João Vitor Torres

**Representações Compactas para
Processos de Decisão de Markov e Sua
Aplicação na Administração de
Impressoras**

Dissertação apresentada à Escola
Politécnica da Universidade de São
Paulo para obtenção do Título de
Mestre em Engenharia.

João Vitor Torres

Representações Compactas para Processos de Decisão de Markov e Sua Aplicação na Administração de Impressoras

Dissertação apresentada à Escola
Politécnica da Universidade de São
Paulo para obtenção do Título de
Mestre em Engenharia.

Área de concentração:
Controle e Automação Mecânica

Orientador:
Prof. Dr.
Fabio Gagliardi Cozman

Este exemplar foi revisado e alterado em relação à versão original, sob responsabilidade única do autor e com anuência de seu orientador.

São Paulo, 12 de junho de 2006.

Assinatura do autor

Assinatura do orientador

FICHA CATALOGRÁFICA

Torres, João Vitor

Representações compactas para processos de decisão de Markov e sua aplicação na administração de impressoras / J.V. Torres. – São Paulo, 2006.

83 p.

Dissertação (Mestrado) — Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos.

1.Processos de decisão de Markov 2.Inteligência artificial
I. Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos II.t.

Aos meus pais, Edílio e Valdete, e aos meus familiares, os quais me deram grande apoio e incentivo durante toda minha vida acadêmica.

Agradecimentos

Agradeço ao meu orientador, Prof. Dr. Fabio Gagliardi Cozman, pelas importantes orientações recebidas durante a condução deste trabalho, por ter me orientado pelos caminhos da pesquisa, e por inspirar grande confiança em seus alunos.

Agradeço a HP-Brasil, ao Instituto de Pesquisas Eldorado e à FDTE pelo apoio financeiro prestado através da Bolsa de Mestrado.

Agradeço ao Engenheiro André Rodrigues, membro da equipe do projeto na HP-Brasil R&D, pelas diversas sugestões fornecidas ao longo do desenvolvimento do projeto.

Agradeço à equipe da T&T Engenharia pela grande colaboração desenvolvendo um simulador para o grupo de impressoras.

Agradeço à Prof. Dra. Leliane Nunes de Barros e ao Prof. Dr. Newton Maruyama pelas sugestões recebidas no Exame de Qualificação.

Finalmente, agradeço à minha família e a todos os amigos que indiretamente me ajudaram na conclusão deste trabalho de mestrado.

Resumo

Os Processos de Decisão de Markov (PDMs) são uma importante ferramenta de planejamento e otimização em ambientes que envolvem incertezas. Contudo a especificação e representação computacional das distribuições de probabilidades subjacentes a PDMs é uma das principais dificuldades de utilização desta ferramenta. Este trabalho propõe duas estratégias para representação destas probabilidades de forma compacta e eficiente. Estas estratégias utilizam redes Bayesianas e regularidades entre os estados e as variáveis. As estratégias apresentadas são especialmente úteis em sistemas onde as variáveis têm muitas categorias e possuem forte inter-relação. Além disso, é apresentada a aplicação destes modelos no gerenciamento de grupos de impressoras (um problema real da indústria e que motivou o desenvolvimento do trabalho) permitindo que estas atuem coletiva e não individualmente. O último tópico discutido é uma análise comparativa da mesma aplicação utilizando Lógica Difusa.

Abstract

Markov Decision Processes (MDPs) are an important tool for planning and optimization in environments under uncertainty. The specification and computational representation of the probability distributions underlying MDPs are central difficulties for their application. This work proposes two strategies for representation of probabilities in a compact and efficient way. These strategies use Bayesian networks and regularities among states and variables. The proposed strategies are particularly useful in systems whose variables have many categories and have strong interrelation. This proposal has been applied to the management of clusters of printers, a real problem that in fact motivated the work. Markov Decision Processes are then used to allow printers to act as a group, and not just individually. The work also presents a comparison between MDPs and Fuzzy Logic in the context of clusters of printers.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas

Lista de Símbolos

1	Introdução	1
1.1	Contribuições	2
1.2	Motivação	2
1.3	Organização	4
2	Probabilidades e Redes Bayesianas	5
2.1	Introdução	5
2.2	Espaço de Possibilidades, Estados e Eventos	5
2.3	Variáveis	6
2.4	Medidas de Probabilidade	6
2.5	Medidas de Probabilidade Condicional	7
2.6	Esperança	7
2.7	Distribuições de Probabilidades	8
2.8	Redes Bayesianas	8
2.8.1	Conceitos Básicos	8
2.8.2	Inferência em Redes Bayesianas	10
3	Processos de Decisão de Markov	11

3.1	Introdução	11
3.2	Estados e Transições de Estados	11
3.3	Ações	16
3.4	Eventos Exógenos	17
3.5	Observações	18
3.6	Trajetória do Sistema e Histórico Observado	20
3.7	Recompensa e Valor	21
3.8	Horizontes e Critério de Sucesso	22
3.9	Critério de Solução	24
3.10	Políticas	25
3.11	Algoritmos de Solução Empregando Programação Dinâmica . . .	26
3.11.1	Iteração de Valor	27
3.11.2	Iteração de Política	28
4	Processos de Decisão de Markov Fatorados	31
4.1	Introdução	31
4.2	Representação dos Estados	32
4.3	Representação das Probabilidades de Transição de Estado	32
4.4	Funções Recompensa e Custo	36
4.5	Resolvendo o PDM	36
4.5.1	Solução Estática	37
4.5.2	Solução Dinâmica	38
4.6	Considerações	39
5	Aplicação de Processos de Decisão de Markov na Administração de Impressoras	41
5.1	Introdução	41
5.2	Funcionamento do Sistema	42

5.3	Modelagem do Sistema Utilizando PDMs: Solução Estática	43
5.3.1	O Espaço de Estados	43
5.3.2	Ações	44
5.3.3	Probabilidades de Transição de Estado	45
5.3.4	Função Recompensa	50
5.3.5	Função Custo	50
5.3.6	Resolvendo o Processo de Decisão de Markov (PDM) . . .	51
5.4	Modelagem do Sistema Utilizando PDMs: Solução Dinâmica . . .	52
5.4.1	O Espaço de Estados	53
5.4.2	Ações	54
5.4.3	Probabilidades de Transição de Estado	54
5.4.4	Funções Recompensa e Custo	57
5.5	Considerações	57
6	Resultados e Considerações Finais	59
6.1	Simulações	59
6.2	Considerações Finais	66
	Anexos	68
A	Um Contraponto Utilizando Lógica Difusa	68
A.1	Introdução	68
A.2	Conceitos Básicos	68
A.2.1	Conjunto Difuso	68
A.2.2	Variáveis Linguísticas	69
A.2.3	Regras Difusas	69
A.2.4	Inferência e Defuzzificação	69
A.3	Metodologia	70
A.4	Cálculo da Nota	70

A.4.1	Variáveis Linguísticas	70
A.4.2	Base de Regras Difusas	74
A.4.3	Um Exemplo de Aplicação das Regras	75
A.4.4	Análise dos Resultados	77
A.5	Considerações	78
Referências Bibliográficas		81

Lista de Figuras

2.1	Um exemplo de Rede Bayesiana (RB).	9
3.1	(a) Um processo estocástico genérico, (b) uma cadeia de Markov e (c) uma cadeia de Markov estacionária.	13
3.2	Um diagrama de transição de estados.	15
3.3	Grafo mostrando as relações de dependência entre ações, estados e observações.	19
3.4	Um processo de decisão com recompensas e custos para as ações.	22
4.1	Uma Rede Bayesiana para ação <i>Imprimir</i> (Exemplo 2).	34
5.1	(a) RB para ação <i>Não Imprimir</i> e (b) RB para ação <i>Imprimir</i> .	46
5.2	(a) RB para ação <i>Não Imprimir</i> e (b) RB para ação <i>Imprimir</i> .	54
6.1	Tempos para tarefas com solicitação de urgência.	65
6.2	Atendimento das solicitações de impressão colorida.	66
A.1	Função de pertinência da variável lingüística <i>Toner</i> .	71
A.2	Função de pertinência da variável lingüística <i>Papel</i> .	72
A.3	Função de pertinência da variável lingüística <i>Fila</i> .	72
A.4	Função de pertinência da variável lingüística <i>TonerColorido</i> .	73
A.5	Função de pertinência da variável lingüística <i>TamanhoJob</i> .	73
A.6	Função de pertinência da variável lingüística <i>Urgencia</i> .	73
A.7	Função de pertinência da variável lingüística <i>JobColorido</i> .	74
A.8	Função de pertinência da variável lingüística <i>Nota</i> .	74
A.9	Exemplo de aplicação das regras de inferência.	76
A.10	<i>Nota</i> em função da <i>Urgencia</i> e da <i>Fila</i> .	78
A.11	<i>Nota</i> em função do <i>Papel</i> e da <i>Urgencia</i> .	79
A.12	<i>Nota</i> em função da <i>Fila</i> e do <i>Toner</i> .	79

A.13 <i>Nota</i> em função do <i>TonerColorido</i> e do <i>JobColorido</i>	80
--	----

Lista de Tabelas

2.1	Relações entre as variáveis da rede Bayesiana da Figura 2.1.	9
3.1	Matriz de probabilidades de transição de estados.	15
4.1	Matriz de probabilidades de transição de estado.	31
4.2	Mapeamento entre o estado s do sistema e as k -tuplas que o caracterizam.	33
5.1	Variáveis do sistema e número de categorias de cada uma.	44
5.2	Exemplos de particionamento das variáveis.	45
5.3	Probabilidades condicionais para variável q	48
5.4	Probabilidades condicionais para variável tr	48
5.5	Probabilidades condicionais para variável tb	49
5.6	Probabilidades condicionais para variável tc	49
5.7	Probabilidades condicionais para variável v	49
5.8	Variáveis binárias e correspondência com a antiga discretização. .	53
5.9	Probabilidades condicionais para variável $\hat{q}$	55
5.10	Probabilidades condicionais para variável $\hat{tr}$	55
5.11	Probabilidades condicionais para variável $\hat{tb}$	56
5.12	Probabilidades condicionais para variável $\hat{tc}$	56
6.1	Resultados da simulação com 10 usuários e 4 impressoras de diferentes velocidades.	61
6.2	Resultados da simulação com 15 usuários e 4 impressoras de diferentes velocidades.	62
6.3	Resultados da simulação com 10 usuários e 4 impressoras de velocidades iguais.	63
6.4	Resultados da simulação com 15 usuários e 4 impressoras de velocidades iguais.	64

Lista de Abreviaturas

IA Inteligência Artificial

PPDDL Probabilistic Planning Domain Definition Language

PDM Processo de Decisão de Markov

PDMs Processos de Decisão de Markov

PDMCO PDM Completamente Observável

PDMCOs PDMs Completamente Observáveis

PDMNO PDM Não Observável

PDMNOs PDMs Não Observáveis

PDMPO PDM Parcialmente Observável

PDMPOs PDMs Parcialmente Observáveis

PO Pesquisa Operacional

RB Rede Bayesiana

RBs Redes Bayesianas

RBD Rede Bayesiana Dinâmica

RBDs Redes Bayesianas Dinâmicas

SMA Sistemas Multiagentes

Lista de Símbolos

- a Uma ação do conjunto de ações de um PDM
- $A, B, C, \dots$ Subconjuntos de Ω ou eventos
- A^c Complemento do conjunto A em relação a Ω
- A_P Ação executada no estado presente de um PDM
- A^t Ação executada no estágio t de um PDM
- $ch(X_i)$ Conjunto dos filhos de X_i
- CA Conjunto de ações de um PDM
- CA_s^t Conjunto aplicável de ações no estado s e estágio t
- $C(s, a)$ Função custo de um PDM
- $E(.)$ Função esperança
- h Um histórico do sistema, elemento do conjunto H_S
- H_S Conjunto de históricos do sistema em um PDM
- H_o Conjunto de históricos observáveis em um PDM
- Imprimir* Ação *Imprimir* do modelo de impressora usando PDMs
- jc Variável de estado discreta da requisição de cor para tarefa de impressão
- $\hat{jc}$ Variável de estado binária da requisição de cor para tarefa de impressão
- Jc Requisição de cor para tarefa de impressão (em %)
- js Variável de estado discreta do tamanho da tarefa de impressão
- $\hat{js}$ Variável de estado binária do tamanho da tarefa de impressão
- Js Tamanho da tarefa de impressão (em páginas)
- ju Variável de estado discreta da urgência da tarefa de impressão

$\hat{j}u$ Variável de estado binária da urgência da tarefa de impressão

Ju Requisição de urgência para tarefa de impressão (em %)

k_i Fator de normalização das expressões de probabilidades

$pa(X_i)$ Conjunto dos pais de X_i

$p_{i,j,k}^h$ Probabilidade da observação o_h dado que o agente executou a ação a_k no estado s_i e chegou no estado s_j

p_{ij} Probabilidade estacionária de transição do estado s_i para o estado s_j

p_{ij}^t Probabilidade de transição do estado s_i no estágio t para o estado s_j no estágio $t + 1$

p_i^0 Probabilidade do estado inicial ser s_i

$p(.)$ Função distribuição de probabilidade

P^t Matriz de probabilidades de transição de estado no estágio t contendo os valores p_{ij}^t

P^0 Vetor de probabilidades para o estado inicial contendo os valores p_i^0

$P(.)$ Função probabilidade

$P(. | .)$ Função probabilidade condicional

P1, P2 e P3 Axiomas da teoria de probabilidades

N Dimensão do espaço de estados em um PDM

Não Imprimir Ação *Não Imprimir* do modelo de impressora usando PDMs

o_h Uma observação do conjunto de observações O

O Conjunto de observações de um PDM

O^t Observação obtida no estágio t

$O(.)$ Ordem de complexidade

q Variável de estado discreta da ocupação da fila de impressão

$\hat{q}$ Variável de estado binária da ocupação da fila de impressão

Q Nível de ocupação da fila de impressão (em %)

$\mathbf{R}(s)$ Função recompensa de um PDM

s Um estado do conjunto de estados de um PDM

$\mathbf{spo}(\mathbf{X}_i)$ Conjunto dos esposos de X_i

$\mathbf{S}$ Conjunto de estados de um PDM

$\mathbf{S}_F$ Estado futuro de um PDM

$\mathbf{S}_P$ Estado presente de um PDM

$\mathbf{S}^t$ Variável aleatória indicadora do estado de um PDM no estágio t

t Estágio de um PDM

$\mathbf{T}$ Conjunto de estágios em um PDM

tb Variável de estado discreta do nível de tinta preta

$\hat{tb}$ Variável de estado binária do nível de tinta preta

Tb Nível de tinta preta (em %)

tc Variável de estado discreta do nível de tinta colorida

$\hat{tc}$ Variável de estado binária do nível de tinta colorida

Tc Nível de tinta colorida (em %)

tr Variável de estado discreta do nível de papel

$\hat{tr}$ Variável de estado binária do nível de papel

Tr Nível de papel (em %)

$\mathbf{T}(s, s', a)$ Distribuição de probabilidades de transição de estados de um PDM

v Variável de estado discreta da velocidade de impressão

V Velocidade de impressão (em páginas/min)

v_i Variável de estado de um PDM

v_{iF} Variável de estado de um PDM no estado futuro

v_{iP} Variável de estado de um PDM no estado presente

$V(h)$ Função valor de um histórico h

$V_n(s)$ Função valor de um estado s em um PDM de horizonte infinito após n iterações

$V_t^\pi(s)$ Função valor de uma política π com t estágios por atravessar no estado s

$V^\pi(s)$ Função valor de uma política de horizonte infinito π no estado s

$V_T^*(s)$ Função valor de uma política ótima com $|T|$ estágios por atravessar no estado s

$V^*(s)$ Função valor de uma política ótima de horizonte infinito no estado s

$VE(.)$ Função valor esperado

ω Um estado do espaço de possibilidades Ω

x Um valor real

$X(\omega)$ e $Y(\omega)$ Variáveis aleatórias

$\mathbf{X}$ Um conjunto de variáveis

$\mathbf{X_E}$ Conjunto das variáveis observadas

$\mathbf{X_q}$ Conjunto das variáveis incógnita

$\mathbf{X} \setminus \mathbf{Y}$ Conjunto de variáveis que pertencem a $\mathbf{X}$ e não pertencem a $\mathbf{Y}$

$\mathbf{X_i}$ Uma variável do conjunto $\mathbf{X}$

Ω Espaço de possibilidades

$\Omega_{\mathbf{X}}$ Espaço de possibilidades da variável aleatória X

$\cap$ Intersecção de eventos probabilísticos

$\cup$ União de eventos probabilísticos

$\sum$ Somatório

$\prod$ Produtório

$\in$ Pertence a

$\Re$ Conjunto dos números reais

$\emptyset$ Conjunto vazio

γ Taxa de desconto

π Uma política em um PDM

$\pi(s)$ Ação fornecida pela política estacionária π no estado s

$\pi(s, t)$ Ação fornecida pela política π no estado s e estágio t

$|\cdot|$ Função dimensão ou número de elementos de um conjunto

1 Introdução

O problema de tomada seqüencial de decisões em um ambiente estocástico, dinâmico e com utilidades associadas é a base de aplicação de Processos de Decisão de Markov (PDM). Esta ferramenta é utilizada em diversas áreas de pesquisa, como teoria de decisão, pesquisa operacional e planejamento (BOUTILIER; DEAN; HANKS, 1999; PUTERMAN, 1994).

Um PDM é uma estrutura constituída basicamente por cinco elementos: um espaço de estados S , um conjunto de ações CA , uma função custo $C(s, a)$ que depende do estado s e da ação a , uma função recompensa $R(s)$ que depende do estado e uma distribuição de probabilidades de transição de estados associada a cada ação $T(s, s', a)$.

O uso desta estrutura permite que um agente planeje o seu comportamento em um ambiente que envolve incertezas de forma otimizada. Para encontrar este plano o agente deve determinar qual é a melhor ação a se executar em cada estado possível, este plano também é chamado de política ótima.

Contudo, o modelo de PDM encontra duas limitações importantes em problemas de maior escala.¹ A primeira dificuldade está na especificação das probabilidades que regem o sistema. Quanto maior a escala, maior será a quantidade de informações necessárias para se especificar as probabilidades. A segunda dificuldade é encontrar uma política ótima. Os algoritmos tradicionais não são eficientes em PDMs com grandes espaços de estados.

Por isso, faz-se necessária a representação dessas probabilidades de forma compacta. Uma maneira conhecida de se fazer essa representação é através de Redes Bayesianas (RBs) (JENSEN, 2001; RUSSEL; NORVIG, 2002), onde explora-se as independências entre as variáveis do sistema de forma a reduzir a quantidade de informações a especificar. Esses modelos são conhecidos na literatura como PDMs fatorados (GUESTRIN et al., 2003).

Porém, apesar de reduzir significativamente a quantidade de informações necessá-

¹Aqui o termo escala denota o tamanho do espaço de estados do problema.

rias para se especificar a distribuição de probabilidades do problema, em muitos casos só o uso de Redes Bayesianas (RBs) ainda não viabiliza o uso de PDMs. Isto ocorre nos casos onde as distribuições de probabilidades condicionais usadas nas RBs possuem muitas informações e a especificação destas distribuições na forma de números é extremamente trabalhosa.

1.1 Contribuições

A primeira contribuição deste trabalho é utilizar RBs combinadas com propriedades entre as variáveis e os estados do PDM na especificação das probabilidades de transição de estado. Através destas propriedades as probabilidades são especificadas de forma compacta usando relações entre as próprias variáveis do sistema. Este método é especialmente útil em sistemas onde as variáveis têm muitas categorias e possuem forte inter-relação.

A segunda contribuição está nos métodos desenvolvidos para solucionar PDMs modelados através das duas estratégias apresentadas neste trabalho. Estes métodos surgem como um importante subproduto das técnicas desenvolvidas.

A terceira contribuição é que todo o estudo realizado foi voltado para um problema real da automação comercial. O trabalho foi realizado em conjunto com especialistas da empresa HP-Brasil, buscando uma solução descentralizada para o problema de alocação de tarefas de impressão entre um conjunto de impressoras conectadas. O objetivo é fazer com que as impressoras cooperem para que a satisfação dos seus usuários seja maximizada.

1.2 Motivação

A pesquisa por uma solução descentralizada para a alocação de tarefas de impressão se justifica no fato de atualmente existirem apenas soluções centralizadas para a alocação deste tipo de serviço. Este tipo de solução depende de um elemento central que coordena o funcionamento de todas as impressoras no grupo e uma falha neste elemento compromete o funcionamento de todo o grupo.

Para se chegar a uma solução descentralizada foram criados modelos de impressora e não de grupos de impressoras. Assim, o grupo de impressoras pode ser visto como um conjunto de agentes, cada um modelando uma impressora, tentando cooperar na execução de tarefas de impressão. Esta situação constitui um problema de Sistemas Multiagentes (SMA) (ALVARES; SICHMAN, 1997). O

método escolhido para fazer os agentes cooperarem foi um esquema de leilão. A cada tarefa de impressão, cada impressora gera uma nota para a execução da tarefa e a impressora com maior nota ganha o leilão desta tarefa.

Diferentes modelos de agente foram construídos para se representar as impressoras. Cada um destes modelos emprega técnicas diferentes na geração dos lances para os leilões. Os modelos construídos e testados são listados abaixo:

- PDM Solução Estática: Este modelo é apresentado no Capítulo 5;
- PDM Solução Dinâmica: Veja o modelo no Capítulo 5;
- Lógica Difusa: Como o próprio nome sugere, este é baseado em Lógica Difusa (ZADEH, 1973; LEE, 1990a; LEE, 1990b);
- Heurística: Este modelo é extremamente simples e é discutido no Capítulo 6.

