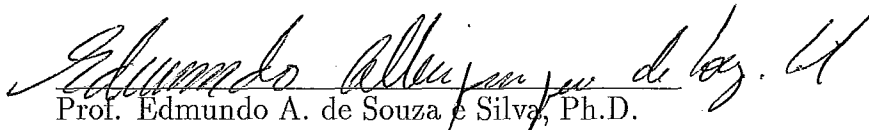


Modelos de *Desempenhabilidade* e Aplicações a Sistemas de Computação

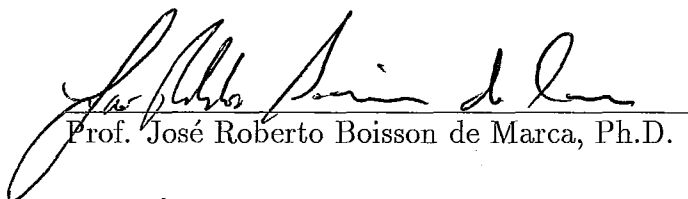
Reinaldo Antonio Vallejos Campos

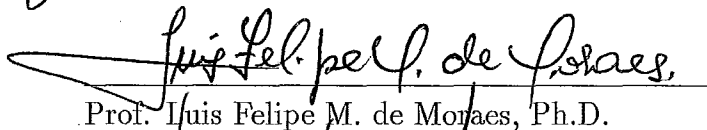
TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS
PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVER-
SIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUI-
SITOS NECESSÁRIOS PARA OBTENÇÃO DO GRAU DE DOUTOR EM
CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

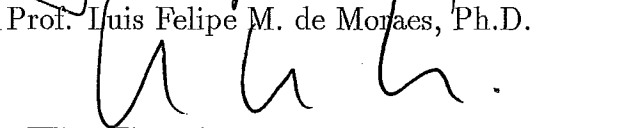
Aprovada por:


Prof. Edmundo A. de Souza e Silva, Ph.D.
(presidente)


Prof. Edil Severiano Tavares Fernandes, Ph.D.


Prof. José Roberto Boisson de Marca, Ph.D.


Prof. Luis Felipe M. de Moraes, Ph.D.


Prof. Paulo H. Aguiar Rodrigues, Ph.D.


Prof. Valmir Carneiro Barbosa, Ph.D.

RIO DE JANEIRO, RJ – BRASIL
JANEIRO DE 1993

VALLEJOS CAMPOS, REINALDO ANTONIO

Modelos de Desempenhabilidade e Aplicações a Sistemas de Computação [Rio de Janeiro] 1993

xii, 146 p., 29.7 cm, (COPPE/UFRJ, D. Sc., Engenharia de Sistemas e Computação, 1993)

TESE – Universidade Federal do Rio de Janeiro, COPPE

1 – Desempenho

2 – Disponibilidade

3 – Modelos Markovianos

4 – Cadeias de Markov com recompensas

I. COPPE/UFRJ II. Título(série).

A minha esposa Marta
e a minhas filhas
Paula, Marisa, Ana Carla e Marcela

Agradecimentos

Ao professor Edmundo de Souza e Silva, por haver-me orientado na pesquisa e por haver-me impulsionado continuamente a realizar um trabalho de alta qualidade.

A minha esposa, pela ajuda ilimitada que me deu na realização desta tese.

As minhas filhas, por termos compartilhado o ideal de sermos eternos aprendizes, o que fez da nossa estadia no Brasil um período maravilhoso em nossas vidas.

Aos meus pais, pelo amor que sempre me deram.

A minha amiga Iara, por ser como ela é.

Aos senhores Alfredo Blanes e Walter Amorim, porque conviver com eles, foi para mim uma ensinância de vida.

A Vera, por ter sacrificado uma semana de suas férias, para me ajudar na correção deste escrito.

A CAPES e a IBM-Brasil, pelo apoio econômico.

Resumo da Tese apresentada à COPPE como parte dos requisitos necessários para à obtenção do grau de Doutor em Ciências (D. Sc)

Modelos de Desempenhabilidade e Aplicações a Sistemas de Computação

Reinaldo Antonio Vallejos Campos

Janeiro de 1993

Orientador: Edmundo A. de Souza e Silva,
Programa: Engenharia de Sistemas e Computação

Neste trabalho estudam-se algumas das medidas mais importantes de ‘desempenhabilidade’ (*Performability*), visando encontrar métodos mais eficientes do que os atualmente usados para o cálculo de tais medidas. Acrescenta-se a este trabalho o estudo de uma aplicação em sistemas de computação.

Em [21] criou-se o termo ‘desempenhabilidade’, considerando o efeito combinado de desempenho com disponibilidade. ($\text{DESEMPENHABILIDADE} = \text{DESEMPENHO} + \text{disponiBILIDADE}$). Desempenho corresponde à medida do rendimento de um sistema em que se presuppõe ausência de falhas, durante um período finito de operação. Por outro lado, o conceito de disponibilidade refere-se a sistemas nos quais pode haver falhas e reparos e diz respeito à fração do tempo em que o sistema está operacional, ou seja, efetuando trabalho útil.

‘Desempenhabilidade’, em termos gerais, corresponde ao desempenho do sistema durante um período finito de operação, sendo que, neste intervalo, o sistema pode falhar, total ou parcialmente, e ser reparado.

Resumem-se, a seguir, os resultados mais relevantes desta tese.

Em primeiro lugar, encontrou-se um novo método para o cálculo da função de distribuição da combinação linear da estatística de ordem de um conjunto de ‘n’ variáveis aleatórias independentes, uniformes no intervalo $[0, t]$. Baseando-se neste resultado e utilizando-se somente uma dedução probabilística, obteve-se o resultado de [9] para a desempenhabilidade, no caso em que a taxa de rendimento do sistema (trabalho realizado por unidade de tempo) depende do estado em que este se encontra. A dedução usada neste trabalho é mais simples e consegue levar ao entendimento dos termos do resultado de [9].

Achou-se, também, uma nova expressão para a desempenhabilidade, onde o rendimento do sistema é obtido somente quando este muda de estado. A expressão matemática aqui encontrada é a mais eficiente desenvolvida até o momento para a avaliação desta medida.

A avaliação numérica da desempenhabilidade com recompensa associada às transições de estado demanda menos recursos computacionais do que o cálculo numérico da desempenhabilidade onde o sistema ganha recompensa a cada unidade de tempo de permanência nos estados. Aproveitando esta diferença de custo computacional, foi proposto um método para avaliar, em forma aproximada, a desempenhabilidade do tipo mais complexo, utilizando-se de um modelo com recompensas associadas as transições. O erro de aproximação é controlável e especificável a priori, existindo um compromisso entre a precisão do resultado obtido e o custo dos recursos computacionais empregados.

Analisou-se, também, aqui, um sistema de Banco de Dados no qual são feitas, periodicamente, gravações de seu estado em um dispositivo de memória secundária. O modelo empregado para esta análise inclui mais detalhes do que os modelos resolvidos analiticamente, que se encontram na literatura especializada. Baseando-se neste modelo foram obtidas expressões analíticas para o valor médio e a função de distribuição da disponibilidade do sistema de Banco de Dados.

Outro importante resultado deste estudo foi demonstrar a existência de uma relação entre o modelo de disponibilidade do Banco de Dados e a área de desempenhabilidade. Esta relação ainda não havia sido encontrada nos artigos publicados sobre o tema.

Abstract of Thesis presented to COPPE as partial fulfillment of the requirements for the degree of Doctor of Science (D. Sc.)

Performability Models and Applications to Computer Systems

Reinaldo Antonio Vallejos Campos

January, 1993

Thesis Supervisor: Edmundo A. de Souza e Silva,
Department: Engenharia de Sistemas e Computação

Some of the main measurements of ‘performability’ are studied within this work which is aimed at findings methods more efficient than those used nowadays for the calculation of such measurements. In addition, this work includes the study of an application in computerized systems.

The term ‘performability’ was created in [21], taking into account the combined effect of performance e availability. (PERFORMABILITY = PERFORMANCE + availaBILITY). Performance corresponds to the measurement of the performance of a system in which we assume the absence of failures during a finite operation time. On the other hand, the concept of availability refers to systems in which we may find failures and repairs and is concerned with the fraction of time during which the system is operative, i.e. carrying out useful work.

In geral terms, ‘performability’ refers to the performance of the system during a finite operation period whereas, within this interval, the system may totally or partially fail an be repaired.

The main results of this thesis are summarized as follows:

In the first place, a new method was found for the calculation of the distribution function of the linear combination of the order statistics of 'n' independent random variables which are uniform during the $[0, t]$ interval. Based on this result, and using only a probabilistic deduction, the result of [9] was obtained for the performability, provided that the performance rate of the system (work carried out per time unit) depends on its current state. The deduction used in this thesis is far more simple and allows us to understand the terms of the result of [9].

A new expression for performability was also found, where the performance of the system is obtained only when it changes its state. The mathematical expression hereby found is the most efficient developed until the present for the evaluation of this measurement.

The numerical evaluation of performability with reward associated to the state transitions requires less computing resources than the numerical evaluation of performability where the system gains reward with each unit of time during which it remains in the states. Taking advantage of this difference, a method of evaluating the more complex performability using a model with rewards associated to transitions was proposed. The approximation error may be controlled and specified a priori, existing a tradeoff between the precision of the result obtained and the cost of the computing resources employed.

Within this work, a Data Base System was also analyzed, whose state is periodically saved in a secondary memory device. The model used for this analysis includes more details than those analytically solved which we may find in the technical literature. Based on this model, analytical expressions were found for the mean value and the distribution function of the Data Base System availability.

Another main result of this study was to demonstrate the existence of a relation between the Data Base availability model and the performability area. This relation had not yet been found in papers published on this subject.

Índice

I	Introdução	1
I.1	O Tipo de Modelo	4
I.2	Medidas de Desempenhabilidade	4
I.3	Técnica de solução.	6
I.4	Visão Geral da Tese	8
II	Combinação Linear de Estatísticas de Ordem (CLEO)	14
II.1	Introdução	14
II.2	Trabalhos Previos	22
II.3	Avaliação da Função de Distribuição de CLEO (FDP-CLEO) . . .	24
II.3.1	Solução da FDP-CLEO	27
II.3.2	Demonstração da solução	28
II.3.3	Inversão da solução desde o plano de Laplace	32
II.3.4	Complexidade Computacional	34
II.4	Exemplos	34
II.4.1	Exemplo 1	34
II.4.2	Exemplo 2	35
II.5	Avaliação da Desempenhabilidade $F_i(t, r)$ a partir da CLEO	37
II.5.1	Introdução	37
II.5.2	Avaliação da Desempenhabilidade	38
II.5.3	Discussão do método de solução de $F_i(t, r)$	46

III Avaliação da Desempenhabilidade para Recompensas Discretas	49
III.1 Introdução	49
III.2 O Modelo	52
III.3 Avaliação da Desempenhabilidade $G_i(t, r)$	53
III.4 Complexidade Computacional	55
III.5 Exemplo	59
III.6 Conclusões	61
IV Aproximação de $F_i(t, r)$ mediante $G_i(t, r)$	62
IV.1 Introdução	62
IV.2 Aproximação Proposta	63
IV.3 Resultado da aproximação para a recompensa adquirida durante a permanência em um único estado.	66
IV.4 Análise de erro da recompensa acumulada em um único estado	69
IV.5 Análise do erro para uma cadeia de Markov geral	72
IV.6 Exemplo.	74
IV.7 Conclusões	75
V <i>Checkpointing and Roll-Back and Recovery</i>	77
V.1 Introdução	77
V.2 Trabalhos Previos	81
V.3 O Modelo	83
V.3.1 Descrição do Modelo e Análise Geral	84
V.4 Avaliação da Disponibilidade Média	87
V.4.1 Avaliação do Tempo Médio de <i>Checkpointing</i>	88
V.4.2 Avaliação de $E[F_i^*(D)]$	88
V.5 Disponibilidade média para uma variante do Modelo.	92
V.6 Avaliação da Função de Distribuição da Disponibilidade	93
V.6.1 Método 1	98

V.6.2	Método 2	105
V.6.3	Método 3	119
V.6.4	Complexidade	127
VI	Conclusões e Trabalhos Futuros	128
VI.1	Conclusões	128
VI.2	Trabalhos Futuros	129
A	Glossário de Termos	134

Lista de Figuras

I.1	Exemplo de cadeia de Markov de parâmetro contínuo	7
I.2	Processo estocástico equivalente ao da figura I.1, obtido por randomização.	7
II.1	Estatística de Ordem de n v.a.'s. com resultados em $[0, t]$	16
II.2	Combinação Linear de Estatísticas de Ordem de n v.a.'s uniformes.	17
II.3	CLEO de v.a.'s uniformes interpretada como a Combinação Linear de v.a.'s Dirichlet.	20
II.4	CLEO de n v.a.'s uniformes interpretada como a recompensa ganha por uma CM com n transições (sem perda de generalidade se assumiu que $r_{ R } > \dots > r_2 > r_1 > 0$)	21
III.1	Protocolo Parada e Espera	49
III.2	Cadeia de Markov que modela o Protocolo <i>Parada e Espera</i> , sem erros	51
III.3	Conjunto de recompensas diferentes atingíveis pelo protocolo.	58
III.4	CM randomizada para o modelo simplificado do protocolo <i>Parada e Espera</i>	59
III.5	Caminhos possíveis para a CM do protocolo <i>Parada e Espera</i>	60
III.6	Recompensas possíveis que pode atingir a CM da figura III.5	60
IV.1	Variáveis associadas ao intervalo $[0, t]$	64
IV.2	Cadeia de Markov de tempo contínuo que modela a recompensa ganha pelo sistema num único estado (modelo 1).	66
IV.3	CM de parâmetro discreto correspondente à CM do modelo 1	67
IV.4	Erro obtido para o modelo 1	72

IV.5 Erro relativo obtido para o modelo 1	73
IV.6 Recompensa acumulada de um sistema tolerante a falhas	76
V.1 Ciclo Típico de CRR	78
V.2 Variáveis associadas a um Ciclo Típico de CRR	85
V.3 Intervalo de tempo disponível (D)	94
V.4 Relação entre $\vec{K}_{n,m}$ e $\vec{K}_{n-1,m}$ (parte a): $\vec{K}_{3,4} = \langle 1, 2, 1, 4 \rangle$	100
V.5 Relação entre $\vec{K}_{n,m}$ e $\vec{K}_{n-1,m}$ (parte b) $\vec{K}_{2,4} = \langle 2, 1, 4 \rangle$	101
V.6 Relação entre $\vec{K}_{n,m}$ e $\vec{K}_{n-1,m}$ (parte c): $\vec{K}_{3,4} = \langle 0, 1, 1, 6 \rangle$	101
V.7 Relação entre $\vec{K}_{n,m}$ e $\vec{K}_{n-1,m}$ (parte d): $\vec{K}_{2,4} = \langle 1, 1, 5 \rangle$	101
V.8 Relação entre $\vec{K}_{n,m}$ e $\vec{K}_{n,m-1}$ (parte e): $\vec{K}_{3,4} = \langle 2, 1, 1, 4 \rangle$	102
V.9 Relação entre $\vec{K}_{n,m}$ e $\vec{K}_{n,m-1}$ (parte f): $\vec{K}_{3,3} = \langle 1, 1, 1, 4 \rangle$	102
V.10 Relação entre $\vec{K}_{n,m}$ e $\vec{K}_{n,m-1}$ (parte g): $\vec{K}_{3,4} = \langle 1, 1, 1, 5 \rangle$	103
V.11 Relação entre $\vec{K}_{n,m}$ e $\vec{K}_{n,m-1}$ (parte h): $\vec{K}_{3,3} = \langle 1, 1, 1, 4 \rangle$	104

Capítulo I

Introdução

Desempenhabilidade (*Performability*) é uma palavra que atualmente não existe na língua portuguesa. Ela foi criada em [21] empregando um prefixo da palavra ‘Desempenho’ (*Performance*) e um sufixo da palavra ‘Disponibilidade’ (*Availability*), da seguinte forma: DESEMPENHABILIDADE = DESEMPENHo + disponiBILIDADE (*PERFORMABILITY = PERFORMANCEe + availaBILITY*), onde as letras maiúsculas destacam as letras componentes da palavra DESEMPE-NHABILIDADE (*Performability*).

O motivo de criar o termo Desempenhabilidade foi representar um novo conceito, que corresponde a uma integração dos conceitos Desempenho e Disponibilidade. Na área de sistemas computacionais, o conceito **Desempenho** refere-se ao rendimento acumulado pelo sistema em um certo intervalo de tempo durante o qual está operando; sendo que o rendimento diz respeito ao aspecto que o usuário está interessado em medir, o qual dependendo da aplicação, pode corresponder a uma grande variedade de medidas, como, por exemplo, o número de transações processadas no intervalo de tempo em que o sistema está sendo usado. Outro exemplo seria associar rendimento à vazão média, com que o sistema processa transações em estado estacionário.

Por outro lado, o conceito de **Disponibilidade** refere-se à percentagem do tempo em que o sistema está operativo, durante o período de observação. Portanto, para medir a disponibilidade de um sistema é necessário definir especificamente quando é que ele está operacional. Assim, esta definição depende do ponto de vista do usuário. Como exemplo, um sistema de N processadores pode ser considerado operacional quando tem, pelo menos, um processador funcionando corretamente ou quando tem, ao menos, K processadores operativos, $2 \leq K \leq N$. O exemplo confirma que considerar o sistema operacional ou não depende da definição do usuário.

Para estudar tanto o Desempenho como a Disponibilidade de sistemas computacionais, faz-se necessário criar modelos que representem em forma abstrata, os sistemas sob estudo. Para facilitar a solução matemática do modelo, este deve ser o mais simples possível, porém a necessidade de representar em forma

realista o sistema leva a incluir no modelo elementos que o tornam complexo de resolver matematicamente. Uma técnica usada na criação de modelos para responder, em forma adequada, ao compromisso existente entre o realismo do modelo e a simplicidade de sua resolução é criar diferentes modelos para estudar diferentes aspectos do mesmo sistema. Por exemplo, o tipo de modelo usado para estudar a Disponibilidade de um sistema é diferente do tipo empregado para avaliar seu rendimento.

Em caso de se desejar estudar o rendimento de um sistema, para simplificar seu modelo, normalmente supõe-se que a estrutura do sistema não muda durante o período de observação deste. Como exemplo, se o objetivo é avaliar o número de transações processadas por um sistema multiprocessador durante um certo intervalo de tempo, uma hipótese normalmente usada para facilitar a solução do modelo é supor que, neste intervalo, nenhum componente do sistema multiprocessador pode falhar. Além disso, existem sistemas que não podem falhar durante sua operação; por isso, são conhecidos como sistemas não tolerantes a falhas (SNTF). Portanto, o tipo de modelo que avalia o desempenho, supondo que não existem falhas nas componentes do sistema, é apropriado para avaliar o rendimento dos SNTF.

Quando o objetivo é estudar somente a Disponibilidade de um sistema, o analista não está interessado em avaliar o rendimento deste. Para isto, o modelo deve representar as diferentes configurações que pode ter o sistema e como estas podem mudar devido a possíveis falhas ou reparos de seus componentes. O tipo de sistema que pode falhar e (possivelmente) ser reparado durante sua operação chama-se sistema tolerante a falhas (STF). Um exemplo de análise de um STF seria avaliar a disponibilidade de um sistema multiprocessador quando este é observado durante um período finito de tempo. Neste caso, entre outros aspectos, o modelo deveria identificar o número de processadores operacionais e falhos, em cada momento em que o sistema é observado, como cada um destes falha ou é reparado, como estas falhas afetam os demais componentes do sistema, etc.

Outro tipo de estudo que pode ser de interesse é avaliar o rendimento de um STF. Neste caso, é necessário considerar as possíveis mudanças de configurações do sistema e a variação de rendimento que ele experimenta em cada uma delas. Por este tipo de estudo incluir, em sua análise, tanto o aspecto de rendimento como o de disponibilidade, ele é conhecido com o nome genérico de **estudo de Desempenhabilidade**.

O cálculo de medidas de rendimento e disponibilidade tem sido uma área de muitas pesquisas nas duas últimas décadas. Quanto a área de Desempenhabilidade, vem recebendo cada vez mais atenção por parte dos pesquisadores nos últimos anos [5, 9, 21], devido à sua utilidade para analisar sistemas computacionais reais [4, 22, 23]. Avaliar medidas de desempenho é, em geral, dispendioso computacionalmente. Isto quer dizer que, para obter uma medida, é preciso ocupar muitos recursos computacionais (memória e tempo de uso do computador). Já que é desejável obter medidas de desempenho e consi-

derando-se que o custo computacional é caro, é fundamental desenvolver novos métodos para avaliar mais eficientemente este tipo de medidas. Este é um dos objetivos principais desta tese.

O estudo de medidas de desempenhabilidade pode ser dividido em duas categorias: **medidas transientes** e **medidas estacionárias**. A primeira visa avaliar o rendimento do sistema computacional, quando este é observado durante um intervalo finito de tempo. A segunda categoria tem como objetivo analisar o rendimento do sistema quando este opera durante tempo ilimitado, no sentido de que este tempo é 'suficientemente grande' para que as variáveis de interesse cheguem a um valor estável. A análise matemática das medidas transientes é, normalmente, bastante mais complexa de realizar do que a análise estacionária. Este tipo de estudo, porém, é importante pois, na prática, os sistemas computacionais não chegam, muitas vezes, a operar em estado estacionário. Em conseqüência, em geral, **as medidas de desempenhabilidade estudadas nesta tese são medidas transientes**.

A forma mais usada para avaliar as medidas de desempenhabilidade é usar cadeias de Markov com recompensas [28]. Em geral, pode-se atribuir diferentes tipos de recompensa (rendimento) a uma determinada cadeia de Markov. Um destes tipos refere-se ao caso em que cada unidade de tempo em que o sistema permanece em um determinado estado ele ganha um rendimento associado a este estado. Em outras palavras, em cada estado diferente em que o sistema pode permanecer, ele ganha rendimento com uma velocidade que depende do estado em que este se encontra. Note-se que, para este tipo de atribuição de recompensa, o rendimento total acumulado pelo sistema, durante um período finito de tempo, é uma variável aleatória contínua. Outra forma de associar recompensa a uma cadeia de Markov consiste em que o sistema ganha rendimento somente quando ele muda de estado, ou seja, para cada transição entre diferentes estados, o sistema ganha uma recompensa (fixa) associada a esta transição. Observe-se que, quando o ganho de recompensa do sistema acontece somente nas transições de estado, o rendimento total acumulado pelo sistema durante um período de tempo finito de observação é uma variável aleatória discreta.

Como já visto, a avaliação numérica de algumas medidas de desempenhabilidade é muito dispendiosa quanto a recursos computacionais; então, tal como foi indicado, um dos objetivos primordiais desta tese é encontrar novas formas de avaliar de modo exato estas medidas. Outra alternativa consiste em desenvolver métodos aproximados que ofereçam alguma vantagem quando comparados com os métodos exatos. Assim sendo, **outro objetivo deste trabalho é desenvolver métodos para o cálculo aproximado de medidas transientes de desempenhabilidade, que ofereçam alguma vantagem quando comparados aos métodos exatos**.

Avaliar medidas de desempenhabilidade somente tem sentido quando elas permitem estudar sistemas computacionais reais; por isso, na literatura especializada têm aparecido vários trabalhos exemplificando como usar estas medidas para estudar sistemas na prática [4, 22, 23, 30, 31]. Portanto, **outra interessante**

contribuição pretendida é encontrar novas possibilidades de aplicar no estudo de sistemas reais, os conceitos e medidas de desempenhabilidade.

A seguir se descreverá o tipo de modelo que foi usado nas distintas partes deste trabalho, tanto para avaliar medidas de desempenhabilidade, quanto para estudar uma aplicação de desempenhabilidade na área de Banco de Dados.

I.1 O Tipo de Modelo

Neste trabalho consideram-se sistemas que podem ser modelados por Processos de Markov Homogêneos com espaço de estados finito [28], onde cada estado pode representar, em geral, uma contenção de recursos ou uma estrutura específica do sistema, como, por exemplo, número de componentes falhos. Isto quer dizer, que as falhas ou reparo dos componentes do sistema durante sua operação, são modelados por mudanças de estado da cadeia de Markov. Em termos específicos: seja $\chi_1 = \{X_1(t), t \geq 0\}$ o processo homogêneo de Markov de tempo contínuo que descreve o comportamento do sistema, e seja $E = \{e_k, k = 1, \dots, |E|\}$ o espaço finito de estados associado ao modelo, onde $|E|$ corresponde ao número total de estados do sistema. Seja $T \subset E \times E$, o conjunto de transições entre estados da cadeia de Markov, onde $t_{ij} \in T$ corresponde à transição desde o estado $e_i (\in E)$ ao estado $e_j (\in E)$. Um vetor é escrito da forma $\vec{v} = \langle v_i \rangle$, e a soma de seus componentes é escrita como $|\vec{v}|$. O componente índice c do vetor \vec{v} é representado como $\vec{v}[c]$. Uma matriz é escrita da forma $M = [m_{ij}]$. Em particular, a matriz Q corresponde às taxas de transição do processo de Markov χ_1 ; por exemplo, q_{ij} da matriz Q pode corresponder a uma taxa de falha ou de reparo de algum componente do sistema sob análise.

Outro aspecto do modelo consiste em que a cadeia de Markov acumula recompensa durante sua operação. A recompensa ganha pela cadeia em um certo intervalo de tempo corresponde a uma abstração do rendimento ganho pelo sistema nesse intervalo. Isto é, diferentes tipos de rendimento do sistema (número de falhas, tarefas processadas no intervalo, etc), do ponto de vista matemático, podem ser conceituados como um único problema: que a cadeia de Markov que modela o sistema ganhe uma determinada quantidade de recompensa nesse intervalo.

I.2 Medidas de Desempenhabilidade

Em geral pode-se atribuir diferentes tipos de recompensas à cadeia de Markov. Um tipo de atribuição de recompensa consiste em associar a cada estado uma taxa de recompensa, isto quer dizer que, por cada unidade de tempo em que o sistema permanece em um determinado estado, ele ganha a recompensa associada a esse estado. Devido a que o modelo usado corresponde a uma cadeia de Markov

de parâmetro contínuo, o tempo que o sistema permanece em um dado estado, é, também, uma variável aleatória contínua. Isto implica em que a recompensa total acumulada pelo sistema em um intervalo de tempo dado, é uma variável aleatória contínua. No restante do texto o tipo de atribuição de recompensa descrito acima, será referenciado como **tipo I**.

Outra forma de atribuição de recompensa à cadeia de Markov, que será referenciado de **tipo II**, consiste em atribuir uma recompensa (fixa) ao sistema cada vez que este efetua uma certa transição t_{ij} . Esta forma de atribuição de recompensa é de tipo **discreta**, e corresponde à recompensa (fixa) ganha pelo sistema cada vez que efetua essa transição.

Os tipos de atribuição de recompensa a uma cadeia de Markov descritas acima serão vistas agora, de uma maneira formal.

O tipo I de atribuição de recompensa consiste em associar a cada estado $e_k \in E$ uma taxa de recompensa $\rho(e_k)$, que corresponde à recompensa ganha pelo sistema por cada unidade de tempo que permanece neste estado. Seja $R = \{r_1, \dots, r_{|R|}\}$ o conjunto de taxas de recompensas diferentes. Em geral tem-se que o número de taxas diferentes é menor o igual ao número de estados da cadeia de Markov ($|R| \leq |E|$), pois diferentes estados podem ser associados a uma mesma taxa de recompensa. Seja $RTA_i(t)$ a recompensa total ganha pelo sistema no intervalo $[0, t]$ quando este inicia sua operação no estado $e_i \in E$, e a atribuição de recompensas é do tipo I. $RTA_i(t)$ é uma variável aleatória (v.a.) devido às possíveis mudanças probabilísticas de rendimento instantâneo do sistema durante sua operação. Como condição inicial normalmente supõe-se que $RTA_i(0) = 0$. Com o objetivo de ter um conhecimento detalhado do rendimento acumulado pelo sistema, é importante calcular a função de distribuição de probabilidade de $RTA_i(t)$, que será representado como $F_i(t, r)$. Mais especificamente, $F_i(t, r)$ corresponde à probabilidade que $RTA_i(t)$ seja menor ou igual a um determinado valor r :

$$F_i(t, r) = P[RTA_i(t) \leq r] \quad (I.1)$$

Um importante caso particular de atribuição de recompensa de tipo I, consiste em associar uma taxa de recompensa igual a 1 a um determinado subconjunto de estados de E e taxa 0 aos demais estados. Um breve raciocínio mostra que neste caso $RTA_i(t)$ corresponde ao tempo acumulado que a cadeia de Markov permanece no referido subconjunto durante o intervalo $[0, t]$, no caso em que ela inicia sua operação no estado $e_i \in E$. Com o objetivo de identificar esta forma particular de atribuição de recompensas, costuma-se escrever $RTA_i(t)$ como $O_i(t)$, onde a letra O é usada para indicar que, usualmente, o subconjunto de estados com taxa de recompensa 1, são estados nos quais o sistema está **Operacional**. Em termos formais, a atribuição de recompensas recém descrita pode ser estabelecida da seguinte forma: seja O um subconjunto de estados de E ($O \subset E$) e seja $I = E - O$. Considere a seguinte atribuição de recompensas:

$$\rho(e_k) = \begin{cases} 1 & \text{se } e_k \in O \\ 0 & \text{caso contrario } (e_k \in I) \end{cases} \quad (\text{I.2})$$

Então:

$$O_i(t) = RTA_i(t)$$

se a atribuição de recompensas é a da equação I.2.

Define-se, também, para este caso, $A_i(t)$ como sendo a fração de tempo que o sistema permanece em estados $\in O$ no intervalo $[0, t]$, quando ele inicia sua operação no estado e_i , isto é:

$$A_i(t) = \frac{O_i(t)}{t} \quad (\text{I.3})$$

Por outro lado, na atribuição de recompensa de tipo II cada transição $t_{ij} \in T$ tem associada uma recompensa fixa $\rho(t_{ij})$, que corresponde à recompensa ganha pelo sistema cada vez que transita do estado e_i ao estado e_j . Seja $\Theta = \{\theta_k, k = 1, \dots, |\Theta|\}$ o conjunto de recompensas fixas distintas. Em geral tem-se que o número de recompensas fixas distintas é menor ou igual ao quadrado do número de estados da cadeia de Markov ($|\Theta| \leq |E|^2$), pois diferentes transições podem ser associadas a uma mesma recompensa e algumas transições entre estados podem ser impossíveis de acontecer. Seja $RTAD_i(t)$ a recompensa total ganha pelo sistema no intervalo $[0, t]$, no caso em que o estado inicial é $e_i \in E$, e o ganho de recompensa do sistema acontece somente nas transições. Evidentemente $RTAD_i(t)$ é uma variável aleatória e normalmente supõe-se que $RTAD_i(0) = 0$. Para conhecer o ganho de recompensa do sistema devido às transições, é interessante calcular $G_i(t, r)$ que corresponde à função de distribuição de $RTAD_i(t)$, isto é:

$$G_i(t, r) = P[RTDA_i(t) \leq r] \quad (\text{I.4})$$

I.3 Técnica de solução.

As soluções dos diferentes modelos apresentados no desenvolvimento desta tese são encontradas usando a técnica matemática conhecida com o nome de **randomização** [3], descrita a seguir.

A randomização, em termos gerais, estabelece que um processo estocástico χ_1 de parâmetro contínuo é equivalente a outro processo estocástico χ_2 de parâmetro discreto, cujas transições acontecem quando ocorrem eventos em um processo de Poisson $N(t)$, com parâmetro Λt , independente de χ_1 (os parâmetros de χ_2 e de $N(t)$ dependem de χ_1 , da forma que será vista mais tarde). Por exemplo, a cadeia de Markov da figura I.1 tem o mesmo comportamento probabilístico que a cadeia de Markov da figura I.2, onde as transições desta última cadeia acontecem somente quando ocorrem eventos no processo de Poisson $N(t)$, da figura I.2. O fato

de que as cadeias de Markov das figuras I.1 e I.2 tenham o mesmo comportamento probabilístico quer dizer que a probabilidade de estar em estados correspondentes é a mesma para ambas as cadeias para qualquer instante de tempo. Por exemplo, para as cadeias de Markov das figuras I.1 e I.2, cumpre-se que $\Pi_1(t) = 1 - e^{-\lambda t}$ e $\Pi_2(t) = e^{-\lambda t}$ no caso em que o parâmetro Λ do Processo de Poisson é maior ou igual a λ .

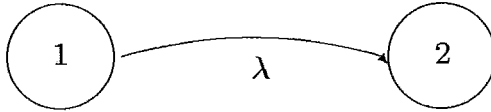


Figura I.1: Exemplo de cadeia de Markov de parâmetro contínuo

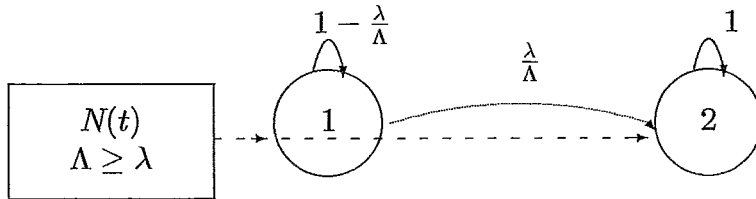


Figura I.2: Processo estocástico equivalente ao da figura I.1, obtido por randomização.

Descreve-se, agora, em termos formais, a técnica de randomização. Seja χ_1 o processo estocástico de tempo contínuo que modela o sistema. Para avaliar o comportamento probabilístico de χ_1 usa-se outro processo estocástico χ_2 de tempo discreto, obtido de χ_1 mediante randomização (uniformização) [3]. É um fato bastante conhecido [3] que ambos processos estocásticos são equivalentes. A randomização do processo χ_1 é feita da seguinte forma: seja \mathbf{Q} a matriz das taxas de transição do processo χ_1 , e Λ um valor maior ou igual à máxima taxa de saída dos estados de \mathbf{Q} , isto é, $\Lambda \geq \max_i \{ |q_{ii}| \}$. Seja $\mathbf{P} = [p_{ij}]$ a matriz estocástica obtida da forma $\mathbf{P} = \mathbf{Q}/\Lambda + \mathbf{I}$, onde \mathbf{I} corresponde à matriz identidade. Seja $\chi_2 = \{\chi_2(n), n = 1, 2, \dots\}$ a cadeia de Markov com espaço de estado E e matriz de probabilidade de transição \mathbf{P} . Seja $\mathbf{N} = \{N(t)\}$ um processo de Poisson independente de χ_2 com parâmetro Λt . Então $\{\chi_2(N(t))\}$ é um processo Markoviano cujo comportamento probabilístico é equivalente ao processo χ_1 , ou seja :

$$Prob[\chi_1(t) = c_k] = Prob[\chi_2(N(t)) = c_k]; \quad c_k \in E \quad (I.5)$$

No restante desta introdução apresentamos uma visão geral do trabalho feito na tese.

