multidisciplinar](#)  
[Baixar livros de Música](#)  
[Baixar livros de Psicologia](#)  
[Baixar livros de Química](#)  
[Baixar livros de Saúde Coletiva](#)  
[Baixar livros de Serviço Social](#)  
[Baixar livros de Sociologia](#)  
[Baixar livros de Teologia](#)  
[Baixar livros de Trabalho](#)  
[Baixar livros de Turismo](#)