O uso de PDMs para representar cada uma das impressoras participando como agentes em um leilão permite que cada impressora planeje individualmente sua ação, imprimir ou não uma tarefa, e pondere o seu lance de forma otimizada considerando seu estado atual, a tarefa atual e as possíveis consequências de suas ações. Ao mesmo tempo o esquema de leilão garante que todas as tarefas serão executadas por alguma e só uma impressora do grupo, mesmo que todas as impressoras escolham a ação imprimir ou todas escolham a ação não imprimir.

Em contraste com o uso de PDMs foi construído um modelo de impressora usando Lógica Difusa. Os PDMs se baseiam na teoria de probabilidades e a Lógica Difusa é uma técnica que também trata problemas envolvendo incertezas, mas não emprega a teoria de probabilidades. Assim, a Lógica Difusa foi utilizada para gerar um lance a cada leilão modelando as incertezas no comportamento dinâmico de uma impressora.

O modelo chamado Heurística serviu como um patamar inferior em comparações de desempenho com os outros modelos. Este modelo foi considerado um patamar inferior, pois não leva em consideração as incertezas no comportamento dinâmico de uma impressora e consiste apenas em uma soma dos recursos disponíveis na impressora.

Outro aspecto a se destacar trata das dificuldades de modelagem de cada uma das técnicas. Observamos de maneira prática o caráter empírico da modelagem através de Lógica Difusa, cuja dificuldade principal é ajustar os parâmetros do modelo. Enquanto isso, a modelagem através de PDMs esbarra na especificação

das probabilidades de transição de estado. Porém, a dificuldade desta última está em como representar estas probabilidades de forma compacta e não em que valores de probabilidades representar.

1.3 Organização

Os temas apresentados acima são discutidos no restante do texto, que se organiza da forma a seguir. No Capítulo 2 é apresentada uma breve revisão de alguns conceitos de probabilidades e de RBs. No Capítulo 3, apresenta-se a estrutura de PDMs em detalhes. No Capítulo 4, mostra-se como unir regularidades entre os estados e as variáveis a RBs na especificação de probabilidades para aplicação em PDMs. No Capítulo 5, é apresentada uma aplicação para a técnica estudada e que motivou o desenvolvimento do trabalho. O Capítulo 6 contém resultados e considerações finais. No Anexo A, mostra-se um contraponto, apresentando o modelo de impressora desenvolvido para a mesma aplicação através de Lógica Difusa.

2 Probabilidades e Redes Bayesianas

2.1 Introdução

Neste capítulo apresenta-se uma breve revisão de probabilidades e RBs. A abordagem de probabilidades apresentada neste texto é subjetivista (PEARL, 2000). Nesta abordagem, as probabilidades são expressões da ignorância, incerteza e preferência de quem as especifica. Isso permite que pessoas diferentes estabeleçam probabilidades diferentes para o mesmo evento.

As redes Bayesianas são representações compactas e eficientes para distribuições de probabilidades conjuntas, podendo representar modelos estatísticos potencialmente grandes por um pequeno número de valores de probabilidade (JENSEN, 2001; RUSSEL; NORVIG, 2002).

2.2 Espaço de Possibilidades, Estados e Eventos

O primeiro item na teoria de probabilidades é o espaço de possibilidades do mundo. Este conjunto é também chamado de espaço amostral. Usaremos aqui a expressão espaço de possibilidades e o denotaremos por Ω . Os elementos ω do conjunto Ω são chamados estados. Um conceito básico é que os estados são mutuamente exclusivos e exaustivos, isto é, apenas um estado pode ocorrer em um experimento particular. Um evento é um subconjunto de Ω . Assim, um evento A ocorre quando um estado $\omega \in A$ é obtido. Os eventos e estados não são difusos, ou seja, um evento ocorre ou não ocorre.

Os eventos são denotados por letras maiúsculas: A , B , C , etc. O sobrescrito c corresponde ao complemento do conjunto com relação a Ω . Como exemplo, A^c é o evento contendo todos os estados que não estão em A .

A intersecção dos eventos A e B é denotada por $A \cap B$, isto é, o evento contém estados que estão ao mesmo tempo em A e B . A união dos eventos A e

B é denotada por $A \cup B$, isto é, o evento contém estados que estão em A ou B ou em ambos.

2.3 Variáveis

A partir de um espaço de possibilidades pode-se definir uma função de Ω para o conjunto dos reais. Qualquer função deste tipo $X : \Omega \rightarrow \mathfrak{R}$ é chamada de variável aleatória. Apesar do nome, a função X é determinística, o argumento ω da função é que é aleatório. Vale ressaltar também que as variáveis têm um número finito de *categorias*, pois supõe-se que o espaço de estados tenha um número finito de estados. Denotá-se por categoria cada um dos valores que a variável aleatória pode assumir.

Como ilustração do conceito acima considere o seguinte exemplo: Uma pessoa ω é selecionada de uma população Ω . A idade dessa pessoa é denotada pela variável $X(\omega)$. A mesma população pode ser utilizada para se definir uma variável diferente Y , onde $Y(\omega)$ é o peso da pessoa (arredondando para valores inteiros em kilogramas) ω selecionada de Ω .

2.4 Medidas de Probabilidade

Uma medida de probabilidade é uma função que associa pesos a eventos. A primeira suposição sobre os valores de probabilidades é que eles são números entre *zero* e *um*. O *zero* está relacionado com a impossibilidade, enquanto o *um* se relaciona com a certeza do acontecimento do evento considerado.

Além disso, é necessária uma suposição adicional. Considere dois eventos A e B , tais que pode-se associar valores precisos de probabilidades a estes, denotando estes valores por $P(A)$ e $P(B)$. Se A e B são eventos disjuntos, então é correto considerar a probabilidade da sua união, $P(A \cup B)$, igual a $P(A) + P(B)$.

No caso em que os eventos A e B não são disjuntos pode-se escrever que:

$$P(A \cup B) = P(A) + P(B) - P(A \cap B)$$

Essas suposições podem ser resumidas pelos três axiomas abaixo:

- P1 - Para qualquer evento A , $P(A) \geq 0$.
- P2 - O espaço de possibilidades Ω tem probabilidade *um*: $P(\Omega) = 1$.

- P3 - A probabilidade da união de dois eventos A e B é dada por:

$$P(A \cup B) = \left\{ \begin{array}{ll} P(A) + P(B) & : \text{se } A \cap B = \emptyset \\ P(A) + P(B) - P(A \cap B) & : \text{caso contrário} \end{array} \right\}$$

onde $\emptyset$ indica o conjunto vazio.

2.5 Medidas de Probabilidade Condicional

Em muitas situações é interessante considerar a probabilidade de um evento A condicionado na ocorrência de outro evento B . Este conceito é captado na medida de probabilidade $P(A|B)$. Dado um espaço de possibilidades finito Ω , uma medida de probabilidade condicional é obtida a partir da função $P(.|.)$, que mapeia pares de eventos para números reais. O primeiro evento, o evento condicionado, é um subconjunto de Ω , enquanto o segundo evento, o evento condicionante, é um subconjunto não vazio de Ω . Não se pode especular sobre a probabilidade de qualquer evento condicional na ocorrência do conjunto vazio, pois o conjunto vazio deve ser visto como uma representação da impossibilidade lógica.

As medidas de probabilidade condicional para qualquer evento B não vazio, $P(.|B)$, também devem atender aos axiomas P1 a P3. Assim a ocorrência do evento B condicional na ocorrência de B deve ser igual a um, $P(B|B) = 1$. Esta é a versão condicional do axioma P2, chamada de normalização.

Um importante axioma que leva ao teorema de Bayes estabelece: Seja um evento B não vazio e $P(B) > 0$, então

$$P(A|B) = \frac{P(A \cap B)}{P(B)}.$$

O teorema de Bayes estabelece que esta mesma quantidade pode ser calculada pela expressão a seguir:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B|A)P(A) + P(B|A^c)P(A^c)}.$$

2.6 Esperança

A partir de uma distribuição de probabilidades sobre o espaço de possibilidades pode-se calcular o valor esperado de qualquer variável aleatória que é função destes estados.

O valor esperado representa o valor médio da variável aleatória considerada,

e é dado por:

$$E(X) = \sum_{\omega \in \Omega} X(\omega)P(\omega)$$

2.7 Distribuições de Probabilidades

Dada uma variável X , existe um espaço de possibilidades Ω_X contendo todos os valores possíveis de X . Isto é, se $X(\omega) = x$ para algum ω , então $x \in \Omega_X$.¹

A função distribuição de probabilidade p para a variável X é uma função tal que:

$$p(x) = P(\{X = x\}).$$

A probabilidade de qualquer evento $A \subseteq \Omega_X$ pode ser calculada usando a função distribuição de X :

$$P(X \in A) = \sum_{x \in \Omega_X, x \in A} p(x).$$

Claramente estas probabilidades são todas derivadas das probabilidades sobre o espaço de possibilidades original Ω .

2.8 Redes Bayesianas

2.8.1 Conceitos Básicos

Uma RB representa uma distribuição de probabilidades sobre um conjunto de variáveis $\mathbf{X}$. A distribuição conjunta é especificada através de um grafo direcionado e acíclico. Cada nó do grafo representa uma variável aleatória X_i em $\mathbf{X}$.² Os pais de X_i são denotados por $pa(X_i)$; os filhos de X_i são denotados por $ch(X_i)$. Além disso, os pais dos filhos de X_i que não tem X_i como pais são denotados por $spo(X_i)$, isto é, esposos de X_i em uma sociedade poligâmica de RBs, veja (COZMAN, 2000).

A semântica do modelo de RBs é determinada pela condição de Markov onde: Toda variável é independente de seus não descendentes dados seus pais. Uma implicação da condição de Markov é que qualquer rede Bayesiana representa uma

¹Observe que as variáveis são maiúsculas e seus valores não, assim X é uma variável e x é um valor de X (um número real).

²É importante salientar que na notação adotada, os conjuntos de variáveis são denotados em negrito; por exemplo, $\mathbf{X}$.

Tabela 2.1: Relações entre as variáveis da rede Bayesiana da Figura 2.1.

X_1	pai de	X_2 e X_3
X_2 e X_3	pais de	X_5
X_6	filho de	X_3 e X_4
X_5 e X_6	filhos de	X_3
X_5, X_6 e X_7	descendentes de	X_3
X_1, X_2 e X_4	não descendentes de	X_3
X_2, X_3, X_5, X_6 e X_7	descendentes de	X_1
X_4	não descendente de	X_1
X_2 e X_4	esposos de	X_3

Figura 2.1: Um exemplo de RB.

única distribuição de probabilidade condicional que pode ser fatorizada como:

$$P(\mathbf{X}) = \prod_i p(X_i | pa(X_i)).$$

Desta forma toda variável aleatória X_i é associada com uma distribuição de probabilidade condicional $p(X_i | pa(X_i))$. Na Figura 2.1 temos um exemplo de rede Bayesiana. Cada um dos nós da rede está associado com uma distribuição de probabilidade: $p(X_1)$, $p(X_2 | X_1)$, $p(X_3 | X_1)$, $p(X_4)$, $p(X_5 | X_2, X_3)$, $p(X_6 | X_3, X_4)$ e $p(X_7 | X_5, X_6)$. Na Tabela 2.1 vemos as relações que as variáveis da rede Bayesiana da Figura 2.1 mantém.

A idéia de RBs, de maneira informal, é usar a direção das setas dos arcos para fornecer um tipo de “significado causal” para a rede. Os arcos indicam influência direta entre as variáveis, tornando fácil quantificar os valores de probabilidades locais. É importante observar que pode-se especificar modelos e probabilidades em qualquer maneira desejada, porém a idéia de se utilizar causas e consequências é bastante poderosa. Por exemplo, em vez de especificar a probabilidade de

uma doença dado um sintoma, pode-se considerar muito mais fácil especificar a probabilidade de um sintoma dada a doença.

2.8.2 Inferência em Redes Bayesianas

Dada uma RB, o evento E denota a *evidência*, ou as *observações*, na rede. Por exemplo, $E = \{X_1 = x_{12}, X_3 = x_{31}\}$ fixa os valores das variáveis X_1 e X_3 . Seja $\mathbf{X}_E$ o conjunto das variáveis observadas; na sentença anterior, $\mathbf{X}_E = \{X_1, X_3\}$.

A inferência com RBs normalmente envolve o cálculo de probabilidades marginais posteriores para um conjunto de variáveis incógnita $\mathbf{X}_q$. Observe que não há restrição no número de variáveis em $\mathbf{X}_q$. Além disso, o símbolo $\mathbf{X}_q$ é usado para denotar tanto as variáveis incógnita como o evento que as variáveis incógnita denotam. A probabilidade posterior de $\mathbf{X}_q$ dado E é:

$$P(\mathbf{X}_q | E) = \frac{P(\mathbf{X}_q, E)}{P(E)} = \frac{\sum_{\mathbf{X} \setminus \{\mathbf{X}_q, \mathbf{X}_E\}} P(\mathbf{X})}{\sum_{\mathbf{X} \setminus \mathbf{X}_E} P(\mathbf{X})},$$

onde a expressão $\sum_{\mathbf{X}} f(\mathbf{X}, \mathbf{Y})$ indica que todas as variáveis em $\mathbf{X}$ são somadas para fora da função $f(\mathbf{X}, \mathbf{Y})$ e o conjunto de todas as variáveis que pertencem a $\mathbf{X}$ mas não pertencem a $\mathbf{X}_E$ é indicado por $\mathbf{X} \setminus \mathbf{X}_E$.

De um ponto de vista computacional, a quantidade importante é o numerador da equação anterior:

$$P(\mathbf{X}_q | E) = \sum_{\mathbf{X} \setminus \{\mathbf{X}_q, \mathbf{X}_E\}} P(\mathbf{X}),$$

pois a partir do numerador, pode-se obter o denominador facilmente. O denominador $P(E)$ é simplesmente uma constante de normalização quando a evidência E é fixada. Assim, uma regra básica na derivação de algoritmos para redes Bayesianas é: Calcule o numerador $P(\mathbf{X}_q, E)$, e obtenha o denominador no último estágio possível.

Observe que cada variável deve aparecer como variável condicionante uma vez para cada um de seus filhos. Isto é, se X_1 é um filho de X_2 , então X_2 deve aparecer como variável condicionante na densidade $P(X_1 | X_2, \dots)$. Note também que qualquer variável X_i aparece somente uma vez como uma variável não condicionante na densidade $p(X_i | pa(X_i))$.

3 Processos de Decisão de Markov

3.1 Introdução

Neste capítulo é feita uma revisão da literatura apresentando a estrutura de PDMs e as relações entre este modelo e os modelos de planejamento clássico em Inteligência Artificial (IA) (BOUTILIER; DEAN; HANKS, 1999). O interesse está em planejar o comportamento de um agente em um sistema estocástico e dinâmico, isto é, um sistema que em qualquer instante de tempo pode estar em um dos estados de um conjunto de estados distintos, e no qual as mudanças de estado do sistema ao longo do tempo dependem do estado atual e dos eventos¹ ocorridos. Uma ação é um tipo particular de evento executado pelo agente com o propósito de mudar o estado do sistema. O modelo assume que o agente tem o controle sobre quando e quais ações são executadas, porém os efeitos das ações podem não ser perfeitamente previsíveis. Ao contrário das ações, os eventos exógenos não estão sob o controle do agente, e a sua ocorrência pode ser apenas parcialmente observável. Esta visão resumida de um agente é consistente com a visão dentro da IA onde o agente é um tomador de decisões autônomo.

3.2 Estados e Transições de Estados

Um estado é uma descrição do sistema em um ponto particular no tempo. O modo de definir cada estado varia com a aplicação particular, porém é comum assumir que o estado captura toda a informação relevante disponível para o processo de decisão do agente. Assume-se um espaço finito de estados $S = \{s_1, \dots, s_N\}$ para os possíveis estados do sistema.² Em muitos casos não há informações completas sobre o estado corrente. Esta incerteza ou parcialidade de informações pode ser

¹O conceito de evento utilizado aqui é diferente do apresentado na teoria de probabilidades.

²A maior parte da discussão deste capítulo também se aplica aos casos onde o espaço de estados é infinito. Para uma discussão sobre problemas com estados infinitos e contínuos veja (PUTERMAN, 1994).

capturada usando uma distribuição de probabilidades sobre os estados em S .

Um sistema dinâmico, estocástico e discreto no tempo, é composto de um espaço de estados e distribuições de probabilidades determinando as possíveis transições de estados, isto é, como o próximo estado do sistema depende dos estados anteriores. Estas distribuições constituem um modelo de como o sistema evolui ao longo do tempo em resposta às ações e aos eventos exógenos, refletindo o fato que os efeitos das ações e eventos exógenos podem não ser perfeitamente preditos, mesmo se o estado atual for conhecido.

Embora, geralmente, o interesse esteja em como o agente escolhe um curso apropriado de ações, no resto desta seção é assumido que o curso de ações do agente está fixo e concentra-se no problema de prever o estado do sistema após a ocorrência de uma seqüência predeterminada de ações. A seção seguinte trata a questão da seleção de ações.

Assume-se que o sistema evolui em estágios, onde a ocorrência de um evento marca a transição de um estágio t para o próximo estágio $t + 1$. Considerando que eventos definem mudanças no estágio e que os eventos usualmente causam transições de estado, usualmente as transições de estágio são equiparadas às transições de estado. Contudo, é possível que a ocorrência de um evento deixe o sistema no mesmo estado.

A progressão do sistema através dos estágios é aproximadamente análoga à passagem do tempo. As duas são idênticas se for assumido que alguma ação (possivelmente uma ação nula) é executada em cada estágio e que cada ação ocupa uma unidade completa de tempo. Desta forma pode-se falar de forma relaxada como se os estágios correspondessem a unidades de tempo, e fazer referência a T de modo permutável como o conjunto de todos os estágios e como o conjunto de todos os pontos no tempo.³

A incerteza pode ser modelada levando-se em conta que o estado do sistema, em um estágio t , é visto como uma variável aleatória S^t que assume valores do espaço de estados S . A suposição de causalidade requer que a variável S^t não dependa diretamente dos valores futuros desta variável S^k tal que $k > t$. Isto requer que o sistema seja modelado de forma que o histórico passado determine diretamente a distribuição sobre o estado corrente, visto que o conhecimento dos estados futuros pode influenciar a estimativa do estado atual apenas indiretamente através das evidências de qual estado corrente pode levar aos estados

³Pode-se encontrar uma literatura considerável na comunidade de Pesquisa Operacional (PO) sobre PDM em tempo contínuo (PUTERMAN, 1994), porém este tema não é abordado aqui.

Figura 3.1: (a) Um processo estocástico genérico, (b) uma cadeia de Markov e (c) uma cadeia de Markov estacionária.

futuros. A Figura 3.1(a) mostra uma perspectiva gráfica de um sistema dinâmico, estocástico e discreto. Os nós são variáveis aleatórias denotando o estado em um instante particular, e os arcos indicam a dependência probabilística direta entre os estados. A descrição completa do sistema deve fornecer também as distribuições condicionais $P(S^t \mid S^0, S^1, \dots, S^{(t-1)})$ para todos os tempos t .

Os estados devem ser vistos como as descrições do sistema sendo modelado, assim surge a questão de quanto detalhe sobre o sistema é capturado na descrição de um estado. Uma maior quantidade de detalhes implica em mais informações sobre o sistema, produzindo como consequência previsões melhores sobre o comportamento futuro. Porém, mais detalhe traz também um conjunto S maior, o que pode aumentar o custo computacional do processo de decisão.

Normalmente é assumido que um estado contém informação suficiente para prever o estado seguinte (BOUTILIER; DEAN; HANKS, 1999). Em outras palavras, qualquer informação no histórico do sistema relevante na previsão do seu futuro é capturada explicitamente no próprio estado. Formalmente, esta suposição, a suposição de Markov, diz que o conhecimento do estado atual torna a informação sobre o passado irrelevante para fazer previsões sobre o futuro:

$$P(S^{(t+1)} \mid S^t, S^{(t-1)}, \dots, S^0) = P(S^{(t+1)} \mid S^t)$$

Os modelos Markovianos podem ser representados graficamente usando uma estrutura como a da Figura 3.1(b), refletindo o fato que o estado presente é suficiente para prever a evolução do estado futuro. ⁴

⁴A propriedade de Markov se aplica a um modelo particular e não ao sistema propriamente, assim, qualquer modelo não Markoviano (de ordem finita, isto é, que depende de no máximo k estados anteriores) de um sistema pode ser convertido em um modelo Markoviano equiva-

Finalmente, é comum assumir que os efeitos de um evento dependem somente do estado corrente, mas não do estágio ou instante no qual o evento ocorreu.⁵ Se a distribuição que prevê o próximo estado é a mesma em cada estágio, o modelo é dito estacionário e pode ser representado esquematicamente usando apenas dois estágios, como na Figura 3.1(c). Neste caso é necessária somente uma distribuição condicional simples. Um sistema de tempo discreto, com um conjunto de estados finito e de dinâmica estocástica é chamado usualmente de Cadeia de Markov.

Pode-se representar uma cadeia de Markov não estacionária com $|T|$ estágios utilizando $|T|$ matrizes de transição, cada uma de tamanho $N \times N$, onde a matriz P^t captura as probabilidades de transição governando o sistema quando ele se move do estágio t para o estágio $t + 1$. Cada matriz é composta pelas probabilidades p_{ij}^t , onde $p_{ij}^t = P(S^{(t+1)} = s_j \mid S^t = s_i)$. Se o processo é estacionário, a matriz de transição é a mesma para todos os estágios e uma matriz será suficiente. Dada uma distribuição inicial sobre os estados P^0 , a distribuição de probabilidades sobre os estados depois de n estágios é $\prod_{i=0}^{n-1} P^i$.

Para completar o modelo é necessário fornecer uma distribuição de probabilidades sobre o estado inicial, refletindo a probabilidade de estar em qualquer estado no estágio 0. Esta distribuição pode ser representada como um vetor linha de valores reais de tamanho $N = |S|$ (uma entrada para cada estado). Denotamos este vetor por P^0 e usamos p_i^0 para denotar cada uma de suas entradas, isto é, a probabilidade de começar no estado s_i .