I.4 Visão Geral da Tese

No capítulo II é encontrada uma nova expressão para a função de distribuição de probabilidade (FDP) das estatísticas de ordem de n variáveis aleatórias (v.a.'s), independentes identicamente distribuídas, uniformes no intervalo $[0, t]$. O motivo fundamental pelo qual foi pesquisado este problema, é que De Souza e Gail em [5] demonstraram que esta combinação linear corresponde ao rendimento total acumulado por uma cadeia de Markov, em um intervalo $[0, t]$, no caso em que o ganho de recompensa é do tipo I e a cadeia de Markov randomizada transita por um caminho conhecido. Outra razão de se avaliar a citada FDP é que ela corresponde, também, à solução de outros problemas, que, aparentemente, não tem relação alguma com a área de desempenhabilidade. Por exemplo, Matsunawa [20] necessitou resolver esta questão, quando estava estudando o problema de *Best-Fit* de variáveis aleatórias. Para colocar o problema em termos específicos, considere o conjunto de variáveis aleatórias $\{U_1, U_2, \dots, U_n\}$ independentes, uniformemente distribuídas no intervalo $[0, t]$. Seja $\{U_{(1)}^n(t), U_{(2)}^n(t), \dots, U_{(n)}^n(t)\}$ o conjunto de v.a.'s obtidas permutando os elementos do conjunto $\{U_i; 1 \leq i \leq n\}$, de forma de se obter uma ordenação crescente, Isto é: $U_{(i)}^n(t)$ é o i -ésimo menor valor entre U_1, \dots, U_n . Para simplificar a notação, no restante deste trabalho as v.a.'s $U_{(k)}^n(t)$, ($1 \leq k \leq n$) serão escritas como $U_{(k)}$, onde os parênteses no subíndice indicam que trata-se de uma estatística de ordem. Somente nos casos em que for possível existir alguma ambigüidade será usada a forma $U_{(k)}^n(t)$. Adicionalmente para estender esta notação define-se $U_{(0)} = 0$ e $U_{(n+1)} = t$.

Seja \vec{a}_n o vetor $\langle a_1, a_2, \dots, a_{n+1} \rangle$, onde $\{a_k : k = 1, 2, \dots, n+1\}$ é um conjunto de constantes pertencentes aos reais. Então define-se:

$$CLEO(\vec{a}_n, t) = \sum_{k=1}^{n+1} a_k U_{(k)} \quad (I.6)$$

A equação acima estabelece que $CLEO(\vec{a}_n, t)$ é uma v.a que corresponde a uma combinação linear do conjunto das estatísticas de ordem $U_{(k)}$, $1 \leq k \leq n+1$

Por outro lado considere o conjunto de variáveis aleatórias $V_{(k)}^n(t)$, e o conjunto de constantes c_k , ($1 \leq k \leq n+1$), definidas como:

$$V_{(k)}^n(t) = U_{(k)}^n(t) - U_{(k-1)}^n(t), \quad 1 \leq k \leq n+1 \quad (I.7)$$

$$c_k = \sum_{j=k}^{n+1} a_j, \quad 1 \leq k \leq n+1 \quad (I.8)$$

Analogamente ao caso das v.a's uniformes, $V_{(k)}^n(t)$, $1 \leq k \leq n + 1$, será escrita como $V_{(k)}$.

O conjunto de v.a's $V_{(k)}$, $1 \leq k \leq n + 1$ definidas em I.7 tem distribuição Dirichlet [20] e $CLEO(\vec{a}_n, t)$ pode ser interpretada como uma combinação linear destas variáveis [8], da seguinte forma:

$$CLEO(\vec{a}_n, t) = \sum_{k=1}^{n+1} c_k V_{(k)} \quad (I.9)$$

Outra interpretação da v.a. $CLEO(\vec{a}_n, t)$ foi dada em [5], onde foi demonstrado que ela corresponde à recompensa total acumulada no intervalo $[0, t]$ por uma cadeia de Markov (CM) que efetua n transições nesse intervalo. Isto é, dado que na CM aconteceram n transições e ela percorreu um determinado caminho \vec{e}_n (onde a i -ésima componente de \vec{e}_n corresponde ao i -ésimo estado visitado pelo sistema no intervalo), em [5] foi demonstrado que:

$$CLEO(\vec{a}_n, t) = [RTA(t)|\vec{e}_n] \quad (I.10)$$

Considere agora $F(\vec{a}_n, t, r)$ a função de distribuição da v.a $CLEO(\vec{a}_n, t)$, isto é:

$$F(\vec{a}_n, t, r) = P[CLEO(\vec{a}_n, t) \leq r] \quad (I.11)$$

Na literatura especializada existem relativamente poucos trabalhos sobre o cálculo de $F(\vec{a}_n, t, r)$. Dentre eles, Demster e Kleyle [8] estudaram o caso onde $a_k > 0$, $1 \leq k \leq n + 1$. Eles conseguiram chegar a uma expressão analítica fechada para $F(\vec{a}_n, t, r)$. Posteriormente Weisberg [32] generalizou o resultado de Demster e Kleyle, para o caso em que as constantes $\{a_k, 1 \leq k \leq n\}$ são valores reais não negativos, e $a_{n+1} = 0$. Um aspecto interessante do resultado de Weisberg é que, ao contrário de Demster e Kleyle, a combinação linear de um subconjunto das diferentes estatísticas de ordem pode ser calculada. Isto se consegue fazendo $a_k = 0$ para todo $U_{(k)}$ que não pertence ao subconjunto de interesse. O resultado de Weisberg entretanto necessita da avaliação recorrente de alguns termos. Posteriormente Matsunawa [20] obteve uma expressão analítica fechada para o cálculo da combinação linear de um subconjunto das estatísticas de ordem. Infelizmente aparecem nesta expressão coeficientes que são muito complicados de calcular. Por este motivo Ramalingam [27] propôs que antes de calcular a função de distribuição (ou a função de densidade) de probabilidade de $CLEO(\vec{a}_n, t)$, tais coeficientes devem ser avaliados usando algum pacote de *software* para o cálculo simbólico de termos matemáticos, que, posteriormente são usados na expressão de Matsunawa.

Uma forma alternativa de avaliar $F(\vec{a}_n, t, r)$, consiste em usar a interpretação desta medida dada em [5]. Deste modo $F(\vec{a}_n, t, r)$ corresponde à recompensa total acumulada por uma cadeia de Markov no intervalo $[0, t]$, quando ela transita por um caminho conhecido. Esta interpretação permite usar diretamente os resultados da área de desempenhabilidade. O menos dispendioso dentre estes

resultados para avaliar numericamente $F(\vec{a}_n, t, r)$ foi publicado em [9]. O resultado de [9] serve para avaliar a FDP da recompensa total acumulada por uma cadeia de Markov, no intervalo $[0, t]$, que é um problema bem mais complexo do que avaliar $F(\vec{a}_n, t, r)$. Usando, porém, a interpretação dada em [5] para $F(\vec{a}_n, t, r)$, citada anteriormente, é possível aplicar diretamente o resultado de [9] para a avaliação desta medida. Esta forma de avaliação de $F(\vec{a}_n, t, r)$ requer $O(n | R |)$ localizações de memória e o número de multiplicações e divisões necessárias (que são as operações mais complexas usadas) é $O(n^2 | R |)$, onde o valor de $| R |$ será formalmente definido no capítulo II. A solução para $F(\vec{a}_n, t, r)$ do capítulo II foi encontrada usando o mesmo método de [9] é até seguindo os mesmos passos do raciocínio feito ali. A vantagem da solução encontrada no capítulo II, com respeito à aplicação direta do resultado de [9] é que ela requer $2n$ localizações de memória e de $O(n^2)$ operações de multiplicação e divisão. Este ganho em eficiência computacional deve-se a que no capítulo II foi usado o método de [9] para resolver diretamente um problema mais simples que o resolvido em [9].

Outra contribuição importante do capítulo II é deduzir, de uma maneira muito mais simples, o resultado de [9]. Como já foi dito anteriormente, o resultado de [9], é importante pois ele permite avaliar $F_i(t, r)$ (ver equação I.1) em forma exata e do modo mais eficiente conhecido até agora. A dedução do resultado de [9] feita no capítulo II, quando comparada com a de aquela referência, emprega um raciocínio muito mais simples. Para que fique claro o motivo desta vantagem serão explicados, a seguir, tanto o método de solução de [9], como o do capítulo II.

O método de solução aparecido em [9], foi o de aplicar duas vezes a transformada de Laplace, chegando-se deste modo, a uma expressão recorrente bastante complexa no duplo plano de Laplace. A solução desta equação recorrente foi estabelecida sem maiores explicações e depois demonstrada por indução. A partir da citada solução e aplicando-se duas vezes a transformada inversa de Laplace, chegou-se à expressão final para $F_i(t, r)$. Este método, além de precisar uma manipulação matemática mais complexa (principalmente devido a usar o duplo plano de Laplace), não permite dar significado a alguns dos termos da expressão final para $F_i(t, r)$, isto porque eles aparecem ao expandir em frações parciais a solução estabelecida no plano de Laplace. por outro lado no capítulo II, o resultado de [9] é obtido em forma sintética (o qual já é um avanço) e usando somente raciocínios probabilísticos simples, que é a vantagem mais importante. Outro benefício é que, devido à dedução probabilística do capítulo II, chega-se a uma interpretação dos diferentes termos que aparecem na solução de [9]. Isto, além de ser conceitualmente importante, poderia oferecer outras vantagens, tal como é discutido no capítulo II.

No capítulo III é desenvolvido um algoritmo para calcular $G_i(t, r)$, que tal como já se viu, corresponde à FDP do rendimento total acumulado por uma cadeia de Markov, no caso em que o ganho de recompensa é do tipo II. Isto é, a cadeia de Markov ganha recompensa somente quando muda de estado. O cálculo numérico da expressão encontrada no capítulo III para $G_i(t, r)$ requer $O(| E | \min\{r, | \Theta | N_{max}\})$ localizações de memória e $O(| E | dN_{max} \min\{r, | \Theta |$

N_{max} }) operações de multiplicação e divisão, onde:

- $|E|$ corresponde ao número de estados da cadeia de Markov.
- d é o número máximo de transações fora de um estado da cadeia de Markov.
- $|\Theta|$ é o número de recompensas diferentes que pode ganhar a cadeia de Markov, devido a mudanças de estado.
- N_{max} corresponde ao máximo número de eventos do processo de Poisson da randomização, considerados na solução de $G_i(t, r)$. Este último valor é próprio da técnica de randomização e permite limitar, de antemão, o erro obtido ao avaliar numericamente as soluções a que se chega aplicando esta técnica (ver [5])

Por outro lado, a solução mais eficiente que se encontra na literatura para avaliar $G_i(t, r)$ encontra-se em [5] e tem um custo de memória $O\left(\left(\begin{matrix} |\Theta| + N_{max} \\ |\Theta| \end{matrix}\right) |E| d\right)$. Concluindo, a contribuição do capítulo III consiste em haver encontrado a expressão de menor custo (quanto ao uso de recursos computacionais) até agora conhecida para avaliar $G_i(t, r)$, se bem que no caso particular de existirem apenas dois valores de recompensas diferentes associadas às transições ($|\Theta| = 2$), tanto o método de [5] como o do capítulo III tem os mesmos requisitos computacionais. Entretanto, quando aumenta o número de recompensas diferentes associadas às transições, o método do capítulo III é significativamente mais eficiente do que o de [5].

No capítulo IV é proposto um novo método de avaliar, em forma aproximada, $F_i(t, r)$. A aproximação consiste em avaliar $G_i(t, r)$ em vez de $F_i(t, r)$, associando recompensas apropriadas às transições da CM randomizada. A atribuição de recompensas às transições da cadeia de Markov usada na aproximação depende do conjunto de taxas de recompensas associada aos estados da cadeia, da forma que será abordada no capítulo IV.

Um motivo de se propor esta aproximação é que o uso de recursos computacionais necessários para avaliar numericamente $G_i(t, r)$, em alguns casos pode ser significativamente menor do que o requerido para o caso de avaliar $F_i(t, r)$. Especificamente, tal como indicado em [9], avaliar $F_i(t, r)$, usando o resultado desta referência, requer de $O(|E| |R| N_{max})$ localizações de memória e de $O(|E| |R| dN_{max}^2)$ operações de multiplicação ou divisão. Por outro lado, avaliar $G_i(t, r)$ usando o resultado do capítulo III, exige $O(|E| \min\{r, |\Theta| N_{max}\})$ localizações de memória e de $O(|E| dN_{max} \min\{r, |\Theta| N_{max}\})$ operações de multiplicação ou divisão, onde para o caso da aproximação, ocorre que $|\Theta|$ é igual a $|R|$. Portanto, nos casos em que $(r \ll |\Theta| N_{max})$ é mais eficiente avaliar $F_i(t, r)$ usando a aproximação do capítulo IV. A vantagem de avaliar $F_i(t, r)$ em forma aproximada, como explicado acima, acontece somente em alguns casos, porém a aproximação fornece outra vantagem de maior abrangência, que consiste em que a aproximação usa somente operações de soma, multiplicação e

divisão, mas a expressão de [9] usa também a subtração, sendo que esta operação aparece entre outros lugares, no denominador de algumas frações. O fato de usar concomitantemente adições e subtrações e o fato de esta última operação aparecer no denominador (principalmente quando são subtraídas quantidades similares), pode levar à ocorrência de problemas de instabilidade numérica e/ou falta de precisão nos resultados obtidos.

Outras características desejáveis que a aproximação oferece são:

- O erro produzido pela aproximação é controlável e especificável de antemão. Isto se obtém escolhendo um valor apropriado da taxa de randomização, tal como será visto no capítulo IV.
- O erro pode ser tão pequeno quanto se desejar, o que na verdade é uma consequência do item anterior.
- Existe um compromisso entre o erro do resultado obtido usando a aproximação e a exigência em recursos computacionais para obter este erro. Em outras palavras, para se chegar ao menor erro no resultado é necessário empregar mais recursos computacionais; alternativamente, se se deseja poupar estes recursos, isto se consegue permitindo um erro maior no resultado obtido.

No capítulo V é estudada uma técnica que ajuda a manter a integridade dos dados em Sistemas de Bancos de Dados. Os sistemas de banco de dados orientados ao processamento de transações são cada dia mais importantes devido a sua grande utilidade em diferentes aplicações. Na maioria desses sistemas a demanda de dados imposta pelos usuários requer uma alta disponibilidade do sistema (fração de tempo em que o sistema está disponível para processar transações) e um pequeno tempo de resposta às transações.

Uma das técnicas mais comuns para conservar a integridade dos dados, aumentar a disponibilidade e/ou diminuir o tempo de resposta, é guardar periodicamente cópias do estado do banco de dados [26]. O estado do sistema inclui todos os arquivos e informações necessárias para devolvê-lo à situação em que se encontrava no momento em que foi feita a cópia. Esta informação é gravada em uma memória secundária. O processo de fazer uma cópia do sistema é conhecido com o nome de *Checkpointing* (C). As transações que modificam o sistema de arquivos a partir do último checkpointing são gravadas em um arquivo especial, conhecido com o nome de *Audit Trial*. No caso de acontecer uma falha no banco de dados é iniciado um reparo. Numa operação de reparo a falha é removida da seguinte forma : em primeiro lugar o sistema é restaurado ao estado em que estava no momento em que foi feito o último C. Esta fase de reparo é conhecida com o nome de operação de retorno (*Roll-Back*). Depois disto o reparo continua com o reprocessamento de todas as transações armazenadas no *Audit Trial*, até deixar o sistema no estado em que estava no momento de acontecer a falha. Esta segunda e última fase do reparo é conhecida pelo nome de recuperação (*Recovery*). O processamento normal é reiniciado logo após o término do reparo.

Na literatura têm aparecido vários modelos para estudar a técnica de *Checkpointing and Roll-Back and Recovery* (CRR) [12, 14, 15, 16, 26, 29]. A maioria deles foi projetada para determinar os instantes em que devem ser iniciados C's com o objetivo a maximizar de disponibilidade [14], [15]. Alguns outros modelos têm considerado o problema de determinar quando devem ser iniciados os C's com o objetivo de minimizar o tempo de resposta das transações [26]. Os instantes de início dos C's que maximizam a disponibilidade, são, em geral, diferentes daqueles que minimizam o tempo de resposta das transações [26].

O objetivo do capítulo V é resolver analiticamente um modelo mais realista do que os publicados anteriormente para a análise da técnica de CRR. As contribuições mais importantes do capítulo V à área de CRR consistem em ter sido usado e resolvido analiticamente um modelo mais realista que os publicados anteriormente, e ter relacionado pela primeira vez as áreas de desempenhabilidade e CRR. A diferença fundamental do modelo empregado neste trabalho com relação aos modelos resolvidos analiticamente na literatura, consiste em que aqui os tempos de RR são representados em forma exata, porém nos trabalhos anteriores supõe-se que este tempo é proporcional ao tempo transcorrido entre o último C e o instante de acontecimento da falha. Esta diferença é muito importante, segundo indicado em [26], pois nesta referência concluiu-se que o erro das medidas obtidas ao representar os tempos de RR em forma não exata, pode ser muito grande.

Outra contribuição do capítulo V é calcular o valor médio e a função de distribuição da disponibilidade do sistema de Banco de Dados, onde é primordial avaliar-se esta última medida, pois ela fornece um conhecimento muito mais detalhado da operação do sistema do que o valor médio. Cabe aqui destacar que apesar das vantagens de se conhecer a função de distribuição, a grande maioria dos trabalhos que estudaram a disponibilidade calculou o valor médio desta medida, devido a que isto é mais simples. Somente em [15] foi obtida a FDP da disponibilidade, representando, porém, os tempos de RR da forma aproximada antes discutida. A complexidade assintótica dos recursos computacionais necessários para avaliar numericamente o valor médio e a FDP da desempenhabilidade, usando os resultados do capítulo V, expressa-se como um polinômio de baixo grau. Isto é importante, pois apesar do cálculo da FDP ser uma medida complexa de se avaliar, o resultado aqui fornecido torna isto acessível aos computadores de uso corrente nos nossos dias.

A contribuição do capítulo V, que possivelmente seja a mais importante, consiste em haver relacionado, pela primeira vez, áreas até agora desconexas, como são as áreas de CRR e de desempenhabilidade. Especificamente, demonstrou-se, que calcular a FDP da disponibilidade é equivalente a avaliar uma nova medida de desempenhabilidade (definida no capítulo V), na qual, a velocidade com que o sistema ganha recompensa depende do estado em que este se encontra e é não homogênea, é dizer, varia dependendo do instante de observação do sistema.

Por último no capítulo VI são apresentadas as conclusões do trabalho feito, e analisadas algumas possíveis extensões deste.

Capítulo II

Combinação Linear de Estatísticas de Ordem (CLEO)

Os resultados principais deste capítulo são dois. Em primeiro lugar, a dedução da equação II.46, que corresponde a uma nova expressão para a combinação linear das estatísticas de ordem de n variáveis uniformes em $[0, t]$. Por outro lado, a partir de II.46 obteve-se a equação II.81, que já havia sido deduzida em [9] e corresponde à expressão mais eficiente conhecida até agora que permite avaliar computacionalmente a desempenhabilidade $F_i(t, r)$ (ver sua definição na equação I.1). A vantagem do método usado neste trabalho para deduzir II.81 em relação ao usado em [9] é que ele é bastante mais simples e dá uma interpretação (usando as equações II.79 e II.80, que são originais deste trabalho) aos diferentes termos que aparecem em II.81, isto último não foi feito em [9].

O restante deste capítulo está organizado da seguinte forma: nas seções II.1 ... II.4 calcula-se a nova expressão para a FDP da combinação linear das estatísticas de ordem de n variáveis aleatórias uniformes, posteriormente nas seções II.5 e II.6, empregando um novo método, chega-se ao resultado principal de [9].

II.1 Introdução

A necessidade de avaliar a função de distribuição da combinação linear de um conjunto de variáveis aleatórias uniformes num certo intervalo de tempo, aparece ao se estudar algumas medidas da área de *Desempenhabilidade* [5]. Porém outros autores, motivados originalmente por interesses diversos, também chegaram à necessidade de avaliar esta FDP. Por exemplo Matsunawa [20] chegou a esta necessidade ao estudar o problema de *Best-Fit* de variáveis aleatórias. Isto mostra que a obtenção de algoritmos eficientes para calcular a FDP da citada combinação linear é importante para diferentes áreas do conhecimento. Em consequência, este será o problema atacado no presente capítulo.

Inicia-se a seguir a discussão com a definição formal do problema a estudar. Seja $\{X_1, X_2, \dots, X_n\}$ um conjunto de variáveis aleatórias (v.a's) contínuas, independentes e identicamente distribuídas (iid) no intervalo $[0, t]$. Seja $\{X_{(1)}, X_{(2)}, \dots, X_{(n)}\}$ o conjunto de valores obtidos permutando os resultados da execução dos n experimentos aleatórios, de forma de se obter uma ordenação crescente. Em outras palavras: $X_{(i)}$ corresponde ao i -ésimo menor resultado da execução dos n experimentos aleatórios. Para facilitar a compreensão do leitor apresenta-se na figura II.1 um exemplo de execução dos n experimentos aleatórios, desenhados acima da linha segmentada, onde E_j corresponde ao j -ésimo experimento e Y_j é o resultado dele. Embaixo da linha segmentada aparecem desenhadas as diferentes estatísticas de ordem dos resultados obtidos nos n experimentos aleatórios. No caso particular deste exemplo o j -ésimo experimento foi o que gerou o resultado de menor valor, por isto $X_{(1)} = Y_j$; de igual modo o segundo menor resultado foi o do n -ésimo experimento, portanto $X_{(2)} = Y_n$, etc.

No caso particular em que a função de densidade de probabilidade (fdp) de cada v.a. X_i tem distribuição uniforme no intervalo $[0, t]$, as v.a's. X_i , $X_{(j)}$, $0 \leq i, j \leq n + 1$, serão denotadas como U_i e $U_{(j)}$ respectivamente. Adicionalmente, nos casos em que seja necessário explicitar os valores de n e t , as v.a's $U_{(j)}$, $0 \leq j \leq n + 1$, serão escritas da forma $U_{(j)}^{(n)}(t)$.

Considere a variável aleatória $CLEO(\vec{a}_n, t)$, definida pela seguinte equação:

$$CLEO(\vec{a}_n, t) = \sum_{k=1}^{n+1} a_k U_{(k)} \quad (\text{II.1})$$

Onde $\{a_k : k = 1, 2, \dots, n + 1\}$ é um conjunto de constantes pertencentes aos reais e $\vec{a}_n = (a_1, a_2, \dots, a_n, a_{n+1})$.

Um aspecto interessante da equação II.1, é que o seu lado direito estabelece que a variável aleatória $CLEO(\vec{a}_n, t)$ corresponde a uma combinação linear do conjunto das ordens estatísticas $U_{(k)}$, $1 \leq k \leq n + 1$. Este fato aparece ilustrado na figura II.2, na qual cada segmento correspondente a uma certa estatística de ordem $U_{(k)}$ ($1 \leq k \leq n + 1$) está multiplicado pelo coeficiente a_k , que aparece escrito ao lado esquerdo do segmento.

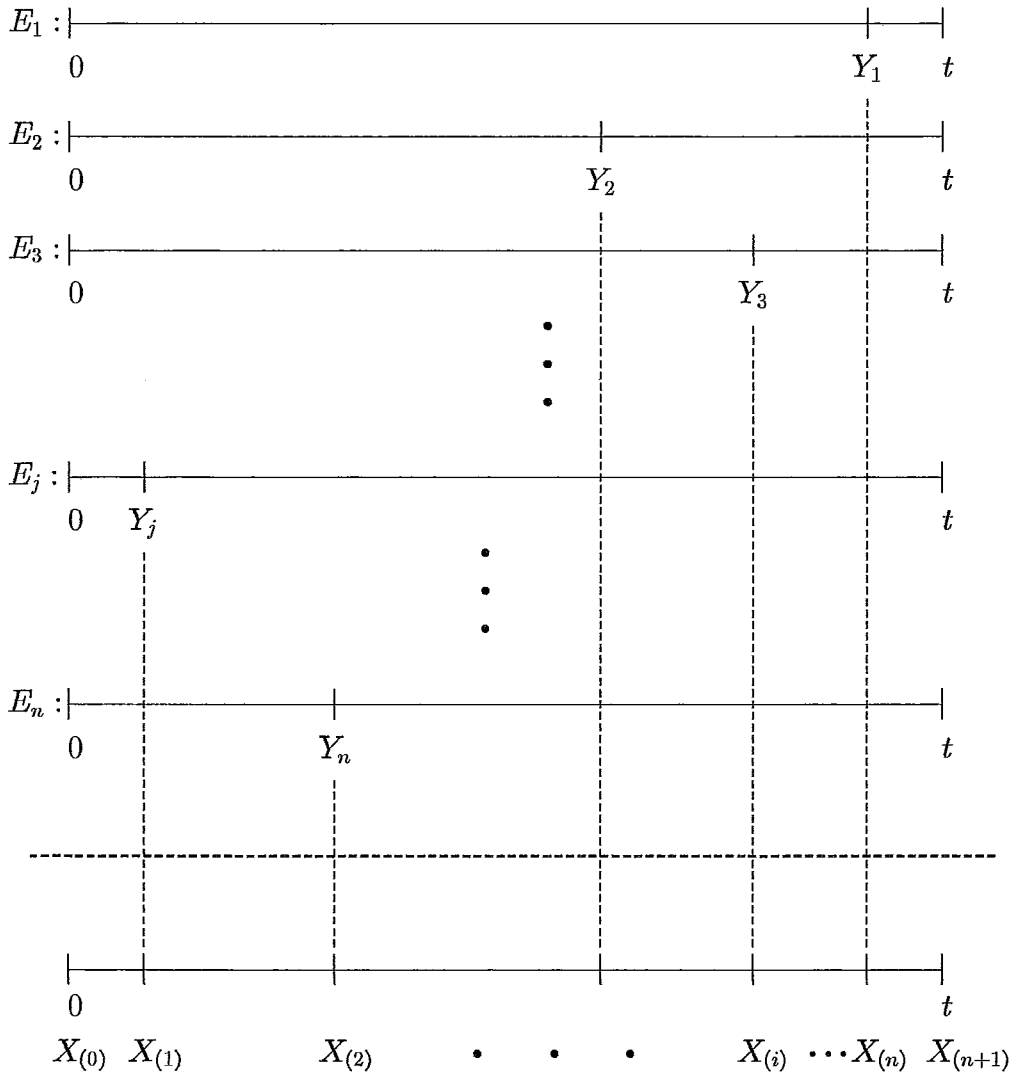


Figura II.1: Estatística de Ordem de n v.a's. com resultados em $[0, t]$.

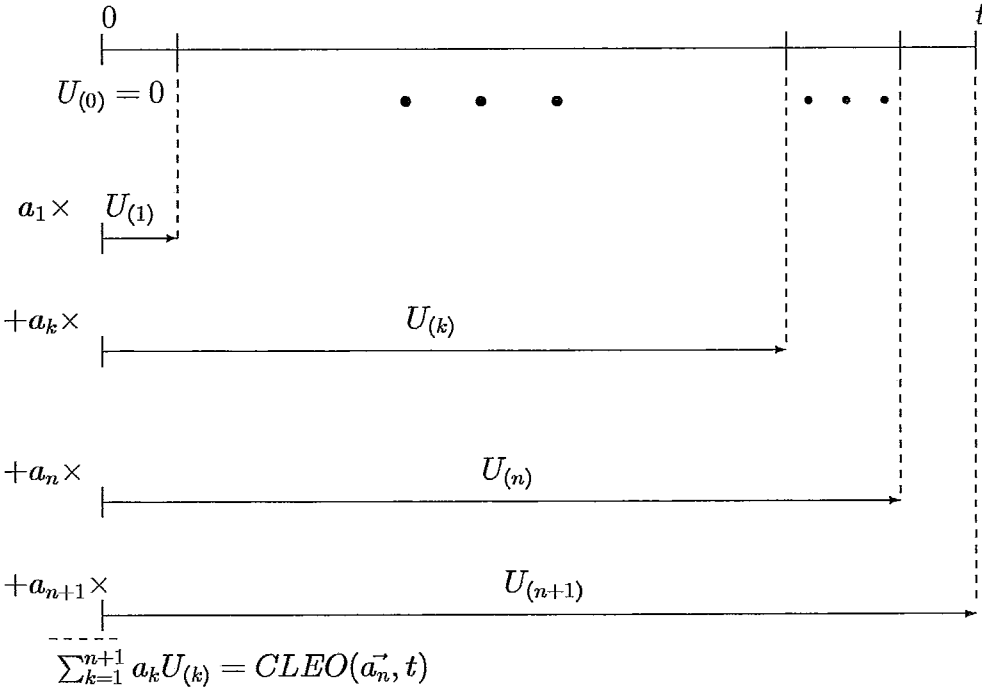


Figura II.2: Combinação Linear de Estatísticas de Ordem de n v.a's uniformes.

Um objetivo importante deste capítulo é calcular $F(\vec{a}_n, t, r)$, que corresponde à F.D.P da v.a. $CLEO(\vec{a}_n, t)$. Em termos formais deseja-se avaliar:

$$\begin{aligned}
 F(\vec{a}_n, t, r) &= P[CLEO(\vec{a}_n, t) \leq r] \\
 &= P\left[\sum_{k=1}^{n+1} a_k U_{(k)} \leq r\right]
 \end{aligned} \tag{II.2}$$

Tal como foi dito, o motivo pelo qual deseja-se avaliar $F(\vec{a}_n, t, r)$ é que em [5] foi demonstrado que esta FDP está intimamente ligada à área de desempenho, que é a área desta tese. Porém, como também já foi comentado, $F(\vec{a}_n, t, r)$ pode ser interpretada de outras formas, algumas das quais se explicitam a seguir.

Em primeiro lugar, considere a figura II.3. Nela aparecem desenhados os conjuntos de variáveis aleatórias $V_{(k)}$, e o conjunto de constantes c_k ($1 \leq k \leq n + 1$) que correspondem a:

$$V_{(k)} = U_{(k)} - U_{(k-1)}, \quad 1 \leq k \leq n + 1 \tag{II.3}$$

$$c_k = \sum_{j=k}^{n+1} a_j, \quad 1 \leq k \leq n + 1 \tag{II.4}$$

Tal como foi indicado em [20], o conjunto de v.a's $V_{(k)}$, $1 \leq k \leq n + 1$, tem distribuição Dirichlet. O interessante da figura II.3 é que em [8] foi indicado que

a v.a. $CLEO(\vec{a}_n, t)$ também corresponde a uma combinação linear das v.a.'s $V_{(k)}$, $1 \leq k \leq n+1$, da seguinte forma:

$$\begin{aligned} CLEO(\vec{a}_n, t) &= \sum_{k=1}^{n+1} a_k U_{(k)} \\ &= \sum_{k=1}^{n+1} c_k V_{(k)} \end{aligned} \quad (\text{II.5})$$

Seja $\vec{c}_n = \{c_1, c_2, \dots, c_n, c_{n+1}\}$. Então:

$$\begin{aligned} F(\vec{a}_n, t, r) &= P[CLEO(\vec{a}_n, t) \leq r] \\ &= P\left[\sum_{k=1}^{n+1} a_k U_{(k)} \leq r\right] \\ &= P\left[\sum_{k=1}^{n+1} c_k V_{(k)} \leq r\right] \end{aligned} \quad (\text{II.6})$$

Em palavras, II.5 estabelece que a v.a. $CLEO(\vec{a}_n, t)$ também corresponde à combinação linear de um conjunto de v.a.'s Dirichlet; concomitantemente II.6 indica que $F(\vec{a}_n, t, r)$ é a FDP desta última combinação linear.

Outra interpretação da v.a. $CLEO(\vec{a}_n, t)$ foi dada em [5], onde foi demonstrado que ela corresponde à recompensa total acumulada no intervalo $[0, t]$ por uma cadeia de Markov (CM) condicionando-se a existência de n transições nesse intervalo. Em termos formais, a observação anterior pode ser estabelecida da seguinte forma. Seja $RTA(t)$ a recompensa total ganha pela CM no intervalo $[0, t]$. Seja n o número de transições da cadeia no intervalo $[0, t]$. Seja $\vec{e}_n = \langle e_{(1)}, e_{(2)}, \dots, e_{(n)}, e_{(n+1)} \rangle$ um dos caminho que pode percorrer a CM. Seja $E = \{e_1, e_2, \dots, e_{|E|}\}$ o conjunto de estados diferentes da CM. Cada estado $e_k \in E$ é associado a uma taxa de recompensa $\rho(e_k)$, isto é, por cada unidade de tempo que o sistema fica no estado e_k ganha uma recompensa igual a $\rho(e_k)$. Seja $R = \{r_1, r_2, \dots, r_{|R|}\}$ o conjunto de taxas de recompensas possíveis que pode ganhar o sistema, onde $|R|$ corresponde ao número de taxas diferentes. Em geral tem-se que $|R| \leq |E|$, já que dois o mais estados podem estar associados a uma mesma taxa de recompensa. Seja $[RTA(t)|\vec{e}_n]$ a recompensa total ganha pelo sistema no intervalo $[0, t]$, dado que na CM aconteceram n transições e ela percorreu o caminho \vec{e}_n . Então, em [5] foi demonstrado que:

$$[RTA(t)|\vec{e}_n] = \sum_{k=1}^{n+1} V_{(k)} \rho(e_{(k)}) \quad (\text{II.7})$$

e portanto, de II.5 e II.7 conclue-se que:

$$CLEO(\vec{a}_n, t) = [RTA(t)|\vec{e}_n] \quad (\text{II.8})$$

Esta equação estabelece que, $CLEO(\vec{a}_n, t)$ também pode ser interpretada como a recompensa total acumulada por uma CM que percorre o caminho \vec{e}_n .

A figura II.4 mostra a interpretação da v.a. $CLEO(\vec{a}_n, t)$ segundo a equação II.8. No eixo vertical aparecem as diferentes taxas de recompensa que pode ganhar o sistema, no eixo horizontal os instantes $\tau_1 \dots \tau_n$ das n transições, e o tamanho dos sucessivos segmentos $V_{(1)}, V_{(2)}, \dots, V_{(n+1)}$. O gráfico corresponde às diferentes taxas de recompensa ganhas pelo sistema no intervalo $[0, t]$. Acima de cada segmento do gráfico são indicados os estados por onde transita o sistema. O fato de que no exemplo $\rho(e_{(1)}) = \rho(e_{(k)}) = r_j$ significa somente que os estados $e_{(1)}$ e $e_{(k)}$ tem a mesma recompensa, e não necessariamente que sejam o mesmo estado, pois diferentes estados podem estar associados a uma mesma recompensa. Outro fato interessante da figura II.4 é que a área do o gráfico corresponde à $[RTA(t)|\vec{e}_n]$. Por último note-se que as equações II.6 ... II.8 implicam que $\rho(e_{(k)})$, corresponde ao coeficiente c_k , $1 \leq k \leq n + 1$ da equação II.6. Este mesmo fato implica que os coeficientes c_k , $1 \leq k \leq n + 1$, podem ser interpretados como as taxas de recompensas associadas aos segmentos $V_{(k)}$, $1 \leq k \leq n + 1$, respectivamente.

Seja $F'(\vec{e}_n, t, r) = P[RTA(t) \leq r|\vec{e}_n]$, então tem-se que:

$$\begin{aligned} F'(\vec{e}_n, t, r) &= P[RTA(t) \leq r|\vec{e}_n] & (\text{II.9}) \\ &= P\left[\sum_{k=1}^{n+1} V_{(k)}\rho(e_{(k)}) \leq r\right] \\ &= P[CLEO(\vec{a}_n, t) \leq r] \\ &= F(\vec{a}_n, t, r) \end{aligned}$$

Onde \vec{a}_n depende de \vec{e}_n da seguinte forma: \vec{e}_n e a função ρ determinam \vec{c}_n , e este vetor mediante a equação II.4 determina \vec{a}_n . A equação II.10 explicita um fato que já tinha sido indicado, já que se $CLEO(\vec{a}_n, t)$ é equivalente a $[RTA(t)|\vec{e}_n]$, evidentemente a FDP destas v.a's deve ser a mesma.

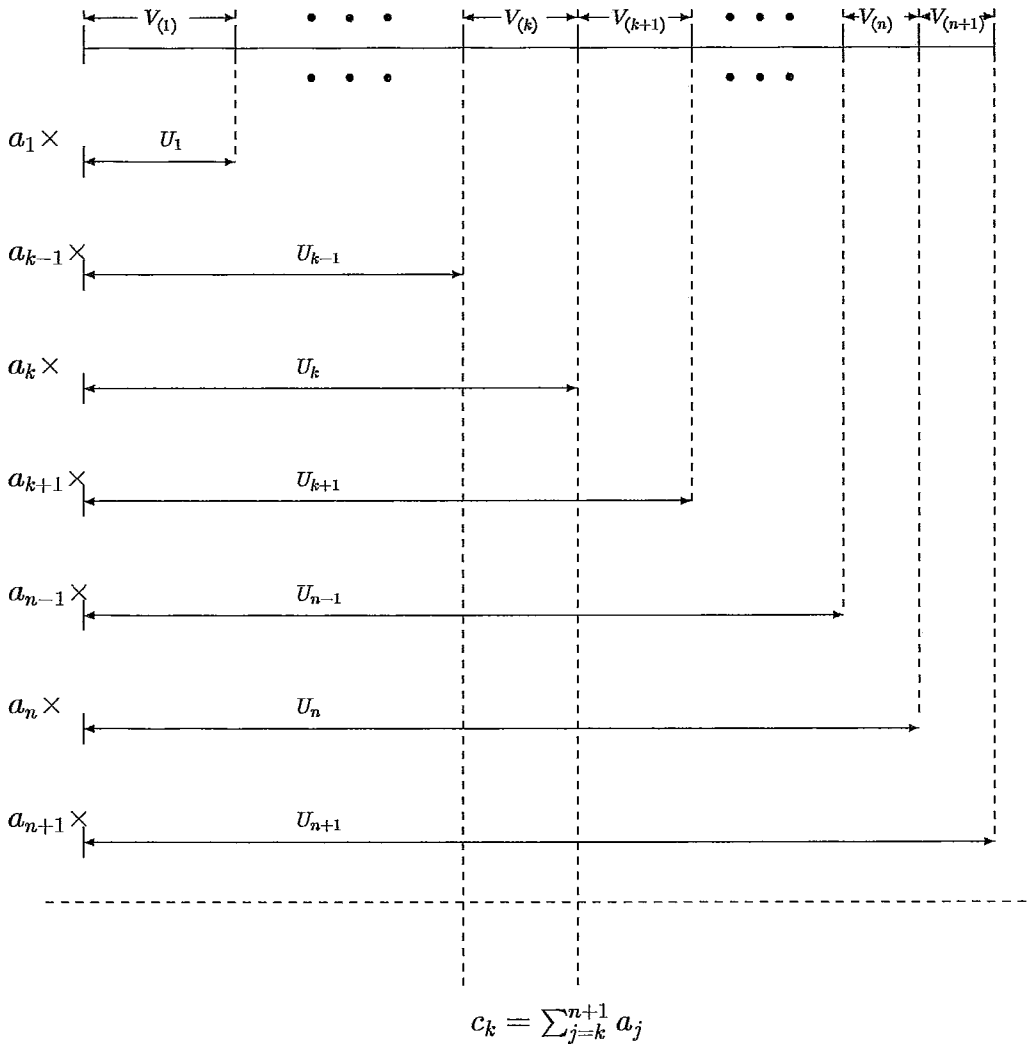


Figura II.3: CLEO de v.a's uniformes interpretada como a Combinação Linear de v.a's Dirichlet.

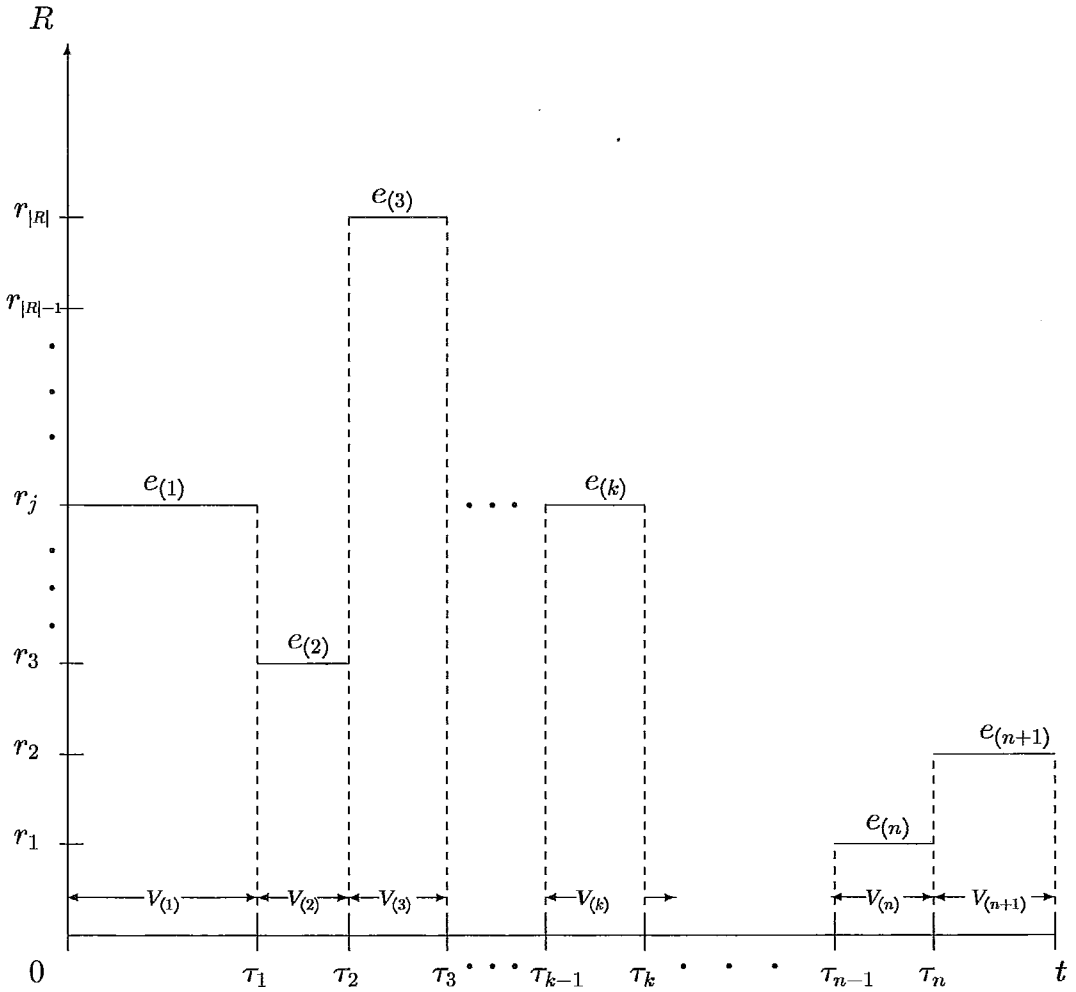


Figura II.4: CLEO de n v.a.'s uniformes interpretada como a recompensa ganha por uma CM com n transições (sem perda de generalidade se assumiu que $r_{|R|} > \dots > r_2 > r_1 > 0$)

II.2 Trabalhos Previos

Na literatura especializada existem relativamente poucos trabalhos sobre o cálculo da função de distribuição de probabilidade da combinação linear das estatísticas de ordem de v.a's uniformes. Dentre eles, Demster e Kleyle [8] estudaram o caso onde $c_1 > c_2 > \dots > c_n > c_{n+1} = 0$. Eles conseguiram chegar a uma expressão analítica fechada para $F(\vec{a}_n, t, r)$. Posteriormente Weisberg [32] generalizou o resultado de Demster e Kleyle, no caso em que o conjunto de constantes $\{c_k, 1 \leq k \leq n\}$ têm somente a restrição de serem valores reais não negativos, e $c_{n+1} = 0$. Um aspecto interessante do resultado de Weisberg é que, ao contrário de Demster e Kleyle, permite calcular a combinação linear de um subconjunto das diferentes ordens estatísticas. Isto se consegue fazendo $c_k = 0$ para todo $U_{(k)}$ que não pertença ao subconjunto de interesse. O resultado de Weisberg entretanto necessita da avaliação recorrente de alguns termos. Posteriormente Matsunawa [20] obteve uma expressão analítica fechada para o cálculo da combinação linear de um subconjunto das ordens estatísticas. Infelizmente nesta expressão aparecem coeficientes que são muito complicados de calcular. Por este motivo Ramalingam [27] sugeriu que antes de calcular a função de distribuição (ou a função de densidade) de probabilidade da v.a. $CLEO(\vec{a}_n, t)$, tais coeficientes sejam avaliados usando algum pacote de 'software' orientado ao cálculo simbólico de termos matemáticos, e posteriormente usar estes valores na expressão de Matsunawa.

Outra forma de avaliar $F(\vec{a}_n, t, r)$ deriva-se ao aplicar a interpretação da v.a $CLEO(\vec{a}_n, t)$ como a recompensa acumulada por um caminho específico de uma CM, da maneira que foi explicada no capítulo I. Especificamente, para avaliar $F(\vec{a}_n, t, r)$, na realidade avalia-se $F'(\vec{c}_n, t, r) = P[\sum_{k=1}^{n+1} c_k V_{(k)} \leq r]$, o que, segundo a equação II.6, é um problema equivalente. Esta forma de avaliação permite usar os resultados obtidos na área de desempenhabilidade, já que o cálculo de $F'(\vec{c}_n, t, r)$ é um problema desse tipo. Para uma visão geral sobre desempenhabilidade referencia-se [7]. O resultado de maior interesse da área de desempenhabilidade para o problema que está discutindo-se, encontra-se em [9], já que a solução apresentada nesse artigo é a de menor custo assintótico computacional aparecida até agora, no caso que o número de taxas diferentes de recompensas associadas aos estados da CM é maior que 2. Seja $R = \{r_1, r_2, \dots, r_{|R|}\}$, o conjunto de valores diferentes que aparecem nos componentes de \vec{c}_n , e seja r_w um componente deste conjunto. Escrevendo, na notação usada aqui, o resultado de [9] para o cálculo de $F'(\vec{c}_n, t, r)$, tem-se que:

$$F'(\vec{c}_n, t, r) = \alpha^{(n)} u(r - r_{\Phi(c_1)} t) + \sum_{h=1}^n \sum_{w=1}^{|R|} \binom{n}{h-1} \beta^{(n)}(w, h) \left(\frac{r - r_w t}{t} \right)^{n-h+1} u(r - r_w t) \quad (\text{II.10})$$

Onde $\Phi : c_k \times [1 \dots |R|]$, $1 \leq k \leq n+1$, é a função que indica a qual elemento do conjunto R corresponde cada um dos elementos do vetor \vec{c}_n , ou seja, $\Phi(c_l) = j$ sse $c_l = r_j$,

$$u(x) = \begin{cases} 0, & \text{se } x < 0 \\ 1, & \text{se } x \geq 0 \end{cases}$$

$$I(e) = \begin{cases} 1, & \text{se o evento 'e' for verdadeiro} \\ 0, & \text{em caso contrario} \end{cases}$$

e os coeficientes $\alpha^{(n)}$ e $\beta^{(n)}(w, h)$ são independentes das variáveis t e r , e avaliados recorrentemente usando as seguintes equações:

$$\alpha^{(0)} = 1 \quad (\text{II.11})$$

$$\alpha^{(n)} = \alpha^{(n-1)} I(r_{\Phi(c_1)} = r_{\Phi(c_2)}); \quad n \geq 1 \quad (\text{II.12})$$

$$\beta^{(n)}(w, n) = -\frac{\alpha^{(n-1)}}{r_w - r_{\Phi(c_1)}}; \quad n \geq 1 \quad (\text{II.13})$$

$w \neq \Phi(c_1)$

$$\beta^{(n)}(w, h) = \frac{\beta^{(n)}(w, h+1)}{(r_w - r_{\Phi(c_1)})} - \frac{\beta^{(n-1)}(w, h)}{(r_w - r_{\Phi(c_1)})} \quad (\text{II.14})$$

$1 \leq h < n; \quad w \neq \Phi(c_1)$

$$\beta^{(n)}(\Phi(c_1), 1) = - \sum_{\substack{w=1 \\ w \neq \Phi(c_1)}}^{|R|} \beta^{(n)}(w, 1); \quad n \geq 1 \quad (\text{II.15})$$

$$\beta^{(n)}(\Phi(c_1), h) = \beta^{(n-1)}(\Phi(c_1), h-1) \quad (\text{II.16})$$

$2 \leq h < n$

Das equações II.10 ... II.16 tem-se que no l -ésimo passo do algoritmo para calcular $F(\vec{a}_n, t, R)$ aplicando o resultado de [9], o maior requisito de memória é o armazenamento dos valores de $\beta^{(l)}(w, h)$, ($1 \leq w \leq |R|$, $1 \leq h \leq l$), o que implica que neste passo é necessário usar $O(l|R|)$ localizações de memória; adicionalmente, devido a que $1 \leq l \leq n$, conclue-se que o máximo requisito de memória, deste algoritmo, é $O(n|R|)$. Por outro lado, dado que as equações II.10 ... II.16 se avaliam em tempo constante e que $1 \leq l \leq n$, chega-se a que para se obter $F(\vec{a}_n, t, r)$ são necessárias $O(n^2|R|)$ operações.

II.3 Avaliação da Função de Distribuição de CLEO (FDP-CLEO)

A seguir desenvolvemos uma nova solução para o cálculo de $F(\vec{a}_n, t, r)$. O método e os passos do raciocínio empregado para resolver este problema são os mesmos que os usados em [9]. A diferença consiste, em que na citada referência, foi resolvido um problema mais complexo, a saber, calcular $F_i(t, r)$. A vantagem principal da solução obtida neste trabalho é que ela é mais eficiente quanto aos requisitos de memória e número de operações do que usar o resultado de [9] para avaliar $F(\vec{a}_n, t, r)$ (da forma indicada no capítulo I).

Para avaliar $F(\vec{a}_n, t, r)$ no caso $n = 0$, por definição cumpre-se que $\vec{a}_n = \langle a_1 \rangle$, e (ver equação II.1) $CLEO(\vec{a}_n, t) = a_1 t$, portanto de II.2:

$$\begin{aligned} F(\vec{a}_0, t, r) &= P[CLEO(\vec{a}_0, t) \leq r] \\ &= P[a_1 t \leq r] \\ &= u[r - a_1 t] \\ &= u[r - r_{\Phi(c_1)} t] \end{aligned} \quad (\text{II.17})$$

Por outro lado, para avaliar $F(\vec{a}_n, t, r)$ para $n \geq 1$, condicionamos no valor de $U_{(1)}^n(t)$. Nesta parte do raciocínio, ao escrever uma estatística de ordem, usam-se explicitamente os argumentos n e t , pois isto é necessário, tal como será visto mais na frente. Condicionando no valor de $U_{(1)}^n(t)$, tem-se que:

$$F(\vec{a}_n, t, r) = \int_0^t F(\vec{a}_n, t, r | U_{(1)}^n(t) = \tau) dP[U_{(1)}^n(t) = \tau], \quad n \geq 1 \quad (\text{II.18})$$

Onde $F(\vec{a}_n, t, r | U_{(1)}^n(t) = \tau)$ é a FDP da CLEO de n v.a's uniformes em $[0, t]$, no caso em que $U_{(1)}^n(t) = \tau$.

É um fato bastante conhecido que (ver [18]):

$$dP[U_{(1)}^n(t) = \tau] = \frac{n(t - \tau)^{n-1}}{t^n} d\tau \quad (\text{II.19})$$

O outro fator do lado direito de II.18, é avaliado usando sua definição, isto é:

$$F(\vec{a}_n, t, r | U_{(1)}^n(t) = \tau) = P \left[\sum_{k=1}^{n+1} a_k U_{(k)}^n(t) \leq r | U_{(1)}^n(t) = \tau \right], \quad n \geq 1 \quad (\text{II.20})$$

Reescrevendo o lado direito de II.20 da seguinte forma:

$$\begin{aligned}
 P \left[\sum_{k=1}^{n+1} a_k U_{(k)}^n(t) \leq r \mid U_{(1)}^n(t) = \tau \right] &= \\
 P \left[\left\{ \sum_{k=1}^{n+1} a_k \left(U_{(k)}^n(t) - U_{(1)}^n(t) \right) \leq r - \sum_{k=1}^{n+1} a_k U_{(1)}^n(t) \right\} \mid U_{(1)}^n(t) = \tau \right] &= \\
 = P \left[\left\{ \sum_{k=2}^{n+1} a_k \left(U_{(k)}^n(t) - \tau \right) \leq r - \sum_{k=1}^{n+1} a_k \tau \right\} \mid U_{(1)}^n(t) = \tau \right] &= \\
 = P \left[\left\{ \sum_{k=2}^{n+1} a_k \left(U_{(k)}^n(t) - \tau \right) \leq r - r_{\Phi(c_1)} \tau \right\} \mid U_{(1)}^n(t) = \tau \right] & \quad (\text{II.21})
 \end{aligned}$$

Pode-se usar o fato que a f.d.p. conjunta de $(U_{(2)}^n(t) - \tau), \dots, (U_{(n+1)}^n(t) - \tau)$, com $n \geq 1$, condicionada a que $U_{(1)}^n(t) = \tau$, é igual a f.d.p. conjunta de $U_{(1)}^{n-1}(t - \tau), \dots, U_{(n)}^{n-1}(t - \tau)$ (ver, por exemplo [18]). Isto implica que neste caso a f.d.p. marginal de $(U_{(k)}^n(t) - \tau)$ (condicionada a $U_{(1)}^n(t) = \tau$) é igual à f.d.p. marginal de $U_{(k-1)}^{n-1}(t - \tau)$, com $n \geq 1$, e $2 \leq k \leq (n + 1)$.

Sustituindo a equação II.21 em II.20, e trocando a variável k por $k - 1$, obtemos que:

$$F(\vec{a}_n, t, r \mid U_{(1)}^n(t) = \tau) = P \left[\sum_{k=1}^n a_{k+1} U_{(k)}^{n-1}(t - \tau) \leq r - r_{\Phi(c_1)} \tau \right], \quad n \geq 1$$

Agora nota-se que o somatório do lado direito desta equação tem a mesma forma da equação II.2, portanto pode ser reescrita da forma:

$$F(\vec{a}_n, t, r \mid U_{(1)}^n(t) = \tau) = F(\vec{a}_{n-1}, t - \tau, r - r_{\Phi(c_1)} \tau), \quad n \geq 1 \quad (\text{II.22})$$

Onde $\vec{a}_{n-1} = \langle a_2, a_3, \dots, a_{n+1} \rangle$. Por último, usando as equações II.19 e II.22 na equação II.18, chega-se à seguinte expressão recorrente:

$$F(\vec{a}_n, t, r) = \int_0^t \frac{n(t - \tau)^{n-1}}{t^n} F(\vec{a}_{n-1}, t - \tau, r - r_{\Phi(c_1)} \tau) d\tau, \quad n \geq 1 \quad (\text{II.23})$$

Resumindo o avanço feito até o momento: as equações II.17 e II.23 fornecem um método recorrente para calcular $F(\vec{a}_n, t, r)$ e foram obtidas usando o mesmo raciocínio de [9].

Continuando o desenvolvimento em forma análoga a [9], o método que será usado para resolver as equações II.17 e II.23, consiste em aplicar duas

vezes a transformada de Laplace, em primeiro lugar sobre a variável r e depois sobre a variável t .

Seja $\mathcal{F}(\vec{a}_n, t, z)$ a transformada de Laplace Stieltjes de $F(\vec{a}_n, t, r)$ sobre a variável r , e seja $\mathcal{L}\mathcal{L}$ o seu operador. Aplicando esta transformada às equações II.17 e II.23, e usando as identidades $u(r-a) \iff e^{-za}$; $f(r-a) \iff e^{-za}\mathcal{L}\mathcal{L}[f(r)]$, onde a é uma constante, obtém-se que:

$$\mathcal{F}(\vec{a}_0, t, z) = e^{-r_{\Phi(c_1)}tz} \quad (\text{II.24})$$

$$\mathcal{F}(\vec{a}_n, t, z) = \frac{n}{t^n} \int_0^t e^{-r_{\Phi(c_1)}\tau z} (t-\tau)^{n-1} \mathcal{F}(\vec{a}_{n-1}, t-\tau, z) d\tau, \quad n \geq 1 \quad (\text{II.25})$$

Seja $\delta(x)$ a derivada da função $u(x)$ definida anteriormente, e seja $\mathcal{H}(s, \vec{a}_n, z)$ a transformada de Laplace de $\mathcal{F}(\vec{a}_n, t, z)$ sobre a variável t . Aplicando esta transformada na equação II.24 e usando a identidade $be^{-at}\delta(t) \iff \frac{b}{s+a}$, onde a e b são constantes, conclui-se que:

$$\mathcal{H}(\vec{a}_0, s, z) = \frac{1}{s + r_{\Phi(c_1)}z} \quad (\text{II.26})$$

Para obter a transformada de Laplace da equação II.25, procede-se da seguinte forma: em primeiro lugar identifica-se que a integral da equação II.25 corresponde à convolução de duas funções, a saber: ' $e^{-r_{\Phi(c_1)}zt}$ ' e ' $t^{n-1}\mathcal{F}(\vec{a}_{n-1}, t, z)$ '. Então denotando ' \otimes ' ao operador convolução, e ' \mathcal{L} ' ao operador da transformada de Laplace, o que precisa ser avaliado é:

$$\mathcal{L} \left[\frac{n}{t^n} \left(e^{-r_{\Phi(c_1)}zt} \otimes t^{n-1} \mathcal{F}(\vec{a}_{n-1}, t, z) \right) \right]$$

Adicionalmente, usando a identidade $\mathcal{L} \left[\frac{f(t)}{t^n} \right] = \int_{b_1=s}^{\infty} \dots \int_{b_n=b_{n-1}}^{\infty} \mathcal{L}[f(t)] db_n \dots db_1$, a transformada de Laplace da equação II.25 fica:

$$\mathcal{H}(\vec{a}_n, t, z) = n \int_{b_1=s}^{\infty} \dots \int_{b_n=b_{n-1}}^{\infty} \mathcal{L} \left[e^{-r_{\Phi(c_1)}zt} \otimes t^{n-1} \mathcal{F}(\vec{a}_{n-1}, t, z) \right] db_n \dots db_1, \quad n \geq 1$$

O fato que a transformada de Laplace de uma convolução é o produto das respectivas transformadas, implica que:

$$\mathcal{H}(\vec{a}_n, t, z) = n \int_{b_1=s}^{\infty} \dots \int_{b_n=b_{n-1}}^{\infty} \mathcal{L} \left[e^{-r_{\Phi(c_1)}zt} \right] \mathcal{L} \left[t^{n-1} \mathcal{F}(\vec{a}_{n-1}, t, z) \right] db_n \dots db_1, \quad n \geq 1$$

Para resolver a equação de acima, usam-se as seguintes identidades:

$$\mathcal{L}[be^{-at}\delta(t)] = \frac{b}{s+a};$$

$$\mathcal{L}[t^{n-1}f(t)] = (-1)^{n-1} \frac{d^{n-1}}{ds^{n-1}} \mathcal{L}[f(t)]$$

Onde $\delta(x)$ corresponde à derivada da função $u(x)$ definida na seção II.2. Usando estas identidades chega-se a que:

$$\mathcal{H}(\vec{a}_n, s, z) = n \int_{b_1=s}^{\infty} \dots \int_{b_n=b_{n-1}}^{\infty} \frac{(-1)^{n-1}}{(b_n + r_{\Phi(c_1)}z)} \frac{d^{n-1}}{db_n^{n-1}} \mathcal{H}(\vec{a}_{n-1}, b_n, z) db_n \dots db_1, \quad n \geq 1 \quad (\text{II.27})$$

II.3.1 Solução da FDP-CLEO

As equações II.26 e II.27, constituem uma solução recorrente ao problema de calcular $F(\vec{a}_n, t, r)$, porém no duplo plano de Laplace 'z' e 's'. A vantagem destas equações com relação às equações II.17 e II.23, é que neste caso elas são mais simples. Portanto, para resolver o problema original, isto é, avaliar $F(\vec{a}_n, t, r)$, o trabalho que falta realizar é encontrar uma solução para as equações II.26 e II.27, e depois, levar esta solução ao plano (t, r) .

Seja $\vec{K}_n = \langle k_1, k_2, \dots, k_{|R|} \rangle$, o vetor definido pela seguinte equação:

$$\vec{K}_n[c] = \sum_{l=1}^{n+1} I[\vec{c}_n[l] = r_c]; \quad 1 \leq c \leq |R| \quad (\text{II.28})$$

Isto é $\vec{K}_n[c]$, $1 \leq c \leq |R|$, corresponde ao número de componentes do vetor \vec{c}_n que são iguais ao valor $r_c \in R$.

O resultado chave desta análise é que a solução das equações II.26 e II.27 está dado pela seguinte equação:

$$\mathcal{H}(\vec{a}_n, s, z) = n! \int_{b_1=s}^{\infty} \dots \int_{b_n=b_{n-1}}^{\infty} \sum_{c=1}^{|R|} \sum_{m=1}^{\vec{K}_n[c]} \frac{g(c, m, \vec{K}_n)}{z^{n+1-m}} \frac{1}{(b_n + r_c z)^m} db_n \dots db_1 \quad (\text{II.29})$$

Onde o conjunto de valores $g(c, m, \vec{K}_n)$, ($1 \leq c \leq |R|$, $1 \leq m \leq \vec{K}_n[c]$), não dependem das variáveis z e s , e serão calculadas mais adiante.

É importante ressaltar que ter encontrado o lado direito de II.29 como sendo a solução de II.26 e II.27, e ter identificado que o método de [9] serve, também, para o cálculo de $F(\vec{a}_n, t, r)$, são as únicas contribuições originais deste trabalho na avaliação desta medida, pois tal como já foi comentado, os passos usados na solução de $F(\vec{a}_n, t, r)$ correspondem exatamente a aqueles de [9].

Entretanto, tal como será visto mais adiante, a solução de II.29 é mais eficiente computacionalmente do que a aplicar o resultado de [9] para avaliar $F(\vec{a}_n, t, r)$. A seguir demonstraremos por indução no valor de n , a validade da equação II.29.

II.3.2 Demonstração da solução

Base da Indução. Para o caso $n = 0$, por definição de \vec{c}_n e \vec{K}_n , tem-se que $\vec{c}_0 = \langle c_1 \rangle$ e $\vec{K}_0 = \vec{1}_{\Phi(c_1)}$. Onde $\vec{1}_c$ é o vetor de dimensão $|R|$, composto somente de zeros, exceto no c -ésimo componente onde vale 1. Em conseqüência, para $n = 0$, a equação II.29 corresponde a:

$$\mathcal{H}(\vec{a}_0, s, z) = \frac{g(\Phi(c_1), 1, \vec{1}_{\Phi(c_1)})}{s + r_{\Phi(c_1)}z} \quad (\text{II.30})$$

Definindo arbitrariamente $g(\Phi(c_1), 1, \vec{1}_{\Phi(c_1)}) = 1$, a equação II.30 coincide com II.26, o que significa que a base da indução foi demonstrada.

Hipotese indutiva. De II.29:

$$\begin{aligned} \frac{d^{n-1}}{db_n^{n-1}} \mathcal{H}(\vec{a}_{n-1}, b_n, z) = & \\ & \frac{d^{n-1}}{db_n^{n-1}} (n-1)! \int_{x_1=b_n}^{\infty} \dots \int_{x_{n-1}=x_{n-2}}^{\infty} \left[\sum_{m=1}^{\vec{K}_{n-1}[\Phi(c_1)]} \frac{g(\Phi(c_1), m, \vec{K}_{n-1})}{z^{n-m}} \right. \\ & \left. + \sum_{\substack{c=1 \\ c \neq \Phi(c_1)}}^{|R|} \sum_{m=1}^{\vec{K}_{n-1}[c]} \frac{g(c, m, \vec{K}_{n-1})}{z^{n-m}} \right] dx_{n-1} \dots dx_1 \end{aligned} \quad (\text{II.31})$$

Onde $\vec{K}_n = \vec{K}_{n-1} + \vec{1}_{\Phi(c_1)}$.

O fato do primeiro segmento estar associado a recompensa c_1 , implica que $\vec{K}_{n-1}[\Phi(c_1)] = (\vec{K}_n[\Phi(c_1)] - 1)$ e $\vec{K}_{n-1}[c] = \vec{K}_n[c]$ para todo $c \neq \Phi(c_1)$. Portanto, no que se segue, estes termos serão usados indistintamente segundo seja conveniente em cada caso.

O fato que os termos r_c , z , e $g(c, m, \vec{K}_{n-1})$ não dependem das variáveis X_1, \dots, X_{n-1} , b_n , implica que o lado direito da equação II.31, em termos abstratos, é uma expressão da forma:

$$\frac{d^{n-1}}{db_n^{n-1}} \int_{x_1=b_n}^{\infty} \dots \int_{x_{n-1}=x_{n-2}}^{\infty} a(x_{n-1} + b)^{-m} dx_{n-1} \dots dx_1$$

onde a e b são constantes.

Usando identidades de cálculo básico, é fácil ver que a expressão acima é igual a $(-1)^{n-1}a(b_n + b)^{-m}$. Isto, junto com o fato dos termos ‘ r_c ’, ‘ z ’ e ‘ $g(c, m, \vec{K}_{n-1})$ ’ não dependerem das variáveis x_1, \dots, x_{n-1}, b_n , permitem reescrever II.31 da forma:

$$\begin{aligned} \frac{d^{n-1}}{db_n^{n-1}} \mathcal{H}(\vec{a}_{n-1}, b_n, z) = \\ (-1)^{n-1} (n-1)! \left[\sum_{m=1}^{\vec{K}_n[\Phi(c_1)]-1} \frac{g(\Phi(c_1), m, \vec{K}_{n-1})}{z^{n-m} (b_n + r_{\Phi(c_1)} z)^m} + \sum_{\substack{c=1 \\ c \neq \Phi(c_1)}}^{|R|} \sum_{m=1}^{\vec{K}_n[c]} \frac{g(c, m, \vec{K}_{n-1})}{z^{n-m} (b_n + r_c z)^m} \right] \end{aligned} \quad (\text{II.32})$$

Sustituindo II.32 em II.27:

$$\begin{aligned} \mathcal{H}(\vec{a}_n, s, z) &= n! \int_{b_1=s}^{\infty} \dots \int_{b_n=b_{n-1}}^{\infty} \left[\sum_{m=1}^{\vec{K}_n[\Phi(c_1)]-1} \frac{g(\Phi(c_1), m, \vec{K}_{n-1})}{z^{n-m} (b_n + r_{\Phi(c_1)} z)^m} \right. \\ &\quad \left. + \sum_{\substack{c=1 \\ c \neq \Phi(c_1)}}^{|R|} \sum_{m=1}^{\vec{K}_n[c]} \frac{g(c, m, \vec{K}_{n-1})}{z^{n-m} (b_n + r_{\Phi(c_1)} z) (b_n + r_c z)^m} \right] db_n \dots db_1 \\ &= n! \int_{b_1=s}^{\infty} \dots \int_{b_n=b_{n-1}}^{\infty} \left[\sum_{m=2}^{\vec{K}_n[\Phi(c_1)]} \frac{g(\Phi(c_1), m-1, \vec{K}_{n-1})}{z^{n+1-m} (b_n + r_{\Phi(c_1)} z)^m} \right. \\ &\quad \left. + \sum_{\substack{c=1 \\ c \neq \Phi(c_1)}}^{|R|} \sum_{m=1}^{\vec{K}_n[c]} \frac{g(c, m, \vec{K}_{n-1})}{z^{n-m} (b_n + r_{\Phi(c_1)} z) (b_n + r_c z)^m} \right] db_n \dots db_1 \end{aligned} \quad (\text{II.33})$$

No processo de demonstrar a validade da equação II.29 chega-se à equação II.33. Portanto, para completar esta demonstração deve-se comprovar que as equações II.29 e II.33 são equivalentes. Para isto reescreve-se II.33 em forma mais conveniente, isto é, usando os mesmos denominadores de II.29. Uma vez que a equação II.33 seja escrita desta forma, a comparação da equivalência entre ambas equações se reduz a comprovar se os numeradores correspondentes a um mesmo denominador são iguais em ambas equações. Para reescrever a equação II.33, da forma indicada, os termos dela são expandidos em frações parciais. Tal como em [9], a expansão é feita usando a seguinte equação, que pode ser demonstrada facilmente por indução no valor de m .

$$\frac{y}{(x+a)(x+b)^m} = \frac{yA}{(x+a)} + \sum_{i=1}^m \frac{yB_i}{(x+b)^i} \quad (\text{II.34})$$

Onde

$$A = \frac{1}{(b-a)^m}; \quad B_i = \frac{-1}{(b-a)^{m-i+1}}$$

Aplicando II.34 em II.33:

$$\begin{aligned} \mathcal{H}(s, \vec{a}_n, z) = & n! \int_{b_1=s}^{\infty} \dots \int_{b_n=b_{n-1}}^{\infty} \left[\sum_{m=2}^{\vec{K}_n[\Phi(c_1)]} \frac{g(\Phi(c_1), m-1, \vec{K}_{n-1})}{z^{n+1-m}} \right. \\ & + \sum_{\substack{c=1 \\ c \neq \Phi(c_1)}}^{|R|} \sum_{m=1}^{\vec{K}_n[c]} \frac{g(c, m, \vec{K}_{n-1})}{z^n (r_c - r_{\Phi(c_1)})^m} \\ & \left. - \sum_{\substack{c=1 \\ c \neq \Phi(c_1)}}^{|R|} \sum_{m=1}^{\vec{K}_n[c]} \sum_{i=1}^m \frac{g(c, m, \vec{K}_{n-1})}{z^{n+1-i} (r_c - r_{\Phi(c_1)})^{m-i+1}} \frac{1}{(b_n + r_c z)^i} \right] db_n \dots db_1 \end{aligned}$$

Usando a identidade $\sum_{m=1}^{\vec{K}_n[c]} \sum_{i=1}^m = \sum_{i=1}^{\vec{K}_n[c]} \sum_{m=i}^{\vec{K}_n[c]}$, e trocando a variável i pela m , a equação de acima pode ser reescrita da forma:

$$\begin{aligned} \mathcal{H}(\vec{a}_n, s, z) = & n! \int_{b_1=s}^{\infty} \dots \int_{b_n=b_{n-1}}^{\infty} \left[\sum_{m=2}^{\vec{K}_n[\Phi(c_1)]} \frac{g(\Phi(c_1), m-1, \vec{K}_{n-1})}{z^{n+1-m}} \right. \\ & + \sum_{\substack{c=1 \\ c \neq \Phi(c_1)}}^{|R|} \sum_{m=1}^{\vec{K}_n[c]} \frac{g(c, m, \vec{K}_{n-1})}{z^n (r_c - r_{\Phi(c_1)})^m} \\ & \left. - \sum_{\substack{c=1 \\ c \neq \Phi(c_1)}}^{|R|} \sum_{m=1}^{\vec{K}_n[c]} \sum_{i=m}^{\vec{K}_n[c]} \frac{g(c, i, \vec{K}_{n-1})}{z^{n+1-m} (r_c - r_{\Phi(c_1)})^{i-m+1}} \frac{1}{(b_n + r_c z)^m} \right] db_n \dots db_1 \quad (\text{II.35}) \end{aligned}$$

Então, tal como foi comentado anteriormente, para completar a demonstração por indução da equação II.29, é necessário que os lados direitos de II.35 e II.29 sejam iguais. Comparando estas expressões, conclue-se que a igualdade é verdadeira somente se os numeradores correspondentes a um mesmo denominador são iguais em ambas equações. Isto implica que, os valores $g(c, m, \vec{K}_n)$, $1 \leq c \leq |R|$; $1 \leq m \leq \vec{K}_n[c]$, da equação II.29, devem ser calculados através das seguintes equações recorrentes:

Da base indutiva temos que:

$$g(\Phi(c_1), 1, \vec{1}_{\Phi(c_1)}) = 1, \quad (\text{II.36})$$

De II.29 e do primeiro termo do lado direito de II.35:

$$g(\Phi(c_1), m, \vec{K}_n) = g(\Phi(c_1), m-1, \vec{K}_{n-1}); \quad n \geq 1 \quad (\text{II.37})$$

$$2 \leq m \leq \vec{K}_n[\Phi(c_1)]$$

De II.29 e do terceiro termo do lado direito de II.35:

$$g(c, m, \vec{K}_n) = - \sum_{\substack{i=m \\ c \neq \Phi(c_1) \\ 1 \leq m \leq \vec{K}_n[c]}}^{\vec{K}_n[c]} \frac{g(c, i, \vec{K}_{n-1})}{(r_c - r_{\Phi(c_1)})^{i-m+1}}; \quad n \geq 1 \quad (\text{II.38})$$

O tempo de cálculo dos termos $g(c, m, \vec{K}_n)$ que aparecem em II.38, pode ser diminuído avaliando-os recorrentemente, de $m = \vec{K}_n[c]$ até 1. Para deduzir estas expressões recorrentes usa-se a própria equação II.38 e isola-se o primeiro termo do somatório. Portanto, lembrando que $\vec{K}_n[c] = \vec{K}_{n-1}[c]$ no caso em que $c \neq \Phi(c_1)$, tem-se:

$$g(c, \vec{K}_n[c], \vec{K}_n) = - \frac{g(c, \vec{K}_{n-1}[c], \vec{K}_{n-1})}{(r_c - r_{\Phi(c_1)})}, \quad n \geq 1 \quad (\text{II.39})$$

$$c \neq \Phi(c_1)$$

e

$$g(c, m, \vec{K}_n) = - \frac{g(c, m, \vec{K}_{n-1})}{(r_c - r_{\Phi(c_1)})} - \sum_{i=m+1}^{\vec{K}_n[c]} \frac{g(c, i, \vec{K}_{n-1})}{(r_c - r_{\Phi(c_1)})^{i-m+1}}; \quad n \geq 1 \quad (\text{II.40})$$

$$c \neq \Phi(c_1)$$

$$1 \leq m < \vec{K}_n[c]$$

Identificando o somatório da equação II.40 em II.38, e dividindo por $(r_c - r_{\Phi(c_1)})$, tem-se que:

$$g(c, m, \vec{K}_n) = \frac{g(c, m+1, \vec{K}_n)}{(r_c - r_{\Phi(c_1)})} - \frac{g(c, m, \vec{K}_{n-1})}{(r_c - r_{\Phi(c_1)})}; \quad n \geq 1 \quad (\text{II.41})$$

$$c \neq \Phi(c_1)$$

$$1 \leq m < \vec{K}_n[c]$$

De II.29 e do segundo termo do lado direito de II.35:

$$g(\Phi(c_1), 1, \vec{K}_n) = \sum_{\substack{c=1 \\ c \neq \Phi(c_1)}}^{|R|} \sum_{i=1}^{\vec{K}_n[c]} \frac{g(c, i, \vec{K}_{n-1})}{(r_c - r_{\Phi(c_1)})^i}, \quad n \geq 1 \quad (\text{II.42})$$

Identificando o segundo somatório desta equação em II.38 e notando que ele é diferente de zero somente se $\vec{K}_n[c] > 0$, a expressão para $g(\Phi(c_1), 1, \vec{K}_n)$ fica:

$$g(\Phi(c_1), 1, \vec{K}_n) = - \sum_{\substack{c=1 \\ c \neq \Phi(c_1); \vec{K}_n[c] > 0}}^{|R|} g(c, 1, \vec{K}_n), \quad n \geq 1 \quad (\text{II.43})$$

Os termos calculados nas equações II.36 ... II.43 são disjuntos e correspondem a todos os termos $g(c, m, \vec{K}_n)$ da equação II.29. Note-se, também, que estas equações confirmam a suposição feita anteriormente de que os termos $g(c, m, \vec{K}_n)$ de II.29 não dependem das variáveis s e z . Estas observações completam a demonstração de II.29.