Um processo estacionário de Markov pode ser representado também usando um diagrama de transição de estados como o da Figura 3.2. Aqui os nós correspondem a estados particulares e os estágios não são representados explicitamente. Os arcos denotam as transições possíveis (aquelas com probabilidade diferente de zero) e são nomeados com a probabilidade de transição $p_{ij} = P(S^{(t+1)} = s_j \mid S^t = s_i)$. O arco de um nó i para o nó j é nomeado com p_{ij} se $p_{ij} > 0$.⁶ O tamanho de tal diagrama é pelo menos $O(N)$ e no máximo $O(N^2)$, dependendo do número de arcos. Esta representação é útil quando o gráfico de transição é esparso, por exemplo, quando a maioria dos estados tem transições imediatas apenas para poucos vizinhos.

Na Tabela 3.1 vemos uma representação diferente para o mesmo modelo apre-

lente, porém maior. Na teoria de controle, isto é chamado conversão para a forma de estados (LUENBERGER, 1979).

⁵Claramente, esta também é uma declaração sobre os detalhes do modelo, dizendo que o estado carrega informação suficiente para fazer o estágio irrelevante na previsão das transições.

⁶É importante notar aqui que os nós não representam variáveis aleatórias como aconteceu nas figuras anteriores.

Figura 3.2: Um diagrama de transição de estados.

Tabela 3.1: Matriz de probabilidades de transição de estados.

	1	2	3	4	5	6	7
1	0.0	0.5	0.0	0.5	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.8	0.0	0.2
3	0.0	0.0	0.0	0.0	0.0	0.1	0.9
4	0.0	0.0	0.0	0.0	0.0	0.0	1.0
5	0.0	0.0	0.0	0.0	0.0	0.0	1.0
6	0.7	0.0	0.3	0.0	0.0	0.0	0.0
7	0.0	0.0	0.4	0.0	0.5	0.1	0.0

sentado na Figura 3.2. Na primeira coluna temos o estado atual do sistema, em cada uma das colunas seguintes temos as probabilidades de o estado da coluna ser o estado futuro. Por exemplo, na segunda linha da tabela, o estado atual é o **1** e os estados **2** e **4** têm probabilidade 0.5 cada um de se tornar o estado futuro. Os demais estados tem probabilidade zero de se tornar o estado futuro para esse estado atual.

3.3 Ações

As cadeias de Markov podem ser usadas para descrever a evolução de um sistema estocástico, mas elas não capturam o fato de que o agente pode escolher executar ações que alteram o estado do sistema. Um elemento chave dos PDMs é o conjunto de ações disponível para o tomador de decisões. Quando uma ação é executada em um estado particular, o estado muda estocasticamente em resposta à ação. Assume-se que o agente toma alguma ação em cada estágio do processo, então o sistema muda o estado de acordo.

Em cada estágio t do processo e em cada estado s , o agente tem disponível um conjunto de ações CA_s^t . Isto é chamado *conjunto aplicável* para o estado s no estágio t . Para descrever os efeitos de $a \in CA_s^t$, devemos fornecer uma distribuição de transição de estado $P(S^{t+1} = s_j \mid S^t = s_i, A^t = a)$ para todas as ações a , estados s e estágios t . Isto forma uma família de distribuições parametrizada por S^t e A^t , sendo que a probabilidade de A^t não é parte do modelo.

Freqüentemente assume-se que o conjunto aplicável de ações é o mesmo para todos os estágios e estados, no qual o conjunto de ações é $CA = \{a_1, \dots, a_K\}$ e cada uma pode ser executada em qualquer tempo. Isto contrasta com a prática de planejamento em IA onde pré-condições são associadas às ações, definindo os estados nos quais elas podem ser executadas de maneira significativa. Aqui assumimos que qualquer ação pode ser executada (ou tentada) em qualquer estado. Quando a ação não tem efeitos quando executada em um estado, ou sua execução leva a efeitos não desejados, isto pode ser apontado na matriz de transição desta ação. As pré-condições das ações, freqüentemente, são uma conveniência computacional e não uma necessidade de representação: elas podem tornar o processo de planejamento mais eficiente ao identificar estados nos quais o planejador não deve considerar a seleção de certas ações. Para ilustrar os conceitos de planejamento abaixo, contudo, nós assumimos em alguns casos que as ações têm pré-condições.

Novamente restringimos nossa atenção a processos estacionários, onde neste caso os efeitos de cada ação dependem somente do estado e não do estágio. A matriz de transição toma a forma $p_{ij} = P(S^{t+1} = s_j | S^t = s_i, A^t = a_k)$, capturando a probabilidade de que o sistema passe para o estado s_j quando a_k é executada em s_i . Nos modelos estacionários uma ação a_k é completamente descrita por uma matriz de transição P^k de dimensões $N \times N$. É importante notar que a matriz de transição para uma ação inclui não só os efeitos diretos da execução da ação, mas também os efeitos dos eventos exógenos que podem ocorrer no mesmo estágio. É possível estimar os efeitos das ações e dos efeitos exógenos separadamente, e então combiná-los dentro de uma matriz de transição simples em certos casos (PUTERMAN; BOUTILIER, 1995).

3.4 Eventos Exógenos

Eventos exógenos são aqueles eventos que estocasticamente causam transição de estado, assim como as ações, porém estes eventos estão fora do controle do tomador de decisões. Os eventos exógenos podem corresponder à evolução natural do processo ou às ações de um outro agente. Os efeitos de uma ação do agente podem ser combinados com os efeitos dos eventos exógenos, assim, as probabilidades de transição de estado incorporam ambas as causas. Para o propósito de tomada de decisão, é precisamente este efeito combinado que é importante na previsão da distribuição sobre os possíveis estados resultantes quando uma ação é executada. Estes modelos são chamados de modelos com *eventos implícitos*, pois os efeitos dos eventos exógenos são modelados nas probabilidades de transição associadas com as ações.

Contudo, freqüentemente é natural ver estas transições como o conjunto de dois eventos separados, cada um tendo seus próprios efeitos no estado. De forma mais geral, freqüentemente pensamos nas transições como o resultado dos efeitos da ação escolhida pelo agente e de certos eventos exógenos além do controle do agente, onde cada um destes pode ocorrer com uma certa probabilidade. Quando os efeitos das ações são decompostos desta maneira, chamamos o modelo de modelo de *eventos explícitos*.

Naturalmente, processos diferentes para a construção de matrizes de eventos implícitos exigirão diferentes suposições sobre a interação entre ações e eventos. Se tais modelos implícitos são construídos ou especificados diretamente sem mencionar explicitamente os eventos exógenos, sempre assumiremos a menos que seja dito o contrário que as matrizes de transição de estados levam em consideração os

efeitos dos eventos exógenos, e então representam a melhor informação do agente sobre o que acontecerá se uma ação particular for executada. Para maiores detalhes sobre eventos exógenos veja (BOUTILIER; DEAN; HANKS, 1999).

3.5 Observações

Embora os efeitos de uma ação possam depender de qualquer aspecto do estado anterior, a escolha da ação pode depender somente do que o agente pode observar sobre o estado atual, do que pode lembrar sobre suas observações anteriores e dos seus objetivos. Modela-se as capacidades de observação ou sensoramento do agente introduzindo-se um conjunto finito de observações $O = \{o_1, \dots, o_h\}$. O agente recebe uma observação deste conjunto em cada estágio antes de escolher sua ação neste estágio. Pode-se modelar esta observação como uma variável aleatória O^t cujo valor é obtido de O . A probabilidade de que um valor particular de O^t seja gerado depende de:

- O estado do sistema em $t - 1$
- A ação executada em $t - 1$
- O estado do sistema em t após a execução da ação em $t - 1$ e depois que os efeitos de qualquer evento exógeno em $t - 1$ são realizados, porém antes da execução da ação em t .

Seja $P(O^t = o_h | S^{t-1} = s_i, A^{t-1} = a_k, S^t = s_j)$ a probabilidade de que o agente observe o_h no estágio t dado que ele executou a ação a_k no estado s_i e chegou ao estado s_j . Assim como com as ações, assume-se que a distribuição sobre as observações seja estacionária (independente do estágio), usando $p_{i,j,k}^h = p(o_h | s_i, a_k, s_j)$ para denotar esta quantidade. As dependências probabilísticas entre os estados, as ações e as variáveis de observação podem ser vistas como um grafo no qual as variáveis indexadas no tempo são os nós e uma variável é diretamente dependente probabilisticamente de outra se existe um arco ligando as duas (Figura 3.3).

Este modelo permite uma ampla variedade de suposições sobre as capacidades de sensoramento do agente. Em um extremo estão os PDMs Completamente Observáveis (PDMCOs), no qual o agente sabe exatamente em qual estado o sistema está em cada estágio t . Este caso pode ser modelado fazendo $O = S$ e então:

$$p(o_h | s_i, a_k, s_j) = \begin{cases} 1 & \text{se } o_h = s_j \\ 0 & \text{nos outros casos} \end{cases} \quad (3.1)$$

Figura 3.3: Grafo mostrando as relações de dependência entre ações, estados e observações.

Assim, o agente recebe respostas precisas sobre os resultados de suas ações e os efeitos dos eventos exógenos — o agente possui atuadores com ruído, porém seus sensores são instantâneos e livres de ruído. A maior parte da pesquisa recente em IA que adota a estrutura do PDM supõe observação completa.

Em outro extremo, podemos considerar sistemas não observáveis, os PDMs Não Observáveis (PDMNOs), nos quais o agente não recebe nenhuma informação sobre o estado do sistema durante execução. Este caso pode ser modelado fazendo $O = \{o\}$. Aqui a mesma observação é reportada em cada estado, sem trazer nenhuma informação sobre o estado, então $p(s_j \mid s_i, a_k, o) = p(s_j \mid s_i, a_k)$. Nestes sistemas de laço aberto (open-loop), o agente não recebe respostas úteis sobre o resultado de suas ações: o agente possui atuadores ruidosos e não possui sensores. Neste caso o agente escolhe suas ações de acordo com um plano que consiste em uma seqüência de ações executadas incondicionalmente. De fato, o agente está baseado em seu modelo de previsões para determinar boas escolhas de ações antes do tempo de execução.

Tradicionalmente, o trabalho de planejamento clássico em IA implicitamente tem feito a suposição de um sistema não observável, freqüentemente acompanhada de omnisciência — que o agente sabe o estado inicial com certeza, pode prever perfeitamente os efeitos de suas ações, e pode prever precisamente a ocorrência de qualquer evento exógeno e seus efeitos. Sob estas circunstâncias, o agente pode prever o resultado exato de qualquer plano removendo, assim, a necessidade de observação. Tal agente pode construir um plano totalmente ordenado, ou seja, uma seqüência de ações a serem executadas sem resposta. Este plano é tão bom quanto qualquer outro cuja execução possa depender de informações colhidas durante o tempo de execução.

Estes dois extremos são casos especiais do modelo geral de observação, o qual permite ao agente receber informações incompletas ou ruidosas sobre o estado do sistema (i.e., PDMs Parcialmente Observáveis (PDMPOs)).

3.6 Trajetória do Sistema e Histórico Observado

Os termos *trajetória* e *histórico* são utilizados de forma equivalente para descrever o comportamento do sistema durante o curso do período de solução do problema, ou talvez em algum segmento deste. O *histórico completo do sistema* é uma seqüência de estados, ações e observações geradas do estágio 0 até algum ponto do tempo de interesse, e pode ser de comprimento finito ou infinito. Históricos completos podem ser representados por seqüências (possivelmente infinitas) de enuplas da forma

$$[(S^0, O^0, A^0), (S^1, O^1, A^1), \dots, (S^{|T|}, O^{|T|}, A^{|T|})]$$

Duas noções alternativas de histórico que contém informações menos completas podem ser definidas. Para algum estágio arbitrário t define-se o *histórico observável* como a seqüência

$$[(O^0, A^0), (O^1, A^1), \dots, (O^{t-1}, A^{t-1})]$$

onde O^0 é a observação do estado inicial. O histórico observável no estágio t contém toda a informação disponível para o agente sobre seu histórico quando ele escolhe sua ação no estágio t . O terceiro tipo de histórico é a *trajetória do sistema*, a qual é a seqüência:

$$[(S^0, A^0), (S^1, A^1), \dots, (S^{t-1}, A^{t-1}), S^t]$$

que descreve o comportamento do sistema independentemente da visão particular do agente sobre o sistema.

No cálculo do desempenho do agente, o interesse está na trajetória do sistema. Uma política para o agente deve ser definida em termos do histórico observável, pois o agente não tem acesso à trajetória do sistema, exceto no caso completamente observável, quando os dois são equivalentes.

3.7 Recompensa e Valor

O problema confrontando o agente é selecionar uma ação a ser executada em cada estágio do problema de decisão, fazendo esta decisão com base no histórico observável. O agente ainda precisa julgar de algum modo a qualidade do curso de ações. Isto é feito pela definição de uma *função valor* $V(\cdot)$, a qual mapeia o conjunto de históricos do sistema H_S no conjunto dos reais; isto é, $V : H_S \rightarrow \mathbb{R}$.⁷ O agente prefere o histórico do sistema h a h' no caso em que $V(h) > V(h')$. Assim, o agente julga seu comportamento bom ou ruim dependendo do seu efeito na trajetória do sistema. Geralmente, o agente não pode prever com certeza qual trajetória o sistema irá percorrer, e o melhor que ele pode fazer é gerar uma distribuição de probabilidades sobre as possíveis trajetórias causadas por suas ações. Neste caso, o agente computa o valor esperado de cada curso de ação candidato e escolhe uma política que maximiza esta quantidade.

Assim como com a dinâmica do sistema, a especificação de uma função valor sobre trajetórias arbitrárias pode ser trabalhosa e não intuitiva. No entanto é importante identificar a estrutura da função valor, pois esta estrutura pode levar a uma representação mais compacta.

Duas suposições sobre as funções valor são comumente adotadas na literatura do PDM: separabilidade no tempo e aditividade. Uma função valor separável no tempo é definida em termos de funções mais primitivas que podem ser aplicadas sobre os estados e ações componentes. A função recompensa $R : S \rightarrow \mathbb{R}$ associa uma recompensa ao fato do sistema estar em um estado s . Custos podem ser associados à execução de ações, definindo uma função custo $C : S \times CA \rightarrow \mathbb{R}$ que associa um custo a execução da ação a no estado s . Recompensas são adicionadas à função valor, enquanto custos são subtraídos.⁸

Uma função valor é separável no tempo se ela é uma combinação simples das recompensas e custos apurados em cada estágio. “Combinação simples” significa que o valor obtido é uma função dos custos e recompensas em cada estágio, onde os custos e recompensas podem depender do estágio t , mas a função que combina este dois deve ser independente do estágio, mas comumente uma combinação linear ou um produto.⁹ Uma função valor é aditiva se a função combinação é

⁷Tecnicamente, o conjunto de históricos de interesse também depende do horizonte escolhido, como é descrito abaixo.

⁸O termo “recompensa” é um pouco confuso pois poderia ser negativo, neste caso “penalidade” seria um nome melhor. De forma semelhante, “custos” podem ser positivos (punitivos) ou negativos (benéficos).

⁹Veja (LUENBERGER, 1973) para uma definição mais precisa de separação no tempo.

Figura 3.4: Um processo de decisão com recompensas e custos para as ações.

uma soma dos custos e recompensas apurados sobre o histórico de estágios. A adição de recompensas e custos das ações em um sistema com valores separáveis no tempo pode ser vista graficamente na Figura 3.4, onde um custo $C(S^t, A^t)$ no estágio t está associado a execução da ação A^{t-1} no estado S^{t-1} e uma recompensa $R(S^t)$ está associada a cada estado S^t .

A suposição de separação no tempo é restritiva. Poderia haver certos objetivos que envolvessem limites de tempo e manutenção, o que requer funções valor que não são separáveis, dada a nossa representação atual do estado. Note, contudo, que a separabilidade - assim como a propriedade de Markov - é uma propriedade de uma representação particular. Assim, podem-se adicionar informações aos estados como o tempo. Com esta informação adicional poderíamos restabelecer uma função valor separável no tempo, mas ao custo de um aumento no número de estados e uma representação das ações mais custosa e detalhada.¹⁰

3.8 Horizontes e Critério de Sucesso

Com o objetivo de calcular um curso de ação particular, é necessário especificar o intervalo (em quantos estágios) de duração da execução. Isto é conhecido como o problema do horizonte. Em problemas de horizonte finito, o desempenho do agente é calculado sobre um número fixo e finito de estágios $|T|$. Comumente, nosso objetivo é maximizar a recompensa total esperada associada com o curso de ação; então se define o valor de qualquer histórico (de horizonte finito) h com

¹⁰Para uma abordagem sistemática da manipulação de certos tipos de funções recompensa que dependem do histórico veja (BACCHUS; BOUTILIER; GROVE, 1996; BACCHUS; BOUTILIER; GROVE, 1997).

comprimento $|T|$ como (BELLMAN, 1957):

$$V(h) = \sum_{t=0}^{|T|-1} (R(S^t) - C(S^t, A^t)) + R(S^{|T|})$$

Um problema de horizonte infinito, por outro lado, requer que o desempenho do agente seja calculado sobre uma trajetória infinita. Neste caso a recompensa total pode ser ilimitada. Nesta situação, pode ser necessário um meio diferente de calcular a trajetória. O mais comum é introduzir um fator de desconto, o qual assegura que as recompensas e custos apurados nos estágios posteriores são inferiores aos apurados nos estágios anteriores. A função valor para um problema de recompensa descontada total esperada é definida como:

$$V(h) = \sum_{t=0}^{\infty} \gamma^t (R(S^t) - C(S^t, A^t))$$

onde γ é uma taxa de desconto fixa ($0 \leq \gamma < 1$). Esta formulação particular é um modo simples de assegurar uma medida limitada de valor sobre um horizonte infinito, porém é importante verificar que a desconto é de fato apropriado. Justificativas econômicas são freqüentemente fornecidas para modelos de desconto — uma recompensa recebida mais cedo é mais valiosa do que uma obtida no futuro, pois a mesma pode de alguma maneira ser investida. O desconto também pode ser adequado para modelar um processo que termina com probabilidade $(1 - \gamma)$ em algum ponto no tempo, ou seja, o processo tem horizonte finito, porém de tamanho incerto (ex., um robô que pode quebrar a qualquer instante). Por estas razões, o desconto é algumas vezes usado também em problemas de horizonte finito.

Outra técnica que trata problemas de horizonte infinito é o cálculo de trajetória baseado na média das recompensas obtidas por estágio, também chamada de ganho. O ganho de um histórico é definido como:

$$V(h) = \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{t=0}^n (R(S^t) - C(S^t, A^t))$$

Refinamentos deste critério também foram propostos (PUTERMAN, 1994).

Algumas vezes o problema por si só assegura que a recompensa total sobre uma trajetória infinita é limitada, e então o critério da recompensa total esperada é bem definido. Considere o caso comum dos planejadores de IA onde a tarefa do agente é levar o sistema para um estado objetivo. Uma recompensa positiva é obtida somente quando o objetivo é atingido, todas as ações causam um custo

não negativo, e quando um objetivo é atingido o sistema entra em um *estado absorvente*, isto é, um estado a partir do qual nenhum custo ou recompensa adicional é apurado e de onde o sistema não pode sair. Se há certeza de que o objetivo é atingido em algum estágio, esta situação pode ser formulada como um problema de horizonte infinito onde a recompensa total é limitada para qualquer trajetória desejada (BERTSEKAS, 1987; PUTERMAN, 1994). Em geral, tais problemas não podem ser formulados como problemas de horizonte finito e fixo a menos que um limite a priori no número de passos necessários para atingir o objetivo possa ser estabelecido. Estes problemas são algumas vezes chamados de problemas de horizonte indefinido: de um ponto de vista prático, o agente continuará a executar ações por um número finito de estágios, mas o número exato não pode ser determinado a priori.

3.9 Critério de Solução

Para completar nossa definição do problema de planejamento é necessário especificar o que constitui uma solução para o problema. Aqui novamente há uma separação entre as formulações explícitas do PDM e o trabalho na comunidade de planejamento em IA. Os problemas clássicos de PDM são geralmente colocados como problemas de otimização: dada uma função valor, um horizonte, e uma métrica de cálculo (ex., a recompensa total esperada, a recompensa descontada total esperada, a média de recompensa esperada por estágio) o agente procura uma política de comportamento que maximiza a função objetivo.

O trabalho em IA freqüentemente busca uma solução de satisfação para tais problemas. Na literatura de planejamento, é geralmente assumido que qualquer plano que satisfaça o objetivo é igualmente desejado tanto quanto qualquer outro plano que satisfaça o objetivo, e que qualquer plano que satisfaça o objetivo é mais adequado do que qualquer outro que não o satisfaça.¹¹ Numa estrutura probabilística, pode-se buscar o plano que satisfaça o objetivo com máxima probabilidade (uma otimização), porém isto pode levar a situações nas quais um plano ótimo tem comprimento infinito se o estado do sistema não é completamente observável. A alternativa de satisfação é buscar qualquer plano que satisfaça o objetivo com uma probabilidade excedendo um limite dado.

¹¹Williamson e Hanks (1994) apresentam uma rerepresentação de planejamento como um problema de otimização e não de satisfação.

3.10 Políticas

Até o momento o termo política (ou curso de ações, ou plano) foi mencionado informalmente, agora será fornecida uma definição formal. O problema de decisão confrontando o agente pode ser visto de maneira mais geral como decidir qual ação executar dado o histórico observado atual. Define-se uma *política* π como um mapeamento do conjunto de históricos observáveis H_o para ações, isto é, $\pi : H_o \rightarrow CA$. Intuitivamente, o agente executa a ação

$$A^t = \pi(\langle \langle O^0, A^0 \rangle \dots, \langle O^{t-1}, A^{t-1} \rangle, O^t \rangle)$$

no estágio t se ele executou as ações $A^0, \dots, A^{t-1}$ e fez as observações $O^0, \dots, O^{t-1}$ nos estágios passados, e acabou de fazer a observação O^t no estágio atual.