Como já indicou-se anteriormente, os passos seguintes (inversão de II.29) forneceram uma equação mais eficiente para o cálculo da combinação linear de estatísticas de ordem do que o uso direto da equação II.10, que corresponde a aplicação direta do resultado de [9].

II.3.3 Inversão da solução desde o plano de Laplace

Para inverter o lado direito da equação II.29 nas variáveis z e s usa-se o fato de que, as expressões de $g(c, m, \vec{K}_n)$ $1 \leq c \leq |R|$; $1 \leq m \leq \vec{K}_n[c]$ não dependem das variáveis s e z .

Seja \mathcal{L}^{-1} é o operador da transformada de Laplace inversa. Em primeiro lugar inverte-se a equação II.29 sobre a variável s . Para isto usa-se a igualdade [19]:

$$\mathcal{L}^{-1} \left[\int_{b_1=s}^{\infty} \dots \int_{b_n=b_{n-1}}^{\infty} \mathcal{L}(f(t)) db_n \dots db_1 \right] = \frac{f(t)}{t^n}$$

Com o qual II.29 fica da forma:

$$\mathcal{F}(\vec{a}_n, t, z) = \frac{n!}{t^n} \sum_{c=1}^{|R|} \sum_{m=1}^{\vec{K}_n[c]} \mathcal{L}^{-1} \left[\frac{g(c, m, \vec{K}_n)}{z^{n+m-1}} \right] \frac{1}{(s + r_c z)^m}$$

Adicionalmente, usando a igualdade [19]:

$$\mathcal{L}^{-1} \left[\frac{b}{(s+a)^m} \right] = b \frac{t^{m-1}}{(m-1)!} e^{-at}$$

Onde a e b são constantes, obtem-se:

$$\mathcal{F}(\vec{a}_n, t, z) = \sum_{c=1}^{|R|} \sum_{m=1}^{\vec{K}_n[c]} \frac{n!g(c, m, \vec{K}_n) e^{-r_c z t}}{(m-1)! t^{n+1-m} z^{n+1-m}}, \quad n \geq 0 \quad (\text{II.44})$$

Agora deve-se inverter o lado direito de II.44 sobre a variável z . Sejam $\mathcal{L}\mathcal{L}^{-1}$ o operador da transformada de Laplace Stieltjes inversa na variável z , e $f(\vec{a}_n, t, r')$ a fdp da v.a. $CLEO(\vec{a}_n, t)$ avaliada no ponto r' . Para inverter o lado direito de II.44 usa-se as seguintes identidades da transformada de Laplace Stieltjes:

$$\mathcal{L}\mathcal{L}^{-1} \left[\frac{f(z)}{z^a} \right] = \underbrace{\int_{-\infty}^x \dots \int_{-\infty}^x}_{a \text{ vezes}} \mathcal{L}\mathcal{L}^{-1}[f(z)](dx)^a;$$

$$\mathcal{L}\mathcal{L}^{-1}[e^{-az}] = \delta(x - a)$$

Onde $\delta(x)$ é a derivada da função $u(x)$ definida anteriormente. Com isto tem-se que:

$$f(\vec{a}_n, t, r') = \sum_{c=1}^{|R|} \sum_{m=1}^{\vec{K}_n[c]} \underbrace{\int_{-\infty}^{r'} \dots \int_{-\infty}^{r'}}_{n+1-m \text{ vezes}} \frac{n!g(c, m, \vec{K}_n) \delta(x - r_c t)}{(m-1)! t^{n+1-m}} (dx)^{n+1-m}$$

Efetutando as $(n+1-m)$ integrais:

$$f(\vec{a}_n, t, r') = \sum_{c=1}^{|R|} \sum_{m=1}^{\vec{K}_n[c]} \frac{n!g(c, m, \vec{K}_n) (r' - r_c t)^{n-m}}{(m-1)!(n-m)! t^{n+1-m}} u[r' - r_c t]; \quad n \geq 0 \quad (\text{II.45})$$

Por último, integrando a equação anterior sobre a variável r' , obtem-se o resultado final, ou seja:

$$F(\vec{a}_n, t, r) = \sum_{c=1}^{|R|} \sum_{m=1}^{\vec{K}_n[c]} \binom{n}{m-1} g(c, m, \vec{K}_n) \left(\frac{r - r_c t}{t} \right)^{n+1-m} u[r - r_c t] \quad (\text{II.46})$$

Onde as expressões $g(c, m, \vec{K}_n)$, $1 \leq c \leq |R|$; $1 \leq m \leq \vec{K}_n[c]$; $n \geq 0$; $\sum_{c=1}^{|R|} \vec{K}_n[c] = n+1$, são avaliadas recorrentemente usando as equações II.36, II.37, II.39, II.41 e II.43. Note-se que a expressão de II.46 para $F(\vec{a}_n, t, r)$ é muito similar à expressão de II.10, que corresponde à aplicação direta do resultado de [9], para esta FDP, segundo foi explicado anteriormente. Porém a expressão da equação II.46 é mais eficiente de calcular numericamente, tal como será visto na seção II.3.4.

É interessante notar que a solução dada pela equação II.46, depende do número de segmentos $V_{(k)}$ (ver figura II.3) associados a cada recompensa e não

da ordem desses segmentos. Esta característica da solução é uma conseqüência do fato dos segmentos $V_{(k)}$ ($1 \leq k \leq n+1$) serem intercambiáveis [18]. Formalmente a observação anterior é equivalente ao que segue. Seja $\vec{p}\vec{c}_n$ um vetor obtido permutando (de alguma forma) os componentes de \vec{c}_n , então a equação II.46 implica que:

$$F'(\vec{c}_n, t, r) = F'(\vec{p}\vec{c}_n, t, r) \quad (\text{II.47})$$

II.3.4 Complexidade Computacional

Para calcular numericamente a equação II.46, o maior custo de memória é o necessário à avaliação dos diferentes termos $g(c, m, \vec{K}_n)$.

Usando o fato que $\sum_{c=1}^{|R|} \sum_{m=1}^{\vec{K}_n[c]} 1 = n+1$, é fácil verificar que existem $n+1$ termos $g(c, m, \vec{K}_n)$. Como o algoritmo para o cálculo de $g(c, m, \vec{K}_l)$ é recorrente em l , e como no passo número l só precisamos dos valores do passo anterior, conclue-se que para avaliar II.46 precisa-se de no máximo $2n$ unidades de memória. Analogamente, das equações II.37, II.39, II.41 e II.43, conclue-se que a avaliação de $F(\vec{a}_n, t, r)$ requer de $O(n^2)$ operações de multiplicação ou divisão. Por outro lado, tal como foi visto anteriormente, a aplicação direta do resultado de [9] para avaliar $F(\vec{a}_n, t, r)$ da forma dada pela equação II.10 requer de $O(n|R|)$ localizações de memória e de $O(n^2|R|)$ operações. Em conclusão, a análise anterior demonstra que o resultado obtido neste trabalho é mais eficiente que a aplicação direta de [9] para o cálculo de $F(\vec{a}_n, t, r)$.

II.4 Exemplos

II.4.1 Exemplo 1

Para ilustrar o cálculo recorrente dos diferentes termos $g(c, m, \vec{K}_n)$, considere o caso em que: $n = 4$, $R = \{1, 2, 4\}$ (o que implica que $|R| = 3$), $\vec{c}_4 = \{1, 2, 4, 2\}$. Portanto: $r_1 = 1, r_2 = 2, r_3 = 4$, $\vec{K}_4 = \langle 1, 2, 1 \rangle$, e para avaliar II.46 é preciso calcular $g(1, 1, \langle 1, 2, 1 \rangle)$, $g(2, 1, \langle 1, 2, 1 \rangle)$, $g(2, 2, \langle 1, 2, 1 \rangle)$, e $g(3, 1, \langle 1, 2, 1 \rangle)$. Onde os 4 termos $g(c, m, \vec{K}_n)$ são avaliados recorrentemente, da seguinte forma:

$$\begin{aligned} g(2, 2, \langle 1, 2, 1 \rangle) &= \text{função de (ver II.39) } g(2, 2, \langle 0, 2, 1 \rangle), \\ g(2, 1, \langle 1, 2, 1 \rangle) &= \text{função de (ver II.41) } g(2, 2, \langle 1, 2, 1 \rangle), g(2, 1, \langle 0, 2, 1 \rangle), \\ g(3, 1, \langle 1, 2, 1 \rangle) &= \text{função de } g(3, 2, \langle 1, 2, 1 \rangle) = 0, g(3, 1, \langle 0, 2, 1 \rangle), \\ g(1, 1, \langle 1, 2, 1 \rangle) &= \text{função de (ver II.43) } g(2, 1, \langle 1, 2, 1 \rangle), g(3, 1, \langle 1, 2, 1 \rangle). \end{aligned}$$

Continuando, precisamos avaliar $g(2, 2, \langle 0, 2, 1 \rangle) = 0$, $g(2, 1, \langle 0, 2, 1 \rangle)$, e $g(3, 1, \langle 0, 2, 1 \rangle) = 0$.

$g(2, 2, < 0, 2, 1 >) =$ função de (ver II.37) $g(2, 1, < 0, 1, 1 >)$, e assim por diante.

II.4.2 Exemplo 2

Neste exemplo usa-se a equação II.46 para avaliar $F(\vec{a}_n, t, r)$, no caso particular que $|R|=2$.

Sem perda de generalidade supõe-se que $r_2 > r_1$. Neste caso, trivialmente tem-se que:

$$F(\vec{a}_n, t, r) = \begin{cases} 1, & r \geq r_2 t > r_1 t \\ 0, & r_2 t > r_1 t \geq r \end{cases} \quad (\text{II.48})$$

No caso não trivial, quando $r_2 t > r > r_1 t$, a equação II.46 fica (note que $u(r - r_2 t) = 0$):

$$F(\vec{a}_n, t, r) = \sum_{m=1}^{\vec{K}_n[1]} \binom{n}{m-1} g(1, m, \vec{K}_n) \left(\frac{r - r_1 t}{t} \right)^{n+1-m} \quad (\text{II.49})$$

Para avaliar $g(1, m, \vec{K}_n)$, aplica-se $(m-1)$ vezes a equação II.37, com o qual chega-se a que:

$$g(1, m, \vec{K}_n) = g(1, 1, \vec{K}_{n+1-m}) \quad (\text{II.50})$$

Agora usando a equação II.43, tem-se que:

$$g(1, m, \vec{K}_n) = -g(2, 1, \vec{K}_{n+1-m}) \quad (\text{II.51})$$

Seja $l = n + 1 - m$. Usando o fato que $\vec{K}_n[1] + \vec{K}_n[2] = n + 1$, e substituindo II.51 em II.49:

$$F(\vec{a}_n, t, r) = - \sum_{l=\vec{K}_n[2]}^n \binom{n}{l} g(2, 1, \vec{K}_l) \left(\frac{r - r_1 t}{t} \right)^l \quad (\text{II.52})$$

A seguir, aplicando 3 vezes sucessivas a equação II.41 e agrupando termos chega-se a que:

$$g(2, 1, \vec{K}_l) = \frac{g(2, 2, \vec{K}_l)}{(r_2 - r_1)} - \frac{g(2, 1, \vec{K}_{l-1})}{(r_2 - r_1)}$$

$$\begin{aligned}
&= \frac{1}{(r_2 - r_1)} \left\{ \frac{g(2, 3, \vec{K}_l)}{(r_2 - r_1)} - \frac{g(2, 2, \vec{K}_{l-1})}{(r_2 - r_1)} - \frac{g(2, 2, \vec{K}_{l-1})}{(r_2 - r_1)} \right. \\
&\quad \left. + \frac{g(2, 1, \vec{K}_{l-2})}{(r_2 - r_1)} \right\} \\
&= \frac{1}{(r_2 - r_1)^2} \left\{ g(2, 3, \vec{K}_l) - 2g(2, 2, \vec{K}_{l-1}) + g(2, 1, \vec{K}_{l-2}) \right\} \\
&= \frac{1}{(r_2 - r_1)^2} \left\{ \frac{g(2, 4, \vec{K}_l)}{(r_2 - r_1)} - \frac{g(2, 3, \vec{K}_{l-1})}{(r_2 - r_1)} - \frac{2g(2, 3, \vec{K}_{l-1})}{(r_2 - r_1)} \right. \\
&\quad \left. + \frac{2g(2, 2, \vec{K}_{l-2})}{(r_2 - r_1)} + \frac{g(2, 2, \vec{K}_{l-2})}{(r_2 - r_1)} - \frac{g(2, 1, \vec{K}_{l-3})}{(r_2 - r_1)} \right\} \\
&= \frac{1}{(r_2 - r_1)^3} \left\{ g(2, 4, \vec{K}_l) - 3g(2, 3, \vec{K}_{l-1}) + 3g(2, 2, \vec{K}_{l-2}) \right. \\
&\quad \left. - g(2, 1, \vec{K}_{l-3}) \right\} \tag{II.53}
\end{aligned}$$

O fator entre chaves do último lado direito da equação anterior tem a forma do binômio. Isto leva à equação (a qual é facilmente demonstrada por indução no valor de l):

$$g(2, 1, \vec{K}_l) = \frac{1}{(r_2 - r_1)^l} \sum_{k=0}^l \binom{l}{k} (-1)^{l-k} g(2, k+1, \vec{K}_k) \tag{II.54}$$

Sustituindo II.54 em II.52, tem-se:

$$F(\vec{a}_n, t, r) = \sum_{l=\vec{K}_n[2]}^n \binom{n}{l} \left(\frac{r - r_1 t}{r_2 t - r_1 t} \right)^l \sum_{k=0}^l \binom{l}{k} (-1)^{l-k+1} g(2, k+1, \vec{K}_k) \tag{II.55}$$

Porém, II.36 e II.37 implicam que: $g(2, k+1, \vec{K}_k) = 1$ para $0 \leq k < \vec{K}_n[2]$, e II.46 implica que $g(2, k+1, \vec{K}_k) = 0$ para $k \geq \vec{K}_n[2]$, logo:

$$g(2, k+1, \vec{K}_k) = \begin{cases} 1, & 0 \leq k < \vec{K}_n[2] \\ 0, & k \geq \vec{K}_n[2] \end{cases} \tag{II.56}$$

Sustituindo II.56 em II.55:

$$F(\vec{a}_n, t, r) = \sum_{l=\vec{K}_n[2]}^n \left(\frac{r - r_1 t}{r_2 t - r_1 t} \right)^l \binom{n}{l} \left\{ - \sum_{k=0}^{\vec{K}_n[2]-1} \binom{l}{k} (-1)^{l-k} \right\} \tag{II.57}$$

O teorema do binômio, estabelece que:

$$0 = (1 - 1)^l = \sum_{k=0}^l \binom{l}{k} (-1)^{l-k} 1^k$$

Portanto:

$$\begin{aligned}
 F(\vec{a}_n, t, r) &= \sum_{l=\vec{K}_n[2]}^n \left(\frac{r - r_1 t}{r_2 t - r_1 t} \right)^l \binom{n}{l} \left\{ - \left[0 - \sum_{k=\vec{K}_n[2]}^l \binom{l}{k} (-1)^{l-k} \right] \right\} \\
 &= \sum_{l=\vec{K}_n[2]}^n \left(\frac{r - r_1 t}{r_2 t - r_1 t} \right)^l \binom{n}{l} \sum_{k=\vec{K}_n[2]}^l \binom{l}{k} (-1)^{l-k}
 \end{aligned} \tag{II.58}$$

Seja $p = \left(\frac{r - r_1 t}{r_2 t - r_1 t} \right)$. Usando o fato de que $\binom{n}{l} \binom{l}{k} = \binom{n}{k} \binom{n-k}{l-k}$, finalmente chega-se a que:

$$F(\vec{a}_n, t, r) = \sum_{l=\vec{K}_n[2]}^n p^l \sum_{k=\vec{K}_n[2]}^l (-1)^{l-k} \binom{n}{k} \binom{n-k}{l-k} \tag{II.59}$$

O que corresponde à solução do exemplo.

É interessante notar que para $|R| = 2$ tem-se que (ver por exemplo [6]):

$$F(\vec{a}_n, t, r) = P \left[U_{(\vec{K}_n[2])}^n(t) \leq \frac{r - r_1 t}{r_2 - r_1} \right] \tag{II.60}$$

Entretanto, é conhecido o fato que (ver [18])

$$P \left[U_{(\vec{K}_n[2])}^n(t) \leq \frac{r - r_1 t}{r_2 - r_1} \right] = \sum_{k=\vec{K}_n[2]}^n \binom{n}{k} p^k (1-p)^{n-k} \tag{II.61}$$

Portanto,

$$F(\vec{a}_n, t, r) = \sum_{k=\vec{K}_n[2]}^n \binom{n}{k} p^k (1-p)^{n-k} \tag{II.62}$$

Por último, é fácil demonstrar a igualdade dos lados direitos das equações II.58 e II.62.

II.5 Avaliação da Desempenhabilidade $F_i(t, r)$ a partir da CLEO

II.5.1 Introdução

O resultado de [9] para o cálculo da Desempenhabilidade de sistemas tolerantes a falhas, tem um custo polinomial de baixo grau, e portanto a solução é eficiente.

Porém o método de solução ali empregado não fornece nenhuma interpretação para os termos da expressão final. Entretanto, interpretar tais termos é relevante por vários motivos, que vão desde o cheque de resultados intermediários durante o cálculo de termos do algoritmo, até a possibilidade de fornecer subsídios para outras aplicações.

O objetivo da seção II.5 é desenvolver um novo método para obter o resultado de [9] e em dar uma interpretação aos termos da solução de [9]. A vantagem do método aqui empregado com relação ao usado em [9] é que este é significativamente mais simples, e usa, somente, argumentação probabilística.

II.5.2 Avaliação da Desempenhabilidade

Por definição de desempenhabilidade $F_i(t, r)$ tem-se que:

$$F_i(t, r) = P[RTA_i \leq r] \quad (\text{II.63})$$

Com o objetivo de calcular $F_i(t, r)$ será usada a técnica de randomização. Para isto, em primeiro lugar condiciona-se no número de eventos do processo de Poisson $N(t)$ de taxa Λ que acontecem no intervalo $[0, t]$. Condicionando e descondicionando, tem-se:

$$F_i(t, r) = \sum_{n=0}^{\infty} e^{-\Lambda t} \frac{(\Lambda t)^n}{n!} F_i(t, r|n) \quad (\text{II.64})$$

Onde $F_i(t, r|n)$ corresponde à desempenhabilidade no caso que $N(t) = n$. Isto é:

$$F_i(t, r|n) = P[RTA_i(t) \leq r | N(t) = n] \quad (\text{II.65})$$

Seja $\vec{K}_n = \langle k_1, k_2, \dots, k_{|R|} \rangle$, tal que $k_c \geq 0$, $1 \leq c \leq |R|$, e $|\vec{K}_n| = \sum_{c=1}^{|R|} k_c = n+1$. Na equação II.28 o vetor \vec{K}_n foi definido implicitamente em função de \vec{c}_n (ou analogamente \vec{a}_n), porém no contexto atual \vec{K}_n não é função de nenhum argumento. Porém sempre que usa-se \vec{K}_n , a única informação de interesse é o valor de seus componentes e não como eles são obtidos, portanto o fato de usar \vec{K}_n com este novo significado não deve causar ambigüidades.

Seja $\Gamma_i[\vec{K}_n]$ a probabilidade agregada de todos os caminhos de tamanho $(n+1)$ que partem no estado e_i , e que transitam por $\vec{K}_n[1]$ estados com recompensa r_1 , $\vec{K}_n[2]$ estados com recompensa r_2 , \dots , e $\vec{K}_n[|R|]$ estados com recompensa $r_{|R|}$. Em outras palavras: o c -ésimo elemento de \vec{K}_n indica o número de segmentos $V_{(l)}$, $1 \leq l \leq n+1$, que tem associada recompensa r_c (ver figura II.4). Para avaliar $F_i(t, r|n)$, na equação II.65 condiciona-se no valor de \vec{K}_n , e aplica-se o teorema de probabilidades totais, de onde:

$$\begin{aligned}
F_i(t, r|n) &= \sum_{\vec{K}_n} \Gamma_i[\vec{K}_n] P[RTA_i(t) \leq r | N = n, \vec{K}_n] \\
&= \sum_{\vec{K}_n} \Gamma_i[\vec{K}_n] P[RTA_i(t) \leq r | \vec{K}_n]
\end{aligned} \tag{II.66}$$

$\Gamma_i[\vec{K}_n]$ pode ser calculado recorrentemente da seguinte forma:

$$\Gamma_i[\vec{K}_n] = \begin{cases} \Gamma_i[\vec{K}_n] = 0, & \text{se } \vec{K}_n[\psi(e_i)] = 0 \\ \sum_{e_j \in E} P_{ij} \Gamma_j[\vec{K}_n - \vec{1}_{\psi(e_i)}], & n \geq 1 \end{cases} \tag{II.67}$$

Onde $\psi(e_i)$ é o índice no conjunto R da recompensa associada ao estado e_i , isto é $\psi(e_i) = c$ sse. $\rho(e_i) = r_c$. Note que $\vec{K}_n[\psi(e_i)] = 0$ é impossível de acontecer, pois o estado do primeiro segmento é e_i .

A equação II.67 é semelhante à equação 5.4.2.1 de [5], exceto que em [5] a condição é feita no último estado visitado, e em II.67 condiciona-se no primeiro estado visitado. Esta nova condição será importante como será visto abaixo.

Para avaliar $P[RTA_i(t) \leq r | \vec{K}_n]$ usa-se a definição do evento $[RTA_i(t) \leq r | \vec{K}_n]$ e o fato que os segmentos $V_{(l)}$, $1 \leq l \leq n+1$ são intercambiáveis [18]. De onde:

$$P[RTA_i(t) \leq r | \vec{K}_n] = P \left[\sum_{c=1}^{|R|} \sum_{m=1}^{\vec{K}_n[c]} r_c V_{(\varphi(c,m))} \leq r | \vec{K}_n \right] \tag{II.68}$$

Onde $\varphi(c, m) = l$, $1 \leq l \leq n+1$, se somente se o m -ésimo segmento com recompensa r_c (contando de esquerda a direita), é o l -ésimo dos $n+1$ segmentos do intervalo $[0, t]$ (ver figura II.4). Note que $\sum_{c=1}^{|R|} \sum_{m=1}^{\vec{K}_n[c]} 1 = n+1$.

As equações II.6 e II.68 implicam que $P[RTA_i(t) \leq r | \vec{K}_n]$ corresponde à FDP da combinação linear de n v.a. uniformes em $[0, t]$, portanto para calcular esta FDP usa-se II.46.

$$P[RTA_i(t) \leq r | \vec{K}_n] = \sum_{c=1}^{|R|} \sum_{m=1}^{\vec{K}_n[c]} \binom{n}{m-1} g(c, m, \vec{K}_n) \left(\frac{r - r_c t}{t} \right)^{n+1-m} u[r - r_c t] \tag{II.69}$$

Tal como já tinha sido comentado, $P[RTA_i(t) \leq r | \vec{K}_n]$ não depende do caminho específico percorrido pela CM, senão que depende somente do fato de que este caminho tenha $\vec{K}_n[1]$ segmentos em estados com recompensas r_1 , $\vec{K}_n[2]$ estados com recompensa r_2 , etc.

Para explicitar a dependência com relação aos diferentes argumentos, a seguir são reproduzidas as equações que permitem calcular recorrentemente o conjunto de termos $g(c, m, \vec{K}_n)$.

$$g(\psi(e_i), 1, \vec{K}_0) = I \left[\vec{K}_0 = \vec{1}_{\psi(e_i)} \right] \quad (\text{II.70})$$

$$g(\psi(e_i), m, \vec{K}_n) = g(\psi(e_i), m-1, \vec{K}_{n-1}); \quad n \geq 1 \quad (\text{II.71})$$

$$2 \leq m \leq \vec{K}_n[\psi(e_i)]$$

$$g(\psi(e_i), 1, \vec{K}_n) = - \sum_{\substack{c=1 \\ c \neq \psi(e_i); \vec{K}_n[c] > 0}}^{|\mathcal{R}|} g(c, 1, \vec{K}_n); \quad n \geq 1 \quad (\text{II.72})$$

$$g(c, \vec{K}_n[c], \vec{K}_n) = - \frac{g(c, \vec{K}_{n-1}[c], \vec{K}_{n-1})}{(r_c - r_{\psi(e_i)})}; \quad n \geq 1 \quad (\text{II.73})$$

$$c \neq \psi(e_i)$$

$$g(c, m, \vec{K}_n) = \frac{g(c, m+1, \vec{K}_n)}{(r_c - r_{\psi(e_i)})} - \frac{g(c, m, \vec{K}_{n-1})}{(r_c - r_{\psi(e_i)})}; \quad n \geq 1 \quad (\text{II.74})$$

$$c \neq \psi(e_i)$$

$$1 \leq m < \vec{K}_n[c]$$

Agora voltando ao cálculo da desempenhabilidade, usa-se a equação II.68 na equação II.66, com o que obtém-se:

$$F_i(t, r|n) = \sum_{\vec{K}_n} \Gamma_i \left[\vec{K}_n \right] \sum_{c=1}^{|\mathcal{R}|} \sum_{m=1}^{\vec{K}_n[c]} \binom{n}{m-1} g(c, m, \vec{K}_n) \left(\frac{r - r_c t}{t} \right)^{n+1-m} u[r - r_c t] \quad (\text{II.75})$$

Como $\sum_{c=1}^{|\mathcal{R}|}$ independe de \vec{K}_n , e no somatório de acima $m \leq \vec{K}_n[c]$, tem-se que:

$$\sum_{\vec{K}_n} \sum_{c=1}^{|\mathcal{R}|} \sum_{m=1}^{\vec{K}_n[c]} = \sum_{c=1}^{|\mathcal{R}|} \sum_{m=1}^{n+1} \sum_{\substack{\vec{K}_n \text{ tal que} \\ \vec{K}_n[c] \geq m \geq 1}}$$

Portanto, trocando a ordem dos somatorios em II.75:

$$F_i(t, r|n) = \sum_{c=1}^{|R|} \sum_{m=1}^{n+1} \sum_{\substack{\vec{K}_n \text{ tal que} \\ \vec{K}_n[c] \geq m \geq 1}} \binom{n}{m-1} \Gamma_i[\vec{K}_n] g(c, m, \vec{K}_n) \left(\frac{r - r_c t}{t} \right)^{n+1-m} u[r - r_c t] \quad (\text{II.76})$$

Porém para todo $c \neq \psi(e_i)$ a desigualdade $n + 1 > \vec{K}_n[c] \geq m \geq 1$, tem que ser satisfeita, pois o primeiro segmento tem recompensa $r_{\psi(e_i)}$. Isto implica que $g(c, n + 1, \vec{K}_n) = 0$, e $g(\psi(e_i), n + 1, \vec{K}_n) \geq 0$. Portanto, isolando a parte correspondente a $g(\psi(e_i), n + 1, \vec{K}_n)$, a equação II.76 pode ser reescrita da forma:

$$F_i(t, r|n) = \Gamma_i \left[n, (n + 1) \vec{1}_{\psi(e_i)} \right] g(\psi(e_i), n + 1, (n + 1) \vec{1}_{\psi(e_i)}) u[r - r_{\psi(e_i)} t] + \sum_{c=1}^{|R|} \sum_{m=1}^n \binom{n}{m-1} \sum_{\substack{\vec{K}_n \text{ tal que} \\ \vec{K}_n[c] \geq m \geq 1 \\ \vec{K}_n \neq (n+1) \vec{1}_{\psi(e_i)}}} \Gamma_i[\vec{K}_n] g(c, m, \vec{K}_n) \left(\frac{r - r_c t}{t} \right)^{n+1-m} u[r - r_c t] \quad (\text{II.77})$$

Note que $a \vec{1}_{\psi(e_i)}$ corresponde ao vetor de dimensão $|R|$ que tem todas as seus componentes iguais zero, exceto o $\psi(e_i)$ -ésimo que vale a .

As equações II.67, II.70 . . . II.72 e II.77 permitem calcular $F_i(t, r|n)$. Porém esta forma de cálculo tem um custo que é proporcional ao número de termos $\Gamma_i(\vec{K}_n)$, que corresponde a todas os modos de distribuir as $|R|$ recompensas nos $n + 1$ segmentos do intervalo $[0, t]$. Isto implica que o número de termos $\Gamma_i(\vec{K}_n)$ e, portanto, o custo de avaliar II.77, cresce combinatoriamente com $|R|$. Este custo combinatorial pode ser muito alto para algumas aplicações. Para evitar este alto custo ver-se-á a seguir como calcular $F_i(t, r|n)$ em forma mais eficiente. Para isto considere-se novamente a equação II.66:

$$F_i(t, r|n) = \sum_{\vec{K}_n} \Gamma_i[\vec{K}_n] F(\vec{K}_n) \quad (\text{II.78})$$

Onde $F(\vec{K}_n) = P[RTA_i \leq r | \vec{K}_n]$.

A equação II.78 permite estabelecer as condições que devem cumprir $\Gamma_i[\vec{K}_n]$ e $F(\vec{K}_n)$ para avaliar $F_i(t, r, n)$ mais eficientemente que usando II.77. Avaliar numericamente II.77 é caro devido a que existem *muitos* termos $\Gamma_i[\vec{K}_n]$ e $g(c, m, \vec{K}_n)$. Então, para calcular II.78 em forma menos dispendiosa, é necessario *agrupar* termos $\Gamma_i[\vec{K}_n]$ e/ou $F(\vec{K}_n)$, de forma tal que exista uma recorrência entre estes grupos, pois, ao existir menos (grupos de) termos para efetuar o cálculo, o uso de recursos computacionais torna-se menor.

Antes de analisar o caso específico da equação II.78, serão vistas algumas condições sob as quais II.78 seria mais fácil de calcular do que II.77. Considerem-se, especificamente, os dois seguintes casos.

Caso 1: Se $\Gamma_i[\vec{K}_n]$ e $F(\vec{K}_n)$ são combinações lineares de $\Gamma_i[\vec{K}_{n-1}]$ e $F(\vec{K}_{n-1})$, respectivamente, $F_i(t, r, n)$ é proporcional a $F_i(t, r, n-1)$ e a alguns termos recorrentes em \vec{K}_n . Portanto, esta forma de avaliação apresenta-se mais eficiente numericamente do que II.77.

Caso 2: $\Gamma_i[\vec{K}_n]$ é uma combinação linear de $\Gamma_i[\vec{K}_{n-1}]$, e $F(\vec{K}_n) = f(\vec{K}_{n-1}, v_1, \dots, v_m)$, onde f é alguma função dada. Neste caso, a agrupação conveniente é da forma $\sum_{\vec{K}_n} \Gamma_i[\vec{K}_n] f(\vec{K}_{n-1}, v_1, \dots, v_m)$, pois estes *grupos* podem ser calculados recorrentemente, em função do mesmo tipo de grupo.

As equações II.67, II.70 ... II.74 correspondem ao caso 2 recém analisado. Portanto, para avaliar $F_i(t, r, n)$ da forma descrita no caso 2, definem-se os termos $\alpha_i^{(n)}$ e $\beta_i^{(n)}(c, m)$ ($1 \leq c \leq |R|$, $1 \leq m \leq n$) da seguinte maneira:

$$\alpha_i^{(n)} = \Gamma_i \left[(n+1) \vec{1}_{\psi(e_i)} \right] g(\psi(e_i), n+1, (n+1) \vec{1}_{\psi(e_i)}) \quad (\text{II.79})$$

$$\beta_i^{(n)}(c, m) = \sum_{\substack{1 \leq c \leq |R|; \\ 1 \leq m \leq n}} \Gamma_i \left[\vec{K}_n \right] g(c, m, \vec{K}_n) \quad (\text{II.80})$$

$\vec{K}_n \neq (n+1) \vec{1}_{\psi(e_i)};$
 $1 \leq m \leq \vec{K}_n[c]$

É importante notar que o lado direito das equações II.79 e II.80 permitem interpretar o significado dos diferentes termos de α 's e β 's. Esta interpretação é original deste trabalho, pois não aparece em [9]. Especificamente II.80 estabelece que os termos β 's correspondem à probabilidade agregada de alguns caminhos (aqueles que cumprem a condição do somatório), multiplicanda por um dos termos da solução da FDP da CLEO de n v.a's uniformes em $[0, t]$. Analogamente II.79 estabelece que $\alpha_i^{(n)}$ corresponde à probabilidade agregada de todos os caminhos compostos somente por estados com a mesma recompensa, multiplicada por um dos termos da solução de FDP da CLEO de n v.a's uniformes em $[0, t]$.

Os diferentes termos β 's definidos em II.80 correspondem a um agrupamento de *muitos* termos da forma ' $\Gamma_i(\vec{K}_n)g(c, m, \vec{K}_n)$ ', e o termo $\alpha^{(n)}$ também tem a mesma estrutura. Por outro lado, as equações II.67, II.70 ... II.72 estabelecem que os termos $\Gamma_i(\vec{K}_n)$ e $g(c, m, \vec{K}_n)$ são recorrentes em n . Em conseqüência, os termos $\alpha^{(n)}$'s e $\beta^{(n)}$'s devem, necessariamente, ser recorrentes em n ; mais ainda, tal como se demonstrará mais adiante, os termos $\alpha^{(n)}$ e $\beta^{(n)}$'s podem ser calculados em função somente dos termos $\alpha^{(n-1)}$ e $\beta^{(n-1)}$'s. Os fatos abordados anteriormente são a chave para calcular eficientemente $F_i(t, r|n)$, pois tal como também será visto

mais adiante, o número de termos $\alpha^{(n)}$'s e $\beta^{(n)}$'s é polinomial em n , e portanto menor do que o número de termos de II.77.

Voltando ao cálculo de $F_i(t, r|n)$; II.79 e II.80 permitem re-escrever II.77 da forma:

$$F_i(t, r|n) = \alpha_i^{(n)} u[r - r_{\psi(e_i)} t] + \sum_{c=1}^{|R|} \sum_{m=1}^n \binom{n}{m-1} \beta_i^{(n)}(c, m) \left(\frac{r - r_c t}{t} \right)^{n+1-m} u[r - r_c t] \quad (\text{II.81})$$

que é idêntica à expressão principal de [9] para o cálculo da desempenhabilidade.

A seguir serão usadas as equações II.67, II.70 . . . II.72 para deduzir como calcular recorrentemente em n os termos $\alpha_i^{(n)}$ e $\beta_i^{(n)}(c, m)$. Analisando todos os casos possíveis, tem-se:

Caso $\alpha_i^{(0)}$: De II.79:

$$\begin{aligned} \alpha_i^{(0)} &= \Gamma_i [\vec{1}_{\psi(e_i)}] g(\psi(e_i), 1, \vec{1}_{\psi(e_i)}) \\ &= 1 \end{aligned} \quad (\text{II.82})$$

Caso $\beta_i^{(0)}(c, m)$: De II.80:

$$\begin{aligned} \beta_i^{(0)}(c, 1) &= \Gamma_i [\vec{K}_0] g(c, 1, \vec{K}_0) \\ \vec{K}_0 \neq \vec{1}_{\psi(e_i)}; \vec{K}_0[c]=1 & \\ &= 0 \end{aligned} \quad (\text{II.83})$$

Caso $\alpha_i^{(n)}$, $n \geq 1$: De II.67, II.71 e II.79, temos:

$$\alpha_i^{(n)} = \sum_{e_j \in E} p_{ij} \Gamma_j [n \vec{1}_{\psi(e_i)}] g(\psi(e_i), n, n \vec{1}_{\psi(e_i)}); \quad n \geq 1 \quad (\text{II.84})$$

Em II.84 não existe restrição enquanto ao valor de e_j . Em particular pode acontecer que $\rho(e_j) = r_{\psi(e_i)}$ ou $\rho(e_j) \neq r_{\psi(e_i)}$. No caso em que $\rho(e_j) \neq r_{\psi(e_i)}$, da equação II.67 temos que $\Gamma_j [n \vec{1}_{\psi(e_i)}] = 0$. Isto é assim pois se $\vec{K}_n = n \vec{1}_{\psi(e_i)}$, todos os estados dos caminhos agregados em $\Gamma_j [n \vec{1}_{\psi(e_i)}]$ deveriam ter recompensa $r_{\psi(e_i)}$, porém está sendo suposto que o primeiro estado (e_j) tem uma recompensa diferente. Este fato, em conjunto com a equação II.79, permitem reescrever II.84 da forma:

$$\alpha_i^{(n)} = \sum_{\substack{e_j \in E \\ \rho(e_j) = \rho(e_i)}} p_{ij} \alpha_j^{(n-1)}; \quad n \geq 1 \quad (\text{II.85})$$

Caso $\beta_i^{(n)}(\psi(e_i), m)$, ($2 \leq m \leq n$): De II.80:

$$\beta_i^{(n)}(\psi(e_i), m) = \sum_{2 \leq m \leq n} \sum_{\substack{\vec{K}_n \\ \vec{K}_n \neq (n+1)\vec{1}_{\psi(e_i)} \\ m \leq k_{\psi(e_i)}}} \Gamma_j [\vec{K}_n] g(\psi(e_i), m, \vec{K}_n); \quad n \geq 1 \quad (\text{II.86})$$

Usando II.67 e II.71 em II.86:

$$\beta_i^{(n)}(\psi(e_i), m) = \sum_{2 \leq m \leq n} \sum_{\substack{\vec{K}_n \\ \vec{K}_n \neq (n+1)\vec{1}_{\psi(e_i)} \\ m \leq k_{\psi(e_i)}}} \sum_{e_j \in E} p_{ij} \Gamma_j [\vec{K}_n - \vec{1}_{\psi(e_i)}] g(\psi(e_i), m-1, \vec{K}_n - \vec{1}_{\psi(e_i)}) \quad (\text{II.87})$$

Agora lembrando que: $\vec{K}_n = \vec{K}_{n-1} + \vec{1}_{\psi(e_i)}$, $|\vec{K}_{n-1}| = n$ e $K_{n-1}[\psi(e_i)] \geq m-1$, II.87 pode ser reescrita da forma:

$$\beta_i^{(n)}(\psi(e_i), m) = \sum_{2 \leq m \leq n} p_{ij} \sum_{\substack{\vec{K}_{n-1} \\ \vec{K}_{n-1} \neq n\vec{1}_{\psi(e_i)} \\ m-1 \leq K_{n-1}[\psi(e_i)]}} \Gamma_j [\vec{K}_{n-1}] g(\psi(e_i), m-1, \vec{K}_{n-1}) \quad (\text{II.88})$$

O segundo somatório de acima corresponde ao lado direito de II.80, portanto:

$$\beta_i^{(n)}(\psi(e_i), m) = \sum_{2 \leq m \leq n} \sum_{e_j \in E} p_{ij} \beta_j^{(n-1)}(\psi(e_i), m-1) \quad (\text{II.89})$$

Caso $\beta_i^{(n)}(c, n)$, $c \neq \psi(e_i)$: As equações II.67, II.73 e II.80 estabelecem que:

$$\begin{aligned} \beta_i^{(n)}(c, n) &= \sum_{\substack{\vec{K}_n \\ c \neq \psi(e_i)}} \sum_{\substack{e_j \in E \\ \vec{K}_n \neq (n+1)\vec{1}_{\psi(e_i)} \\ \vec{K}_n |c| \geq n}} p_{ij} \Gamma_j [\vec{K}_{n-1}] \left\{ -\frac{g^{(n-1)}(c, n, \vec{K}_{n-1})}{(r_c - r_{\psi(e_i)})} \right\} \\ &= - \sum_{e_j \in E} \frac{p_{ij}}{(r_c - r_{\psi(e_i)})} \sum_{\substack{\vec{K}_{n-1} \\ \vec{K}_{n-1} \neq n\vec{1}_{\psi(e_i)} \\ \vec{K}_{n-1} |c| = n}} \Gamma_j [\vec{K}_{n-1}] g(c, n, \vec{K}_{n-1}) \end{aligned} \quad (\text{II.90})$$

Como $\vec{K}_{n-1}[c] = n$, e o número de transições da CM é $(n - 1)$, evidentemente, cumpre-se que $\vec{K}_{n-1} = n\vec{1}_c$. Por outro lado, como $\vec{K}_{n-1} = n\vec{1}_c$ e o estado inicial dos caminhos correspondentes a $\Gamma_j[\vec{K}_{n-1}]$ é o estado e_j , deve cumprir-se que $\rho(e_j) = r_c$. Estas observações implicam que a equação II.90 pode ser reescrita da forma:

$$\beta_i^{(n)}(c, n) = - \sum_{\substack{c \neq \psi(e_i) \\ e_j \in E \\ \rho(e_j) = r_c}} \frac{p_{ij}}{(r_c - r_{\psi(e_i)})} \Gamma_j [n\vec{1}_c] g(c, n, n\vec{1}_c) \quad (\text{II.91})$$

Usando II.79, tem-se que:

$$\beta_i^{(n)}(c, n) = - \sum_{\substack{c \neq \psi(e_i) \\ e_j \in E \\ \rho(e_j) = r_c}} \frac{p_{ij} \alpha_j^{(n-1)}}{(r_c - r_{\psi(e_i)})} \quad (\text{II.92})$$

Caso $\beta_i^{(n)}(c, m)$, $c \neq \psi(e_i)$, $1 \leq m < n$: Usando II.74 em II.80:

$$\beta_i^{(n)}(c, m) = \sum_{\substack{c \neq \psi(e_i) \\ 1 \leq m < n \\ \vec{K}_n \neq (n+1)\vec{1}_{\psi(e_i)} \\ m \leq \vec{K}_n[c]}} \Gamma_i [\vec{K}_n] \left\{ \frac{g(c, m+1, \vec{K}_n) - g(c, m, \vec{K}_{n-1})}{(r_c - r_{\psi(e_i)})} \right\} \quad (\text{II.93})$$

Manipulando a equação de acima e usando II.67, tem-se:

$$\beta_i^{(n)}(c, m) = \frac{1}{(r_c - r_{\psi(e_i)})} \sum_{\substack{\vec{K}_n \\ \vec{K}_n \neq (n+1)\vec{1}_{\psi(e_i)} \\ m \leq \vec{K}_n[c]}} \Gamma_i [\vec{K}_n] g(c, m+1, \vec{K}_n) - \sum_{\substack{e_j \in E \\ \vec{K}_{n-1} \\ \vec{K}_{n-1} \neq n\vec{1}_{\psi(e_i)} \\ m \leq \vec{K}_{n-1}[c]}} \frac{p_{ij} \Gamma_j [\vec{K}_{n-1}] g(c, m, \vec{K}_{n-1})}{(r_c - r_{\psi(e_i)})} \quad (\text{II.94})$$

Porém a equação II.29 implica que $g(c, m+1, \vec{K}_n)$ vale 0 para todo \vec{K}_n no qual $\vec{K}_n[c] = m$. Isto implica que o primeiro somatório de II.94 varia para $\vec{K}_n[c] \geq m+1$, o que em conjunto com II.80 implicam que:

$$\beta_i^{(n)}(c, m) = \frac{\beta_i^{(n)}(c, m+1)}{(r_c - r_{\psi(e_i)})} - \sum_{e_j \in E} \frac{p_{ij} \beta_i^{(n-1)}(c, m)}{(r_c - r_{\psi(e_i)})} \quad (II.95)$$

$c \neq \psi(e_i)$
 $1 \leq m < n$

Caso $\beta_i^{(n)}(\psi(e_i), 1)$: Usando II.72 em II.80:

$$\begin{aligned} \beta_i^{(n)}(\psi(e_i), 1) &= \sum_{\substack{\vec{K}_n \\ \vec{K}_n \neq (n+1)\vec{1}_{\psi(e_i)} \\ \vec{K}_n|c \geq 1}} \Gamma_i[\vec{K}_n] \left\{ - \sum_{\substack{c=1 \\ c \neq \psi(e_i); \vec{K}_n|c > 0}}^{|R|} g(c, 1, \vec{K}_n) \right\} \\ &= - \sum_{\substack{c=1 \\ c \neq \psi(e_i)}}^{|R|} \sum_{\substack{\vec{K}_n \\ \vec{K}_n \neq (n+1)\vec{1}_{\psi(e_i)} \\ \vec{K}_n|c \geq 1}} \Gamma_i[\vec{K}_n] g(c, 1, \vec{K}_n) \end{aligned} \quad (II.96)$$

Identificando o segundo somatório em II.80, tem-se que:

$$\beta_i^{(n)}(\psi(e_i), 1) = - \sum_{\substack{c=1 \\ c \neq \psi(e_i); \vec{K}_n|c > 0}}^{|R|} \beta_i^{(n)}(c, 1) \quad (II.97)$$

II.82, II.83, II.85, II.89, II.92, II.95 e II.97 permitem calcular em forma recorrente todos os termos $\alpha_i^{(n)}$ e $\beta_i^{(n)}(c, h)$. Estas equações, em conjunto com II.81 correspondem exatamente ao resultado de [9].

II.5.3 Discussão do método de solução de $F_i^r(t, r)$.

Para discutir em termos mais específicos os desenvolvimentos feitos em [9] e neste trabalho, a seguir reproduzem-se em forma abstrata ambos métodos. Tal como foi indicado anteriormente, o método empregado em [9] para deduzir II.81 e as expressões recorrentes dos termos α 's e β 's, foi usado nesta tese para deduzir $F(\vec{a}_n, t, r)$ (seção II.3), que de aqui em frente será denominado Método A e que basicamente é o seguinte:

Método A

1. Randomiza-se a cadeia de Markov que modela o sistema.

2. Condicionando-se no valor de $N(t)$ e de $U_{(1)}$, chega-se a uma expressão recorrente para a medida sob análise.
3. Para simplificar a expressão do passo 2, aplica-se a transformada de Laplace, conseguindo-se desta forma outra expressão recorrente, mais simples que a anterior, porém no plano de Laplace.
4. Estabelece-se uma solução para a expressão recorrente obtida no passo 3. Para a realização deste passo não existe uma sistemática, senão que o êxito dele depende da experiência ou intuição do pesquisador.
5. Demonstra-se, usando indução, a solução estabelecida no passo 4.
6. Deduzem-se expressões recorrentes para os possíveis termos que aparecem na solução estabelecida no ponto 4, que ainda não foram calculados.
7. Leva-se a solução do ponto 4 para o plano do tempo (mundo real), com o qual completa-se a solução do problema sob análise.

Por outro lado, o método empregado, nesta seção (II.5), para avaliar $F_i(t, r)$ que de aqui em frente denominaremos método B, é o seguinte:

Método B

1. Randomiza-se a cadeia de Markov que modela o sistema.
2. Condicionando-se no valor de $N(t)$ e no número de segmentos do intervalo $[0, t]$ que recebem a mesma recompensa, chega-se a uma solução do problema sob estudo. Lamentavelmente, esta solução, normalmente é muito cara de ser avaliada computacionalmente.
3. Para diminuir o custo da solução obtida no passo anterior, agrupam-se termos dela de forma tal que eles possam ser avaliados recorrentemente (em função do mesmo tipo de grupo). Este é o passo chave do método B e é o que permite chegar a soluções eficientes de serem avaliadas computacionalmente.
4. Obtem-se expressões recorrentes para os grupos definidos no passo anterior, com o qual conclue-se a solução do problema sob análise.

Ao comparar os métodos A e B, vê-se que este último têm as seguintes vantagens:

1. É mais simples, principalmente porque não precisa usar transformada de Laplace.
2. É mais sistemático, pois o método A requer 'intuir' a solução do problema no plano de Laplace e não existe nenhuma sistemática para conseguir isto. Este tipo de situação não acontece no método B.

3. O método B permite entender o significado dos termos que aparecem na expressão final da medida calculada (termos α 's e β 's de $F_i(t, r)$), o que não acontece quando é usado o método A.

Para finalizar este capítulo é interessante destacar que o método A como o método B permitem, também, resolver outros problemas da área de desempenhabilidade. De fato o método A foi usado em [9] para avaliar $F_i(t, r)$ e neste capítulo para avaliar $F(\vec{a}_n, t, r)$, mais ainda, mais na frente nesta mesma tese ambos métodos serão usados para resolver um problema mais complexo que os enfrentados até agora.

Capítulo III

Avaliação da Desempenhabilidade para Recompensas Discretas

III.1 Introdução

Neste capítulo estuda-se outro tipo de medidas de desempenhabilidade. Especificamente o interesse é calcular a Função Distribuição de Probabilidade do Rendimento Total Acumulado durante um período de tempo finito, por um sistema que ganha recompensa fixa cada vez que transita entre um mesmo par de estados.

A utilidade deste tipo de medida pode ser ilustrada pelo seguinte exemplo: suponha que deseja-se medir o número de um certo tipo de falhas que acontecem no sistema no período $[0, t]$. Para medir o número destas falhas, na cadeia de Markov que modela o sistema atribui-se recompensa '1' às transições que correspondem ao tipo de falha de interesse, e recompensa zero às demais transições. Em conseqüência: o número de transições marcadas (com recompensa igual a '1') que acontecem no intervalo $[0, t]$, corresponde à recompensa total ganha pelo sistema neste intervalo.

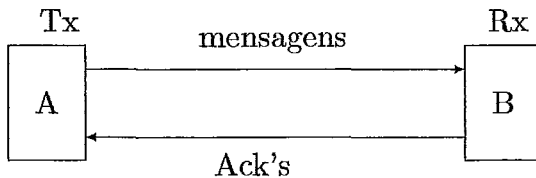


Figura III.1: Protocolo Parada e Espera

Outro exemplo: deseja-se medir a vazão (*throughput*) do protocolo *Parada e Espera* (*Stop-and-Wait*). Este protocolo, cujos componentes aparecem na figura III.1, é talvez o mais simples que permite transferir informação entre

transmissor (Tx) e um receptor (Rx). A idéia básica dele, é ter certeza de que cada pacote tenha sido corretamente recebido antes de iniciar a transmissão do seguinte pacote. Quando o Rx recebe um pacote sem erros, envia ao Tx um pedido requerindo o próximo pacote. Caso o Tx não receba este pedido, após um determinado tempo de espera (*time-out*), envia novamente o pacote anterior. O esquema de operação entre o Rx e o Tx, recém descrito, implica que é necessário distinguir os diferentes pacotes transmitidos, isto é feito atribuindo uma identidade diferente a cada um deles. A forma mais simples de se fazer esta atribuição de identidades, de forma que não sejam confundidos uns com os outros, é numerá-los ciclicamente com 0 e 1, é dizer, o primeiro pacote transmitido recebe identidade 0, o segundo identidade 1, o terceiro identidade 0, etc. É importante notar que, todas as retransmissões de um mesmo pacote, já seja devido a erros no transmissor ou a tempo de espera, levam a mesma identidade de pacote (isto é evidente, pois está sendo retransmitido o mesmo pacote). Como consequência da operação do protocolo descrito anteriormente, o Tx possui 2 estados, que serão chamados de 0 e 1. Estes estados correspondem ao número (módulo 2) do pacote que está sendo enviado. Em outras palavras, se o Tx inicia sua operação no estado 0, ele permanece nesse estado até que o receptor lhe informe que o pacote que está sendo transmitido foi corretamente recebido; nesse momento o Tx muda seu estado ao estado 1. Analogamente ao caso anterior, o Tx permanece no estado 1 até que o Rx lhe informe que o pacote que está sendo transmitido foi corretamente recebido, nesse momento o Tx muda seu estado para 0. As mudanças de estado do Tx continuam ciclicamente da forma recém descrita, até acabar a sessão. Por outra parte, o Rx também têm 2 estados possíveis, chamados 0 e 1. Estes estados correspondem ao número (mod 2) do pacote que está sendo esperado pelo Rx. Analogamente à operação do Tx, o Rx somente muda de estado quando recebe, livre de erros, o pacote que ele está esperando. Como consequência do descrito anteriormente, o estado conjunto do Tx e Rx é um par ordenado (A, B) , com A e $B \in \{0, 1\}$. Considere-se que inicialmente o estado conjunto é $(0, 0)$, o que significa que o Tx está enviando um pacote com identidade 0, o qual está sendo esperado pelo Rx. Quando o Rx recebe corretamente o pacote, muda ao estado 1, o que faz com que o estado conjunto mude ao estado $(0, 1)$; neste estado o Rx envia ao Tx o pedido de transmissão do seguinte pacote. O estado conjunto não muda até que o Rx receba corretamente este pedido, nesse momento o Tx muda ao estado 1 o que faz que o estado conjunto mude ao estado $(1, 1)$. Continuando a análise, é fácil ver que as sucessivas mudanças do estado conjunto, do sistema Tx / Rx, acontecem da forma desenhada na figura III.2; em outras palavras, o estado conjunto muda ciclicamente seguindo a seqüência $(0, 0)$, $(0, 1)$, $(1, 1)$, $(1, 0)$, $(0, 0)$, etc. Adicionalmente, agregando-se a hipótese de que os tempos de permanência no Tx e no Rx são exponencialmente distribuídos com parâmetros λ_1 e λ_2 respectivamente, o sistema pode ser modelado pela cadeia de Markov da figura III.2

Sea $T_h(t)$ a vazão (*throughput*) no intervalo $[0, t]$ do protocolo *Parada e Espera*. A avaliação de T_h pode ser mapeada ao problema de desempenho estudado neste capítulo, fazendo uso da seguinte equação:

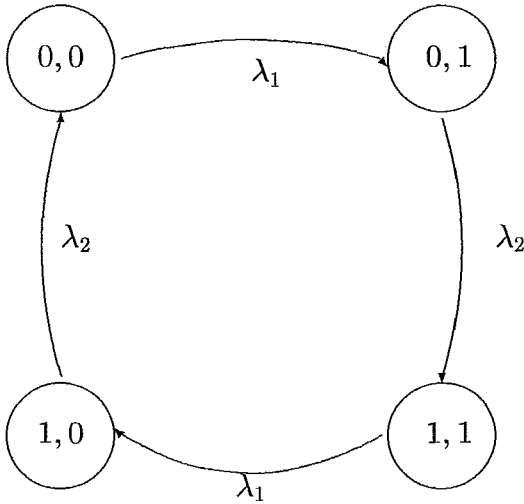


Figura III.2: Cadeia de Markov que modela o Protocolo *Parada e Espera*, sem erros

$$P[T_h(t) \leq r] = P[\# \text{ de pacotes transmitido em } [0, t] \leq r] \quad (\text{III.1})$$

Portanto, avaliar a FDP da vazão do protocolo é equivalente a avaliar a FDP do número de pacotes transmitidos em $[0, t]$. Se atribuímos recompensa 1 às transições que correspondem à transmissão de um pacote (as que acontecem com taxa λ_1 na figura III.2), e recompensa 0 às demais transições, temos que a recompensa total acumulada pelo sistema no intervalo $[0, t]$, corresponde ao número de pacotes transmitidos neste intervalo.

Retoma-se, agora, o objetivo principal deste capítulo que consiste em calcular $G_i(t, r)$, isto é: calcular a função de distribuição de probabilidade do rendimento total acumulado em $[0, t]$ por uma cadeia de Markov, cujas transições tem associadas recompensas fixas.

Em [5] foi encontrada uma expressão para $G_i(t, r)$ cuja avaliação numérica requer de $O\left(|E| \binom{N_{max} + |\Theta|}{\Theta}\right)$ localizações de memória e de $O\left(|E|d \binom{N_{max} + |\Theta|}{\Theta}\right)$ operações de multiplicação ou divisão. Por outra parte, as equações III.8 ... III.10 deste capítulo, permitem calcular $G_i(t, r)$ usando $O(|E| \min\{N_{max}|\Theta|, r\})$ localizações de memória e $O(|E|d \min\{N_{max}|\Theta|, r\})$ operações de multiplicação e divisão.

Ao comparar-se os resultados de [5] e deste trabalho, ve-se que,

no caso particular em que $|\Theta| = 2$, ambos métodos têm a mesma complexidade, porém ao aumentar o valor de $|\Theta|$ é notavelmente mais eficiente avaliar $G_i(t, r)$, usando as equações III.8 . . . III.10, o que constitui a contribuição mais importante deste capítulo à área de desempenhabilidade.

III.2 O Modelo

Consideram-se sistemas que podem ser modelados por Processos de Markov Homogeneos [3]. Cada transição entre um par de estados do sistema tem associada uma recompensa fixa, isto é, cada vez que o sistema efetua uma certa transição, recebe uma determinada recompensa. Em geral, diferentes transições podem ter associadas recompensas diferentes. É importante notar que as recompensas ganhas independem do tempo que o sistema fica em cada um dos estados.

É interessante notar que, tal como será visto no capítulo IV, em alguns casos é útil definir uma função de atribuição de recompensas, na qual o sistema ganha uma recompensa fixa cada vez que permanece num certo estado, de forma que estados diferentes podem ter associadas recompensas diferentes. Porém, um breve raciocínio mostra que esta forma de atribuir recompensas é um caso particular do tipo anterior e portanto estudar-se-á somente o tipo descrito no parágrafo acima.

A seguir descreve-se parte da notação que usaremos. Considere um processo contínuo e homogêneo de Markov $\chi_1 = \{X_1(t), t \geq 0\}$ que descreve o comportamento do sistema. Seja $E = \{e_j, j = 1, \dots, |E|\}$ o espaço finito de estados associado ao modelo, onde $|E|$ corresponde ao número total de estados do sistema. Seja $T \subset E \times E$, o conjunto de transições entre estados da cadeia de Markov randomizada. Denota-se t_{ij} à transição do estado $e_i \in E$ ao estado $e_j \in E$. Cada transição t_{ij} tem associada uma recompensa $\rho(t_{ij})$, que corresponde a recompensa ganha pelo sistema cada vez que ele transita do estado e_i ao estado e_j . Seja $\Theta = \{\theta_k, k = 1, \dots, \theta_{|\Theta|}\}$ o conjunto de recompensas distintas associadas às transições de T . Em geral tem-se que $|\Theta| \leq |E|^2$, pois diferentes transições podem (ou não) ter associada uma mesma recompensa, e algumas transições entre estados podem ser impossíveis de acontecer. Seja $RTAD_i(t)$ a recompensa total ganha pelo sistema no intervalo $[0, t]$, no caso em que o estado inicial é e_i , e o ganho de recompensas acontece nas transições segundo a função ρ antes descrita. Devido a que no intervalo de observação, a cadeia de Markov pode transitar por caminhos diversos, evidentemente $RTAD_i(t)$ é uma variável aleatória, porém é usual (embora não necessário) supor que $RTAD_i(0) = 0$.

O objetivo deste capítulo é calcular $G_i(t, r)$, que corresponde à função de distribuição de probabilidade de $RTAD_i(t)$, isto é:

$$G_i(t, r) = P[RTAD_i(t) \leq r] \quad (\text{III.2})$$

Onde r é um valor real.

III.3 Avaliação da Desempenhabilidade $G_i(t, r)$

Tal como nos casos anteriores, para calcular $G_i(t, r)$ usaremos a técnica de randomização. Portanto, condicionando no valor de $N(t)$ e aplicando o teorema de probabilidades totais, tem-se que:

$$G_i(t, r) = \sum_{n=0}^{\infty} G_i(t, r|N(t) = n) P[N(t) = n] \quad (\text{III.3})$$

Onde $P[N(t) = n]$ corresponde à probabilidade que no processo de Poisson tenham acontecido n eventos no intervalo $[0, t]$, e portanto esta dada pela equação [28]:

$$P[N(t) = n] = e^{-\Lambda t} \frac{(\Lambda t)^n}{n!} \quad (\text{III.4})$$

Por outra parte, $G_i(t, r|N(t) = n)$, corresponde a $G_i(t, r)$ no caso em que $N(t) = n$, o que em conjunto com III.2 implicam que:

$$G_i(t, r|N(t) = n) = P[RTAD_i(t) \leq r|N(t) = n] \quad (\text{III.5})$$

Agora, usando o fato que, por definição $RTAD_i(t)$ corresponde à soma das recompensas ganhas nas transições que efetua a cadeia de Markov randomizada tem-se que:

$$G_i(t, r|N(t) = n) = P\left[\sum_{l=1}^n \rho(t(l)) \leq r | \chi_2(0) = e_i\right] \quad (\text{III.6})$$

Onde $t(l)$ corresponde à l -ésima transição da cadeia de Markov randomizada, isto é $t(l) = t_{jk}$ no caso em que $\chi_2(l-1) = e_j$, e $\chi_2(l) = e_k$.

O lado direito de III.6 indica que $G_i(t, r|N(t) = n)$ não depende de t , e portanto no que segue esta variável será escrita simplesmente da forma $G_i(r|n)$.

É interessante destacar que as equações III.2, III.3, e o fato que $G_i(r|n)$ independe de t , implicam que: **$G_i(t, r)$ depende de t somente através do processo de Poisson $N(t)$ e depende do caminho percorrido pela cadeia de Markov randomizada somente através de $G_i(r|n)$.**

Seja \mathcal{R}_i^n o conjunto de todas as recompensas diferentes que pode acumular o sistema no caso em que o estado inicial é e_i , e $N(t) = n$. Em outras palavras: \mathcal{R}_i^n corresponde aos diferentes valores que pode tomar o somatório

$\sum_{l=1}^n [\rho(t(l)) | \chi_2(0) = e_i]$. Seja w um dos valores pertencentes a \mathcal{R}_i^n . Evidentemente o valor de w depende do caminho percorrido pela cadeia de Markov randomizada, porém, note-se que um mesmo valor de w pode corresponder a diferentes caminhos. Seja r_i^n o subconjunto de \mathcal{R}_i^n composto por todas as recompensas $w \in \mathcal{R}_i^n$ tal que $w \leq r$.

A modo de exemplo, a figura III.3 mostra os diferentes valores de \mathcal{R}_i^n para o caso particular em que $n = 0, 1, 2, 3, 4$, e $\Theta = \{0, 2, 5\}$; estes valores aparecem marcados com um ponto. A figura foi desenhada supondo que em cada transição o sistema pode ganhar qualquer uma das recompensas pertencentes a Θ , o que para muitos sistemas pode não ser verdade. Portanto, para o conjunto Θ dado, os pontos desenhados na figura III.3, correspondem a um limite superior para o conjunto R_i^n ,

Outro aspecto interessante da figura: no caso de $r = 8$, tem-se que $r_i^3 = \{0, 2, 4, 5, 6, 7\}$, e $\mathcal{R}_i^3 = r_i^3 \cup \{9, 10, 12, 15\}$. Isto ilustra o fato que $r_i^n \subseteq \mathcal{R}_i^n$.

Voltando ao cálculo de $G_i(r|n)$, define-se $RTAD_{i,n}$ como a recompensa total ganha pelo sistema no caso em que este inicie sua operação no estado e_i , e $N(t) = n$. É claro então que:

$$G_i(r|n) = \sum_{w \in r_i^n} P[RTAD_{i,n} = w] \quad (\text{III.7})$$

A probabilidade $P[RTAD_{i,n} = w]$ pode facilmente ser calculada em forma recorrente no valor de n . Para $n = 0$ evidentemente $\mathcal{R}_i^n = \{0\}$, portanto:

$$P[RTAD_{i,0} = w] = I[w = 0] \quad (\text{III.8})$$

Para avaliar $P[RTAD_{i,n} = w]$ no caso $n \geq 1$, condiciona-se na primeira transição da cadeia de Markov randomizada, de onde:

$$P[RTAD_{i,n} = w] = \sum_{j \in [1..E]} p_{ij} P[RTAD_{j,n-1} = w - \rho(t_{ij})] \quad (\text{III.9})$$

Onde p_{ij} corresponde à probabilidade que a CM randomizada transite, num passo, do estado e_i ao estado e_j .

Por último, usando III.4 e III.7, em III.3:

$$G_i(t, r) = \sum_n e^{-\Lambda t} \frac{(\Lambda t)^n}{n!} \sum_{w \in r_i^n} P[RTAD_{i,n} = w] \quad (\text{III.10})$$

Onde $P[RTAD_{i,n} = w]$ é calculada recorrentemente usando as equações III.8 e III.9.

III.4 Complexidade Computacional

Custo de Memória.

A quantidade de memória usada para avaliar $G_i(t, r)$, segundo a equação III.10, é a necessária para armazenar os valores de $P[RTAD_{i,n} = w]$ para todo $w \in r_i^n$ e todo $e_i \in E$. Isto é assim devido a que para avaliar $P[RTAD_{i,n} = w]$, segundo as equações III.8 e III.9, é preciso armazenar somente os valores do passo anterior. Portanto, no n -ésimo passo do algoritmo é necessário armazenar $|E| \cdot |r_i^n|$ valores diferentes. Onde $|r_i^n|$ corresponde à cardinalidade do conjunto r_i^n . Por outro lado, já que evidentemente, $|r_i^{n+1}| \geq |r_i^n|$, a máxima quantidade de memória requerida pelo algoritmo é $O(|E| \cdot |r_i^{N_{max}}|)$.

O valor exato de $|r_i^n|$ depende do conjunto Θ , do valor de r , da cadeia de Markov, e do estado e_i . Porém, para alguns casos, avaliar $|r_i^n|$ é bastante simples. Por exemplo se $\Theta = \{0, 1, 2, \dots, |\Theta|\}$ e em cada transição a cadeia de Markov pode ganhar qualquer uma das recompensas $\in \Theta$, é fácil ver que $r_i^n = \min\{n \cdot |\Theta| + 1, r + 1\}$. Para o caso geral, no qual os elementos de Θ pertencem aos reais, o cálculo exato de $|r_i^n|$ pode implicar algumas operações tediosas e não muito esclarecedoras. Uma análise mais simples consiste em encontrar uma função que seja um limite superior de $|r_i^n|$, o que será feito a seguir.

Cálculo de um limite superior para $|r_i^n|$.

O problema de calcular um limite superior para $|r_i^n|$ pode ser mapeado em outro problema, da seguinte forma. A cada recompensa $\theta \in \Theta$ associa-se uma nota de dinheiro de valor θ . O fato do sistema ganhar a recompensa θ devido a uma transição é associado à extração da nota de valor θ . Então, dado que o conjunto de notas distintas existentes é $\Theta = \{\theta_k, k = 1, 2, \dots, |\Theta|\}$, o problema de calcular um limite superior para $|r_i^n|$, é equivalente a encontrar o número de quantias diferentes de dinheiro menores ou iguais a r que pode-se acumular com n notas. A vantagem deste mapeamento é que ele transforma o problema de calcular $|r_i^n|$, em um problema combinatorial. A seguir procede-se a resolver este problema.

Seja $f(n)$ a resposta ao problema de acima. Para situá-lo num contexto geral, supõe-se que os elementos $\theta \in \Theta$ são valores racionais. Então sem perda de generalidade pode-se assumir que:

- $\min_k \{\theta_k, 1 \leq k \leq |\Theta|\} = 0$. Isto se consegue subtraindo o valor da menor recompensa do conjunto Θ original a todos os seus elementos.
- Os elementos de Θ são inteiros não negativos. Isto se deve a que podemos multiplicar os elementos do conjunto obtido no passo anterior pela menor potencia de 10, de maneira a eliminar os algarismos decimais.
- O mínimo denominador comum (*mdc*) dos elementos de Θ é 1, isto é:

$mdc(\{\theta_1, \theta_2, \dots, \theta_{|\Theta|}\}) = 1$, pois podemos dividir os elementos do conjunto obtido no passo anterior pelo mdc deles.