Uma política induz uma distribuição $P(h \mid \pi)$ sobre o conjunto de históricos do sistema H_s ; esta distribuição de probabilidades depende da distribuição inicial P^0 . O valor esperado de uma política é definido como:

$$VE(\pi) = \sum_{h \in H_s} V(h)P(h \mid \pi)$$

Gostaríamos que o agente adotasse uma política que ou maximizasse o valor esperado ou, em um contexto de satisfação, tenha um valor esperado satisfatoriamente alto.

A forma geral de uma política, dependendo de como o agente faz uma observação arbitrária, pode levar a políticas ou algoritmos de construção de políticas muito complicados. Em casos especiais, contudo, suposições sobre a capacidade de observação e a estrutura da função valor pode resultar em políticas ótimas que têm uma forma muito mais simples.

No caso de um PDM completamente observável com uma função valor separável no tempo, a ação ótima em cada estágio pode ser computada usando apenas informações sobre o estado e o estágio atuais, isto é, podemos restringir as políticas para ter uma forma mais simples $\pi : S \times T \rightarrow CA$ sem o perigo de atuar de modo sub-ótimo. Isto se deve ao fato de que a capacidade de observação completa permite que o estado seja completamente observado, e a suposição de Markov torna o histórico anterior irrelevante.

No caso não observável, o histórico observado contém um vácuo de observações e o agente deve escolher suas ações usando só o conhecimento de suas ações prévias e do estágio; contudo, como incorpora as ações prévias, a política

toma a forma $\pi : T \rightarrow CA$. Esta forma de política corresponde a uma seqüência linear e incondicional de ações $\langle A^1, A^2, \dots, A^T \rangle$, ou um plano totalmente ordenado na nomenclatura de IA.¹²

3.11 Algoritmos de Solução Empregando Programação Dinâmica

Nesta seção descreve-se técnicas para resolver PDMCOs, as quais exploram a estrutura da função valor como a separação no tempo e a aditividade. Essas técnicas abordam o problema através de programação dinâmica.

Suponha que é dado um PDM Completamente Observável (PDMCO) com uma função valor separável no tempo e aditiva. Em outras palavras, é dado um espaço de estados S , um conjunto de ações CA , uma matriz de transição de estados $P(s' | s, a)$ para cada ação a em CA , uma função recompensa $R(s)$ e uma função custo $C(s, a)$. Nós começamos com o problema de encontrar uma política que maximize a recompensa total esperada para algum horizonte finito e fixo $|T|$. Suponha que é dada uma política π tal que $\pi(s, t)$ é a ação a ser executada pelo agente no estado s com t estágios restantes para agir (tal que $0 \leq t \leq |T|$).¹³ Pode-se mostrar que o valor esperado de tal política em qualquer estado pode ser computada usando o conjunto de funções valor com t estágios ainda por atravessar V_t^π (BELLMAN, 1957). Define-se que $V_0^\pi(s) = R(s)$, então temos:

$$V_t^\pi(s) = R(s) + C(\pi(s, t), s) + \sum_{s' \in S} \{P(s' | \pi(s, t), s) V_{t-1}^\pi(s')\} \quad (3.2)$$

Esta definição da função valor para π torna clara a sua dependência em relação ao estado inicial.

Diz-se que uma política π é ótima se $V_T^\pi(s) \geq V_T^{\pi'}(s)$ para todas as políticas π' e todo $s \in S$. A função valor ótima com $|T|$ estágios por atravessar, denotada por V_T^* , é simplesmente a função valor da política ótima com horizonte $|T|$. O princípio ótimo de Bellman (1957) forma a base dos algoritmos de programação dinâmica estocásticos usados para resolver PDMs, estabelecendo a seguinte relação entre a função valor ótima no estágio t e a função valor ótima

¹²A maioria dos algoritmos da literatura de IA produz uma seqüência de ações parcialmente ordenada. Porém, estes planos não envolvem execuções condicionais ou não determinísticas. Em vez disso, eles representam o fato que qualquer seqüência linear consistente com a ordem parcial resolverá o problema. Então, um plano parcialmente ordenado é uma representação concisa para um conjunto de planos totalmente ordenados.

¹³Lembre-se que para PDMCOs outros aspectos do histórico não são relevantes.

no estágio anterior:

$$V_t^*(s) = R(s) + \max_{a \in CA} \{C(a, s) + \sum_{s' \in S} \{P(s' \mid a, s) V_{t-1}^*(s')\}\} \quad (3.3)$$

3.11.1 Iteração de Valor

A Equação (3.3) forma a base do algoritmo de iteração de valor para problemas de horizonte finito. A iteração de valor começa com a função valor $V_0^* = R$, e usa a Equação (3.3) para calcular na sequência as funções valor para os intervalos de tempo maiores, até um horizonte $|T|$. Qualquer ação que maximize o lado direito da Equação (3.3) pode ser escolhida como um elemento da política $\pi(s, t)$. A política resultante é ótima para o estágio $|T|$, para o PDMCO, e também para qualquer horizonte mais curto $t < |T|$.

É importante notar que a política descreve o que deve ser feito em cada estágio e para todo estado do sistema, mesmo que o agente não possa atingir certos estados dada a configuração inicial do sistema e as ações disponíveis.

Frequentemente nos deparamos com problemas que não possuem um horizonte finito. Nestes problemas as tarefas são executadas quando a necessidade surge, ou seja, uma ação deve ser executada apenas em determinados estados e o sistema pode retornar a estes estados. Tais problemas são melhor modelados como problemas de horizonte infinito.

Consideramos aqui o problema de construir uma política que maximiza a soma descontada das recompensas esperadas sobre um horizonte infinito.¹⁴ Mostra-se que sempre existe uma política estacionária ótima para tais problemas (HOWARD, 1960). Intuitivamente, este é o caso, pois não importa qual é o estágio do processo, sempre existe um número infinito de estágios restando; desta forma a ação ótima em cada estado é independente do estágio. Assim, podemos restringir nossa atenção a políticas que escolhem a mesma ação para um estado de forma independente do estágio do processo. Sob essa restrição, a política terá o mesmo tamanho $|S|$ independente do número de estágios sobre os quais a política é executada - a política π tem a forma $\pi : S \rightarrow CA$. Em contraste, políticas ótimas para problemas de horizonte finito são geralmente não estacionárias.

Howard (1960) também mostra que o valor da política π satisfaz a seguinte

¹⁴Este é de longe o problema mais comumente estudado na literatura, embora seja argumentado em (PUTERMAN; BOUTILIER, 1995; MAHADEVAN, 1994; SCHWARTZ, 1993) que tais problemas são frequentemente melhor modelados usando a recompensa média por estágio como um critério ótimo. Para uma discussão sobre a recompensa média como critério ótimo e suas muitas variantes e refinamentos, veja (PUTERMAN, 1994).

recorrência:

$$V^\pi(s) = R(s) + C(\pi(s), s) + \gamma \sum_{s' \in S} \{P(s' | \pi(s), s) V^\pi(s')\} \quad (3.4)$$

e que a função valor ótima V^* satisfaz:

$$V^*(s) = R(s) + \max_{a \in CA} \{C(a, s) + \gamma \sum_{s' \in S} \{P(s' | a, s) V^*(s')\}\} \quad (3.5)$$

O valor de uma política fixa π pode ser computado usando o método de aproximações sucessivas, o qual é quase idêntico ao processo descrito na Equação (3.2) acima. Começa-se com uma atribuição de valores para $V_0^\pi(s)$, então define-se:

$$V_t^\pi(s) = R(s) + C(\pi(s, t), s) + \gamma \sum_{s' \in S} \{P(s' | \pi(s, t), s) V_{t-1}^\pi(s')\} \quad (3.6)$$

A sequência de funções V_t^π converge linearmente para a função valor verdadeira V^π (PUTERMAN, 1994).

O algoritmo de iteração de valor pode ser levemente alterado de forma a construir políticas ótimas para problemas de horizonte infinito descontados. O algoritmo começa com uma função valor V_0 que atribui um valor arbitrário para cada $s \in S$. Dado o valor estimado $V_t(s)$ para cada estado s , $V_{t+1}(s)$ é calculado como:

$$V_{t+1}(s) = R(s) + \max_{a \in CA} \{C(a, s) + \gamma \sum_{s' \in S} \{P(s' | a, s) V_t(s')\}\} \quad (3.7)$$

A sequência de funções V_t converge linearmente para uma função valor ótima $V^*(s)$. Depois de um número finito de iterações n , a escolha da ação que maximiza a função valor para cada s forma uma política ótima π e V_n aproxima seu valor.

15

3.11.2 Iteração de Política

O algoritmo de iteração de política (HOWARD, 1960) é uma alternativa a iteração de valor para problemas de horizonte infinito. Ao invés de melhorar iterativamente a função valor estimada, o algoritmo modifica a política diretamente. Começa-se com uma política arbitrária π_0 , então computa-se π_{i+1} a partir de π_i .

Cada iteração do algoritmo compreende dois passos, avaliação da política e melhora da política:

¹⁵O número de iterações n é baseado no critério de parada que geralmente envolve a medição da diferença entre V_t e V_{t+1} . Para maiores detalhes sobre o critério de parada e convergência do algoritmo, veja Puterman (1994).

1. (Avaliação da política) Para cada $s \in S$, calcule a função valor $V^{\pi_i}(s)$ baseado na política corrente.
2. (Melhora da política) Para cada $s \in S$, encontre a ação a^* que maximiza

$$Q_{t+1}(a, s) = R(s) + C(a, s) + \gamma \sum_{s' \in S} \{Pr(s'|a, s)V^{\pi_i}(s')\} \quad (3.8)$$

Se $Q_{i+1}(a^*, s) > V^{\pi_i}(s)$, faça $\pi_{i+1} = a^*$; caso contrário $\pi_{i+1} = \pi_i$.¹⁶ O algoritmo itera até que $\pi_{i+1}(s) = \pi_i(s)$ para todos os estados s . O passo 1 avalia a política atual resolvendo o sistema linear $N \times N$ representado pela Equação (3.4) (uma equação para cada $s \in S$), o que pode ser computacionalmente custoso. Contudo, o algoritmo converge para uma política ótima ao menos linearmente e sob certas condições converge superlinearmente ou quadraticamente (PUTERMAN, 1994). Na prática, a iteração de política tende a convergir em um número de iterações muito menor do que a iteração de valor. A iteração de política gasta mais tempo computacional em cada estágio individual, resultando que menos estágios precisam ser computados.¹⁷

O algoritmo de iteração de política modificado (PUTERMAN; SHIN, 1978) provê um meio termo entre a iteração de política e a iteração de valor. A estrutura do algoritmo é exatamente a mesma do de iteração de política, alternando as fases de avaliação e melhora. A idéia chave é que a iteração de política modificada não precisa avaliar de forma exata a política antes de melhorá-la. Portanto, a fase de avaliação envolve um número de iterações (usualmente pequeno) de aproximações sucessivas (i.e., fazendo $V^\pi = V_t^\pi$ para algum t pequeno, usando a Equação (3.7)). Fazendo alguns ajustes no valor de t usado em cada iteração, a iteração de política modificada pode ser mais eficiente na prática do que a iteração de valor e a iteração de política (PUTERMAN, 1994). Tanto a iteração de valor quanto a iteração de política são casos especiais da iteração de política modificada, correspondendo a $t = 0$ e $t = \infty$, respectivamente.

Diversas outras variantes tanto da iteração de valor quanto da iteração de política têm sido propostas (BOUTILIER; DEAN; HANKS, 1999). Por exemplo, versões assíncronas destes algoritmos não requerem que a função valor seja construída, ou a política melhorada, em cada estado no processo. A convergência é garantida no caso da iteração de valor para problemas de horizonte infinito, na medida em que cada estado é atualizado com frequência suficiente. Garantias si-

¹⁶A função Q definida pela Equação (3.8), e assim chamada devido ao seu uso no *aprendizado-Q* (WATKINS; DAYAN, 1992), fornece o valor da execução da ação a no estado s assumindo que a função valor V^π reflete precisamente o valor futuro.

¹⁷Veja Littman, Dean e Kaelbling (1995) para mais detalhes da complexidade do algoritmo.

milares podem ser fornecidas para formas assíncronas da iteração de política. Tais variantes são ferramentas importantes para o entendimento de várias abordagens para solução de PDMs em tempo de execução (BERTSEKAS; TSITSIKLIS, 1996). Para uma discussão sobre programação dinâmica assíncrona, veja (BERTSEKAS, 1987; BERTSEKAS; TSITSIKLIS, 1996).

4 Processos de Decisão de Markov Fatorados

4.1 Introdução

Em muitos dos problemas modelados através de PDMs frequentemente utilizam-se formas fatoradas para sua representação (GUESTRIN et al., 2003; BOUTILIER; DEAN; HANKS, 1999). PDMs fatorados constituem um método para se representar PDMs grandes de forma compacta. Nesta estrutura, um estado é descrito implicitamente por uma atribuição a um conjunto de *variáveis de estado* e o modelo de transição de estados é representado usando-se *Redes Bayesianas Dinâmicas (RBDs)* (DEAN; KANAZAWA, 1989).

Neste capítulo é proposto um novo método baseado em modelos fatorados para a representação de forma compacta e eficiente das distribuições de probabilidades de transição de estado empregadas nos PDMs. A justificativa para isto está no fato que em muitos sistemas o espaço de estados do domínio é tão grande que a especificação dessas distribuições na forma tradicional ou mesmo na forma fatorada simples se torna inviável. Considere um espaço de estados de tamanho N . Para cada ação é necessário especificar uma matriz com $N \times N$ números, pois para cada estado atual possível existe uma probabilidade de transição para cada um dos N estados no espaço de estados. Este conceito é ilustrado na Tabela 4.1.1, onde p_{ij} indica a probabilidade de o sistema ir do estado i para o estado j , S_F é o estado futuro e S_P o estado atual.

Tabela 4.1: Matriz de probabilidades de transição de estado.

S_P/S_F	s_1	s_2	...	s_N
s_1	p_{11}	p_{12}	...	p_{1N}
s_2	p_{21}	p_{22}	...	p_{2N}
...	...	...	...	...
s_N	p_{N1}	p_{N2}	...	p_{NN}

A forma utilizada para representar estas distribuições de maneira compacta emprega RBDs e propriedades das *variáveis de estado* do sistema. As RBDs são utilizadas aqui para representar cadeias de Markov empregando RBs (PEARL, 1988). O termo *variáveis de estado*¹ indica o conjunto de características que permite determinar o estado atual do sistema conforme ilustrado na Seção 4.2.

É importante mencionar também que uma técnica bastante utilizada para a representação compacta de distribuições de probabilidade é o modelo Noisy-OR. Porém, este não é utilizado neste trabalho devido a influência mútua que as variáveis do problema estudado mantém. Para maiores detalhes sobre esta técnica veja Jensen (2001).

4.2 Representação dos Estados

Considere um sistema que é caracterizado por k variáveis $v_1, v_2, \dots, v_k$. Nesta situação o estado do sistema s é então determinado pela k -tupla $(v_1, v_2, \dots, v_k)$ formada pelas variáveis que o caracterizam. Suponha que cada uma destas variáveis pode assumir j valores inteiros no intervalo $[1, j]$, ou seja, cada variável possui j categorias. Assim o espaço de estados S fica com uma dimensão $|S| = j^k$.

Exemplo 1 *Considere que uma impressora é modelada por três variáveis: fila de impressão — queue — (q), nível de papel — tray — (tr) e nível de tinta — toner — (to). Suponha que estas variáveis são binárias, indicando a disponibilidade de recursos. O estado da impressora é então indicado por (q, tr, to) e existem $2^3 = 8$ estados. O mapeamento entre as variáveis e o estado do sistema é mostrado na Tabela 4.2.1.*

4.3 Representação das Probabilidades de Transição de Estado

S_P e S_F podem assumir, a princípio, quaisquer valores do espaço de estados S . O interesse aqui está em especificar a probabilidade de o sistema ir para o estado s_i dado que o estado atual é s_l e a ação executada foi a_i .

$$P(S_F = s_i \mid S_P = s_l, A_P = a_i)$$

¹Estas *variáveis* são chamadas frequentemente de *fluentes* na literatura em IA, veja McCarthy e Hayes (1969)

Tabela 4.2: Mapeamento entre o estado s do sistema e as k -tuplas que o caracterizam.

s	q	tr	to
s_1	0	0	0
s_2	0	0	1
s_3	0	1	0
s_4	0	1	1
s_5	1	0	0
s_6	1	0	1
s_7	1	1	0
s_8	1	1	1

Conforme visto anteriormente, para se especificar essa probabilidade para todos os pares s_i e s_l possíveis na forma tradicional é necessário uma matriz com $N \times N$ números. No caso em que o sistema possui k variáveis onde cada uma possui j categorias, então o espaço de estados tem dimensão $N = j^k$ e quantidade de números a especificar é j^{2k} para cada ação.

A primeira medida para compactar essa representação é utilizar RBs. Para trabalhar com esta ferramenta utiliza-se as variáveis do sistema para indicar o seu estado, assim, o estado futuro S_F é denotado pela k -tupla $(v_{1F}, v_{2F}, \dots, v_{kF})$ e o estado presente pela k -tupla $(v_{1P}, v_{2P}, \dots, v_{kP})$. Desta maneira podemos escrever que

$$\begin{aligned}
 P(S_F = s_i \mid S_P = s_k, A_P = a_i) = & P(v_{1F} \mid v_{1P}, v_{2P}, \dots, v_{kP}, A_P = a_i) \times \\
 & P(v_{2F} \mid v_{1P}, v_{2P}, \dots, v_{kP}, A_P = a_i) \times \\
 & \dots \times \\
 & P(v_{kF} \mid v_{1P}, v_{2P}, \dots, v_{kP}, A_P = a_i)
 \end{aligned}$$

Para especificar cada termo do produto são necessários j^{k+1} números, pois existem j^k combinações para a k -tupla $S_P = (v_{1P}, v_{2P}, \dots, v_{kP})$ e cada variável no estado futuro ainda pode assumir j possíveis valores. Assim, utilizando RBs seriam necessários kj^{k+1} números em vez de j^{2k} como no caso tradicional para se especificar a distribuição de probabilidades de transição de estado.

Outra vantagem de se utilizar RBs é que pode-se explorar a independência entre as variáveis reduzindo-se consideravelmente a quantidade de números necessários para se especificar a distribuição de probabilidades. Esta característica pode ser explorada em sistemas onde uma ou mais variáveis em S_F não depende de todas as variáveis em S_P . Para ilustrar esta situação, imagine um sistema onde

Figura 4.1: Uma Rede Bayesiana para ação *Imprimir* (Exemplo 2).

a variável v_{1F} só depende de v_{1P} . Neste caso o primeiro termo do produto acima poderia ser escrito como $P(v_{1F} \mid v_{1P}, A_P = a_i)$ e seria necessário apenas j^2 números para especificá-lo. Se a mesma idéia puder ser aplicada a todas as variáveis teríamos que

$$\begin{aligned} P(S_F = s_i \mid S_P = s_k, A_P = a_i) = & P(v_{1F} \mid v_{1P}, A_P = a_i) \times \\ & P(v_{2F} \mid v_{2P}, A_P = a_i) \times \\ & \dots \times \\ & P(v_{kF} \mid v_{kP}, A_P = a_i) \end{aligned}$$

e toda a distribuição poderia ser especificada com kj^2 números.

Exemplo 2 Considere uma impressora que é modelada por duas variáveis: fila de impressão — queue — (q) e tamanho da tarefa de impressão — job size — (js) e um conjunto de ações $CA = \{\text{Imprimir}, \text{Não Imprimir}\}$. Assim, podemos escrever que $S_P = (q_P, js_P)$ e $S_F = (q_F, js_F)$. Se a ação é *Imprimir*, suponha que q_F depende de q_P e js_P , mas js_F não depende do estado anterior. Então a probabilidade de transição de estado para ação *Imprimir* pode ser escrita como:

$$P(S_F = (q_F, js_F) \mid S_P = (q_P, js_P)) = p(q_F \mid q_P, js_P) \times p(js_F)$$

e para especificar as transições de estado basta especificar cada termo no produto. A Figura 4.1 mostra a RB resultante.

Até aqui foram utilizados métodos já conhecidos na literatura e obtivemos um redução considerável, mas a quantidade de kj^2 números a especificar ainda pode ser inviável em muitas aplicações.

Uma das contribuições deste trabalho é propor uma compactação maior fazendo-se três suposições sobre o comportamento das variáveis e como elas se influenciam.

Uma primeira proposta a se considerar é a de que muitas das variáveis em um sistema dinâmico tem uma variação limitada, isto é, a cada transição ela pode

subir ou descer uma quantidade limitada de níveis.² Assim, muitas transições teriam probabilidade zero. Chamaremos esta propriedade de *variação limitada*.

Exemplo 3 *Considere a variável fila de impressão (q) em nosso modelo da impressora. Suponha que esta variável possui 10 categorias variando de 1 a 10. É plausível assumir que o nível da fila de impressão pode variar apenas dois níveis para cima ou para baixo a cada transição.*

Outra proposta estabelece que a probabilidade de uma variável v_i passar do nível x no instante presente para o nível $x + y$ no instante futuro, $P(v_{iF} = x + y \mid v_{iP} = x)$, independe do valor de x . Essa suposição pode ser utilizada para valores positivos e negativos de y e não se aplica apenas nas extremidades da escala da variável. Dependendo da natureza do sistema pode-se ainda utilizar simetrias, como por exemplo estabelecendo que a probabilidade de subir um nível seja igual a de descer a mesma quantidade. Esta propriedade será chamada de *variação homogênea*.