Note-se que as três operações indicadas acima modificam o conjunto Θ original, deixando outro conjunto que será denotado como Θ' . Em outras palavras: como consequência das modificações indicadas acima, o conjunto Θ pode ser mapeado no conjunto $\Theta' = \{0, 1, \theta'_3, \dots, \theta'_{|\Theta|}\}$, onde, por simplicidade de notação, supõe-se que $\theta'_{k-1} < \theta'_k$, $1 \leq k \leq |\Theta|$. O fato de modificar o conjunto Θ implica que também é necessário modificar o valor de r (deixando um certo valor de r') de forma tal que $|r_i^n|$ (obtido a partir de Θ) seja igual a $|r'_i^n|$ (obtido a partir de Θ'). Para que se cumpra esta igualdade, os três passos que alteram o conjunto Θ (gerando o conjunto Θ'), implicam respectivamente os seguintes três passos para modificar o valor de r (gerando o valor r'):

- de r subtrai-se o valor: $n \min_k (\theta_k, 1 \leq k \leq |\Theta|)$.
- Multiplica-se o valor obtido no passo anterior pela mesma potência de 10 usada para multiplicar os elementos de Θ .
- Divide-se o valor obtido no passo anterior pelo mesmo mdc com o qual foram divididos os elementos de Θ .

É importante notar que $\theta'_{|\Theta|}$ depende da precisão, isto é do número de algarismos significativos das recompensas atribuídas às transições. Especificamente, no caso de S algarismos significativos, cumpre-se que $\theta'_{|\Theta|} \leq 10^S$. Isto, tal como será visto mais adiante, implica que existe um compromisso entre a precisão das recompensas atribuídas ao sistema e a complexidade computacional.

A análise que segue será feita para o conjunto Θ' . Neste caso é fácil demonstrar que $f(n)$ está limitada superiormente pela função $g(n) = \min((\theta'_{|\Theta|} + 1)n, r' + 1)$, onde $r' \leq 10^S$, e S é a quantidade de algarismos decimais dos elementos de Θ . Para entender o porquê deste limite superior consideremos o seguinte problema: encontrar todas as quantias de dinheiro diferentes menores ou iguais a r' que podem ser obtidas fazendo n extrações com reposição do conjunto $\mathcal{Q} = \{0, 1, 2, \dots, \theta'_{|\Theta|}\}$. É óbvio que a solução a este problema é $g(n)$, e como $\Theta' \subseteq \mathcal{Q}$, conclui-se que $f(n) \leq g(n)$, o que implica que $|r_i^n|$ é $O(g(n))$. Com este novo antecedente é possível retomar o cálculo da complexidade computacional necessária para avaliar numericamente as expressões III.8 ... III.10. Anteriormente demonstrou-se que o custo assintótico no uso de memória é $O(|E||r_i^{N_{max}}|)$, o que em conjunto com o fato que $|r_i^n|$ é o $O(\min\{\theta'_{|\Theta|}n, r'\})$, implicam que o custo de memória é $O(|E|\min\{\theta'_{|\Theta|}N_{max}, r'\})$.

Número de Operações.

A complexidade do algoritmo quanto a número de operações está dada pelo cálculo de todos os termos $P[RTAD_{i,n}]$. Da equação III.9 tem-se que o

número de multiplicações (que é a operação mais lenta a ser realizada) é $O(|E|)$ para todo estado $e_i \in E$ e toda recompensa $w \in r_i^n$. Portanto, considerando que n varia de 0 até N_{max} , conclue-se que o número de multiplicações e divisões do algoritmo é $O(|E|^2 N_{max} \min(\theta'_{|\Theta|} N_{max}, r'))$.

Outro fato a notar é que em geral as matrizes de transição de estados de modelos de sistemas reais são esparsas. Isto quer dizer que se d é o número medio de transições de um estado da cadeia de Markov, então, em geral $d \ll |E|$. Neste caso o número de operações do algoritmo é $O(|E|dN_{max} \min(\theta'_{|\Theta|} N_{max}, r'))$.

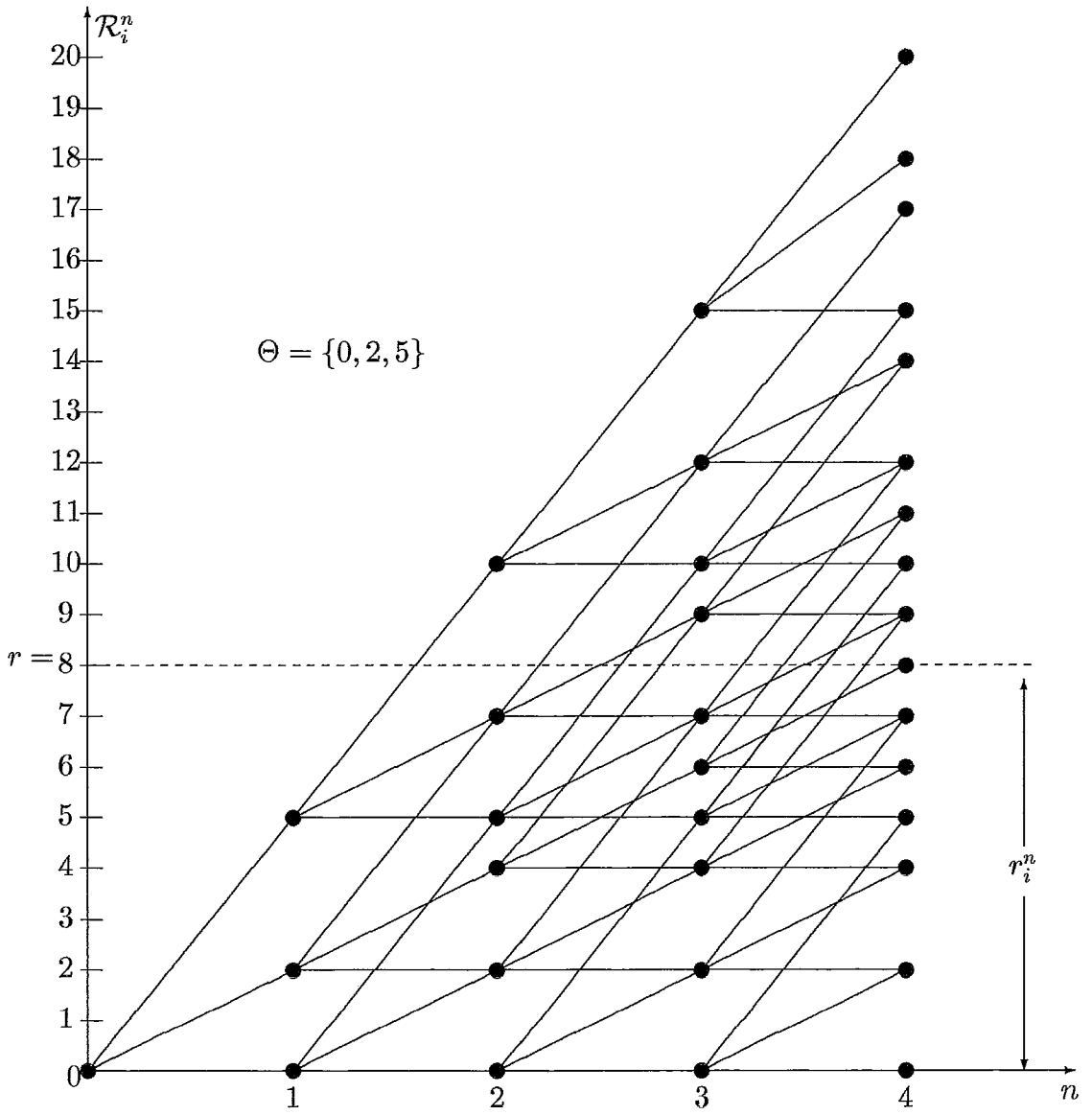


Figura III.3: Conjunto de recompensas diferentes atingíveis pelo protocolo.

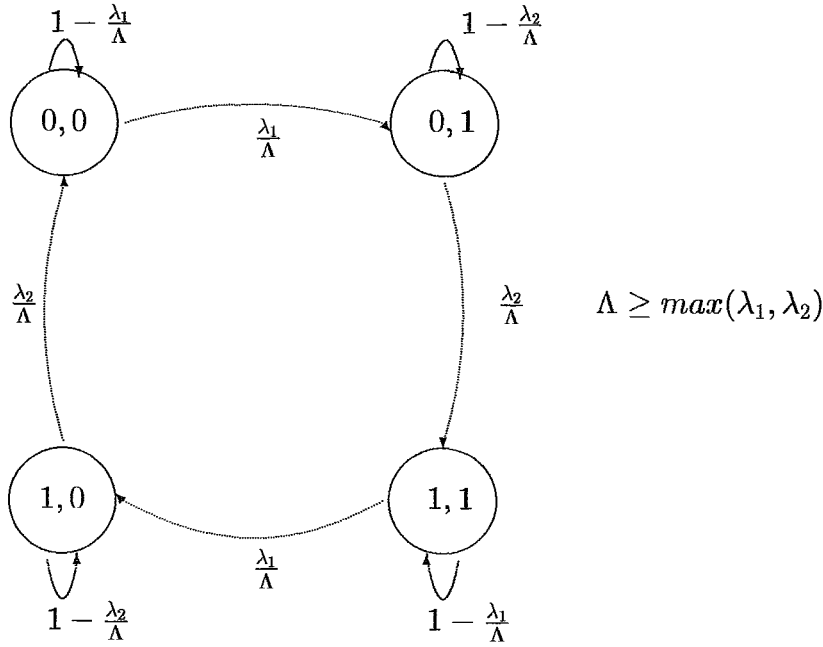


Figura III.4: CM randomizada para o modelo simplificado do protocolo *Parada e Espera*.

III.5 Exemplo

Para ilustrar a agregação de caminhos feita pelo algoritmo proposto, considere o problema de calcular a função de distribuição da vazão no intervalo $[0, t]$ do modelo simplificado do protocolo *Parada e Espera* (ver seção III.1.) Segundo foi visto, a equação III.1 estabelece como mapear este problema num problema de desempenhabilidade com recompensas nas transições. Portanto, para calcular a vazão desejada, é necessário avaliar $P[RTAD_i(t) \leq r]$, onde $RTAD_i(t)$ corresponde à recompensa total acumulada pelo sistema (para a atribuição de recompensas explicada no exemplo 2 da introdução).

O primeiro passo para resolver este problema consiste em randomizar a cadeia de Markov da figura III.2 que modela o protocolo. Randomizando obtém-se a cadeia de Markov de tempo discreto da figura III.4. A função ρ para esta cadeia é tal que as transições que acontecem com probabilidade λ_1/Λ tem associada recompensa 1, e as demais transições tem recompensa 0.

A figura III.5 ilustra a agregação de caminhos utilizada pelo algoritmo de cálculo de $G_i(t, r)$ proposto neste trabalho. Nela as transições com recompensa 1 estão marcadas com o símbolo *, e as transições com recompensa 0 não estão marcadas. Desta figura é fácil ver que o número de caminhos diferentes

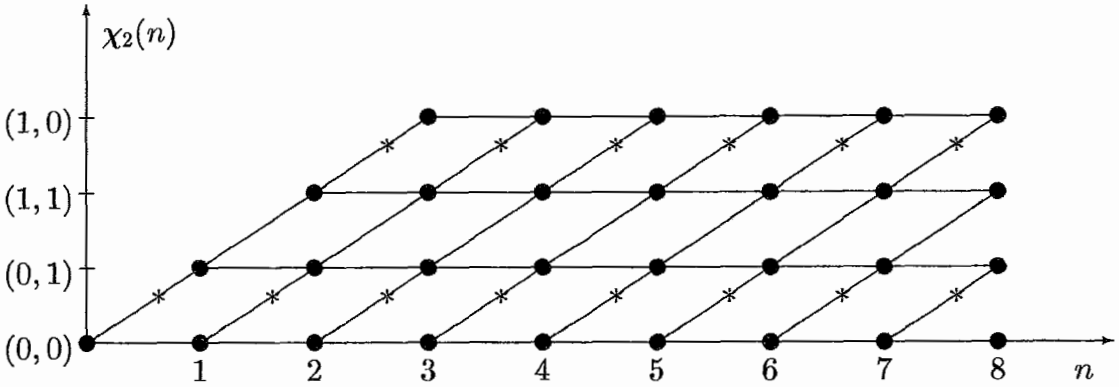


Figura III.5: Caminhos possíveis para a CM do protocolo *Parada e Espera*.

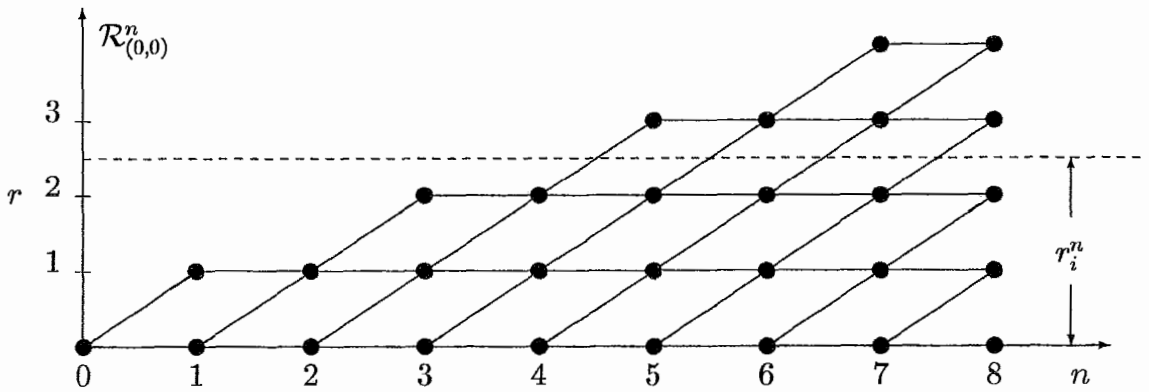


Figura III.6: Recompensas possíveis que pode atingir a CM da figura III.5

para $N(t) = n$ é $O(2^n)$, pois para $n \geq 3$ o sistema pode estar em qualquer um dos 4 estados, e de cada um deles, num passo, pode transitar a dois estados.

Por outro lado a figura III.6, ilustra que para $n \geq 5$ cumpre-se que $|r_i^n| \leq r$. Isto é muito importante pois indica que avaliar $G_i(t, r)$, em geral, requer de $O(|E|r')$ localizações de memória e $o(|E|dr')$ operações de multiplicação; em outras palavras: na maioria dos casos cumpre-se que $r' < N_{max}|\Theta|$.

III.6 Conclusões

O objetivo deste capítulo foi encontrar um algoritmo para calcular $G_i(t, r)$, com custo de memória $O(|E| \min(\theta_{|\Theta|} N_{max}, r'))$ e número de operações $O(|E| d N_{max} \min(\theta_{|\Theta|} N_{max}, r'))$. Em geral estes custos representam uma melhora significativa com relação ao resultado de [5], que tem custo de memória

$O\left(\binom{|\Theta| + N_{max} + 1}{N_{max+1}} |E|\right)$ e número de operações $O\left(\binom{|\Theta| + N_{max} + 1}{N_{max+1}} |E| d\right)$.

No caso particular que $\Theta = \{0, 1\}$ (este caso corresponde a problemas onde é necessário "contar" o número de um determinado evento em $[0, t]$), o custo de memória do algoritmo apresentado aqui é de $O(|E| \min(N_{max}, r))$ e número de operações $O(|E| d N_{max} \min(N_{max}, r))$; e no caso de [5] o custo de memória e de operações são idênticos.

Capítulo IV

Aproximação de $F_i(t, r)$ mediante $G_i(t, r)$

IV.1 Introdução

Ultimamente o cálculo de $F_i(t, r)$ tem recebido bastante atenção por parte dos pesquisadores, devido principalmente à grande aplicabilidade desta medida. Portanto, encontrar novos métodos, que sejam mais eficientes para calculá-la, implica um avanço na solução de muitos problemas reais. O tipo de problema que a desempenhabilidade pode modelar varia dependendo principalmente da interpretação física que se da às taxas de recompensa associadas aos diferentes estados. Por exemplo: se as taxas de recompensas atribuídas para cada estado correspondem a: vazão, número de unidades operacionais ou custo por unidade de tempo; então, a desempenhabilidade corresponde, respectivamente, a um determinado valor nas seguintes variáveis: número de jobs processados, tempo operacional total acumulado por todas as unidades, custo total acumulado.

Na literatura tem sido publicados métodos cada vez mais eficientes quanto ao uso de memória e número de operações para calcular $F_i(t, r)$, e, como visto no capítulo anterior, o método mais eficiente publicado até agora requer $O(|E||R|N_{max})$ localizações de memória e $O(|E||R|dN_{max}^2)$ operações, onde d é o máximo número de transições de saída de um estado da cadeia de Markov. Entretanto em muitos modelos de sistemas reais a cadeia de Markov possui um número de estados ($|E|$) muito grande, e o valor N_{max} também pode ser grande. Isto implica um custo computacional muito alto para avaliar $F_i(t, r)$. portanto, novos resultados para calcular $F_i(t, r)$, que resultem em algum benefício comparativo em relação aos resultados anteriores, são ainda necessários.

A contribuição deste capítulo consiste em desenvolver um novo método para avaliar $F_i(t, r)$ em forma aproximada. A idéia básica da aproximação consiste em calcular $G_i(t, r)$ em vez de $F_i(t, r)$, usando uma atribuição especial de recompensas às transições da cadeia de Markov randomizada. Esta atribuição

será discutida posteriormente neste capítulo. Algumas características desejáveis que oferece a aproximação são:

- O erro induzido por ela é controlável e pode ser especificado a priori. Em consequência, é possível fazer com que ele seja desprezível.
- Existe um compromisso entre o erro do resultado obtido e a exigência de recursos computacionais para obter este erro, isto é, para que o resultado da aproximação seja mais próximo ao valor exato de $F_i(t, r)$, é necessário usar mais recursos computacionais.

Evidentemente que o calcular $F_i(t, r)$ em forma aproximada, tem a desvantagem de se obter uma medida com um certo erro, porém, a aproximação proposta neste capítulo oferece algumas vantagens quando comparada com o cálculo exato de $F_i(t, r)$, usando o resultado de [9]. Estas vantagens são:

- Em geral usa menos recursos computacionais. Isto devido a que, tal como foi visto nos capítulos II e III, $G_i(t, r)$ é mais barato de ser avaliado do que $F_i(t, r)$.
- Usa somente operações de soma, multiplicação e divisão de números positivos. No entanto, a expressão de [9] usa adicionalmente a operação subtração, o que em alguns casos pode levar a problemas de instabilidade numérica.

IV.2 Aproximação Proposta

Para o caso em que o número de eventos do processo de Poisson usado na randomização seja n , isto é $N(t) = n$, considere a seguinte atribuição de recompensas à CM randomizada: cada vez que o sistema visita o estado e_i , ganha uma recompensa $r_i(n)$ igual a $\frac{r_i t}{n+1}$. Isto quer dizer que ao visitar um certo estado, a recompensa ganha pelo sistema não depende somente do estado específico no qual ele fica, mas também do número de eventos do processo de Poisson $N(t)$ no intervalo $[0, t]$. Note-se que a aproximação somente muda a atribuição de recompensas associada à cadeia de Markov randomizada. Esta cadeia e o processo de Poisson que a randomiza permanecem inalterados, isto é, são os mesmos que os usados no cálculo exato de $F_i(t, r)$.

O tipo de recompensas associado à cadeia de Markov, recém descrito, é um tipo particular de atribuição de recompensas discretas discutido no capítulo III. Tal como foi indicado nesse capítulo, atribuir recompensa fixa à visita de um estado é equivalente a atribuir essa recompensa às transições que levam (ou saem) desse estado. Isto implica que a probabilidade que a recompensa total acumulada pelo sistema no intervalo $[0, t]$ seja menor ou igual a um certo valor r , corresponde a calcular $G_i(t, r)$ do capítulo III. Em outras palavras: neste

capítulo propõe-se aproximar $F_i(t, r)$ por $G_i(t, r)$, mediante uma atribuição de recompensas apropriadas.

Para explicar o motivo pelo qual a atribuição de recompensas proposta permite que $G_i(t, r)$ aproxime $F_i(t, r)$, em primeiro lugar definem-se algumas variáveis. Considere o caso em que $N(t) = n$. Na figura IV.1 mostra-se o intervalo $[0, t]$ e as variáveis definidas. Seja τ_k , $1 \leq k \leq n$, o instante $\in [0, t]$ de acontecimento da k -ésima transição da cadeia de Markov randomizada χ_2 (equivalentemente τ_k corresponde ao instante de acontecimento do k -ésimo evento do processo de Poisson $N(t)$). Por simplicidade da notação, define-se também $\tau_0 = 0$, e $\tau_{n+1} = t$. Seja $V_{(k)}$, $1 \leq k \leq n+1$, o segmento $[\tau_{k-1}, \tau_k)$. segundo foi visto no capítulo II, $V_{(k)}$ tem distribuição Dirichlet. Seja $\Delta t = \frac{t}{n+1}$; e seja t_k , $1 \leq k \leq n$, o instante $k\Delta t \in [0, t]$.

É muito importante notar que segundo foi comentado no capítulo II, τ_k corresponde à k -ésima estatística de ordem de n variáveis aleatórias uniformes em $[0, t]$, e em contraposição as variáveis t_k são deterministas.

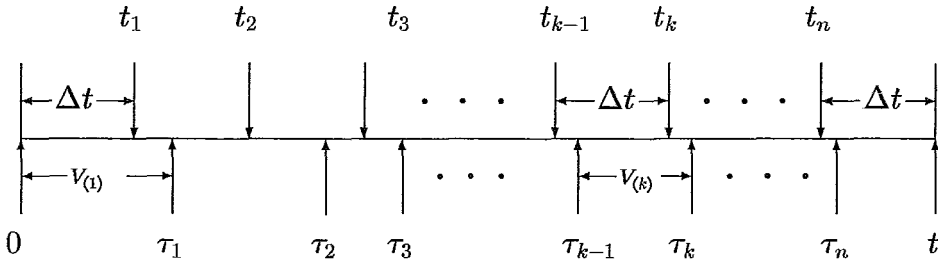


Figura IV.1: Variáveis associadas ao intervalo $[0, t]$

Seja $RTA_i(t)$ a recompensa total real ganha pelo sistema no intervalo $[0, t]$ e $RTAD_i(t)$ a recompensa total ganha pelo sistema no intervalo $[0, t]$ segundo a aproximação. É importante destacar que a recompensa atribuída ao intervalo $[t_{k-1}, t_k)$, devido a que a CM e o processo de Poisson usados na aproximação são os mesmos usados no cálculo exato de $F_i(t, r)$. é a mesma do intervalo $[\tau_{k-1}, \tau_k)$. Da figura IV.1 temos que:

$$RTA_i(t) = \sum_{k=1}^{n+1} V_{(k)} \rho(\chi_2(k) | \chi_2(0) = e_i) \quad (\text{IV.1})$$

$$RTAD_i(t) = \sum_{k=1}^{n+1} \Delta t \rho(\chi_2(k) | \chi_2(0) = e_i) \quad (\text{IV.2})$$

Seja ΔR_i o erro da aproximação em prever o valor $RTA_i(t)$, isto é:

$$\Delta R_i = RTA_i(t) - RTAD_i(t) \quad (\text{IV.3})$$

Das equações IV.1 ... IV.3, tem-se que:

$$\Delta R_i = \sum_{k=1}^{n+1} \rho(\chi_2(k) | \chi_2(0) = e_i) [V_{(k)} - \Delta t] \quad (\text{IV.4})$$

Esta equação mostra que o erro em prever a recompensa ganha pelo sistema segundo a aproximação, é proporcional a $[V_{(k)} - \Delta t] = [V_{(k)} - \frac{t}{n+1}]$. Isto quer dizer que, para entender o erro da aproximação (ou equivalentemente, entender porquê ela funciona) é necessário compreender a diferença entre a v.a. $V_{(k)}$ e o segmento determinístico de tamanho $\frac{t}{n+1}$. Isto é o que será feito a seguir.

Segundo foi indicado no capítulo II, a v.a. $V_{(k)}$ tem distribuição Dirichlet, portanto [20]:

$$\begin{aligned} P[V_{(k)} \leq s] &= P[V_{(1)} \leq s] \\ &= \sum_{i=1}^n \binom{n}{i} \left(\frac{s}{t}\right)^i \left(1 - \frac{s}{t}\right)^{n-i} \end{aligned} \quad (\text{IV.5})$$

A primeira igualdade de acima deve-se à intercambiabilidade dos segmentos $V_{(k)}$ [18]. A equação de acima implica que:

$$P[V_{(1)} > s] = \left(1 - \frac{s}{t}\right)^n \quad (\text{IV.6})$$

De onde:

$$\begin{aligned} E[V_{(1)}] &= \int_0^{\infty} P[V_{(1)} > s] ds \\ &= \int_0^t P[V_{(1)} > s] ds \\ &= \frac{t}{n+1} \end{aligned} \quad (\text{IV.7})$$

A equação IV.4 mostra que o erro em prever a recompensa ganha pelo sistema segundo a aproximação é proporcional a $[V_{(k)} - \Delta t] = [V_{(k)} - \frac{t}{n+1}]$, e IV.7 mostra que o valor medio de $V_{(k)}$ é precisamente $\frac{t}{n+1}$. Adicionalmente pode ser demonstrado que a variância de $V_{(k)}$ é igual a $\frac{nt^2}{(n+1)^2(n+2)}$ e portanto diminui ao aumentar n . Os valores da esperança e a variância de $V_{(k)}$ implicam que se o valor n for muito grande, o fator $[V_{(k)} - \frac{t}{n+1}]$ tende a ser muito pequeno e portanto o erro da aproximação diminui. O interessante é que existe uma forma de fazer que com alta probabilidade $N(t)$ seja um valor grande. Isto se consegue simplesmente randomizando por um valor Λ suficientemente grande.

A análise do tipo de resultado que se consegue usando-se a aproximação proposta neste capítulo, será feito a seguir usando o seguinte método: em primeiro lugar, para uma cadeia de Markov específica, comparar-se-á o resultado obtido pela aproximação com respeito ao caso exato, para uma cadeia de Markov específica. Esta cadeia é composta por dois estados: o estado inicial e um estado absorvente, e será usada para modelar a recompensa total acumulada pelo sistema durante a visita a um estado. Posteriormente, usando as conclusões obtidas para o caso mencionado, se analisará o resultado da aproximação para uma cadeia de Markov em geral.

IV.3 Resultado da aproximação para a recompensa adquirida durante a permanência em um único estado.

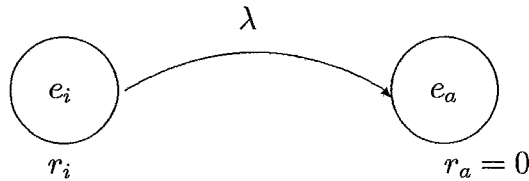


Figura IV.2: Cadeia de Markov de tempo contínuo que modela a recompensa ganha pelo sistema num único estado (modelo 1).

O objetivo desta seção é calcular a recompensa ganha para cadeia de Markov C durante todo o tempo que ela permanece (considerando somente uma passagem) em um determinado estado, o qual será denotado como e_i . Para realizar este cálculo usar-se-á o modelo da da figura IV.2, que é obtido da seguinte forma: os estados e_i e e_a da figura IV.2 representam, respectivamente, o estado e_i e a agrupação de todos os demais estados (diferentes de e_i) de C ; o tempo de permanência no estado e_i da figura IV.2 é exponencialmente distribuído com parâmetro λ , onde λ corresponde à soma das taxas de saída do estado e_i da CM. Isto implica que o tempo de permanência no estado e_i é o mesmo em ambas cadeias. O estado e_i tem associada a mesma taxa de recompensa r_i em ambas cadeias. Uma vez que o sistema abandona o estado e_i de C , ele não ganha mais recompensa por essa visita. Isto é modelado na figura IV.2 pela transição ao estado absorvente e_a o qual tem associada uma taxa de recompensa r_a igual a zero.

Para o caso da cadeia de Markov da figura IV.2, é sabido que [28]

$F_i(t, r)$ está dada por:

$$F_i(t, r) = \begin{cases} 0 & , r < 0 \\ 1 - e^{-\lambda \frac{r}{r_i}} & , 0 \leq r < r_i t \\ 1 & , r \geq r_i t \end{cases} \quad (\text{IV.8})$$

Seja $G_i(t, r)$ a função de distribuição da desempenhabilidade obtida segundo a aproximação acima. Tal como em outros casos, para calcular $G_i(t, r)$ usar-se-á a técnica de randomização.

Randomizando a cadeia de Markov da figura IV.2 com taxa $\Lambda \geq \lambda$, é obtida a cadeia de Markov da figura IV.3. Nessa figura é indicado que a recompensa ganha (segundo a aproximação) em cada visita ao estado e_i , no caso que $N(t) = n$, é $r_i(n) = \frac{r_i t}{n+1}$. Além do mais $P_{ii} = (1 - \frac{\lambda}{\Lambda})$; $P_{ia} = \frac{\lambda}{\Lambda}$; $r_a(n) = 0$; e $P_{aa} = 1$.

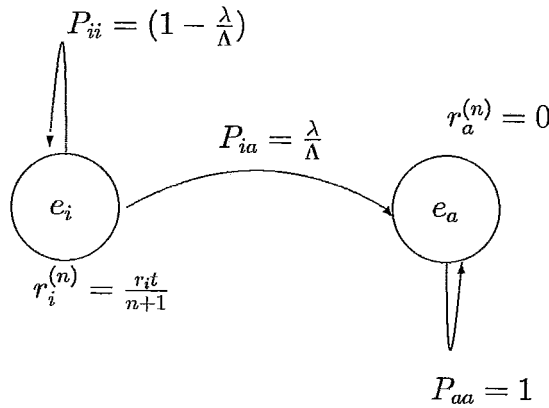


Figura IV.3: CM de parâmetro discreto correspondente à CM do modelo 1

Para a cadeia de Markov da figura IV.3, condicionando no valor de $N(t)$ e aplicando o teorema de probabilidades totais, obtém-se:

$$G_i(t, r) = \sum_{n=0}^{\infty} e^{-\Lambda t} \frac{(\Lambda t)^n}{n!} G_i(t, r|n) \quad (\text{IV.9})$$

$G_i(t, r|n)$ é avaliado condicionando no número m de transições do estado e_i a si mesmo ($0 \leq m \leq n$), de onde:

$$G_i(t, r|n) = \sum_{m=0}^n P[m|n] G_i(t, r|n, m) \quad (\text{IV.10})$$

O fato do estado e_a ser um estado absorvente, implica que se acontecem m transições do estado e_i a si mesmo, estas devem ser as m primeiras transições. Portanto, como a probabilidade de deixar e_i é $\frac{\lambda}{\Lambda}$:

$$P[m|n] = \begin{cases} \frac{\lambda}{\Lambda} \left(1 - \frac{\lambda}{\Lambda}\right)^m, & 0 \leq m < n \\ \left(1 - \frac{\lambda}{\Lambda}\right)^n, & m = n \end{cases} \quad (\text{IV.11})$$

Agora, para calcular $G_i(t, r|n, m)$, usa-se o fato que se existirem m transições de e_i a si mesmo, o sistema fica $(m + 1)$ vezes no estado e_i , e portanto ganha uma recompensa total igual a $\frac{(m+1)r_i t}{n+1}$. Formalmente:

$$\begin{aligned} G_i(t, r|n, m) &\stackrel{\text{def}}{=} P[RTAD_i(t) \leq r|n, m] \\ &= P\left[\frac{(m+1)r_i t}{n+1} \leq r\right] \\ &= u\left[r - \frac{(m+1)r_i t}{n+1}\right], \quad 0 \leq m \leq n \end{aligned} \quad (\text{IV.12})$$

Seja $m' = m + 1$. Substituindo IV.12 e IV.11 em IV.10; e isolando o caso $m = n$:

$$\begin{aligned} G_i(t, r|n) &= \left(1 - \frac{\lambda}{\Lambda}\right)^n u[r - r_i t] + \sum_{m=0}^{n-1} \frac{\lambda}{\Lambda} \left(1 - \frac{\lambda}{\Lambda}\right)^m u\left[r - \frac{(m+1)r_i t}{n+1}\right] \\ &= \left(1 - \frac{\lambda}{\Lambda}\right)^n u[r - r_i t] + \frac{\frac{\lambda}{\Lambda}}{\left(1 - \frac{\lambda}{\Lambda}\right)} \sum_{m'=1}^n \left(1 - \frac{\lambda}{\Lambda}\right)^{m'} u\left[r - \frac{m' r_i t}{n+1}\right] \end{aligned} \quad (\text{IV.13})$$

Seja $\alpha = \frac{r}{r_i t}$. Note-se que α corresponde a uma percentagem da máxima recompensa possível de ser acumulada pela CM de figura IV.2. Para evitar trivialidades estudar-se-á somente o caso $0 \leq r \leq r_i t$. Neste caso cumpre-se que: $0 \leq \alpha \leq 1$. Seja $m^*(n)$ o máximo valor de m' tal que $r \geq \frac{m' r_i t}{n+1}$, então usando o fato que $(1 \leq m' \leq n)$:

$$m^*(n) = \min(n, \lfloor \alpha(n+1) \rfloor) \quad (\text{IV.14})$$

Substituindo IV.14 em IV.13:

$$\begin{aligned} G_i(t, r|n) &= \left(1 - \frac{\lambda}{\Lambda}\right)^n u[r - r_i t] + \frac{\frac{\lambda}{\Lambda}}{\left(1 - \frac{\lambda}{\Lambda}\right)} \sum_{m=1}^{m^*(n)} \left(1 - \frac{\lambda}{\Lambda}\right)^{m'} \\ &= \left(1 - \frac{\lambda}{\Lambda}\right)^n u[r - r_i t] + \frac{\frac{\lambda}{\Lambda}}{\left(1 - \frac{\lambda}{\Lambda}\right)} \left[\frac{1 - \left(1 - \frac{\lambda}{\Lambda}\right)^{m^*(n)+1}}{1 - \left(1 - \frac{\lambda}{\Lambda}\right)} - 1 \right] \\ &= \left(1 - \frac{\lambda}{\Lambda}\right)^n u[r - r_i t] + \frac{1 - \left(1 - \frac{\lambda}{\Lambda}\right)^{m^*(n)+1} - \frac{\lambda}{\Lambda}}{1 - \frac{\lambda}{\Lambda}} \end{aligned}$$

Portanto:

$$G_i(t, r|n) = 1 - \left(1 - \frac{\lambda}{\Lambda}\right)^{m^*(n)} + \left(1 - \frac{\lambda}{\Lambda}\right)^n u[r - r_i t] \quad (\text{IV.15})$$

Sustituindo IV.15 em IV.9:

$$\begin{aligned} G_i(t, r) &= \sum_{n=0}^{\infty} e^{-\Lambda t} \frac{(\Lambda t)^n}{n!} \left\{ 1 - \left(1 - \frac{\lambda}{\Lambda}\right)^{m^*(n)} + \left(1 - \frac{\lambda}{\Lambda}\right)^n u[r - r_i t] \right\} \\ &= 1 - \sum_{n=0}^{\infty} e^{-\Lambda t} \frac{(\Lambda t)^n}{n!} \left(1 - \frac{\lambda}{\Lambda}\right)^{m^*(n)} + \sum_{n=0}^{\infty} e^{-\Lambda t} \frac{[t(\Lambda - \lambda)]^n}{n!} u[r - r_i t] \end{aligned} \quad (\text{IV.16})$$

Como $e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!}$:

$$G_i(t, r) = 1 - \sum_{n=0}^{\infty} e^{-\Lambda t} \frac{(\Lambda t)^n}{n!} \left(1 - \frac{\lambda}{\Lambda}\right)^{m^*(n)} + e^{-\lambda t} u[r - r_i t] \quad (\text{IV.17})$$

onde $m^*(n) = \min\{n, [\alpha(n+1)]\}$, e $\alpha = \frac{r}{r_i t}$. A equação IV.17 corresponde à expressão final para $G_i(t, r)$ da figura IV.3.

IV.4 Análise de erro da recompensa acumulada em um único estado

A seguir analisa-se o erro induzido ao aproximar o valor de $F_i(t, r)$ da cadeia de Markov C pelo valor de $G_i(t, r)$ da cadeia de Markov da figura IV.3. A análise mostra que o erro depende do valor de r , e portanto ela será dividida em diferentes casos, que correspondem aos diferentes valores de r .

1. Caso $r = 0$.

Neste caso temos que $\alpha = 0$, portanto:

$$m^*(n) = 0$$

$$u[-r_i t] = 0$$

Sem perda de generalidade pode-se considerar que $r_i > 0$, portanto:

$$\begin{aligned} G_i(t, 0) &= 1 - \sum_{n=0}^{\infty} e^{-\Lambda t} \frac{(\Lambda t)^n}{n!} \\ &= 0 \end{aligned} \quad (\text{IV.18})$$

Comparando as equações IV.8 e IV.18, vê-se que no caso $r = 0$ a aproximação corresponde ao caso exato.

2. **Caso** $r = r_i t$. Neste caso:

$$\alpha = 1$$

$$m^*(n) = n$$

$$u[r - r_i t] = 1$$

Portanto, de IV.17 temos que:

$$G_i(t, r_i t) = 1 - \sum_{n=0}^{\infty} e^{-\Lambda t} \frac{(\Lambda t)^n}{n!} \left\{ \left(1 - \frac{\lambda}{\Lambda}\right)^n - \left(1 - \frac{\lambda}{\Lambda}\right)^n \right\} \quad (\text{IV.19})$$

$$= 1 \quad (\text{IV.20})$$

O que, novamente corresponde ao caso exato, já que de IV.8: $F_i(t, r_i t) = 1$.

3. **Caso** $0 < r < r_i t$.

Neste caso:

$$0 < \alpha < 1$$

$$m^*(n) = \min(n, [\alpha(n+1)])$$

$$= [\alpha(n+1)]$$

$$u[r - r_i t] = 0$$

Portanto:

$$G_i(t, r) = 1 - \sum_{n=0}^{\infty} e^{-\Lambda t} (\Lambda t)^n \frac{\left(1 - \frac{\lambda}{\Lambda}\right)^{[\alpha(n+1)]}}{n!} \quad (\text{IV.21})$$

O valor de $[\alpha(n+1)]$ pode ser limitado por ambos lados da seguinte forma:

$$\begin{aligned} \alpha(n+1) &= \alpha n + \alpha \\ &\geq [\alpha(n+1)] \end{aligned} \quad (\text{IV.22})$$

e:

$$\begin{aligned} [\alpha(n+1)] &\geq \alpha(n+1) - 1 \\ &\geq \alpha n - (1 - \alpha) \end{aligned} \quad (\text{IV.23})$$

Usando IV.22 e IV.23 em IV.21, ve-se que:

$$G_i(t, r) \leq 1 - \sum_{n=0}^{\infty} e^{-\Lambda t} \frac{(\Lambda t)^n \left(1 - \frac{\lambda}{\Lambda}\right)^{\alpha n + \alpha}}{n!} \quad (\text{IV.24})$$

$$G_i(t, r) \geq 1 - \sum_{n=0}^{\infty} e^{-\Lambda t} \frac{(\Lambda t)^n \left(1 - \frac{\lambda}{\Lambda}\right)^{\alpha n - (1 - \alpha)}}{n!} \quad (\text{IV.25})$$

Portanto:

$$G_i(t, r) \leq 1 - \left(1 - \frac{\lambda}{\Lambda}\right)^\alpha \sum_{n=0}^{\infty} e^{-\Lambda t} \frac{\left[\Lambda t \left(1 - \frac{\lambda}{\Lambda}\right)^\alpha\right]^n}{n!} \quad (\text{IV.26})$$

$$G_i(t, r) \geq 1 - \left(1 - \frac{\lambda}{\Lambda}\right)^{-(1-\alpha)} \sum_{n=0}^{\infty} e^{-\Lambda t} \frac{\left[\Lambda t \left(1 - \frac{\lambda}{\Lambda}\right)^\alpha\right]^n}{n!} \quad (\text{IV.27})$$

Como $e^x = \sum_{x=0}^{\infty} \frac{x^n}{n!}$:

$$G_i(t, r) \leq 1 - \left(1 - \frac{\lambda}{\Lambda}\right)^\alpha e^{-\Lambda t} e^{\Lambda t \left(1 - \frac{\lambda}{\Lambda}\right)^\alpha} \quad (\text{IV.28})$$

$$G_i(t, r) \geq 1 - \left(1 - \frac{\lambda}{\Lambda}\right)^{-(1-\alpha)} e^{-\Lambda t} e^{\Lambda t \left(1 - \frac{\lambda}{\Lambda}\right)^\alpha} \quad (\text{IV.29})$$

Seja $\varepsilon = (F_i(t, r) - G_i(t, r))$, o erro obtido ao calcular $F_i(t, r)$ mediante $G_i(t, r)$. De IV.28 e IV.29:

$$\begin{aligned} \varepsilon &\leq \left[\left(1 - \frac{\lambda}{\Lambda}\right)^{\alpha-1} - \left(1 - \frac{\lambda}{\Lambda}\right)^\alpha \right] e^{-\Lambda t + \Lambda t \left(1 - \frac{\lambda}{\Lambda}\right)^\alpha} \\ &= \left(1 - \frac{\lambda}{\Lambda}\right)^\alpha \left[\frac{1}{1 - \frac{\lambda}{\Lambda}} - 1 \right] e^{-\Lambda t + \Lambda t \left(1 - \frac{\lambda}{\Lambda}\right)^\alpha} \end{aligned} \quad (\text{IV.30})$$

Devido a que $0 \leq \alpha \leq 1$ e $\lambda < \Lambda$:

$$\varepsilon \leq \left(1 - \frac{\lambda}{\Lambda}\right)^\alpha \left(1 - \frac{\lambda}{\Lambda}\right) \quad (\text{IV.31})$$

Portanto:

$$\varepsilon \leq \frac{\lambda}{\Lambda - \lambda} \quad (\text{IV.32})$$

IV.31 e IV.32 estabelecem que o erro da aproximação, para o caso $0 < r < r_i t$, depende dos seguintes fatores:

- Ao aumentar Λ o erro diminua.
- Ao aumentar α o erro diminua. Isto é importante, pois os sistemas normalmente são projetados para terem alta disponibilidade e portanto é interessante avaliar $F_i(t, r)$ para valores de α cercanos a 1.

Como α depende de r , r_i , e t , conclue-se que:

- Ao variar r desde zero até $r_i t$ (mantendo os demais parâmetros constantes) o erro diminui.

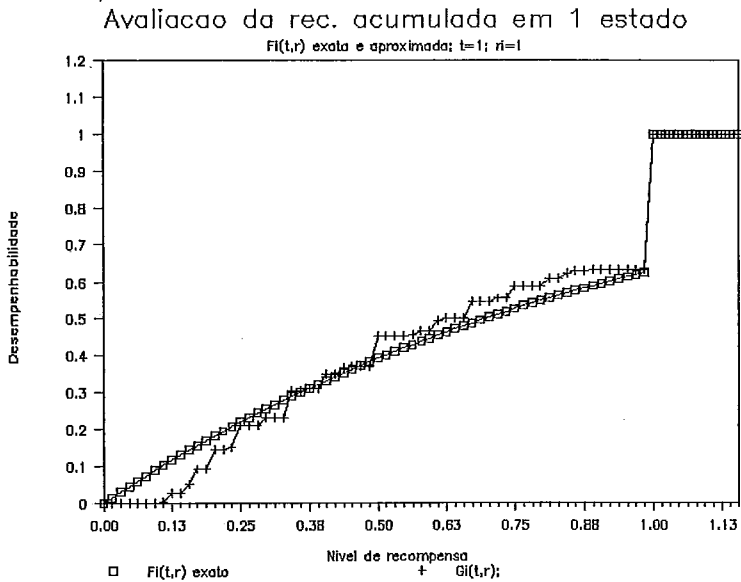


Figura IV.4: Erro obtido para o modelo 1

- O erro aumenta ao aumentar r_i (mantendo os demais parâmetros constantes).
- O erro aumenta ao aumentar t (mantendo os demais parâmetros constantes).

A modo de exemplo, considere-se agora a aproximação de $F_i(t, r)$ mediante $G_i(t, r)$ para o conjunto de valores dos parâmetros que se indicam na figura IV.4. Nesta figura vê-se que para o caso em que $r_i = 1$, $\lambda = 1$, $t = 1$ e $\Lambda = 4$, a função de erro dada pela equação IV.32 indica que $\varepsilon \leq \frac{1}{4}$, e o gráfico mostra que $\varepsilon \leq 0.1$, o que mostra que IV.32 entrega limites cercanos aos reais. Por outro lado, na figura IV.5 se mostra o erro relativo da aproximação para o caso em que: $r_i = 1$, $\lambda = 1$, $t = 1$ e $\Lambda = 1024$. Neste último exemplo, devido a que o erro é muito pequeno foi necessário desenhar o erro relativo, definido como: $(F_i(t, r) - G_i(t, r))/F_i(t, r)$. Na figura IV.5 vê-se, mais uma vez, que o limite da equação IV.32 é correto. Note-se que para valores de r cercanos a 0, $F_i(t, r)$ também é cercano a 0 e portanto o erro relativo da figura IV.5 tende a ser ilimitado; porém, isto se deve somente a que está sendo graficado o erro relativo, o erro propriamente tal continua sendo limitado pela equação IV.32.

IV.5 Análise do erro para uma cadeia de Markov geral

Seja ε_n o erro da aproximação para uma CM qualquer, no caso em que $N(t) = n$. Condicionando no valor de $N(t)$ e aplicando o teorema de probabilidades totais,

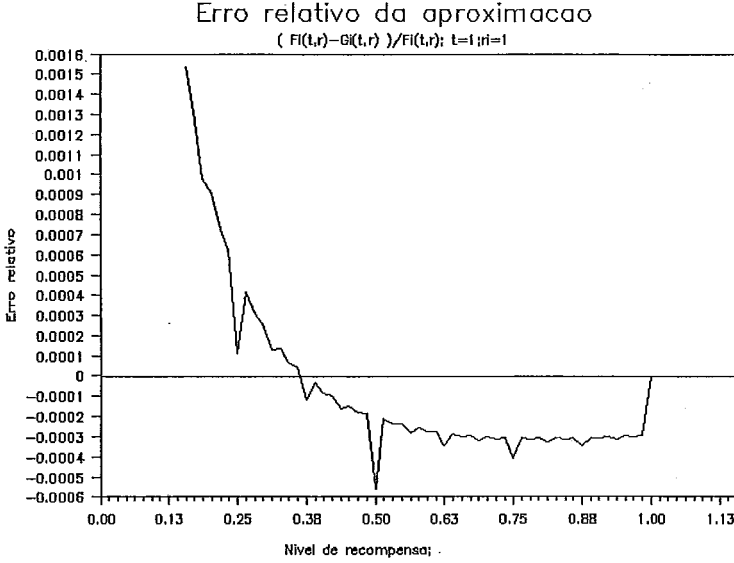


Figura IV.5: Erro relativo obtido para o modelo 1

tem-se que:

$$\varepsilon = \sum_{n=0}^{\infty} P[N(t) = n] \varepsilon_n \quad (\text{IV.33})$$

Seja m_i o número de transições fora do estado e_i , dadas n transições da CM randomizada, e seja $m = \sum_{e_i \in \mathcal{E}} m_i$. Esta definição indica que m corresponde ao número de transições *reais* da CM no intervalo $[0, t]$. Portanto $n = m + n_{tv}$, onde n_{tv} corresponde ao número de transições virtuais no intervalo $[0, t]$. Condicionando e descondicionando no valor de m , chega-se a que:

$$\varepsilon = \sum_{n=0}^{\infty} P[N(t) = n] \sum_{m=0}^n P[m|n] \varepsilon_{n,m} \quad (\text{IV.34})$$

Onde $\varepsilon_{n,m}$ corresponde ao erro da aproximação no caso em que $N(t) = n$ e acontecem m transições reais no intervalo $[0, t]$. Seja $\hat{\varepsilon}$ o máximo erro da aproximação devido a m_i transições para fora do estado e_i , portanto:

$$\begin{aligned} \varepsilon_{n,m} &\leq \sum_{e_i} m_i \varepsilon_i \\ &\leq m \hat{\varepsilon} \end{aligned} \quad (\text{IV.35})$$

Logo:

$$|\varepsilon| \leq \sum_{n=0}^{\infty} P[N(t) = n] \sum_{m=0}^n P[m|n] m \hat{\varepsilon} \quad (\text{IV.36})$$

Onde $\hat{\varepsilon} = \max_{e_i \in E} \{\hat{\varepsilon}_i\}$. Seja γ a máxima taxa de saída de um estado

da cadeia de Markov (sem randomizar) que modela o sistema. Seja N^* o número médio de mudanças de estado reais da cadeia de Markov no intervalo $[0, t]$, então evidentemente cumpre-se que:

$$N^* \leq \gamma t \quad (\text{IV.37})$$

Usando IV.37 em IV.36 conclue-se que:

$$|\varepsilon| \leq |\hat{\varepsilon}| \gamma t \quad (\text{IV.38})$$

Por outro lado, a equação IV.32 implica que:

$$|\hat{\varepsilon}| \leq \frac{\gamma}{\Lambda - \gamma} \quad (\text{IV.39})$$

Finalmente, de IV.38 e IV.39 conclue-se que:

$$|\varepsilon| \leq \frac{\gamma^2 t}{\Lambda} \quad (\text{IV.40})$$

A equação IV.40 demonstra que para uma cadeia de Markov qualquer, fazendo com que a taxa Λ de randomização seja suficientemente grande, o erro obtido ao calcular $F_i(t, r)$ mediante $G_i(t, r)$ pode ser tão pequeno quanto se quiser.

IV.6 Exemplo.

Considere um sistema tolerante a falhas [9] composto por N processadores e tamanho de memória B . Seja μ_p a taxa de falha de cada processador, e suponha que cada processador falha independentemente. A falha de um processador implica que o sistema é degradado a outro sistema com um processador a menos. Seja μ_b a taxa de falhas de cada componente da memória. Cada um destes componentes falha independentemente e sua falha deixa o sistema inoperante. Quando o sistema está inoperante não podem acontecer mais falhas. Cada vez que um componente falha (processador o etapa de memória), ele é isolado do resto do sistema e deixado no conjunto de componentes falhos. Os componentes falhos são retornados ao sistema somente quando todos eles são reparados. O tempo médio de

falha para reparar j componentes falhos é j/μ_r . Se adicionalmente supõe-se que os tempos de falha e reparo têm distribuição exponencial, então o estado do sistema pode ser modelado por um processo markoviano homogêneo $\{X(\tau), U(\tau)\}$, onde $X(\tau)$ é o número de processadores operacionais no instante τ , e $U(\tau)$ vale 1 ou 0 dependendo se a memória está operacional ou não respectivamente. Para analisar o rendimento deste sistema associa-se a cada estado $(p, 1)$, $0 \leq p \leq N$, uma taxa de recompensa igual à vazão média em estado estacionário de um sistema de filas $M/M/p/p + B$, com taxa de chegadas μ_a e tempo médio de serviço $1/\mu_s$; a taxa de recompensa associada com os estados $(-, 0)$ e com o estado $(0, 1)$ é 0.

Como conseqüência da atribuição de recompensas aos diferentes estados do sistema, descrita acima, a recompensa total acumulada por ele no intervalo $[0, t]$ corresponde ao número de trabalhos que processa durante esse período de tempo.

A figura IV.6 considera a avaliação da desempenhabilidade para o sistema recém descrito, em função de diferentes tamanhos de memória, no caso em que os valores dos diferentes parâmetros são: $\mu_a = 80$, $\mu_s = 20$, $N = 4$, $\mu_p = 0.01/semana$, e $1/\mu_r = 6 horas$. O gráfico superior corresponde à máxima recompensa atingível pelo sistema durante um dia de operação, isto é, a recompensa ganha pelo sistema no caso em que não acontecem falhas durante sua operação. O gráfico inferior corresponde ao número de tarefas que são processadas num dia de operação com probabilidade menor o igual a 0.99; este gráfico é uma reprodução do apresentado em [9], sendo que naquele trabalho foi obtido usando o valor exato da desempenhabilidade $F_i(t, r)$. Os dois gráficos do meio correspondem ao número de tarefas processadas em um dia com probabilidade menor o igual a 0.99, sendo que este número é calculado usando a aproximação proposta neste capítulo. O gráfico mais próximo ao caso exato foi obtido usando uma taxa de randomização igual a quatro vezes a máxima taxa de saída dos estados do sistema, sendo que o outro gráfico foi obtido usando uma taxa de randomização duas vezes maior que a máxima taxa de saída do sistema. Da figura observa-se que, como era de se esperar, a medida que a taxa de randomização aumenta, o erro da aproximação diminui. Outro fato que se observa na figura é que ainda para valores pequenos do fator de aproximação (taxa de randomização dividida pela máxima taxa de saída do sistema), os resultados obtidos pela aproximação são bastantes satisfatórios.

IV.7 Conclusões

Neste capítulo foi demonstrado que mediante uma atribuição apropriada de recompensas, $G_i(t, r)$ pode aproximar $F_i(t, r)$ com o grau de precisão que se deseje. A vantagem fundamental desta aproximação consiste em que é mais eficiente avaliar $G_i(t, r)$ do que $F_i(t, r)$. Isto devido a que tal como indicado em [9], avaliar $F_i(t, r)$ requer de $O(|E||R|N_{max})$ localizações de memória e $O(|E|d|R|N_{max}^2)$ operações, onde d é o máximo número de transições incidente a um estado na cadeia de Markov. Por outro lado no capítulo III foi demonstrado que avaliar $G_i(t, r)$ requer

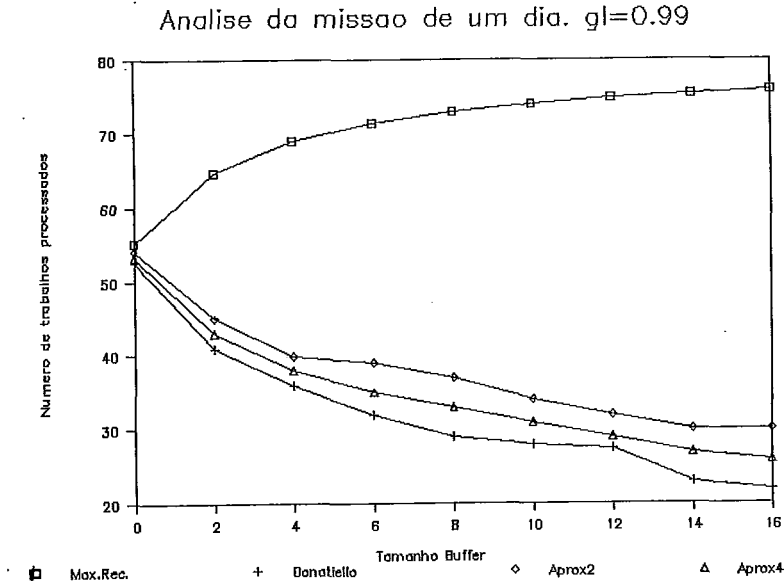


Figura IV.6: Recompensa acumulada de um sistema tolerante a falhas

de $O(|E| \min(\theta'_{|e|} N_{max}, r'))$ localizações de memória e de $O(|E| d N_{max} \min(\theta'_{|e|} N_{max}, r'))$ operações, onde $\theta'_{|e|}$ e r' são valores que dependem do conjunto de recompensas discretas diferentes associadas aos estados.

Outra vantagem da aproximação, que em alguns casos pode ser importante, é que ela não usa a operação subtração, senão que emprega somente as operações soma, multiplicação e divisão. Porém, em [9] adicionalmente é usada a operação subtração. Isto pode levar a problemas numéricos, devido às eventuais subtrações de quantidades similares, particularmente quando isto acontece no denominador.

Capítulo V

Checkpointing and Roll-Back and Recovery

V.1 Introdução

Os sistemas de banco de dados orientados ao processamento de transações são cada dia mais importantes devido à sua grande utilidade em diferentes aplicações. Na maioria desses sistemas, a demanda de dados imposta pelos usuários requer uma alta disponibilidade das informações e um baixo tempo de resposta das transações. Define-se a disponibilidade como a fração do tempo que o sistema está disponível para processar transações. É interessante notar que o ‘tempo disponível’ significa que o sistema está disponível para processar transações; porém o usuário pode ou não usar o sistema quando este está disponível. Do ponto de vista do usuário, o ‘tempo disponível’ é composto pelo ‘tempo efetivamente usado’ e pelo tempo em que o sistema permanece ocioso (porém disponível). Em contraposição ao tempo disponível está o ‘tempo indisponível’, que corresponde ao tempo em que o sistema não pode atender transações de usuários.

Uma das técnicas mais comuns para conservar a integridade dos dados, aumentar a disponibilidade ou diminuir o tempo de resposta, é guardar cópias do estado do banco de dados periodicamente, da forma que será detalhada abaixo [12]. O estado do sistema inclui todos os arquivos e informações necessárias para restaurar o sistema à situação em que ele se encontrava no momento em que foi feita a cópia. Esta informação é gravada em um dispositivo secundário livre de erro. O processo de fazer uma cópia do sistema é conhecido com o nome de *Checkpointing*. Durante as operações de *Checkpointing* o sistema não está disponível para processar transações.

Depois de finalizado o *Checkpointing*, as transações que modificam o sistema de arquivos são gravadas em um arquivo especial, conhecido com o nome de *Audit Trail*. No caso de acontecer uma falha no banco de dados, seja devido a falha de *hardware*, *software* ou má operação, supõe-se que ela é detectada ins-

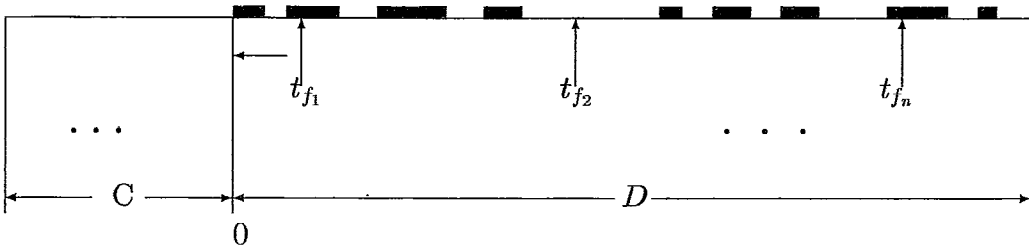


Figura V.1: Ciclo Típico de CRR

tantâneamente e é iniciado um reparo.

Numa operação de reparo a falha é removida da seguinte forma: em primeiro lugar o sistema é restaurado ao estado em que estava no momento em que foi feito o último. Esta fase de reparo é conhecida com o nome de ‘Retorno’ (*Roll-Back*). Depois, a ação de reparo continua com o reprocessamento de todas as transações armazenadas no *Audit Trail*, até deixar o sistema no estado em que estava justo antes do momento de acontecer a falha. Esta segunda e última fase do reparo é conhecida pelo nome de ‘Recuperação’ (*Recover*). Durante a operação de reparo o sistema não está disponível para processar transações dos usuários. O processamento normal é reinicializado logo após o término com sucesso do reparo. É interessante notar que a discussão anterior implica que o tempo em que o sistema está indisponível corresponde à soma dos tempos usados em repará-lo e o tempo gasto em operações de *Checkpointing*.

Na figura V.1 mostra-se um ciclo típico de *Checkpointing and Roll-back and Recover* (CRR). Este ciclo se inicia com a operação de *Checkpointing* de duração C . Durante este intervalo o sistema está indisponível. Depois vem o intervalo disponível de duração D . Na figura foram omitidos os tempos de reparo, mas estes aparecem em forma implícita como veremos a seguir. Dentro do intervalo D , os segmentos de tempo em que o sistema está atendendo transações foram desenhados como um segmento grosso no eixo do tempo, o resto do intervalo D corresponde ao tempo disponível não usado em processar transações. Na figura o instante de acontecimento da primeira falha, medido a partir do final do *checkpoint*, está marcado com o símbolo t_{f_1} . A flecha de retorno desde t_{f_1} ao início do intervalo D , significa que depois da falha, o processamento de transações deve recomeçar no estado em que se encontrava o sistema ao final do *checkpoint*. A operação de reparo constituída pelas fases de Retorno e de Recuperação não aparecem desenhadas. A fase de Retorno corresponde a leer o estado em que estava o sistema ao final do *Checkpoint*. A fase de Recuperação da primeira falha, corresponde ao reprocessamento das transações que tinham sido processadas no segmento $[0, t_{f_1}]$ do intervalo D . O tempo de Recuperação da primeira falha em geral é menor que t_{f_1} , pois nesta recuperação são descartados os segmentos de tempo disponíveis não utilizados em processar transações. Em termos gráficos: o tempo de recuperação da primeira falha corresponde à soma dos segmentos grossos do intervalo $[0, t_{f_1}]$ do tempo disponível. Depois da recuperação da primeira falha, o sistema fica como estava antes da falha, isto é como se estivesse novamente no instante t_{f_1} . As demais falhas e reparos acontecem em forma análoga à primeira

falha descrita acima.

Tal como já foi dito, o objetivo principal da técnica de CRR, é tolerar falhas; porém outros objetivos importantes são aumentar a disponibilidade e diminuir o tempo de resposta das transações. Aumentar a frequência com que são feitos os *Checkpoints* aumenta o tempo gasto nestas operações, porém (em geral) diminui o tempo usado em reparos, pois diminui a probabilidade de acontecer falhas entre 2 *checkpoints*, e caso estas aconteçam, o tempo com que são feitos os RR em geral é menor. Por outro lado, caso se diminua a frequência de *Checkpoints*, diminui o tempo gasto nesta operação, porém aumenta o tempo de RR. Isto implica que existe um intervalo ótimo entre os instantes de inicialização das operações de *checkpoint*, de maneira a maximizar a disponibilidade do sistema. Define-se como política de *Checkpointing* à especificação dos instantes no tempo em que são iniciados os *Checkpoints*. A política de *Checkpointing* que maximiza a disponibilidade do sistema não necessariamente é a mesma que minimiza o tempo de resposta das transações [26].

Na literatura tem aparecido vários modelos para analisar a técnica de CRR [26, 15, 16, 29, 12, 14]. A maioria deles foram projetados para determinar a política de *Checkpointing* que maximiza a disponibilidade [12], [14],[16], [15]. Alguns outros modelos tem considerado o problema de determinar os instantes de tempo nos quais devem ser iniciados os *Checkpointing* com o objetivo de minimizar o tempo de resposta das transações [26].

Quando se analisam sistemas submetidos a CRR, dois aspectos muito importantes a modelar são a política de *Checkpointing*, e o tempo necessário para efetuar os reparos, isto é os tempos de RR. O tempo necessário para efetuar as operações de RR depende tanto do intervalo de tempo entre o último *Checkpointing* e o instante de acontecimento da falha, como do número de transações armazenadas no *Audit Trial* (mais precisamente do tempo necessário para processar estas transações).

Com o objetivo de simplificar a análise matemática, os diferentes modelos introduzem algumas hipóteses. Estas hipóteses devem satisfazer o duplo objetivo de simplificar a análise matemática, e ainda representar corretamente os aspectos mais importantes do sistema. Algumas das suposições de *Checkpointing* que correspondem a modelos de sistemas reais, são:

- C1** : O tempo entre dois *Checkpoints* consecutivos é uma variável aleatória com distribuição exponencial.
- C2** : O tempo disponível entre o final de um *Checkpointing* e o início do outro é constante (i.e, não considerando os tempos de recuperação e reparo caso haja uma falha).
- C3** : Número de transações processadas entre dois *Checkpoints* consecutivos é constante.

A suposição C1 é a menos realista, porém é a mais fácil de analisar matematicamente. Por outro lado, em [12] foi demonstrado que a suposição C2 é ótima no sentido de maximizar a disponibilidade com relação a todas as políticas que possuem tempo entre *Checkpointing* aleatório com distribuição geral, no caso em que a taxa de falhas durante os períodos de disponibilidade é constante. A suposição C3 é dependente da carga, já que especifica que o número de transações a serem processadas entre dois *Checkpointing* consecutivos é constante, portanto quanto maior é a carga menor é o tempo entre *Checkpoints*. Esta suposição tem a desvantagem de que, no caso em que o sistema esteja com uma carga muito baixa, o tempo entre *Checkpoints* pode chegar a ser desnecessariamente grande. Na prática, uma política de *Checkpointing* pode ser uma combinação das suposições descritas acima, ou alguma outra. Por exemplo uma política que do ponto de vista de implementação prática parece razoável, é iniciar *Checkpointing* cada certo intervalo de tempo fixo, por exemplo todos os dias às 18:00 hrs. Esta política parece razoável já que, em geral os sistemas são projetados para ter uma alta disponibilidade, e portanto ela deve comportar-se em forma semelhante à suposição C2.

Outro aspecto importante dos modelos de CRR é a forma de modelar a dependência existente entre os tempos de RR e as suposições feitas para o intervalo entre *Checkpoints*. Esta dependência determina a solução do problema de otimização descrito anteriormente. Algumas das formas de modelar os tempos de RR são os seguintes:

- R1** : Dependência Paramétrica.
- R2** : Dependência Estocástica.
- R3** : Dependência Determinística.

Nos modelos do tipo R1 supõe-se que os tempos de RR são variáveis aleatórias independentes e identicamente distribuídas (v.a.i.i.d.'s), e os parâmetros da função de densidade de probabilidade (f.d.p.) de cada uma destas v.a.'s dependem dos parâmetros da política de *Checkpoint*. Por exemplo, o mais comum dentre os modelos deste tipo aparecidos na literatura é supor que o tempo de RR é proporcional ao tempo entre dois *Checkpoints* consecutivos [12], [14],[16], [15]. Este modelo é usado normalmente devido à simplicidade da análise matemática. Nos modelos do tipo R2 o tempo de RR é determinado por uma f.d.p. cujos parâmetros dependem do número de transações processadas desde o último *Checkpoint* até o momento da falha. A idéia do modelo R2 consiste em que, no caso que aconteça uma falha depois de terem sido processadas '*n*' transações, o tempo de RR corresponde ao processamento de outras '*n*' transações (diferentes das já processadas), porém o tempo de processamento de cada uma destas últimas transações é igualmente distribuído ao de cada uma das primeiras transações. Ao contrário do modelo R1, no modelo R2 os tempos de RR de duas falhas consecutivas não são independentes, já que as transações a serem processadas devido à uma falha posterior incluem aquelas correspondentes a uma falha anterior. Por exemplo na

figura V.1 o reprocessamento devido à falha que acontece em t_{f_2} inclui as transações processados em $[0, t_{f_1}]$, e adicionalmente inclui outras transações (as processadas no intervalo (t_{f_1}, t_{f_2})). O modelo R3 é uma representação exata do que acontece na realidade, ou seja o tempo de reprocessamento é igual ao tempo acumulado processando transações desde o último *Checkpointing* até o momento da falha. Evidentemente, tal como no caso do modelo R2, os períodos de RR consecutivos neste modelo são dependentes.

Para obter alguma medida de desempenho/ desempenhabilidade (disponibilidade, tempo de resposta, etc) de um sistema de banco de dados submetido à política de CRR, além de modelar a política de *Checkpointing* e os tempos de RR, é necessário representar no modelo os aspectos mais importantes do sistema, tais como: taxa de chegada de jobs, taxa de processamento, máxima capacidade de armazenamento de jobs no sistema, etc. Nos modelos que se encontram na literatura aparecem diferentes combinações entre políticas de *Checkpointing* e tempos de RR, assim como diferentes parâmetros do sistema. A quantidade de informação (parâmetros) representada do sistema varia dependendo se o objetivo é avaliar a disponibilidade, o tempo de resposta, ou ambos.

A análise para os modelos C1, R1 são mais simples e tem permitido o tratamento analítico [13], [25], [24]. Por outro lado, a análise das políticas C2, C3, R2, R3 são mais difíceis de se realizar e, em geral, o tratamento matemático tem sido feito mediante técnicas numéricas, ou simulação, ou uma mistura de ambas [26].

Uma das contribuições importantes deste trabalho é resolver analiticamente um modelo do tipo (C2,R3), o que até o momento não tinha sido conseguido na literatura, devido à dificuldade da análise matemática [26]. Especificamente, foi calculado o valor médio e a FDP da disponibilidade. As expressões encontradas para estas variáveis tem um custo computacional polinomial de baixo grau, o que Isto permite a implementação de uma ferramenta computacional eficiente para o cálculo destas medidas. Outra contribuição importante deste trabalho consiste em ter relacionado áreas até agora desconexas, como são as áreas de CRR e desempenhabilidade. Isto foi feito demonstrando que o cálculo da FDP da disponibilidade é equivalente a avaliar uma nova medida de desempenhabilidade, que será formalmente definida posteriormente.

V.2 Trabalhos Previos

O estudo de sistemas submetidos à política de CRR tem recebido considerável atenção ultimamente. A seguir revisam-se os trabalhos mais significativos publicados nesta área.

Em [16] foi calculada a transformada de Laplace-Stieltjes da função de distribuição de probabilidade da disponibilidade, durante um período finito de observação, de um sistema submetido a uma política de CRR. A partir desta

transformada é possível obter os diferentes momentos de disponibilidade, ou alternativamente é possível usar algum método de inversão numérica para avaliar a função de distribuição da disponibilidade. Os tempos de RR foram modelados da forma R1 e os tempos de início de *checkpoints* da forma C2. Adicionalmente as seguintes suposições foram feitas: o tempo necessário para efetuar um *checkpoint* é constante, as falhas acontecem de acordo com a um processo de Poisson e são detetados instantaneamente, durante as operações de CRR não podem acontecer falhas. Previamente os mesmos autores em [9] analisaram um problema similar ao anterior no sentido que usaram o mesmo modelo e hipóteses que em [16], porém a análise foi limitada à avaliação de distribuição da disponibilidade entre dois *checkpoints* consecutivos (e não durante um dado período de tempo). Neste caso, os autores conseguiram inverter a expressão da distribuição obtida no plano de Laplace.

Em [26] foram estudados diferentes modelos de sistemas submetidos a CRR, com o objetivo de identificar quais deles correspondem a uma representação realista do sistema e a forma em que eles são analisáveis. O método de solução empregado foi analítico, numérico ou de simulação, segundo a dificuldade do modelo. Em particular foi analisado um modelo no qual os *checkpoints* são realizados de acordo com um processo de Poisson (C1), porém se o número de transações processadas chega até um certo limite é feito um *checkpoint* especial. Portanto, esta política de *checkpoint* estudada corresponde a uma mistura das políticas C1 e C3. O método de solução empregado para o modelo anterior foi derivar um conjunto de expressões para a disponibilidade e o número médio estacionário de transações no sistema; as incógnitas das expressões obtidas são avaliadas numericamente. Uma conclusão importante deste trabalho é que os modelos de tipo C2 e R3 são bastante realistas. Este tipo de modelo foi avaliado usando simulação, pois segundo os autores o tratamento analítico do modelo é difícil.