Exemplo 4 *Considere novamente a variável fila de impressão (q) de nosso modelo da impressora. Podemos dizer que a probabilidade da fila de impressão aumentar um nível é independente do nível atual:*

$$P(q_F = 2 \mid q_P = 1) = P(q_F = 3 \mid q_P = 2).$$

Isto não é verdade na extremidade, quando $q_P = 10$ e $P(q_F = 11 \mid q_P = 10) = 0$.

A última proposta considerada é a de que a probabilidade condicional de qualquer uma das variáveis tem uma relação proporcional com as variáveis condicionantes e essa relação depende da natureza do sistema. Para ilustrar este conceito considere um sistema onde a probabilidade da variável v_1 subir um nível é diretamente proporcional ao valor v_{2P} e inversamente proporcional ao valor de v_{3P} . Logo teríamos

$$P(v_{1F} = x + 1 \mid v_{2P} = x, v_{3P} = y) = kx/y$$

onde k é uma constante de normalização. Esta última propriedade será chamada de *variação proporcional*.

Exemplo 5 *Considere novamente o modelo de impressora apresentado no Exemplo 2. A probabilidade da fila de impressão (q) aumentar um nível é diretamente*

²Lembrando que as variáveis consideradas aqui assumem valores inteiros no intervalo $[1, j]$.

proporcional ao tamanho da tarefa de impressão (js): $p(q_F = q_P + 1 \mid q_P, js_P) = k \times js_P$, onde k é uma constante de normalização.

A utilização de todas essas propriedades, *variação limitada*, *variação homogênea* e *variação proporcional*, aliadas as RBs permite uma representação bastante compacta da distribuição de probabilidades de transição de estados. A aplicação destas idéias em um problema real é mostrada no capítulo seguinte.

4.4 Funções Recompensa e Custo

Em sistemas com grandes espaços de estados, como os de interesse nesse texto, faz-se necessário a representação das funções Recompensa e Custo também de forma compacta. A solução utilizada emprega as variáveis de estado para estabelecer funções que definem a recompensa de um estado e o custo de uma certa ação em um determinado estado.

Exemplo 6 *Considere novamente o modelo de impressora do Exemplo 2. Podemos escrever que o custo de executar a ação Print é proporcional ao nível da fila de impressão (q) e inversamente proporcional ao tamanho do job (js):*

$$C((q_P, js_P), Print) = k \times q_P / js_P.$$

4.5 Resolvendo o PDM

Conforme salientado diversas vezes, a metodologia apresentada nesta seção se aplica a problemas com grandes espaços de estados. A aplicação direta dos algoritmos tradicionais, como *iteração de valor* e *iteração de política*, na solução de PDMCOs com horizonte infinito descontado demanda uma capacidade de processamento computacional excessiva e muitas vezes inviabilizadora em problemas com essa característica. O número de operações por iteração é polinomial, $O(|S|^2|CA| + |S|^3)$, e o número de iterações necessário para convergência para uma política ótima também é polinomial em relação a $|S|$, $|CA|$ e $\frac{1}{1-\gamma}$ (BOUTILIER; DEAN; HANKS, 1999; LITTMAN; DEAN; KAEHLING, 1995).

Para tratar esta questão propomos abaixo duas soluções alternativas, cada uma com suas vantagens e desvantagens.

4.5.1 Solução Estática

Um importante subproduto do método proposto de construção do PDM é a facilidade de se identificar quão conectado é o espaço de estados. Em muitos problemas é possível fragmentar o espaço de estados e resolver cada um dos subespaços de forma independente.

Exemplo 7 *Considere um modelo de impressora constituído por duas variáveis: fila de impressão — queue — (q) e velocidade de impressão — velocity — (v). Suponha que a variável (q) tem 10 categorias e a variável (v) tem 5 categorias. Esse modelo captura impressoras com diferentes níveis de ocupação da fila de impressão e também com diferentes velocidades. Porém, a velocidade de impressão de uma impressora não varia, então é possível dividir o espaço de estados original com 50 estados em 5 problemas independentes com 10 estados cada um.*

Em outras situações, a solução do PDM através da divisão deste em PDMs menores e totalmente independentes não é possível. Porém, pode-se fragmentar o espaço de estados de maneira a obter-se um núcleo tal que as transições de estado possíveis ocorrem apenas no sentido de o sistema entrar neste núcleo e nunca de sair dele. A partir da solução deste núcleo, o espaço de estados é ampliado gradualmente e a cada etapa aproveita-se a solução pronta do espaço antigo.

Exemplo 8 *Considere que uma impressora é modelada por duas variáveis: fila de impressão — queue — (q) e nível de papel — tray — (tr) e um conjunto de ações $CA = \{\text{Imprimir}, \text{Não Imprimir}\}$. Suponha que a variável (q) tem 10 categorias e a variável (tr) tem 10 categorias. Isto produz um espaço de estados de tamanho 100. Para cada tarefa de impressão, a impressora deve decidir entre imprimi-la ou não. Caso a ação seja Imprimir, o nível da fila pode subir ou descer e o nível de papel pode diminuir. Caso a ação seja Não Imprimir, então (q) e (tr) podem apenas diminuir a partir do consumo das tarefas que já estão na fila. Aproveitando o fato que (tr) pode apenas diminuir seu nível, podemos reduzir o espaço de estados a um novo espaço onde esta variável está fixada em seu valor mais baixo, gerando um espaço de estados de tamanho 10. Após a solução deste PDM reduzido, monta-se um novo espaço de estados onde (tr) tem dois níveis possíveis. A solução já obtida para os estados do problema reduzido não muda e é utilizada na solução deste novo espaço de estados. Este processo é repetido adicionando-se níveis até que se obtenha a solução completa do problema original.*

A redução da carga computacional obtida pelos processos acima depende da fragmentação alcançada para o espaço de estados. Dividindo-se o espaço de estados S em n pedaços, então o número de operações por iteração diminui para $O(\frac{|S|^2|CA|}{n} + \frac{|S|^3}{n^2})$. Além disso, como a quantidade de iterações necessária para convergência é polinomial em relação a $|S|$, o número de iterações para convergência em cada fragmento é menor do que ao considerar todo o PDM de uma só vez. Apesar desta redução, este método pode não ser aplicável em problemas que exigem solução em tempo real. Neste caso, a aplicação do método restringe-se a situações onde é possível resolver o PDM e armazenar a função valor e a política obtida. Nos PDMs onde a função valor é separável no tempo e aditiva, a política é estacionária e o espaço necessário para o armazenamento da solução é $O(|S|)$ (BOUTILIER; DEAN; HANKS, 1999).

4.5.2 Solução Dinâmica

Outra importante contribuição deste trabalho é fornecer um novo método para se enfrentar a questão do tamanho do espaço de estados. Este método é baseado em uma transformação das variáveis de estado. Partimos de uma discretização inicial das variáveis onde estas contenham uma quantidade de categorias suficiente para capturar as informações necessárias do sistema, por exemplo a discretização obtida na Solução Estática. Então, faz-se um mapeamento, uma para uma, das variáveis iniciais para variáveis binárias em um novo modelo que chamaremos Solução Dinâmica.

Esta transformação reduz drasticamente o tamanho do espaço de estados. Porém, é necessário redefinir as probabilidades de transição de estados. Aqui as probabilidades de transição de estados são as probabilidades de alteração nas variáveis binárias do novo modelo. A inovação do método está em construir as expressões algébricas destas probabilidades utilizando as variáveis da discretização inicial. Isto reduz a perda de informação causada pela transformação, contudo a distribuição de probabilidades deixa de ser estacionária. Isto é, dependendo dos valores das variáveis da discretização inicial temos distribuições de probabilidades de transição de estados diferentes na Solução Dinâmica.

Exemplo 9 *Considere o nosso modelo de impressora do Exemplo 8. Vamos fazer uma nova discretização das variáveis (q) e (tr) tornando-as binárias. A característica importante da nova variável ($\hat{q}$) é informar se a fila de impressão está cheia ou não. Assim no novo modelo consideraremos que ($\hat{q} = 1$) se na discretização original ($q = 10$) e ($\hat{q} = 0$) para os demais valores de (q). Para*

a nova variável nível de papel ($\hat{tr}$), o importante é identificar se ainda há papel disponível, então no novo modelo ($\hat{tr} = 0$) se ($tr = 1$) — menor nível na discretização original — e ($\hat{tr} = 1$) para os demais valores de (tr). Podemos então escrever que a probabilidade do papel acabar, ou seja, ($\hat{tr}$) passar de 1 para 0 é proporcional ao tamanho da fila (q) e inversamente proporcional à quantidade de papel (tr):

$$P(\hat{tr}_F = 0 \mid \hat{tr}_P = 1) = k_1 q / tr$$

e a probabilidade do papel não acabar é:

$$P(\hat{tr}_F = 1 \mid \hat{tr}_P = 1) = k_1$$

onde k_1 é um fator de normalização e neste caso vale $k_1 = 1/(1 + q/tr)$.

É importante salientar que ao fazermos essa nova discretização a propriedade da *variação homogênea* deixa de valer, porém ainda podemos utilizar a propriedade da *variação proporcional*. Outro ponto importante é o fato que a distribuição de probabilidades de transição de estados deixa de ser estacionária e passa a variar de acordo com os valores das variáveis que regem as expressões das probabilidades de transição de estados. Com isso a cada instante é preciso recalcular a solução para o problema.

Este método de solução contrasta com o anterior, pois reduz de fato o tamanho do espaço de estados. A redução obtida depende da quantidade de categorias das variáveis na discretização original. Se houver poucas variáveis com muitos níveis a redução será grande, caso existam muitas variáveis com poucos níveis cada uma então a redução não será tão significativa. O ganho em tempo de processamento na solução do PDM depende desta redução do tamanho do espaço de estados. No Exemplo 9, a redução foi significativa, 25 vezes, e o tamanho diminuiu de 100 para 4 estados.

4.6 Considerações

Diversos autores têm sugerido diferentes formas de representação compacta para distribuições de probabilidades através de modelos fatorados, veja Guestrin et al. (2003), Cozman et al. (2004), Jaeger (1997), Glesner e Koller (1995), Poole (1993), porém é importante salientar que o modelo apresentado aqui foi desenvolvido a partir de um problema real. Assim, foi possível fazer suposições baseadas em características do problema em questão, formulando uma nova forma de compactação das probabilidades.

Outro ponto a se destacar é o fato que os algoritmos tradicionais demandam uma grande capacidade de processamento computacional na solução de PDMs com grandes espaços de estados como os que se pretende modelar através desta metodologia. Neste trabalho apresentamos dois métodos alternativos para se lidar com essa questão.

É importante salientar também que algoritmos, para PDMs, mais eficientes que os tradicionais já existem na literatura. Guestrin et al. (2003) apresentam métodos eficientes para solução aproximada de PDMs fatorados. Bonet e Geffner (2006) apresentam um método eficiente para se encontrar políticas parcialmente ótimas a partir de um estado inicial. Esta solução é baseada em métodos de busca com heurísticas atualizadas ao longo do processo usando-se programação dinâmica.

5 Aplicação de Processos de Decisão de Markov na Administração de Impressoras

5.1 Introdução

As técnicas desenvolvidas neste trabalho e apresentadas no capítulo anterior foram inspiradas pela análise de um problema real da indústria. Neste capítulo são mostrados dois modelos obtidos a partir da aplicação destas técnicas no problema analisado.

Os modelos foram concebidos em conjunto com especialistas da empresa HP-Brasil a partir de discussões a respeito das características mais importantes de uma impressora e de quais relações essas características mantêm entre si ao longo do tempo. Baseado nas conclusões obtidas foi possível construir modelos probabilísticos do comportamento dinâmico de impressoras.

A motivação da HP-Brasil para o desenvolvimento de tal solução é que atualmente não existe um algoritmo descentralizado que atue em impressoras de um grupo de forma que elas compartilhem os seus recursos na execução de tarefas de impressão. Em uma solução dessa natureza é desejável flexibilidade, ou seja, que esta solução mantenha o seu funcionamento normal com a entrada ou a saída de qualquer uma das impressoras membro do grupo ou “cluster”.

Com o objetivo de atender estas necessidades foi utilizado um sistema de alocação de tarefas de impressão baseado em um leilão. Para cada tarefa de impressão recebido pelo grupo todas as impressoras do mesmo são informadas e então cada uma faz o seu lance.

O processo de alocação de uma tarefa de impressão corresponde à escolha da impressora melhor qualificada dentro de um grupo de impressoras para executar

esta tarefa. Esta determinação depende de diversos fatores como a condição atual de cada impressora, as características da tarefa de impressão e as preferências do usuário para aquela tarefa de impressão.

O método utilizado para calcular o lance, nota, de cada impressora, baseada nestas características, foi o Processo de Decisão de Markov. A motivação para o uso desta técnica está no caráter dinâmico e estocástico que as variáveis envolvidas no problema mantêm. Isto será melhor apreendido no decorrer deste capítulo.

5.2 Funcionamento do Sistema

Um grupo de impressoras consiste de um conjunto de impressoras conectadas em rede e compartilhando recursos. O grupo pode ser acessado através de qualquer uma das impressoras que o constitui.

O processamento de cada tarefa de impressão pelo grupo de impressoras ocorre da seguinte forma:

1. O usuário envia uma tarefa de impressão através de um PC para qualquer uma das impressoras do grupo (P0);
2. P0 recebe a tarefa de impressão e repassa as características desta para as demais impressoras do grupo;
3. Cada uma das impressoras P_i , inclusive P0, processa as informações da tarefa de impressão e de seu estado, verificando o valor e a política correspondente ao estado atual do seu PDM. Em seguida, cada P_i envia sua correspondente ação e valor para P0;
4. P0 seleciona a maior nota entre as impressoras cuja ação é *Imprimir* e envia a tarefa de impressão junto com uma ordem de execução para a impressora correspondente. Se todas as impressoras escolherem a ação *Não Imprimir*, ainda assim a impressora com a maior nota é a escolhida. Em caso de empate a impressora que informou a nota primeiro vence.

É importante salientar que cada impressora é modelada pelo seu PDM e a interação entre elas ocorre apenas através da nota gerada pelo PDM de cada impressora. Assim o problema é modelado de forma descentralizada e não existe uma superestrutura tentando controlar todo o grupo.

5.3 Modelagem do Sistema Utilizando PDMs: Solução Estática

Na construção de um PDM é necessário especificar um espaço de estados S , um conjunto de ações CA , uma matriz $T(s, s', a_i)$ para cada ação a_i em CA , contendo as probabilidades de transição de estado dado que a ação a_i foi executada, uma função prêmio $R : S \rightarrow \mathbb{R}$ e uma função custo $C : S \times CA \rightarrow \mathbb{R}$. A partir desta estrutura é possível determinar de maneira ótima o valor que cada estado possui e qual é a melhor ação a se executar buscando atingir como resultado a maior recompensa possível.

5.3.1 O Espaço de Estados

Conforme visto no capítulo anterior uma maneira usual de se determinar o estado em que o PDM se encontra é através das variáveis que caracterizam o sistema estudado.

Na modelagem de uma impressora foram utilizadas oito variáveis. Estas variáveis receberam nomes em inglês para facilitar o diálogo com os especialistas da HP-Brasil e são apresentadas abaixo.

- Q — queue — nível de ocupação da fila de impressão (em %);
- Tr — tray — nível de papel na impressora (em %);
- Tb — black toner — nível de tinta preta (em %);
- Tc — color toner — nível de tinta colorida (em %). Considera-se aqui o menor nível entre o das três cores: azul, vermelho e amarelo;
- Js — job size — tamanho da tarefa de impressão em páginas;
- Jc — job color — requisição do usuário quanto à tarefa de impressão ser colorida ou não (em %);
- Ju — job urgency — requisição do usuário quanto à tarefa de impressão ser urgente ou não (em %);
- V — printer velocity — velocidade de impressão da impressora em páginas por minuto.

Tabela 5.1: Variáveis do sistema e número de categorias de cada uma.

Variável	Número de níveis	Intervalo para os níveis	Variável indicadora do nível
Q	10	1, 2, ..., 10	q
Tr	10	1, 2, ..., 10	tr
Tb	10	1, 2, ..., 10	tb
Tc	10	1, 2, ..., 10	tc
Js	5	1, 2, ..., 5	js
Jc	2	0, 1	jc
Ju	2	0, 1	ju
V	5	1, 2, ..., 5	v

As variáveis do sistema Q , Tr , Tb , Tc , Jc e Ju são contínuas e para construirmos o espaço de estados foi feita uma discretização com o número de categorias de cada uma destas variáveis mostrado na Tabela 5.1. Na quarta coluna desta tabela estão as correspondentes variáveis discretas e na terceira coluna os valores que estas podem assumir. As variáveis Js e V não são contínuas, mas foram mapeadas para as variáveis js e v com 5 categorias cada uma.

Esta configuração de variáveis resulta em um espaço de estados com 10^6 ($= 10 \times 10 \times 10 \times 10 \times 5 \times 2 \times 2 \times 5$) estados.

A Tabela 5.2 mostra o mapeamento das variáveis contínuas Q , Js e Ju para as variáveis discretas, q , js e ju que indicam o nível dessas quantidades.

Considere a variável Q na segunda coluna da Tabela 5.2, enquanto Q assume valores entre 0 e 100 %, a variável q assume valores entre 1 e 10 de acordo com a tabela. O mesmo raciocínio vale para as variáveis contínuas Tr , Tb e Tc e suas correspondentes discretas tr , tb e tc . Na terceira coluna da Tabela 5.2 temos o mapeamento entre Js em páginas e js , se Js tiver mais de 100 páginas então js fica no nível 5. O mesmo mapeamento feito entre Js e js vale para V e v , porém a unidade de V é em páginas por minuto. Na última coluna da Tabela 5.2 temos o mapeamento entre Ju em % e ju que é binária. O mesmo mapeamento feito entre Ju e ju vale para Jc e jc .

5.3.2 Ações

As ações possíveis são $CA = \{Imprimir, Não Imprimir\}$.

Em cada leilão, cada impressora verifica qual é a melhor ação a executar, ou seja, qual ação trará um retorno maior em termos da função valor de acordo com

Tabela 5.2: Exemplos de particionamento das variáveis.

Q e Ju em % e Js em n.º de págs.	Nível da variável Q indicado pela variável q	Nível da variável Js indicado pela variável js	Nível da variável Ju indicado pela variável ju
0 a 10	1	1	0
11 a 20	2	1	0
21 a 30	3	2	0
31 a 40	4	2	0
41 a 50	5	3	0
51 a 60	6	3	1
61 a 70	7	4	1
71 a 80	8	4	1
81 a 90	9	5	1
91 a 100	10	5	1

o estado atual que é dado pelas variáveis vistas acima.

Então, a impressora envia sua ação e o valor do seu estado presente como lance, participando do leilão em curso. Em seguida aguarda um novo leilão, repetindo o processo. O valor do estado presente S_P da impressora é dado pela função valor ótima $V^*(S_P)$ obtida na solução do PDM.

5.3.3 Probabilidades de Transição de Estado

O problema modelado necessita de duas matrizes de transição, uma para cada ação, com dimensão $10^6 \times 10^6$ cada uma, ou seja, 10^{12} números a especificar para cada ação. Isto torna necessária uma representação compacta dessas matrizes. Uma parte da solução do problema é a utilização de RBs.

Na representação das probabilidades de transição de estado por RBs trabalhamos com as variáveis que definem o estado do sistema, buscando identificar independências entre estas.

De maneira geral a probabilidade de transição de estado dada uma ação específica pode ser escrita como:

$$\begin{aligned}
 P(S_F | S_P) = & P(q_F | S_P)P(tr_F | S_P)P(tb_F | S_P) \\
 & P(tc_F | S_P)P(js_F | S_P)P(ju_F | S_P)P(jc_F | S_P)P(v_F | S_P)
 \end{aligned} \tag{5.1}$$

onde, S_P denota a n-tupla $(q_P, tr_P, tb_P, tc_P, js_P, ju_P, jc_P, v_P)$ que representa o estado presente e S_F denota a n-tupla $(q_F, tr_F, tb_F, tc_F, js_F, ju_F, jc_F, v_F)$ que

Figura 5.1: (a) RB para ação *Não Imprimir* e (b) RB para ação *Imprimir*.

representa o estado futuro¹.

Assim, cada termo do produto na Equação (5.1) contribui com 10×10^6 , 10×10^6 , 10×10^6 , 10×10^6 , 5×10^6 , 2×10^6 , 2×10^6 , 5×10^6 números a especificar respectivamente. Isto reduz o número total de números a especificar para construir cada matriz de transição a 54×10^6 .

Porém considerando as independências entre as variáveis podemos reduzir ainda mais a quantidade de números a especificar. Na Figura 5.1 (a) abaixo temos a RB para a ação *Não Imprimir*.

Nesta rede vemos que muitas variáveis são independentes e que nenhuma das variáveis que definem S_F depende de todas as que definem S_P . Considerando esse fato podemos escrever a probabilidade de transição de estado, dada a ação *Não Imprimir*, pela Equação (5.2). Nesta expressão vemos que cada termo do produto contribui respectivamente com $10 \times 10 \times 10 \times 5$, $10 \times 10 \times 10 \times 5$, $10 \times 10 \times 10 \times 10 \times 5$, $10 \times 10 \times 10 \times 10 \times 5$, 5 , 2 , 2 , 5×5 números a especificar. Isto resulta num total de 110.034 números a especificar.

$$\begin{aligned}
 P(S_F | S_P) = & P(q_F | q_P, tr_P, v_P) \times P(tr_F | q_P, tr_P, v_P) \times \\
 & P(tb_F | q_P, tr_P, tb_P, v_P) \times P(tc_F | q_P, tr_P, tc_P, v_P) \times \\
 & P(js_F) \times P(jc_F) \times P(ju_F) \times P(v_F | v_P)
 \end{aligned} \tag{5.2}$$

¹Lembrando que os instantes de interesse para se verificar o estado atual são aqueles nos quais a impressora averigua sua ação e nota para uma tarefa de impressão.

Na Figura 5.1 (b) é apresentada a RB para a ação *Imprimir*.

A Equação (5.3) mostra como calcular as probabilidades de transição de estado dada a ação *Imprimir*. Nesta expressão vemos que cada termo do produto contribui respectivamente com $10 \times 10 \times 10 \times 5 \times 5$, $10 \times 10 \times 10 \times 5 \times 5$, $10 \times 10 \times 10 \times 10 \times 5 \times 5$, $10 \times 10 \times 10 \times 10 \times 5 \times 5 \times 2$, 5, 2, 2, 5×5 números a especificar. Isto resulta num total de 800.034 números a especificar.

$$\begin{aligned}
 P(S_F | S_P) = & P(q_F | q_P, tr_P, js_P, v_P) \times P(tr_F | q_P, tr_P, js_P, v_P) \times \\
 & P(tb_F | q_P, tr_P, tb_P, js_P, v_P) \times \\
 & P(tc_F | q_P, tr_P, tc_P, js_P, jc_P, v_P) \times \\
 & P(js_F) \times P(jc_F) \times P(ju_F) \times P(v_F | v_P)
 \end{aligned} \tag{5.3}$$

Vemos que a quantidade de números a especificar para se obter a distribuição de probabilidades condicional ainda é muito grande. Porém, pode-se reduzir esse número fazendo-se uso de algumas propriedades adicionais conforme proposto no capítulo anterior.

Considere por exemplo a variável q , digamos que q no instante presente, ou seja, q_P , está no nível x (um valor entre 1 e 10). Assim, q no instante futuro, q_F , pode estar nos níveis $x - k$, $x - k + 1$, ..., x , ..., $x + j - 1$, $x + j$, onde $x - k = 0$ e $x + j = 10$. Utilizando-se a propriedade de *variação homogênea* resulta que a probabilidade da variável q passar do nível x no instante presente para o nível $x + i$ no instante futuro, $P(q_F = x + i | q_P = x)$, independe do valor de x . Assim teríamos, por exemplo: $P(q_F = 2 | q_P = 1) = P(q_F = 3 | q_P = 2) = P(q_F = 4 | q_P = 3)$. Esta propriedade é extensível às variáveis q , tr , tb e tc .

A propriedade de *variação limitada* trata de quantos níveis uma dessas variáveis pode subir ou descer entre dois instantes consecutivos. Lembrando que os instantes de interesse são aqueles onde a impressora gera a sua nota para uma tarefa de impressão, isto é, os instantes dos leilões. Supõe-se aqui que a variável q pode subir ou descer no máximo dois níveis entre dois instantes consecutivos. As variáveis tr , tb e tc podem apenas descer no máximo dois níveis ou ficar constantes. Supõe-se também que a velocidade de impressão, v , de uma impressora não se altera entre dois leilões.

A última propriedade utilizada, a de *variação proporcional*, estabelece que a probabilidade condicional de qualquer uma das variáveis tem uma relação proporcional com as variáveis condicionantes. Por exemplo, a probabilidade do nível da fila aumentar é diretamente proporcional ao tamanho da tarefa de impressão e inversamente proporcional à velocidade de impressão.

Tabela 5.3: Probabilidades condicionais para variável q .

Ação <i>Imprimir</i>			Ação <i>Não Imprimir</i>		
q_F	$P(q_F/q_P, tr_P, js_P, v_P)$		q_F	$P(q_F/q_P, tr_P, v_P)$	
	$tr_P = 1$	$tr_P \neq 1$		$tr_P = 1$	$tr_P \neq 1$
$q_P + 2$	$k_0 js_P$	$k_1 js_P / v_P$			
$q_P + 1$	$4k_0 js_P$	$2k_1 js_P / v_P$			
q_P	$5k_0 / js_P$	$4k_1 js_P v_P$	q_P	1	$5k_2 / v_P$
$q_P - 1$	0	$2k_1 v_P / js_P$	$q_P - 1$	0	$4k_2 v_P$
$q_P - 2$	0	$k_1 v_P / js_P$	$q_P - 2$	0	$k_2 v_P$

Tabela 5.4: Probabilidades condicionais para variável tr .

Ação <i>Imprimir</i>		Ação <i>Não Imprimir</i>	
tr_F	$P(tr_F/q_P, tr_P, js_P, v_P)$	tr_F	$P(q_F/q_P, tr_P, v_P)$
tr_P	$70k_3 / ((q_P + js_P)v_P)$	tr_P	$70k_4 / (q_P v_P)$
$tr_P - 1$	$2k_3 (q_P + js_P)v_P$	$tr_P - 1$	$2k_4 q_P v_P$
$tr_P - 2$	$k_3 (q_P + js_P)v_P$	$tr_P - 2$	$k_4 q_P v_P$

Usando-se estas três propriedades e as relações de independência obtidas anteriormente pelas RBs pode-se representar cada fator do produto que define a probabilidade de transição de estado condicionado no estado atual e na ação pelas Tabelas 5.3, 5.4, 5.5, 5.6 e 5.7.

Para exemplificar como ler as tabelas considere a Tabela 5.3 e suponha que queremos calcular o termo $P(q_F \mid q_P, tr_P, js_P, v_P)$ dada a ação *Imprimir*. Se $tr_P = 1$, então q_F assume os níveis $q_P + 2$, $q_P + 1$ e q_P , com probabilidades $k_0 js_P$, $4k_0 js_P$ e $5k_0 / js_P$ respectivamente. Se $tr_P \neq 1$, então q_F assume os níveis $q_P + 2$, $q_P + 1$, q_P , $q_P - 1$ e $q_P - 2$ com probabilidades $k_1 js_P / v_P$, $2k_1 js_P / v_P$, $4k_1 js_P v_P$, $2k_1 v_P / js_P$ e $k_1 v_P / js_P$ respectivamente. Este raciocínio está resumido nas três primeiras colunas da Tabela 5.3. As expressões algébricas que definem as probabilidades possuem diferentes pesos numéricos, os quais foram obtidos a partir de discussões com especialistas, como por exemplo o fator 4 na expressão $4k_1 js_P v_P$. Na mesma tabela são representadas também as probabilidades condicionais dada a ação *Não Imprimir*.

O mesmo raciocínio pode ser aplicado na leitura das Tabelas 5.4, 5.5, 5.6 e 5.7 para as demais variáveis.

As demais probabilidades independem da ação escolhida e das outras variáveis e valem: $P(js_F = x) = 1/5$, $P(jc_F = y) = \frac{1}{2}$ e $P(ju_F = z) = \frac{1}{2}$, onde x , y e z estão dentro dos valores permitidos para as variáveis js_F , jc_F e ju_F respectiva-

Tabela 5.5: Probabilidades condicionais para variável tb .

Ação Imprimir			Ação Não Imprimir		
tb_F	$P(tb_F/q_P, tr_P, tb_P, js_P, v_P)$		tb_F	$P(tb_F/q_P, tr_P, tb_P, v_P)$	
	$tr_P = 1$	$tr_P \neq 1$		$tr_P = 1$	$tr_P \neq 1$
tb_P	1	$70k_5/((q_P + js_P)v_P)$	tb_P	1	$70k_6/(q_P v_P)$
$tb_P - 1$	0	$2k_5(q_P + js_P)v_P$	$tb_P - 1$	0	$2k_6 q_P v_P$
$tb_P - 2$	0	$k_5(q_P + js_P)v_P$	$tb_P - 2$	0	$k_6 q_P v_P$

Tabela 5.6: Probabilidades condicionais para variável tc .

Ação Imprimir				Ação Não Imprimir		
tc_F	$P(tc_F/q_P, tr_P, tc_P, js_P, jc_P, v_P)$			tc_F	$P(tc_F/q_P, tr_P, tc_P, v_P)$	
	$tr_P = 1$	$tr_P \neq 1$			$tr_P = 1$	$tr_P \neq 1$
		$jc_P = 0$	$jc_P = 1$			
tc_P	1	$\frac{70k_7}{q_P v_P}$	$\frac{70k_8}{(q_P + js_P)v_P}$	tc_P	1	$\frac{70k_9}{q_P v_P}$
$tc_P - 1$	0	$2k_7(q_P v_P)$	$2k_8(q_P + js_P)v_P$	$tc_P - 1$	0	$2k_9 q_P v_P$
$tc_P - 2$	0	$k_7(q_P v_P)$	$k_8(q_P + js_P)v_P$	$tc_P - 2$	0	$k_9 q_P v_P$

Tabela 5.7: Probabilidades condicionais para variável v .

Ação Imprimir		Ação Não Imprimir	
v_F	$P(v_F/v_P)$	v_F	$P(v_F/v_P)$
v_P	1	v_P	1

mente.

Mais duas considerações são feitas a respeito das Tabelas 5.3 a 5.7:

- $k_0, k_2, \dots, k_9$ são fatores para normalização das probabilidades condicionais;
- tomando a variável q na Tabela 5.3, caso q_P valesse 1, então q_F não poderia assumir os valores $-1(= q_P - 2)$ e $0(= q_P - 1)$. A probabilidade de transição para estes valores são acumuladas no valor mais próximo permitido, que neste caso seria 1. Assim temos:

$$P(q_F = 1/q_P = 1) = 4k_1j_{s_P}v_P + 2k_1v_P/j_{s_P} + k_1v_P/j_{s_P}$$

O mesmo raciocínio se aplica para todas as variáveis tanto nos seus limites superior quanto inferior.

5.3.4 Função Recompensa

Em princípio não parece interessante acrescentarmos uma função recompensa no nosso problema, pois não há um estado ou um conjunto de estados objetivo que deva receber uma recompensa ao serem atingidos.

5.3.5 Função Custo

Tanto a impressão quanto a não impressão de uma tarefa de impressão devem gerar um custo. Abaixo são apresentadas duas expressões para se gerar o custo de cada ação em cada estado em função das variáveis de estado da impressora.

A função custo para ação *Imprimir* pode ser analisada da seguinte forma:

$$C(\text{Imprimir}) = \frac{j_{c_P}}{t_{c_P}} + \frac{j_{u_P}q_Pj_{s_P}}{tr_Ptb_Pv_P} + \frac{1}{tb_P}$$

- se a tarefa de impressão for “colorida”, então a primeira parcela, $\frac{j_{c_P}}{t_{c_P}}$, faz o custo subir na medida em que o nível de tinta colorida diminui;
- a segunda parcela, $\frac{j_{u_P}q_Pj_{s_P}}{tr_Ptb_Pv_P}$, faz o custo subir, se a tarefa de impressão for “urgente”, na medida em que a fila e o tamanho da tarefa de impressão aumentam e toner, papel e velocidade de impressão diminuem;
- a terceira parcela, $\frac{1}{tb_P}$, atribui maiores custos para impressoras com menos toner.

A análise da função custo para ação *Não Imprimir* é semelhante:

$$C(NãoImprimir) = j_{cptc_P} + \frac{tr_P tb_P v_P}{q_P j_{s_P}}$$

- a primeira parcela, j_{cptc_P} , aumenta o custo proporcionalmente ao nível de toner colorido se a tarefa de impressão for colorido;
- a segunda parcela, $\frac{tr_P tb_P v_P}{q_P j_{s_P}}$, torna o custo tanto maior quanto maior for a quantidade de recursos disponíveis: toner, papel, velocidade de impressão, etc.

5.3.6 Resolvendo o PDM

Nosso objetivo é encontrar uma política que minimize o custo de manter uma impressora operando continuamente. Assim, o horizonte do problema é infinito.

Os algoritmos tradicionais para tais problemas têm dificuldade significativa em tratar PDMs com 10^6 estados. Mesmo as soluções baseadas em modelos fatorados não apresentam grande eficiência nesta escala e fornecem apenas soluções aproximadas, conforme salientado no fim do Capítulo 4.

Além disso, podemos observar que a representação desenvolvida neste trabalho não é apenas uma técnica descritiva, mas o seu uso pode simplificar a busca de uma política ótima. Quando o PDM é construído utilizando propriedades comuns ao domínio, é fácil identificar regularidades que podem ser exploradas na sua solução. Assim, esta técnica fornece um importante subproduto.

Analizando o PDM da impressora nota-se que este pode ser dividido em cinco PDMs independentes. Isto é possível, pois as velocidades de impressão das impressoras não se alteram ao longo do tempo. Assim, cada novo PDM tem um valor fixo para a variável v . O conceito usado aqui é o mesmo apresentado no Exemplo 7 do Capítulo 4.

Além disso, exploramos o fato que as variáveis tr , tb e tc podem apenas diminuir o seu valor. Assim, começamos a solução do problema fixando o valor dessas variáveis em seu nível mais baixo. Esta é a aplicação do conceito apresentado no Exemplo 8 do Capítulo 4. Ao fazê-lo, o espaço de estados resultante possui apenas 200 estados e pode ser resolvido pelos algoritmos tradicionais como o de iteração de valor. O próximo passo é “alargar” o espaço de estados permitindo que uma das variáveis assuma um nível mais alto. Agora, usamos o resultado inicial para resolver o PDM ampliado. Este processo é repetido até que todo o PDM seja resolvido. É importante notar que cada nova categoria adiciona 200

estados ao espaço de estados, porém apenas esses novos estados ainda não foram resolvidos em cada passo.

Este processo é ordens de magnitude mais rápido do que tentar resolver todo o PDM de uma só vez e ainda assim conserva a otimalidade. O número de operações por iteração no método tradicional fica $O((10^6)^2 \times 2 + (10^6)^3) = O(10^{18})$. Fragmentando o espaço de estados, temos 5×10^3 fragmentos com 200 estados e o número de operações por iteração em cada fragmento fica $O((200)^2 \times 2 + (200)^3) = O(10^6)$. Considerando o número de operações em todos os fragmentos em cada iteração temos $O((10^6)^6 \times 5 \times (10^3)) = O(10^{10})$ operações. Além disso, o número de iterações necessário para convergência de cada fragmento também é significativamente menor do que o número de iterações necessário para a convergência do PDM todo, pois esta quantidade é polinomial em relação $|S|$ e o tamanho de cada fragmento é 5×10^3 menor do que do o espaço de estados do PDM todo.

Apesar de reduzir significativamente a carga computacional necessária para a solução do PDM, este processo não permite sua resolução em tempo real. Assim, é necessário o cálculo e armazenamento prévio da política e da função valor resultantes. Vimos que o espaço necessário para armazenamento demanda uma quantidade $O(|S|)$. Cada impressora tem de armazenar a solução para uma velocidade fixa, resultando numa quantidade de memória $O(2 \times 10^5)$. Experimentalmente verificou-se que foram necessários 9 MBytes de memória para o armazenamento da solução de cada impressora. Esta solução foi armazenada na forma: (estado;ação;valor do estado), onde 1 indica a ação *Imprimir* e 0 a ação *Não Imprimir*. Exemplo:

$$((6, 10, 1, 7, 4, 1, 0, 5); 1; -1.00853334996).$$

5.4 Modelagem do Sistema Utilizando PDMs: Solução Dinâmica

Conforme apresentado no capítulo anterior, na abordagem dinâmica o tamanho do espaço de estados é reduzido sensivelmente fazendo-se uma nova discretização que torna todas as variáveis ou tantas quanto possível binárias. Contudo, as expressões das probabilidades de transição de estado continuam sendo funções da discretização inicial.

Em consequência da redução do tamanho do espaço de estados temos também uma grande redução das transições possíveis, o que por sua vez diminui a quantidade de expressões de probabilidades que se precisa especificar. Outra con-

Tabela 5.8: Variáveis binárias e correspondência com a antiga discretização.

	$\hat{q}$	$\hat{tr}$	$\hat{tb}$	$\hat{tc}$	$\hat{js}$	$\hat{jc}$	$\hat{ju}$
0	$q: 1 \text{ à } 9$	$tr: 1$	$tb: 1$	$tc: 1$	$js: 1 \text{ à } 2$	$jc: 0$	$ju: 0$
1	$q: 10$	$tr: 2 \text{ à } 10$	$tb: 2 \text{ à } 10$	$tc: 2 \text{ à } 10$	$js: 3 \text{ à } 5$	$jc: 1$	$ju: 1$

sequência importante já vista é o fato de as probabilidades de transição de estados não serem constantes e dependerem dos valores das variáveis da discretização inicial. Isto significa que temos um PDM não estacionário.

5.4.1 O Espaço de Estados

Indicaremos por $\hat{x}$ a nova variável binária correspondente a nova discretização da variável x . A Tabela 5.8 mostra como as variáveis $\hat{q}$, $\hat{tr}$, $\hat{tb}$, $\hat{tc}$, $\hat{js}$, $\hat{ju}$ e $\hat{jc}$ do nosso modelo de impressora foram mapeadas para as variáveis binárias $\hat{q}$, $\hat{tr}$, $\hat{tb}$, $\hat{tc}$, $\hat{js}$, $\hat{ju}$ e $\hat{jc}$. O mapeamento das variáveis $\hat{ju}$ e $\hat{jc}$ é direto, pois na discretização anterior estas já eram binárias. Para exemplificar a leitura da Tabela 5.8, considere a segunda coluna desta onde vemos que $\hat{q} = 0$ corresponde a q entre 1 e 9 e $\hat{q} = 1$ corresponde a $q = 10$.

Conforme notado anteriormente, uma impressora não altera sua velocidade ao longo do tempo. Assim, a variável v também não precisa ser rediscretizada, já que seu valor uma vez estabelecido fica fixo.

O novo espaço de estados tem agora $2^7 = 128$ estados. A redução do tamanho do espaço de estados foi de $2 \times 10^6 / 128 = 1562,5$ vezes. Este PDM tem uma dimensão que pode ser tratada pelo algoritmo de iteração de valor em tempo real na aplicação de impressoras. Analisando o número de operações por iteração temos: $O((128)^2 \times 2 + (128)^3) = O(10^6)$. É importante observar que o número de iterações necessárias também é reduzido de forma polinomial em relação a $|S|$.

Nessa modelagem com variáveis binárias indicamos apenas situações extremas da impressora. Assim, $\hat{q}$ indica se a fila está completamente cheia, $\hat{q} = 1$, ou não, $\hat{q} = 0$. $\hat{tr}$ indica se há papel, $\hat{tr} = 1$, ou não, $\hat{tr} = 0$. $\hat{tb}$ indica se há tinta preta, $\hat{tb} = 1$, ou não, $\hat{tr} = 0$. $\hat{tc}$ indica se há tinta colorida, $\hat{tc} = 1$, ou não, $\hat{tr} = 0$. $\hat{jc}$ indica se o usuário deseja uma impressão colorida, $\hat{jc} = 1$, ou não, $\hat{jc} = 0$. E $\hat{ju}$ indica se o usuário tem urgência para a impressão, $\hat{ju} = 1$, ou não, $\hat{ju} = 0$. Apenas a variável $\hat{js}$ não tem uma interpretação desse tipo, mas sua nova discretização não causa grande perda de informação, pois sua discretização anterior ainda é utilizada nas expressões das probabilidades de transição de estados.

Figura 5.2: (a) RB para ação *Não Imprimir* e (b) RB para ação *Imprimir*.

5.4.2 Ações

O conjunto de ações é exatamente o mesmo do modelo antigo $CA = \{Imprimir, Não Imprimir\}$.

5.4.3 Probabilidades de Transição de Estado

Aqui o objetivo é construir expressões que informem a probabilidade da impressora entrar ou sair de umas das situações extremas descritas acima. As novas variáveis binárias mantêm as mesmas relações de independência expressas nas RBs do modelo anterior, veja a figura Figura 5.2.

As probabilidades de transição de estado são dadas pelas equações abaixo. A Equação (5.5) mostra a expressão caso a ação *Imprimir* seja escolhida e a Equação (5.4) mostra a expressão no caso da ação *Não Imprimir*.

$$\begin{aligned}
 P(\hat{S}_F | \hat{S}_P) = & P(\hat{q}_F | \hat{q}_P, \hat{tr}_P, \hat{v}_P) \times P(\hat{tr}_F | \hat{q}_P, \hat{tr}_P, \hat{v}_P) \times \\
 & P(\hat{tb}_F | \hat{q}_P, \hat{tr}_P, \hat{tb}_P, \hat{v}_P) \times P(\hat{tc}_F | \hat{q}_P, \hat{tr}_P, \hat{tc}_P, \hat{v}_P) \times \\
 & P(\hat{js}_F) \times P(\hat{jc}_F) \times P(\hat{ju}_F) \times P(v_F | v_P)
 \end{aligned} \tag{5.4}$$

Tabela 5.9: Probabilidades condicionais para variável $\hat{q}$.

(a) Ação *Imprimir*.

$P(\hat{q}_F \hat{q}_P, \hat{tr}_P, \hat{js}_P, \hat{v}_P)$				
	$\hat{q}_P = 0$		$\hat{q}_P = 1$	
$\hat{q}_F$	$\hat{tr}_P = 0$	$\hat{tr}_P = 1$	$\hat{tr}_P = 0$	$\hat{tr}_P = 1$
0	k_0	k_1	0	$k_2 v_P / j s_P$
1	$k_0 j s_P q_P$	$k_1 j s_P q_P / v_P$	1	k_2

(b) Ação *Não Imprimir*.

$P(\hat{q}_F \hat{q}_P, \hat{tr}_P, \hat{v}_P)$				
	$\hat{q}_P = 0$		$\hat{q}_P = 1$	
$\hat{q}_F$	$\hat{tr}_P = 0$	$\hat{tr}_P = 1$	$\hat{tr}_P = 0$	$\hat{tr}_P = 1$
0	1	1	0	$k_3 v_P$
1	0	0	1	k_3

Tabela 5.10: Probabilidades condicionais para variável $\hat{tr}$.

(a) Ação *Imprimir*.

$P(\hat{tr}_F \hat{q}_P, \hat{tr}_P, \hat{js}_P, \hat{v}_P)$		
$\hat{tr}_F$	$\hat{tr}_P = 0$	$\hat{tr}_P = 1$
0	1	k_4
1	0	$k_4(j s_P + q_P) v_P / tr_P$

(b) Ação *Não Imprimir*.

$P(\hat{tr}_F \hat{q}_P, \hat{tr}_P, \hat{v}_P)$		
$\hat{tr}_F$	$\hat{tr}_P = 0$	$\hat{tr}_P = 1$
0	1	k_5
1	0	$k_5 q_P v_P / tr_P$

$$\begin{aligned}
P(\hat{S}_F | \hat{S}_P) = & P(\hat{q}_F | \hat{q}_P, \hat{tr}_P, \hat{js}_P, \hat{v}_P) \times P(\hat{tr}_F | \hat{q}_P, \hat{tr}_P, \hat{js}_P, \hat{v}_P) \times \\
& P(\hat{tb}_F | \hat{q}_P, \hat{tr}_P, \hat{tb}_P, \hat{js}_P, \hat{v}_P) \times \\
& P(\hat{tc}_F | \hat{q}_P, \hat{tr}_P, \hat{tc}_P, \hat{js}_P, \hat{jc}_P, \hat{v}_P) \times \\
& P(\hat{js}_F) \times P(\hat{jc}_F) \times P(\hat{ju}_F) \times P(v_F | v_P)
\end{aligned} \tag{5.5}$$

O próximo passo é especificar cada um dos termos dos produtos acima. As Tabelas 5.9, 5.10, 5.11 e 5.12 apresentam os termos referentes as variáveis $\hat{q}_F$, $\hat{tr}_F$, $\hat{tb}_F$ e $\hat{tc}_F$ respectivamente. Os termos k_1 a k_{10} que aparecem nas tabelas são fatores de normalização.

Os termos $P(\hat{js}_F)$, $P(\hat{jc}_F)$ e $P(\hat{ju}_F)$ são independentes da ação escolhida e das variáveis que definem o estado atual e todos estes termos valem 1/2.

Tabela 5.11: Probabilidades condicionais para variável $\hat{t}b$.

(a) Ação *Imprimir*.

$\mathbf{P}(\hat{\mathbf{t}}b_F \hat{\mathbf{q}}_P, \hat{\mathbf{tr}}_P, \hat{\mathbf{t}}b_P, \hat{\mathbf{j}}s_P, \hat{\mathbf{v}}_P)$				
	$\hat{t}b_p = 0$		$\hat{t}b_p = 1$	
$\hat{\mathbf{t}}b_F$	$\hat{tr}_P = 0$	$\hat{tr}_P = 1$	$\hat{tr}_P = 0$	$\hat{tr}_P = 1$
0	1	1	0	$k_6(q_P + j s_P)v_P/tb_P$
1	0	0	1	k_6

(b) Ação *Não Imprimir*.

$\mathbf{P}(\hat{\mathbf{t}}b_F \hat{\mathbf{q}}_P, \hat{\mathbf{tr}}_P, \hat{\mathbf{t}}b_P, \hat{\mathbf{v}}_P)$				
	$\hat{t}b_p = 0$		$\hat{t}b_p = 1$	
$\hat{\mathbf{t}}b_F$	$\hat{tr}_P = 0$	$\hat{tr}_P = 1$	$\hat{tr}_P = 0$	$\hat{tr}_P = 1$
0	1	1	0	$k_7 q_P v_P / tb_P$
1	0	0	1	k_7

Tabela 5.12: Probabilidades condicionais para variável $\hat{t}c$.

(a) Ação *Imprimir*.

$\mathbf{P}(\hat{\mathbf{t}}c_F \hat{\mathbf{q}}_P, \hat{\mathbf{tr}}_P, \hat{\mathbf{t}}c_P, \hat{\mathbf{j}}s_P, \hat{\mathbf{j}}c_P, \hat{\mathbf{v}}_P)$					
	$\hat{t}c_p = 0$			$\hat{t}c_p = 1$	
$\hat{\mathbf{t}}c_F$	$\hat{tr}_P = 0$	$\hat{tr}_P = 1$	$\hat{tr}_P = 0$	$\hat{tr}_P = 1$	
				$\hat{j}c = 0$	$\hat{j}c = 1$
0	1	1	0	$k_8 q_P v_P / tc_P$	$k_9(q_P + j s_P)v_P / tc_P$
1	0	0	1	k_8	k_9

(b) Ação *Não Imprimir*.

$\mathbf{P}(\hat{\mathbf{t}}c_F \hat{\mathbf{q}}_P, \hat{\mathbf{tr}}_P, \hat{\mathbf{t}}c_P, \hat{\mathbf{v}}_P)$				
	$\hat{t}c_p = 0$		$\hat{t}c_p = 1$	
$\hat{\mathbf{t}}c_F$	$\hat{tr}_P = 0$	$\hat{tr}_P = 1$	$\hat{tr}_P = 0$	$\hat{tr}_P = 1$
0	1	1	0	$k_{10} q_P v_P / tc_P$
1	0	0	1	k_{10}

Considerando o fato de a velocidade da impressora não variar estabelecemos que o termo $P(v_F | v_P)$ vale 1 caso $v_F = v_P$ e 0 em contrário.

5.4.4 Funções Recompensa e Custo

Assim como no modelo anterior, não há uma função Recompensa, pois não há estados objetivo.

As funções Custo são bastante semelhantes às do modelo estático e são apresentadas abaixo:

$$C(Imprimir) = \frac{j_{c_P}}{t_{c_P}} + \frac{j_{u_P q_P j_{s_P}}}{tr_P t b_P v_P} + \frac{1}{t b_P} + q_P j_{s_P}$$

$$C(NãoImprimir) = j_{c_P} t_{c_P} + \frac{tr_P t b_P v_P}{q_P j_{s_P}}$$

A única diferença é a parcela $q_P j_{s_P}$ na função Custo da ação *Imprimir*. Esta parcela foi adicionada para aumentar o custo desta ação nas impressoras com grande ocupação da fila de impressão mesmo que a tarefa não seja urgente. Esta parcela não está presente no modelo estático, pois foi uma alteração posterior a resolução deste modelo e esta resolução demanda um tempo considerável de processamento computacional.

5.5 Considerações

Neste capítulo foram apresentados dois modelos de impressora baseados em PDMs. Estes modelos representam uma impressora cada um, ou seja, temos um modelo descentralizado e não existe uma macro estrutura tentando controlar todo o grupo de impressoras. Esta forma de modelagem permite a entrada e saída de qualquer impressora no grupo sem paralisar o seu funcionamento, pois não há nenhuma entidade centralizadora.

Os dois modelos empregam RBs e as propriedades das variáveis de estado vistas no capítulo anterior para sua representação compacta. É importante salientar também que o algoritmo básico empregado foi o de *iteração de valor*. O resultado obtido na solução destes PDMs é globalmente ótimo para cada impressora, mas é localmente ótimo para o grupo de impressoras. Isto ocorre porque as impressoras interagem apenas através do sistema de leilão.

O primeiro modelo, chamado *Solução Estática*, possui um grande espaço de estados. Vimos que a solução deste PDM exigiu um procedimento especial que

explora regularidades existentes nas variáveis de estado. Este procedimento reduziu a carga computacional necessária em ordens de grandeza, mas ainda não possibilita sua solução em tempo real. Isto torna necessário o cálculo prévio e armazenamento da função valor e da política resultante.

O segundo modelo, chamado *Solução Dinâmica*, possui um espaço de estados muito menor podendo ser tratado em tempo real. Esta redução do tamanho do espaço de estados foi obtida a partir do abandono da propriedade de estacionariedade das probabilidades de transição de estados. Esta solução se torna interessante em impressoras onde a quantidade de memória disponível limita o armazenamento da função valor e da política do modelo anterior.

6 Resultados e Considerações Finais

6.1 Simulações

Para testar o desempenho dos modelos desenvolvidos foi construído um programa simulador. Este programa simula um ambiente onde as impressoras compartilham recursos através do sistema de leilão na execução das tarefas de impressão de diversos usuários.

As simulações empregaram quatro esquemas diferentes para a geração da nota no leilão. O primeiro esquema é baseado em uma heurística simples. Essa heurística é uma soma da quantidade de papel, de tinta e de quanto a fila está livre. O resultado da soma é empregado como nota. A quantidade de tinta é dada pela média dos níveis de tinta nas impressoras coloridas e pelo nível de tinta preta nas não coloridas.

Dois esquemas baseados em PDMs, um utilizando a Solução Estática e o outro empregando a Solução Dinâmica, foram usados. O último esquema utilizado para gerar a nota é baseado no modelo empregando Lógica Difusa apresentado no Anexo A.

A razão da utilização dos modelos empregando PDMs é o fato de esta ser a técnica baseada na teoria de probabilidades mais utilizada em problemas de planejamento envolvendo incertezas. O uso da Lógica Difusa se justifica, pois esta é a principal técnica para tratar problemas com incerteza que não emprega a teoria de probabilidades. Por último, o modelo chamado Heurística foi usado para servir como um patamar inferior na medida de desempenho. Este modelo foi considerado um patamar inferior de desempenho, pois é uma simplificação que não modela as incertezas existentes no comportamento dinâmico de uma impressora.

Os testes de desempenho foram realizados para estes 4 esquemas em 4 configurações diferentes. A Tabela 6.1 apresenta os resultados no caso com 10

usuários e 4 impressoras de diferentes velocidades. Duas dessas impressoras são coloridas e as outras duas imprimem apenas em preto. A Tabela 6.2 apresenta os resultados para a mesma configuração de impressoras, porém agora com 15 usuários.

Os resultados da terceira configuração estão na Tabela 6.3. Aqui foram utilizadas 4 impressoras coloridas com a mesma velocidade de impressão e 10 usuários. Na Tabela 6.4 estão os resultados para a mesma configuração de impressoras com 15 usuários.

A escolha destas configurações de teste tem as seguintes justificativas:

- Testar duas configurações de impressoras: uma contendo um grupo homogêneo de impressoras onde todas elas têm os mesmos recursos e outra onde as impressoras têm recursos diferentes;
- Testar duas situações de usuários: uma onde há um nível médio de carga de trabalho com 10 usuários e outra onde o grupo está sob uma carga maior com 15 usuários;
- Configurações com mais usuários ou impressoras não foram realizadas por limitações de disponibilidade de hardware.

As Tabelas 6.1, 6.2, 6.3 e 6.4 resumem as estatísticas dos testes. Temos de cima para baixo o número de tarefas: enviadas ao grupo de impressoras, impressas pelo grupo, recebidas pelo grupo contando as enviadas e as redirecionadas, as redirecionadas e as rápidas. As tarefas rápidas são aquelas com tempo de impressão abaixo da média. Logo abaixo constam as quantidades de recargas de papel e de tinta. Em seguida temos os tempos de impressão para as diferentes classes de tarefas. Na parte inferior das tabelas temos o número de tarefas com solicitação de cor e o percentual que realmente foi impresso em cores.

A análise dos dados mostra que o esquema baseado na heurística traz os melhores resultados de tempo para as tarefas com urgência no caso em que todas as impressoras são iguais. Porém, apresenta os piores resultados no quesito de atendimento das requisições de impressões coloridas.

A Figura 6.1 mostra uma comparação entre cada um dos esquemas de geração de nota no atendimento do quesito urgência de impressão. A Figura 6.2 mostra uma comparação entre estes esquemas no atendimento ao quesito impressão colorida, ou seja, qual a porcentagem das tarefas de impressão coloridas foram realmente impressas em cor e não apenas em preto e branco devido a falta de tinta

Tabela 6.1: Resultados da simulação com 10 usuários e 4 impressoras de diferentes velocidades.

Tarefas	Heurística	Lógica Difusa	PDM Estático	PDM Dinâmico
Enviadas	600	600	600	600
Impressas	600	600	600	600
Recebidas	986	1033	1052	1052
Redirecionadas	386	433	452	452
Rápidas	504	400	463	427
Recargas	Heurística	Lógica Difusa	PDM Estático	PDM Dinâmico
de Papel	153	153	154	154
de Tinta	28	29	26	28
Tempos (min)	Heurística	Lógica Difusa	PDM Estático	PDM Dinâmico
Máximo	192,5000	48,2000	7,4000	53,6167
Mínimo	0,2833	0,0500	0,0333	0,0500
Médio	4,8518	2,7253	0,7512	3,5522
Tempos para Urgentes (min)	Heurística	Lógica Difusa	PDM Estático	PDM Dinâmico
Máximo	192,0333	6,0167	1,9500	11,8167
Mínimo	0,3833	0,5333	0,4667	0,4000
Médio	15,9107	2,7917	0,7708	3,4693
Tempos para Coloridos (min)	Heurística	Lógica Difusa	PDM Estático	PDM Dinâmico
Máximo	192,5000	6,0167	3,7500	22,0667
Mínimo	0,3833	0,0500	0,3833	0,1167
Médio	13,7771	2,8922	0,9769	3,5811
Tarefas Coloridas	Heurística	Lógica Difusa	PDM Estático	PDM Dinâmico
Enviadas	290	291	283	287
% Atendida	73,45	99,31	100,00	99,30

Tabela 6.2: Resultados da simulação com 15 usuários e 4 impressoras de diferentes velocidades.

Tarefas	Heurística	Lógica Difusa	PDM Estático	PDM Dinâmico
Enviadas	900	900	900	900
Impressas	900	900	900	900
Recebidas	1477	1561	1585	1569
Redirecionadas	577	661	685	669
Rápidas	863	680	691	727
Recargas	Heurística	Lógica Difusa	PDM Estático	PDM Dinâmico
de Papel	230	231	231	231
de Tinta	45	42	40	44
Tempos (min)	Heurística	Lógica Difusa	PDM Estático	PDM Dinâmico
Máximo	1091,3167	30,2500	9,6833	140,5667
Mínimo	0,0167	0,1500	0,2333	0,0333
Médio	12,4496	2,2853	0,6426	4,6132
Tempos para Urgentes (min)	Heurística	Lógica Difusa	PDM Estático	PDM Dinâmico
Máximo	24,6833	8,5167	3,8167	5,0167
Mínimo	0,3833	0,3833	0,3167	1,5167
Médio	2,0877	1,6299	1,2304	2,7722
Tempos para Coloridos (min)	Heurística	Lógica Difusa	PDM Estático	PDM Dinâmico
Máximo	26,2500	4,1500	9,6833	6,7667
Mínimo	0,3167	0,3833	0,3167	1,5167
Médio	2,0600	1,2762	1,4571	3,2694
Tarefas Coloridas	Heurística	Lógica Difusa	PDM Estático	PDM Dinâmico
Enviadas	419	436	407	441
% Atendida	94,75	99,77	99,26	99,09

Tabela 6.3: Resultados da simulação com 10 usuários e 4 impressoras de velocidades iguais.

Tarefas	Heurística	Lógica Difusa	PDM Estático	PDM Dinâmico
Enviadas	600	600	600	600
Impressas	600	600	600	600
Recebidas	1015	1053	1045	1059
Redirecionadas	415	453	445	459
Rápidas	395	436	389	376
Recargas	Heurística	Lógica Difusa	PDM Estático	PDM Dinâmico
de Papel	151	152	153	154
de Tinta	29	29	28	29
Tempos (min)	Heurística	Lógica Difusa	PDM Estático	PDM Dinâmico
Máximo	7,3500	15,6333	11,0167	10,1500
Mínimo	0,5500	0,7000	0,6333	0,9833
Médio	2,0454	2,5182	2,7362	2,4278
Tempos para Urgentes (min)	Heurística	Lógica Difusa	PDM Estático	PDM Dinâmico
Máximo	3,1167	3,6000	11,0167	4,9500
Mínimo	0,7833	0,9333	0,8667	1,2833
Médio	1,6396	1,8468	3,3411	2,1903
Tempos para Coloridos (min)	Heurística	Lógica Difusa	PDM Estático	PDM Dinâmico
Máximo	4,6167	3,3000	3,3333	2,8333
Mínimo	0,8500	0,9333	1,1000	1,4000
Médio	1,7559	1,5773	1,7295	2,0042
Tarefas Coloridas	Heurística	Lógica Difusa	PDM Estático	PDM Dinâmico
Enviadas	277	266	268	296
% Atendida	94,58	94,36	99,25	100,00

Tabela 6.4: Resultados da simulação com 15 usuários e 4 impressoras de velocidades iguais.

Tarefas	Heurística	Lógica Difusa	PDM Estático	PDM Dinâmico
Enviadas	900	900	900	900
Impressas	900	900	900	900
Recebidas	1474	1537	1582	1577
Redirecionadas	574	637	682	677
Rápidas	554	571	535	556
Recargas	Heurística	Lógica Difusa	PDM Estático	PDM Dinâmico
de Papel	229	227	226	227
de Tinta	45	44	44	45
Tempos (min)	Heurística	Lógica Difusa	PDM Estático	PDM Dinâmico
Máximo	7,9667	15,0000	22,8167	12,8667
Mínimo	0,7833	0,6333	0,7000	0,7500
Médio	2,1274	3,4445	4,5874	2,8520
Tempos para Urgentes (min)	Heurística	Lógica Difusa	PDM Estático	PDM Dinâmico
Máximo	4,2167	7,9000	4,0667	2,8167
Mínimo	0,9333	0,9500	0,7000	1,0333
Médio	1,6961	2,0726	1,8389	1,7200
Tempos para Coloridos (min)	Heurística	Lógica Difusa	PDM Estático	PDM Dinâmico
Máximo	1,9500	7,9000	4,0667	3,6333
Mínimo	1,0167	1,0167	0,7000	0,8167
Médio	1,3424	2,7849	1,9679	1,8500
Tarefas Coloridas	Heurística	Lógica Difusa	PDM Estático	PDM Dinâmico
Enviadas	437	441	438	421
% Atendida	94,74	98,87	99,54	99,29

Figura 6.1: Tempos para tarefas com solicitação de urgência.

colorida no momento da solicitação.

Notamos que as soluções baseadas em PDMs não trazem os melhores resultados em todas as configurações testadas quanto ao quesito urgência de impressão, porém atendem quase em 100% as solicitações de impressões coloridas. As demais soluções não apresentam este mesmo desempenho neste último quesito e em certas configurações também não mostram os melhores resultados no quesito urgência.

Observamos também que a Solução Dinâmica é inferior à Estática no quesito tempo de impressão nas configurações com um grupo heterogêneo de impressoras, mas o inverso ocorre nas configurações com um grupo homogêneo de impressoras. Uma das razões para o desempenho inferior da Solução Dinâmica é o tempo necessário para cada impressora resolver seu PDM antes de obter sua nota para a tarefa de impressão do momento. Isto não ocorre na solução Estática, pois esta já tem a solução do PDM pronta no momento em que recebe a tarefa.

Figura 6.2: Atendimento das solicitações de impressão colorida.

6.2 Considerações Finais

Conforme ressaltado anteriormente, o trabalho realizado traz três contribuições importantes. A primeira delas é o desenvolvimento de uma nova metodologia para especificação de PDMs. Este método utiliza RBs combinadas com regularidades entre as variáveis e os estados na especificação das distribuições de probabilidades de transição de estados no PDM. Através destas regularidades as probabilidades condicionais de cada nó das RBs são especificadas de forma compacta usando relações entre as próprias variáveis do sistema.

A segunda contribuição é que todo o estudo realizado foi voltado para a solução de um problema real de automação comercial. O trabalho foi realizado em conjunto com especialistas da empresa HP-Brasil, buscando uma solução descentralizada para o problema de alocação de tarefas de impressão entre um conjunto de impressoras conectadas. A solução procurada faz com que as impressoras cooperem para que a satisfação dos seus usuários seja maximizada.

A terceira contribuição deste trabalho é apresentar o método chamado Solução Dinâmica de modelagem e resolução de PDMs. Através deste método é possível modelar problemas que a princípio teriam grandes espaços de estados de forma

reduzida. Isso permitiu também que nosso modelo fosse resolvido em tempo real. De fato, a Solução Dinâmica exige uma solução em tempo real, pois, conforme explanado, a sua distribuição de probabilidades de transição de estados não é estacionária.

Outro ponto importante deste trabalho é a utilização de duas abordagens bastante distintas para tratar de um problema envolvendo incertezas: uma utilizando PDMs e outra empregando Lógica Difusa. Observamos que através de PDMs foi possível construir um modelo mais simples e fiel a realidade do que usando a Lógica Difusa, pois os modelos com PDMs se basearam apenas na abstração de características da impressora, enquanto o modelo usando Lógica Difusa exigiu o ajuste de seus diversos parâmetros repetidas vezes em um processo iterativo até que o comportamento desejado foi alcançado.

Vale destacar ainda o efeito que o uso de expressões algébricas traz na compactação da representação de probabilidades. Isto pode ser percebido em nosso modelo de impressora. É importante lembrar que uma linguagem de especificação como a linguagem Probabilistic Planning Domain Definition Language (PPDDL) (LITTMAN; YOUNES, 2004) não possui esta capacidade. Sugerimos que se modifique esta linguagem para permitir o uso de expressões algébricas na especificação dos valores de probabilidades. Na versão atual da linguagem, os efeitos probabilísticos das ações têm a forma:

$$\text{probabilistic } \langle \text{probability} \rangle \langle \text{effect} \rangle$$

onde: $\langle \text{probability} \rangle$ é um número entre 0 e 1, e $\langle \text{effect} \rangle$ é o efeito. A sugestão proposta é que o termo $\langle \text{probability} \rangle$ acomode expressões algébricas que sejam funções dos próprios fluentes. Esta alteração daria a linguagem a capacidade de representar PDMs com grandes espaços de estados de forma mais simples conforme apresentado neste trabalho.

Anexo A – Um Contraponto Utilizando Lógica Difusa

A.1 Introdução

Neste anexo é apresentada uma solução alternativa para o problema de administração de impressoras já descrito no Capítulo 5. Da mesma forma que a solução baseada em PDMs, aqui também será adotado um sistema de alocação de tarefas de impressão baseado em um leilão. Para cada tarefa de impressão recebida pelo grupo, todas as impressoras do mesmo são informadas e então cada uma faz o seu lance. Este lance é uma nota que varia de 0 a 10.

O processo de alocação de uma tarefa de impressão corresponde à escolha da impressora melhor qualificada dentro de um grupo de impressoras para executar a tarefa. Esta determinação depende de diversos fatores como a condição atual de cada impressora, as características da tarefa de impressão e as preferências do usuário para aquela tarefa.

O contraponto apresentado aqui é que o método utilizado para calcular o lance, nota, de cada impressora, baseada nestas características, foi a Lógica Difusa.¹ A motivação para uso desta técnica está no caráter vago e impreciso das relações que as variáveis envolvidas no problema mantém. Isto será melhor apreendido no decorrer do texto abaixo.

A.2 Conceitos Básicos

A.2.1 Conjunto Difuso

Um conjunto difuso D é caracterizado pela função de pertinência $f_D(x)$ que associa cada ponto em U , universo de discurso da variável, a um número real no

¹Para uma discussão sobre Lógica Difusa veja (ZADEH, 1973; LEE, 1990a; LEE, 1990b).

intervalo $[0,1]$, com o valor de $f_D(x)$ em x representando o grau de pertinência de x em D , (ZADEH, 1965).

A operação chamada *fuzzificação* é realizada por um conjunto de transformações que produz o grau de pertinência de cada variável aos respectivos conjuntos difusos a partir da leitura do valor da variável no universo de discurso. Estas transformações são realizadas pelas funções de pertinência destes conjuntos.

A.2.2 Variáveis Linguísticas

A principal função das variáveis lingüísticas é fornecer um modo sistemático para a caracterização aproximada de um fenômeno complexo e mal definido. Os valores de uma variável lingüística são etiquetas de subconjuntos difusos de U , os quais têm a forma de frases ou sentenças em uma linguagem natural ou artificial.

Exemplo: Se U é a coleção de inteiros $U = \{1, 2, \dots, 100\}$ e idade é uma variável lingüística com etiquetas x , então os valores de x podem ser jovem, não jovem, muito jovem, não muito jovem, velho, não velho, não muito velho, etc.

A.2.3 Regras Difusas

São declarações da forma ‘SE A ENTÃO B’ que descrevem relações entre condições de duas variáveis lingüísticas, onde A e B são as condições atuais das variáveis lingüísticas. Exemplo: SE Toner é cheio ENTÃO Nota é alta.

A.2.4 Inferência e Defuzzificação

O processo de inferência corresponde ao cálculo do valor de pertinência de cada conjunto difuso da variável lingüística de saída. O método de inferência utilizado foi o proposto por Mandani (LEE, 1990b).

O processo de defuzzificação corresponde ao cálculo de um valor determinístico para a variável de saída a partir dos graus de pertinência dos conjuntos difusos desta variável. Exemplo: um sistema controlando a vazão em um duto deve fornecer um valor preciso para a abertura de uma válvula, como 90% que corresponde a algo como “bastante aberta”.

A.3 Metodologia

Para o desenvolvimento de qualquer aplicação utilizando Lógica Difusa é necessário determinar as variáveis envolvidas no problema e a partição destas variáveis em conjuntos difusos, ou seja, a especificação das funções de pertinência de cada conjunto. Além disso, é necessário especificar as regras que definem as relações entre as variáveis envolvidas no problema.

A partir daí, as leituras das variáveis de entrada do problema são convertidas para conjuntos difusos com as pertinências obtidas de acordo com as respectivas funções de pertinência estabelecidas (processo chamado de fuzzificação), e aplicando-se as regras chega-se ao grau de pertinência das variáveis de saída a cada conjunto difuso que as particiona (processo chamado de inferência). Por fim, é necessário fornecer um valor para cada variável de saída que possa ser entendido e aplicado, processo chamado de defuzzificação.

A maior parte do trabalho de desenvolvimento de um sistema utilizando Lógica Difusa se concentra em ajustar os parâmetros acima de modo que o sistema se comporte da maneira desejada.

A obtenção dos parâmetros para construção de um sistema difuso normalmente é conseguida a partir de um especialista ou de experimentos que permitem aferir as relações entre as variáveis do problema. No caso apresentado aqui essas relações foram obtidas a partir de discussões com especialistas da empresa HP-Brasil e de vários ajustes baseados em testes.

A.4 Cálculo da Nota

A.4.1 Variáveis Linguísticas

O cálculo da nota de cada impressora é baseado no estado atual da impressora, nas características da tarefa de impressão e nas preferências do usuário.

Para representar o estado atual da impressora são consideradas quatro variáveis de entrada:

- *Toner* - representando o nível de toner preto e podendo variar dentro do intervalo $[0, 100]\%$;
- *Papel* - representando o nível de papel e variando também no intervalo $[0, 100]\%$;

Figura A.1: Função de pertinência da variável linguística *Toner*.

- *Fila* - representando o nível de ocupação da fila de tarefas de impressão em execução na impressora e variando também no intervalo $[0, 100]\%$;
- *TonerColorido* - representando o nível de toner colorido e podendo variar dentro do intervalo $[0, 100]\%$.²

A única característica da tarefa de impressão representada é o seu tamanho e é considerado na variável *TamanhoJob*. Esta variável pode assumir valores no intervalo $[0, 100]$ páginas. Por hipótese foi assumido um tamanho máximo de 100 páginas por tarefa de impressão.

Por fim, foram utilizadas duas variáveis para representar as preferências do usuário:

- *Urgência* - informando quanto o fator tempo até o fim da impressão deve influenciar na nota. Esta variável pode assumir valores no intervalo $[0, 1]$;
- *JobColorido* - informando o desejo que o usuário tem de que o documento impresso seja colorido. Esta variável pode assumir valores no intervalo $[0, 1]$.

Todas as funções de pertinência das variáveis linguísticas de entrada são representadas abaixo. A Figura A.1 mostra a partição da variável linguística *Toner*. Seus valores linguísticos são: baixo, médio e cheio. O universo de discurso é $0 - 100\%$.

A Figura A.2 mostra a partição da variável linguística *Papel*. Seus valores linguísticos são: baixo, médio e cheio. O universo de discurso é $0 - 100\%$.

²Adotamos aqui a mesma simplificação do Capítulo 5 onde o valor da variável *TonerColorido* é considerado como sendo o menor valor das três cores de toner (azul, vermelho e amarelo). Assim, não é necessário considerar as três variáveis ao mesmo tempo.

Figura A.2: Função de pertinência da variável linguística *Papel*.

Figura A.3: Função de pertinência da variável linguística *Fila*.

A Figura A.3 mostra a partição da variável linguística *Fila*. Seus valores lingüísticos são: vazia, media, grande e cheia. O universo de discurso é 0 – 100%.

A Figura A.4 mostra a partição da variável linguística *TonerColorido*. Seus valores lingüísticos são: vazio e cheio. O universo de discurso é 0 – 100%.

A Figura A.5 mostra a partição da variável linguística *TamanhoJob*. Seus valores lingüísticos são: pequeno, médio e grande. O universo de discurso é 0 – 100%.

A Figura A.6 mostra a partição da variável linguística *Urgencia*. Seus valores lingüísticos são: baixa e alta. O universo de discurso é 0 - 1.

A Figura A.7 mostra a partição da variável linguística *JobColorido*. Seus valores lingüísticos são: não, indiferente e sim. O universo de discurso é 0 - 1.

Na Figura A.8 temos a variável linguística de saída, a *Nota*. Seus valores lingüísticos são: baixa, media e alta. O universo de discurso é 0 - 10. O valor desta variável representa quão qualificada uma impressora está para executar uma

Figura A.4: Função de pertinência da variável linguística *TonerColorido*.

Figura A.5: Função de pertinência da variável linguística *TamanhoJob*.

Figura A.6: Função de pertinência da variável linguística *Urgencia*.

Figura A.7: Função de pertinência da variável linguística *JobColorido*.

Figura A.8: Função de pertinência da variável linguística *Nota*.

determinada tarefa.

A.4.2 Base de Regras Difusas

A partir de discussões com especialistas da HP-Brasil chegou-se a um conjunto de características desejadas para o comportamento do grupo de impressoras. Baseado nestas características foi possível elaborar um conjunto de regras difusas. Segue abaixo o conjunto de regras obtidas e as suas justificativas.

Regras:

1. Se *Toner* é cheio então *Nota* é alta;
2. Se *Toner* é médio então *Nota* é média;
3. Se *Toner* é baixo então *Nota* é baixa;
4. Se *TonerColorido* é vazio e *JobColorido* é sim então *Nota* é baixa;

5. Se *Fila* é grande e *Urgencia* é alta então *Nota* é baixa;
6. Se *Fila* é média e *Urgencia* é alta então *Nota* é baixa;
7. Se *Papel* é baixo e *Urgencia* é alta então *Nota* é baixa;
8. Se *Papel* é médio e *TamanhoJob* é grande e *Urgencia* é alta então *Nota* é baixa;
9. Se *Fila* é cheia então *Nota* é baixa;
10. Se *Fila* é vazia e *Urgencia* é alta então *Nota* é alta.

As três primeiras regras premiam impressoras com um nível de toner mais alto. A finalidade dessa política é tentar utilizar as impressoras do grupo de forma uniforme. Isto permitiria a manutenção de todas as impressoras do grupo em um mesmo dia, o que representa economia. Por exemplo: trocar todos os toners no mesmo dia implica em uma única visita do técnico.

O desgaste do toner colorido não é tão relevante quanto o toner preto, pois é menos utilizado. Porém, o nível deste tornasse importante para uma tarefa de impressão em que o usuário tem um grande desejo de que esta tarefa seja colorida, justificando a quarta regra.

As regras 5, 6 e 10 relacionam a urgência da tarefa de impressão com o nível da fila da impressora à nota. A idéia é direcionar tarefas urgentes para impressoras com menor fila. Aqui cabe salientar pelo menos dois usos para a variável urgência: o primeiro seria onde, supondo usuários honestos, estes escolhem a urgência com bom senso, já o segundo seria num cenário onde a urgência está ligada a posição do usuário (por exemplo, o diretor de uma empresa tem prioridade sobre seus subalternos).

As regras 7 e 8 levam em conta o possível atraso gerado por falta de papel. Uma impressora com pouco papel não deve ter preferência na execução de uma tarefa de impressão com urgência, pois este tem uma chance considerável de acabar no meio da execução.

Por fim a regra 9 tenta evitar a perda de tarefas por estouro na fila de execução.

A.4.3 Um Exemplo de Aplicação das Regras

Na Figura A.9 temos um exemplo de aplicação das regras. Neste exemplo apenas as regras 1, 2, 5, 6 e 8 são ativadas.

Figura A.9: Exemplo de aplicação das regras de inferência.

Em cada regra onde os antecedentes estão ligados pelo conectivo *and*, o operador de intersecção é aplicado, retornando o mínimo dos graus de pertinência dos antecedentes. Este valor é então aplicado no conseqüente, ou seja, na variável *Nota* em relação à etiqueta lingüística presente na regra.

Em seguida faz-se a união, operador máximo, dos conseqüentes de cada regra. O resultado é uma nova função de pertinência para a saída. Esta função é mostrada à direita na última linha da última coluna da Figura A.9.

Por fim a partir da curva obtida faz-se a defuzzificação desta variável lingüística obtendo um valor crisp. Este valor será então repassado para impressora que originou a tarefa de impressão.

Como não há um procedimento sistemático para escolher uma estratégia de defuzzificação (LEE, 1990a; LEE, 1990b), o método do centro de área foi adotado, dado que o mesmo considera todo o universo de saída ponderado pelos graus de pertinência.

A.4.4 Análise dos Resultados

Nesta seção é apresentado o comportamento obtido para a variável de saída. Nesta análise as variáveis de entrada foram consideradas duas a duas mantendo o valor das demais fixo e igual à metade do seu universo de discurso. A apresentação dos gráficos abaixo justifica-se no fato de a modelagem empregando Lógica Difusa ser feita iterativamente a partir dos resultados obtidos na variável de saída. É o comportamento da variável de saída que orienta os ajustes a serem feitos nas partições das variáveis e nas regras. Apesar disso a imposição de um determinado comportamento de forma monotônica não é trivial.

Na Figura A.10 podemos observar a influência conjunta das variáveis de entrada *Urgência* e *Fila* na variável de saída. Como era de se esperar para *Urgência* alta (igual a 1) e *Fila* cheia (igual a 100%) a *Nota* é a mais baixa. Notamos também que a variável *Fila* só afeta a nota quando atinge seu extremo superior ou no caso em que a *Urgência* é relevante.

Na Figura A.11 vemos o comportamento da *Nota* de acordo com os valores das variáveis *Papel* e *Urgência*. Podemos observar que para um valor de *Fila* à medida que a *Urgência* aumenta há um decremento na *Nota*. Outro aspecto a ser notado é a influência conjunta de *Papel* baixo (próximo de zero) e *Urgência* alta (próximo de um) diminuindo ainda mais a *Nota*.

Na Figura A.12 podemos observar a influência conjunta das variáveis de en-

Figura A.10: *Nota* em função da *Urgência* e da *Fila*.

trada *Toner* e *Fila* na variável de saída. Como era de se esperar níveis mais baixos de *Fila* e mais altos de *Toner* fornecem maiores notas.

Na Figura A.13 podemos observar a influência conjunta das variáveis de entrada *TonerColorido* e *JobColorido* na variável de saída. Observa-se que estes parâmetros só têm relevância quando o usuário tem grande necessidade de que a tarefa de impressão seja colorido e ao mesmo tempo o *TonerColorido* está baixo.

A.5 Considerações

A modelagem apresentada aqui foi obtida através de discussões com especialistas da empresa HP-Brasil e de testes iterativos para se obter o comportamento desejado, o qual foi passado por estes especialistas. Após a definição das variáveis lingüísticas e de um conjunto inicial de regras, várias modificações nas partições das variáveis e ajustes nas regras foram testadas. Uma das grandes dificuldades na modelagem foi impor o comportamento desejado da variável de saída monotônico em relação as variáveis de entrada.

Comparando-se esta solução com a solução anterior baseada em PDMs, nota-se que aqui não é necessária a especificação de uma distribuição de probabilidades de transição de estado. Porém, há uma correspondência entre as duas metodologias. Em ambas definimos variáveis, chamadas de variáveis de estado no PDM e de variáveis lingüísticas na Lógica Difusa. Em seguida discretizamos as variáveis

Figura A.11: *Nota* em função do *Papel* e da *Urgência*.

Figura A.12: *Nota* em função da *Fila* e do *Toner*.

Figura A.13: *Nota* em função do *TonerColorido* e do *JobColorido*.

de estado, assim como particionamos as variáveis linguísticas em conjuntos difusos. Então, estabelecemos as funções custo e recompensa, as ações e as correspondentes distribuições de probabilidade de transição de estado no PDM, enquanto na Lógica Difusa definimos as regras que relacionam as variáveis linguísticas.

A diferença fundamental é que ao construir o modelo baseado em Lógica Difusa faz-se ajustes diretos nos parâmetros de forma que o comportamento ocorra conforme desejado. Essa abordagem na verdade é uma solução aproximada, ou seja, não ótima, obtida manualmente para o PDM. A construção do PDM é baseada nas características probabilísticas do sistema e sem acesso direto ao comportamento resultante do modelo construído. Esse comportamento é obtido somente quando a estrutura, após construída, é resolvida obtendo uma política e uma função valor.

Considerando o desempenho nos testes apresentados no Capítulo 6, observamos que o modelo baseado em Lógica Difusa apresenta um nível de atendimento das solicitação de impressão colorida com um nível bem próximo dos modelos baseados em PDM. Apenas nas configurações com impressoras iguais é que o desempenho foi ligeiramente inferior, mas não foi identificada uma razão para este comportamento. Quanto ao atendimento das solicitações urgentes, observamos que o desempenho médio considerando as 4 configurações foi similar ao desempenho da Heurística, inferior ao desempenho do PDM Estático e superior ao do PDM Dinâmico.

Referências Bibliográficas

- ALVARES, L. O.; SICHMAN, J. S. Introdução aos sistemas multiagentes. In: *Jornadas de Atualização em Informática (XVI JAI)*. Brasília, Brasil: Medeiros, C. M. B. (Ed.), 1997. p. 1–38.
- BACCHUS, F.; BOUTILIER, C.; GROVE, A. Rewarding behaviors. In: *Proceedings of the Thirteenth National Conference on Artificial Intelligence*. Portland, OR: AAAI, 1996. p. 1160–1167.
- BACCHUS, F.; BOUTILIER, C.; GROVE, A. Structured solution methods for non-Markovian decision processes. In: *Proceedings of the Fourteenth National Conference on Artificial Intelligence*. Providence, RI: AAAI, 1997. p. 112–117.
- BELLMAN, R. *Dynamic Programming*. Princeton, NJ: Princeton University Press, 1957.
- BERTSEKAS, D. P. *Dynamic Programming*. Englewood Cliffs, NJ: Prentice-Hall, 1987.
- BERTSEKAS, D. P.; TSITSIKLIS, J. N. *Neuro-dynamic Programming*. Belmont, MA: Athena, 1996.
- BONET, B.; GEFFNER, H. Learning Depth-First Search: A unified approach to heuristic search in deterministic and non-deterministic settings, and its application to MDPs. *16th Int. Conf. on Automated Planning and Scheduling*, Lake District, UK, 2006.
- BOUTILIER, C.; DEAN, T.; HANKS, S. Decision-theoretic planning: Structural assumptions and computational leverage. *Journal of Artificial Intelligence Research*, JAIR, p. 1–94, 1999.
- COZMAN, F. G. Generalizing variable elimination in Bayesian networks. *Workshop on Probabilistic Reasoning in Artificial Intelligence*, Atibaia, Brasil, 2000.
- COZMAN, F. G. et al. Propositional and relational Bayesian networks associated with imprecise and qualitative probabilistic assessments. In: *Conference on Uncertainty in Artificial Intelligence*. Banff, Canadá: AUAI Press, 2004. p. 104–111.
- DEAN, T.; KANAZAWA, K. A model for reasoning about persistence and causation. *Computational Intelligence*, v. 5, n. 3, p. 142–150, 1989.
- GLESNER, S.; KOLLER, D. Constructing flexible dynamic belief networks from first-order probabilistic knowledge bases. *ECSQARU*, p. 217–226, 1995.
- GUESTIN, C. et al. Efficient solution algorithms for factored MDPs. *Journal of Artificial Intelligence Research*, v. 19, p. 399–468, 2003.

- HOWARD, R. A. *Dynamic Programming and Markov Processes*. Cambridge, Massachusetts: MIT Press, 1960.
- JAEGER, M. Relational Bayesian networks. *UAI*, p. 266–273, 1997.
- JENSEN, F. V. *Bayesian Networks and Decision Graphs*. New York: Springer, 2001.
- LEE, C. C. Fuzzy logic in control systems: Fuzzy logic controller - part I. *IEEE Transactions on Systems, Man, and Cybernetics*, v. 20, n. 2, 1990.
- LEE, C. C. Fuzzy logic in control systems: Fuzzy logic controller - part II. *IEEE Transactions on Systems, Man, and Cybernetics*, v. 20, n. 2, 1990.
- LITTMAN, M. L.; DEAN, T. L.; KAEHLING, L. P. On the complexity of solving Markov decision problems. In: *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*. Montreal, Canadá: AUAI, 1995. p. 294–402.
- LITTMAN, M. L.; YOUNES, H. L. S. PPDDL1.0: An extension to PDDL for expressing planning domains with probabilistic effects. *International Planning Competition*, 2004.
- LUENBERGER, D. G. *Introduction to Linear and Nonlinear Programming*. Reading, Massachusetts: Addison-Wesley, 1973.
- LUENBERGER, D. G. *Introduction to Dynamic Systems: Theory, Models, and Applications*. New York: Wiley, 1979.
- MAHADEVAN, S. To discount or not to discount in reinforcement learning: A case study in comparing R-learning and Q-learning. *Proceedings of the Eleventh International Conference on Machine Learning*, New Brunswick, NJ, p. 164–172, 1994.
- MCCARTHY, J.; HAYES, P. J. Some philosophical problems from the standpoint of artificial intelligence. *Machine Intelligence*, v. 4, p. 463–502, 1969.
- PEARL, J. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Los Angeles, California: Morgan Kaufmann, 1988.
- PEARL, J. *Causality: Models, Reasoning, and Inference*. Cambridge: Cambridge University Press, 2000.
- POOLE, D. Probabilistic Horn abduction and Bayesian networks. *Artificial Intelligence*, v. 64, p. 81–129, 1993.
- PUTERMAN, M. L. *Markov Decision Process*. New York: J. Wiley & Sons, 1994.
- PUTERMAN, M. L.; BOUTILIER, C. Process-oriented planning and average-reward optimality. In: *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*. Montreal, Canada: IJCAI, 1995. p. 1096–1103.
- PUTERMAN, M. L.; SHIN, M. Modified policy iteration algorithms for discounted Markov decision problems. *Management Science*, v. 24, p. 1127–1137, 1978.

RUSSEL, S.; NORVIG, P. *Artificial Intelligence: a Modern Approach*. New Jersey: Prentice-Hall, 2002.

SCHWARTZ, A. A reinforcement learning method for maximizing undiscounted rewards. *Proceedings of the Tenth International Conference on Machine Learning*, Amherst, MA, p. 298–305, 1993.

WATKINS, C. J. C. H.; DAYAN, P. Q-learning. *Machine learning*, v. 8, p. 279–292, 1992.

WILLIAMSON, M.; HANKS, S. Optimal planning with a goal-directed utility model. *Proceedings of the Second International Conference on AI Planning Systems*, Chicago, IL, p. 176–180, 1994.

ZADEH, L. A. Fuzzy sets. *Information Control*, v. 8, p. 338–353, 1965.

ZADEH, L. A. Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC3, n. 1, 1973.