Em [12] foi demonstrado que para maximizar a disponibilidade, o intervalo de tempo que o sistema está disponível para atender transações (não necessariamente atendendo transações) entre o final de um *checkpoint* e o início do próximo deve ser constante. Este resultado foi obtido supondo que durante o período de disponibilidade as falhas acontecem segundo um processo de Poisson, e que o tempo de uma operação de RR é proporcional ao tempo transcorrido entre o último *checkpoint* e o instante da falha (modelo R1). Posteriormente em [14] foi demonstrado que o resultado de [12] também é válido para o caso em que a taxa de falhas não é constante (como em [12]), senão que varia durante o período de disponibilidade de acordo a um processo de Poisson com envelhecimento, isto é: uma falha acontece no intervalo $[t, t + \Delta t]$ com probabilidade $\gamma(Y_t)\Delta t + o(\Delta t)$, onde Y_t corresponde ao tempo disponível acumulado pelo sistema desde o último *checkpoint* até o instante da falha, e $\gamma(Y_t)$ é uma função que modela o ‘envelhecimento’ do sistema.

Por outra parte, o método usado neste capítulo para analisar políticas de CRR permite a solução de um modelo mais realista que os modelos publicados

anteriormente. Especificamente, neste modelo os tempos de recuperação são representados em forma exata (política R3), e o intervalo de tempo **disponível** compreendido entre um *checkpoint* e o início do seguinte é considerado constante (política C2). Um dos resultados importantes foi encontrar uma expressão para o valor médio estacionário da disponibilidade e a função de distribuição da disponibilidade durante um ciclo de operação. Ambas expressões contêm termos que são calculados recorrentemente.

A seguir especificamos formalmente o modelo, e calculamos o valor médio e a função de distribuição da disponibilidade.

V.3 O Modelo

O objetivo deste capítulo é avaliar um modelo de um sistema de banco de dados submetido a CRR, que inclui os tipos de modelos C2 e R3 descritos anteriormente. Mais especificamente o interesse é avaliar o valor médio da disponibilidade do sistema em estado estacionário e a distribuição desta v.a. durante um ciclo de operação. A análise será feita para um ciclo típico de operação. Em [9] foi calculada a função de distribuição da disponibilidade de um ciclo de operação para um modelo mais simples, conforme ali foi indicado esta medida é de interesse, pois permite especificar as condições sob as quais o sistema opera com uma disponibilidade maior ou igual a um determinado valor a , o que não pode ser feito mediante um análise de valor médio. Em linhas gerais, o resultado deste capítulo é de interesse devido aos seguintes motivos:

1. O modelo emprega a política C2 de *checkpoint*, a qual é ótima no caso em que as falhas acontecem de acordo a um processo de Poisson [12].
2. Conforme indicado acima, o cálculo da função de distribuição da disponibilidade de um ciclo é uma medida de interesse prático.
3. Os tempos de RR são modelados em forma exata.
4. Se obteve uma expressão fechada para o valor médio da disponibilidade, e a função de distribuição desta v.a. é avaliada recursivamente. Este tipo de resultado além de original é importante, conforme indicado em [26], a disponibilidade obtida usando o modelo R1 (que é o normalmente usado na literatura) corresponde somente em forma aproximada ao valor obtido quando se usa simulação (padrão de referência). Porém, de acordo com a mesma referência os resultados obtidos usando o modelo R3, se ajustam bastante bem à realidade (em [26] o modelo R3 foi resolvido usando simulação).
5. As medidas calculadas fornecem uma informação bastante completa da operação do sistema. Até agora somente em [15] foi obtida a FDP da disponibilidade, porém para um modelo mais simples e menos realista, pois os tempos de RR foram modelados com a política R1. Adicionalmente,

naquele trabalho foram mostrados os benefícios de se dispor da FDP da disponibilidade.

V.3.1 Descrição do Modelo e Análise Geral

Tal como já foi discutido, um sistema de banco de dados submetido a políticas de CRR opera em forma cíclica. O ciclo se inicia com uma operação de *Checkpointing*, depois da qual vem um período de atendimento aos usuários. Uma vez finalizado este período de atendimento o ciclo é iniciado novamente. Durante ambas as fases do ciclo podem acontecer falhas, as quais devem ser reparadas para que o sistema possa continuar operando. Com o objetivo de identificar diferentes eventos que acontecem num ciclo típico, define-se a seguir algumas variáveis de interesse. Para simplificar a exposição usa-se a figura V.2, na qual aparece desenhado um ciclo típico de CRR. O valor de algumas variáveis não depende das possíveis falhas que possam acontecer; para distinguir este tipo de variáveis usa-se o símbolo ‘*’, isto é: se o valor de uma certa variável z não depende das falhas no ciclo, então ela é denotada como z^* . Na figura V.2 aparecem três tipos de intervalos diferentes. O segmento esquerdo, de tamanho C , corresponde à parte do ciclo na qual se efetua o *checkpoint*. O segmento horizontal direito, de tamanho D , denominado período de disponibilidade, corresponde ao componente do ciclo no qual o sistema está disponível para atender transações. Neste período de disponibilidade não são considerados os intervalos de tempo usados em operações de RR, os quais estão desenhados verticalmente a partir do instante (do período de disponibilidade) no qual acontece a falha. Esta forma gráfica de desenhar os tempos de RR permite indicar que depois da operação de RR, o sistema continua sua operação no mesmo estado em que se encontrava antes de acontecer a falha.

Seja C^* a v.a. correspondente ao tempo necessário para efetuar um *Checkpointing* no caso em que não acontecem falhas durante esta operação. Seja m a v.a. correspondente ao número de falhas que acontecem durante a operação de *Checkpointing* antes que esta seja concluída com sucesso. Cada vez que acontece uma falha, o *Checkpointing* deve ser reinicializado. Seja τ_{fl} ($1 \leq l \leq m$) o instante de acontecimento da l -ésima falha na operação de *Checkpointing*, medido a partir do início do ciclo. Seja C a v.a. correspondente ao tempo total necessário para efetuar com êxito a operação de *Checkpointing*. Evidentemente $C = C^* + \tau_{fm}$ (ver figura V.2).

Seja D o tempo total que o sistema está disponível para processar transações no ciclo. Note que, como é usado o modelo C2, D tem um valor constante que independe das possíveis falhas que possam acontecer no período de disponibilidade. Seja n a v.a. correspondente ao número de falhas ocorridas durante o período em que o sistema está disponível (no valor de n não são consideradas as falhas acontecidas durante o *Checkpoint*). Seja t_{fl} ($1 \leq l \leq n$) o instante de acontecimento da l -ésima falha na fase de disponibilidade, medido a partir do final do *Checkpointing*. Tal como aparece na figura V.2, no valor de t_{fl} não são considerados os períodos de CRR. Para simplificar a notação, definimos

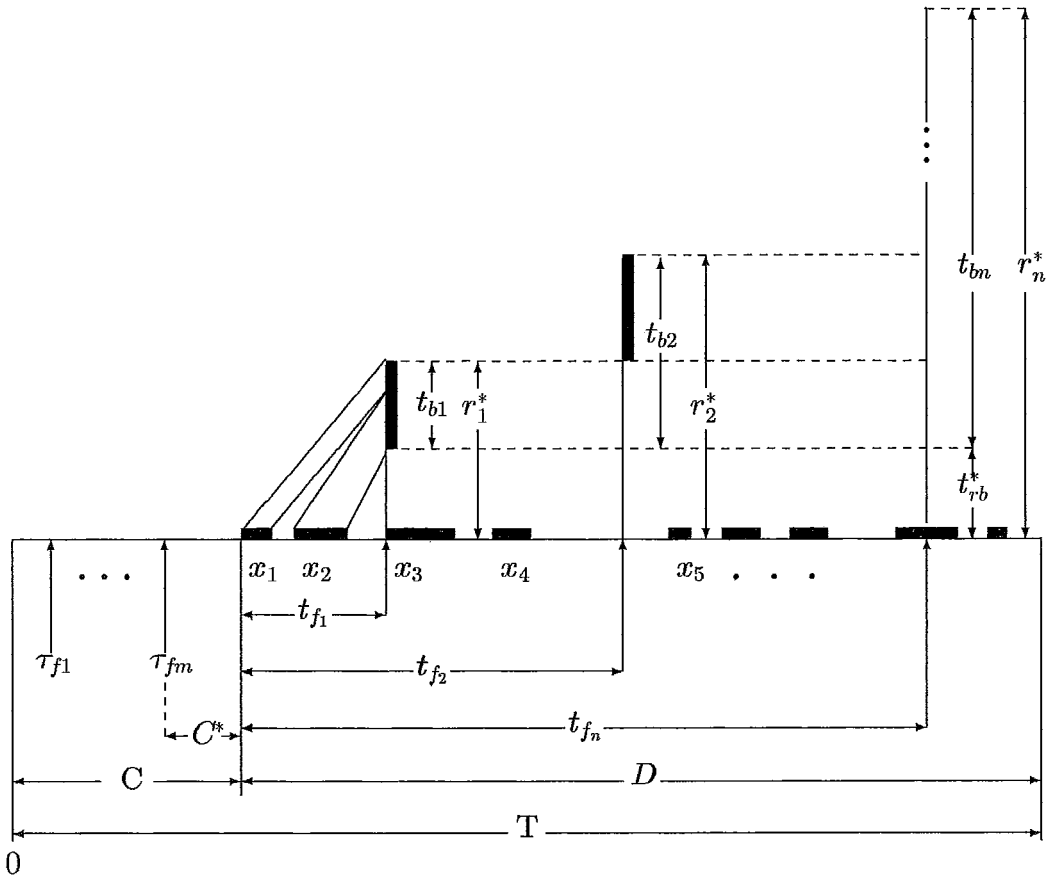


Figura V.2: Variáveis associadas a um Ciclo Típico de CRR

adicionalmente $t_{f_0} = 0$ e $t_{f_{n+1}} = D$.

Seja x_i o i -ésimo segmento de tempo útil, isto é o i -ésimo segmento do tempo disponível usado no atendimento consecutivo de uma ou mais transações. Na figura V.2 os segmentos de tempo útil aparecem desenhados com uma linha preta grossa. Seja t_{bl} ($1 \leq l \leq n$) o tempo útil acumulado até o acontecimento da l -ésima falha na fase de disponibilidade. Por exemplo, o segmento grosso vertical esquerdo da figura V.2, ilustra o fato de que t_{b1} corresponde ao tempo acumulado atendendo transações no intervalo $[0, t_{f1}]$ do período de disponibilidade, isto é: $t_{b1} = x_1 + x_2$; análogamente t_{b2} corresponde ao tempo acumulado atendendo transações no intervalo $[0, t_{f2}]$ do período de disponibilidade, portanto: $t_{b2} = x_1 + x_2 + x_3 + x_4 = t_{b1} + x_3 + x_4$, etc. Evidentemente $t_{bl} \leq t_{fl}$, já que no intervalo $[0, t_{fl}]$ podem existir subintervalos de tempo ocioso. Seja t_{rb}^* a v.a correspondente ao tempo necessário para efetuar uma operação de Retorno supondo que nesta operação não acontecem falhas. Seja r_l^* o tempo necessário para efetuar a l -ésima operação de retorno e recuperação considerando que nesta operação não acontece falha. Da figura V.2: $r_l^* = t_{rb}^* + t_{bl}$. Seja $F^*(D)$ o tempo total gasto no ciclo em operações de RR no intervalo $[0, D]$, considerando que cada uma destas operações **não teve falhas**. Na figura V.2, $F^*(D)$ corresponde à soma de todos os intervalos de tempo desenhados verticalmente; isto é: $F^*(D) = \sum_{i=1}^n r_i^*$. Sejam r_l e $F(D)$ as v.a's correspondentes ao tempo do l -ésimo RR e o tempo total gasto em operações de RR respectivamente, supondo que nestas operações **podem** acontecer falhas. Por exemplo suponha que ao efetuar a primeira operação de Retorno (que, sem considerar as falhas, dura r_1^*) acontece uma falha na metade dela (isto é $r_1^*/2$ unidades de tempo depois de iniciada) e que no segundo intento esta operação se executa corretamente, neste caso cumpre-se que: $r_1 = 1.5r_1^*$.

Nos intervalos de tempo em que o sistema está disponível, a sua evolução pode ser modelada por uma Cadeia de Markov (CM) homogênea de tempo contínuo. Seja $E = \{e_1, e_2, \dots, e_{|E|}\}$ o espaço de estados do sistema, e seja $e_i \in E$ o estado do sistema ao início do ciclo. Os estados da cadeia de Markov são tais que contêm a informação necessária para modelar a evolução do sistema. Uma modelagem simples seria considerar o estado e_k , ($1 \leq k \leq |E|$), igual a k ; onde k corresponde ao número de transações em espera para serem ou sendo processadas. Outra alternativa é considerar que o estado, além de conter o número de transações a serem processadas, contêm informação representando a estrutura (a qual pode variar) do sistema. Portanto, para não particularizar a análise a nenhuma interpretação específica, no restante deste capítulo, o k -ésimo estado será denotado como e_k . Seja $\eta(e_k)$ o número de transações a serem (ou sendo) atendidas no estado $e_k \in E$. Dependendo da definição particular de estados da CM pode existir mais de um estado que tenha associado um mesmo número de transações, ou seja, pode acontecer que $\eta(e_k) = \eta(e_j)$ para $e_k \neq e_j$. Seja $\mathcal{U} \subset E$, o conjunto de estados nos quais o sistema está realizando trabalho útil, isto é está atendendo transações. Seja $\mathcal{I} = E - \mathcal{U}$, o conjunto de estados nos quais o sistema está ocioso, isto é, nos estados pertencentes a \mathcal{I} o sistema está disponível para atender transações, porém não está sendo utilizado.

O valor de algumas variáveis depende do estado do sistema ao

início do ciclo. Para identificar este tipo de variáveis, elas serão denotadas com o subíndice i no caso em que o estado do sistema ao início do ciclo seja o estado e_i . Por exemplo a variável F^* realmente deve ser escrita como $F_i^*(D)$, já que ela depende do estado do sistema ao início do ciclo.

Com o objetivo de simplificar a análise do modelo, são feitas as seguintes suposições: nos intervalos de *Checkpointing* e disponibilidade as falhas acontecem com taxa constante igual a γ_c e γ_d respectivamente; nas operações de RR não acontecem falhas, isto implica que $F_i(D) = F_i^*(D)$; durante os períodos de disponibilidade, os tempos entre chegadas de transações ao sistema são v.a. exponencialmente distribuídas, com parâmetro λ ; durante as operações de CRR não são permitidas as chegadas de transações ao sistema; nos intervalos de disponibilidade o tempo de serviço das transações é uma v.a. exponencialmente distribuída com parâmetro μ ; o sistema tem a capacidade de armazenar como máximo N transações, incluindo aquela que está em serviço; o tempo necessário para efetuar um *Checkpointing* dado que não acontecem erros (C^*) é constante; Esta última hipótese é razoável já que o *Checkpointing* corresponde a fazer uma gravação de todos os arquivos do sistema e apesar de que a informação contidas neles varia, esta variação é percentualmente pequena devido ao grande volume de informação do Banco de Dados. Por este mesmo motivo supõe-se que o tempo necessário para efetuar o Retorno é constante, em consequência no restante deste capítulo este último tempo será denotado com o símbolo t_{rb}^* .

Por último, para avaliar as diferentes medidas de interesse usar-se-á a técnica de randomização descrita anteriormente. A seguir iniciamos a análise calculando o valor esperado da disponibilidade.

V.4 Avaliação da Disponibilidade Média

Seja $A_i(D)$ a v.a. correspondente à fração de tempo que o sistema está disponível (disponibilidade do sistema) para o atendimento de usuários, dado que o tempo total disponível no ciclo é igual a D e o estado do sistema no início do ciclo é e_i . O objetivo desta seção é avaliar $E[A_i(D)]$, o valor médio estacionário da disponibilidade em um ciclo.

Os instantes de início de ciclo nos quais o sistema está no estado e_i constituem pontos de renovação do sistema. Portanto, para avaliar $E[A_i(D)]$ usam-se os resultados da teoria de renovação com recompensas [28], a qual estabelece que:

$$\begin{aligned} E[A_i(D)] &= \frac{E[D]}{E[T]} \\ &= \frac{E[D]}{E[C] + E[D] + E[F_i^*(D)]} \end{aligned} \quad (V.1)$$

O modelo especifica que $E[D]$ é constante, com valor igual a D . Portanto:

$$E[A_i(D)] = \frac{D}{E[C] + D + E[F_i^*(D)]} \quad (\text{V.2})$$

Isto implica que para avaliar $E[A_i(D)]$ precisa-se calcular $E[C]$ e $E[F_i^*(D)]$, o que será feito a seguir.

V.4.1 Avaliação do Tempo Médio de *Checkpointing*

Tal como já foi discutido, cada vez que acontece uma falha durante uma operação de *Checkpointing*, esta deve ser reinicializada. Portanto, condicionando no instante τ da primeira falha depois de ter iniciado o *Checkpoint*, temos que:

$$E[C] = C^* e^{-\gamma_c C^*} + \int_{\tau=0}^{C^*} \gamma_c e^{-\gamma_c \tau} (\tau + E[C]) d\tau \quad (\text{V.3})$$

O primeiro termo do lado direito da equação V.3 corresponde ao caso que não acontecem falhas durante o *Checkpointing*. O segundo termo corresponde ao caso em que depois de ter iniciado o *Checkpoint*, acontece ao menos uma falha. Neste caso o tempo total para efetuar o *Checkpointing* é igual ao tempo transcorrido até acontecer a primeira falha (τ), mais o tempo necessário para efetuar o *Checkpointing* depois desta falha. Este último tempo corresponde exatamente a $E[C]$, já que depois da falha o sistema está na mesma situação que tinha em $t = 0$.

Resolvendo a integral da equação V.3, chega-se a que:

$$E[C] = C^* e^{-\gamma_c C^*} + \frac{1}{\gamma_c} - e^{-\gamma_c C^*} \left(\frac{1}{\gamma_c} + C^* \right) + E[C] (1 - e^{-\gamma_c C^*})$$

De onde conclui-se que:

$$E[C] = \frac{e^{\gamma_c C^*} - 1}{\gamma_c} \quad (\text{V.4})$$

É interessante notar que no caso em que não podem acontecer falhas durante o *Checkpointing*, ou seja $\gamma_c = 0$, o valor de $E[C]$ é igual a C^* , como era de se esperar. Este fato se deduz facilmente de V.4 tomando o limite quando $\gamma_c \rightarrow 0$.

V.4.2 Avaliação de $E[F_i^*(D)]$.

Para avaliar $E[F_i^*(D)]$ condiciona-se no número n de falhas que acontecem durante o tempo que o sistema está disponível, com o qual obtem-se:

$$E[F_i^*(D)] = \sum_{n=0}^{\infty} E[F_i^*(D)|N(t) = n]P[N(t) = n] \quad (\text{V.5})$$

Para calcular $E[F_i^*(D)|n]$ usa-se a hipótese de que as falhas acontecem segundo um processo de Poisson independente do estado do sistema. Neste caso pode-se usar o seguinte teorema [18]: dado que no intervalo $[0, D]$ acontecem n eventos do processo de Poisson, então a distribuição conjunta dos instantes destes eventos é igual à distribuição conjunta das estatísticas de ordem de n variáveis aleatórias independentes uniformes em $[0, D]$. Seja $F_{i,k}'^n(D)$ o tempo acumulado pelo sistema atendendo transações no intervalo $[0, U_{(k)}^n(D)]$, no caso em que o estado inicial é e_i , e $U_{(k)}^n(D)$ é a k -ésima estatística de ordem de n v.a's uniformes em $[0, D]$, então o teorema anterior implica que:

$$E[F_i^*(D)] = E[F_{i,1}'^n(D) + F_{i,2}'^n(D) + \dots + F_{i,n}'^n(D)] \quad (\text{V.6})$$

Devido à definição de $F_{i,k}'^n(D)$, $1 \leq k \leq n$, e sua relação com $U_{(k)}^n(D)$:

$$E[F_{i,1}'^n(D) + F_{i,2}'^n(D) + \dots + F_{i,n}'^n(D)] = nE[F_{i,1}'^1(D)] \quad (\text{V.7})$$

Sustituindo V.6 e V.7 em V.5:

$$\begin{aligned} E[F_i^*(D)] &= \sum_{n=0}^{\infty} nE[F_{i,1}'^1(D)]P[N(t) = n] \\ &= E[F_{i,1}'^1(D)] \sum_{n=0}^{\infty} nP[N(t) = n] \end{aligned} \quad (\text{V.8})$$

O somatório de acima corresponde ao valor médio do processo de Poisson, isto é:

$$\sum_{n=0}^{\infty} nP[N(t) = n] = \gamma_d D \quad (\text{V.9})$$

Usando V.9 em V.8:

$$E[F_i^*(D)|n] = \gamma_d D E[F_{i,1}'^1(D)] \quad (\text{V.10})$$

Para avaliar $E[F_{i,1}'^1(D)]$, condiciona-se no instante τ (do período de disponibilidade) no qual acontece a falha, com o qual obtem-se que:

$$E[F_{i,1}'(D)] = \int_{\tau=0}^D E[F_{i,1}'(D)|\tau] P[\text{instante da falha} = \tau] d\tau \quad (\text{V.11})$$

onde $E[F_{i,1}'(D)|\tau]$ corresponde à $E[F_{i,1}'(D)]$ para o caso em que a única falha acontece no instante $\tau \in [0, D]$. Agora, usando o fato que o instante de acontecimento da falha é uma v.a. uniformemente distribuída no intervalo $[0, D]$, a equação V.11 fica:

$$E[F_{i,1}'(D)] = \frac{1}{D} \int_{\tau=0}^D E[F_{i,1}'(D)|\tau] d\tau \quad (\text{V.12})$$

Como os tempos de RR estão sendo modelados em forma exata, $E[F_{i,1}'(D)|\tau]$ corresponde ao tempo t_{rb}^* necessário para efetuar a operação de Retorno, mais o tempo total acumulado pelo sistema atendendo transações no intervalo $[0, \tau]$ do período de disponibilidade, (este último tempo será denotado t_{br}), isto é:

$$E[F_{i,1}'(D)|\tau] = E[t_{rb}^* + t_{br}] \quad (\text{V.13})$$

Como t_{rb}^* é constante, e independente de t_{br} , conclue-se que:

$$E[F_{i,1}'(D)|\tau] = t_{rb}^* + E[t_{br}] \quad (\text{V.14})$$

Por definição, t_{br} corresponde ao tempo total acumulado pelo sistema no conjunto de estados $\in \mathcal{U}$ (i.e, estados onde o sistema executa trabalho útil) no intervalo $[0, \tau]$ do período de disponibilidade. Acontece que esta medida foi calculada anteriormente em [6]. Na equação 5.5 dessa referência estabelece-se (usando a notação deste escrito) que:

$$E[t_{br}] = \sum_{m=0}^{\infty} e^{-\Lambda\tau} \frac{\tau^{m+1} \Lambda^m}{m!} \sum_{l=1}^{m+1} \frac{l}{m+1} \Omega_i[m, l] \quad (\text{V.15})$$

Onde Λ é a taxa de randomização, e $\Omega_i[m, l]$ corresponde à probabilidade que a CM randomizada transite por l estados $\in \mathcal{U}$ no caso em que acontecem m transições e o estado inicial é o estado e_i . $\Omega_i[m, l]$ é calculado recorrentemente usando as seguintes equações [6]:

$$\Omega_i[m, l] = \sum_{j=1}^{|E|} p_{ij} \Omega_j[m-1, l-1], \quad \text{se} \quad e_i \in \mathcal{U} \quad (\text{V.16})$$

$$\Omega_i[m, l] = \sum_{j=1}^{|E|} p_{ij} \Omega_j[m-1, l], \quad \text{se } e_i \in \mathcal{I} \quad (\text{V.17})$$

Onde \mathbf{P} é a matriz de transição da CM randomizada, e $p_{ij} \in \mathbf{P}$ corresponde à probabilidade de transição em um passo do estado e_i ao estado e_j . As condições iniciais para $\Omega_i[m, l]$ são [6]:

$$\Omega_i[1, 0] = \begin{cases} 1, & \text{se } e_i \in \mathcal{U} \\ 0, & \text{se } e_i \in \mathcal{I} \end{cases} \quad (\text{V.18})$$

$$\Omega_i[0, 0] = \begin{cases} 0, & \text{se } e_i \in \mathcal{U} \\ 1, & \text{se } e_i \in \mathcal{I} \end{cases} \quad (\text{V.19})$$

Sustituindo V.14 e V.15 em V.12, obtem-se:

$$E[F_{i,1}'(D)] = t_{rb}^* + \frac{1}{\Lambda^2 D} \sum_{m=0}^{\infty} \sum_{l=1}^{m+1} l \Omega_i[m, l] \int_{\tau=0}^D \Lambda e^{-\Lambda\tau} \frac{(\Lambda\tau)^{m+1}}{(m+1)!} d\tau \quad (\text{V.20})$$

A expressão dentro da integral corresponde à função de densidade de probabilidade de uma v.a. Erlang com parâmetros $(\Lambda\tau, m+1)$. Portanto a integral corresponde à função de distribuição, no instante D da v.a. Erlang. Sustituindo na equação V.20 a expressão da função de distribuição da v.a. Erlang [28]:

$$E[F_i^*(D)] = t_{rb}^* + \frac{1}{\Lambda^2 D} \sum_{m=0}^{\infty} \sum_{l=1}^{m+1} l \Omega_i[m, l] \left\{ 1 - e^{-\Lambda D} \sum_{i=0}^m \frac{(\Lambda D)^i}{i!} \right\} \quad (\text{V.21})$$

Agora, substituindo V.21 e V.9 em V.8, obtem-se:

$$E[F_i^*] = \gamma_d D t_{rb}^* + \frac{\gamma_d}{\Lambda^2} \sum_{m=0}^{\infty} \left\{ 1 - e^{-\Lambda D} \sum_{i=0}^m \frac{(\Lambda D)^i}{i!} \right\} \sum_{l=1}^{m+1} l \Omega_i[m, l] \quad (\text{V.22})$$

Por último, substituindo as equações V.4 e V.22 na equação V.2:

$$\begin{aligned} E[A_i(D)] &= \gamma_c D \left[e^{\gamma_c D} - 1 + (1 + \gamma_d t_{rb}^*) \gamma_c D \right. \\ &\quad \left. + \frac{\gamma_c \gamma_d}{\Lambda^2} \sum_{m=0}^{\infty} \left\{ 1 - e^{-\Lambda D} \sum_{i=0}^m \frac{(\Lambda D)^i}{i!} \right\} \sum_{l=1}^{m+1} l \Omega_i[m, l] \right]^{-1} \end{aligned} \quad (\text{V.23})$$

Onde $\Omega_i[m, l]$ é calculado em forma recorrente usando as equações V.16...V.19. A equação V.23 corresponde à solução do problema.

V.5 Disponibilidade média para uma variante do Modelo.

Uma possível variante do modelo recém analisado é supor que durante o *Checkpoint* **não podem** acontecer falhas, porém durante esta operação podem chegar novas transações, com taxa constante $= \lambda$.

A análise deste caso é essencialmente a mesma que a feita anteriormente, porém é necessário fazer as seguintes modificações:

Em primeiro lugar a equação V.4 deve ser trocada por:

$$E[C] = C^* \quad (\text{V.24})$$

Por outro lado, para avaliar $E[F_i^*(D)]$ condiciona-se no estado em que fica o sistema logo após o *Checkpoint*. Condicionando e descondicionando:

$$E[F_i^*(D)] = \sum_{e_j \in E} P[\chi_1(C^*) = e_j | \chi_1(0) = e_i] E[F_j^*(D)] \quad (\text{V.25})$$

Onde $\chi_1(t)$ corresponde ao estado do sistema no instante t , medido a partir do início do ciclo e $F_j^*(D)$ corresponde ao tempo total gasto pelo sistema em operações de RR dado que ao final do *Checkpoint* ele fica no estado e_j .

Considerando que o problema permanece essencialmente o mesmo, para avaliar $E[F_j^*(D)]$, evidentemente usam-se as mesmas equações que foram obtidas para o cálculo de $E[F_i^*(D)]$, só que neste caso é necessário usar o estado e_j como estado inicial.

A probabilidade do sistema encontrar-se no estado e_j após o *Checkpoint* dado que o sistema estava em e_i no instante inicial (isto é $P[\chi_1(C^*) = e_j | \chi_1(0) = e_i]$), é calculada condicionando no número de transações submetidas ao sistema durante esta operação, da seguinte forma:

$$P[\chi_1(C^*) = e_j | \chi_1(0) = e_i] = \begin{cases} 0, & \eta(e_j) < \eta(e_i) \\ e^{-\gamma_c C^*} \frac{(\gamma_c C^*)^l}{l!} p_{ij}^l, & \eta(e_j) = \eta(e_i) + l < N \\ \sum_{l=N-\eta(e_i)}^{\infty} e^{-\gamma_c C^*} \frac{(\gamma_c C^*)^l}{l!} p_{ij}^l, & \eta(e_i) < \eta(e_j) = N \end{cases} \quad (\text{V.26})$$

Onde, segundo ja foi definido, $\eta(e_i)$, é uma função que entrega o número de transações no sistema quando este se encontra no estado e_i , e p_{ij}^l corresponde à probabilidade de transitar do estado e_i ao estado e_j em l transições da CM randomizada. O valor de p_{ij}^l é o i, j -ésimo componente da matriz P^l , logo [19]:

$$p_{ij}^l = \begin{cases} p_{ij}, & l = 1 \\ \sum_{e_k \in E} p_{ik} p_{kj}^{l-1}, & l > 1 \end{cases} \quad (\text{V.27})$$

Por último, substituindo as equações V.24 . . . V.26 na equação V.2, chega-se a que:

$$E[A_i(D)] = \frac{D}{C^* + D + \sum_{e_j \in E} P[\chi_1(C^*) = e_j | \chi_1(0) = e_i] E[F_j^*(D)]} \quad (\text{V.28})$$

Onde $P[\chi_1(C^*) = e_j | \chi_1(0) = e_i]$ e $E[F_j^*(D)]$ são avaliados usando as equações V.26, V.27 e V.6 . . . V.22 respectivamente.

V.6 Avaliação da Função de Distribuição da Disponibilidade

O objetivo desta seção é avaliar $FA_i(D, a)$, que corresponde à função de distribuição da disponibilidade de um sistema de banco de dados submetido a política de CRR. Mais especificamente:

$$FA_i(D, a) = P[A_i(D) \leq a] \quad (\text{V.29})$$

Onde a é um valor real não negativo.

O sistema será representado pelo modelo 2 da seção V.5, com as restrições adicionais de que $t_{rb}^* = 0$ e que nas operações de RR não podem acontecer falhas.

Com o intuito de facilitar a leitura, a seguir reescrevem-se as suposições do modelo empregado: os tempos entre *Checkpoints* estão dados pela política C2, os tempos de recuperação correspondem ao modelo R3, o tempo de Retorno supõe-se zero ($t_{rb}^* = 0$), durante as operações de *Checkpointing* não podem acontecer falhas ($C = C^*$), durante os períodos de disponibilidade as falhas acontecem com taxa constante γ_d , nas operações de RR não acontecem falhas ($F_i = F_i^*$), durante os períodos de disponibilidade as transações chegam e são atendidas com taxa constante igual a λ e μ respectivamente, o sistema tem capacidade de armazenar como máximo N transações (incluindo aquele que está em serviço), o tempo necessário para efetuar um *Checkpoint* é constante ($C^* = \text{constante}$) e nos intervalos de tempo em que o sistema está disponível a evolução dele é modelada por uma cadeia de Markov homogênea de tempo contínuo.

Para avaliar $FA_i(D, a)$, em primeiro lugar condiciona-se no estado em que fica o sistema logo após o *Checkpoint*. Portanto, análogamente à equação V.25 tem-se que:

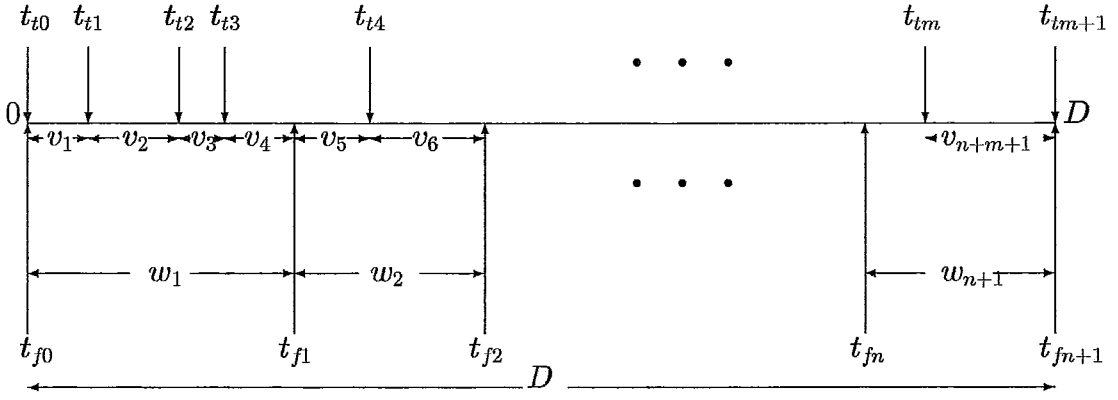


Figura V.3: Intervalo de tempo disponível (D)

$$FA_i(D, a) = \sum_{e_j \in E} P[\chi_1(C^*) = e_j | \chi_1(0) = e_i] FA_j(D, a) \quad (\text{V.30})$$

Onde $FA_j(D, a)$ corresponde à FDP da disponibilidade dado que o estado do sistema ao final do *Checkpoint* ($\chi_1(C^*)$) é o estado e_j . A probabilidade de transição entre estados durante o *Checkpoint* ($P[\chi_1(C^*) = e_j | \chi_1(0) = e_i]$) foi calculada na equação V.26.

Para simplificar as explicações, a partir deste ponto, o símbolo D (ver figura V.3) será usado para denotar tanto o intervalo de tempo em que o sistema está disponível para atendimento a usuários, quanto o seu comprimento, (este pequeno abuso de notação não deve causar confusões). Lembra-se também que no intervalo D não se consideram os tempos usados em operações de CRR.

Seja t_{f_i} e t_{i_j} os instantes de tempo $\in D$ nos quais acontecem a i -ésima falha do sistema e a j -ésima transição da cadeia de Markov respectivamente. Seja $\mathcal{N}(D)$ o processo de Poisson independente do estado do sistema com taxa γ_d , correspondente ao número de falhas do sistema no intervalo ID. Seja $\mathcal{M}(D)$ o processo de Poisson com taxa Λ independente do estado do sistema, correspondente ao número de transições da CM randomizada no intervalo D . Então define-se o processo de Poisson $\mathcal{W}(D)$, com taxa $\nu = \gamma_d + \Lambda$, como a composição dos processos $\mathcal{N}(D)$ e $\mathcal{M}(D)$. Para definir $\mathcal{W}(D)$ foi usado o conhecido fato de que a composição de dois processos de Poisson é outro processo de Poisson com taxa igual à soma das taxas dos processos componentes [28].

Para avaliar $FA_j(D, a)$, em primeiro lugar condiciona-se no número de eventos do processo $\mathcal{W}(D)$, e aplica-se o teorema de probabilidades totais, de onde tem-se que:

$$FA_j(D, a) = \sum_{z=0}^{\infty} e^{-\nu D} \frac{(\nu D)^z}{z!} FA_j(D, a | \mathcal{W}(D) = z) \quad (\text{V.31})$$

Agora condicionando e descondicionando no valor de $\mathcal{N}(D)$:

$$FA_j(D, a) = \sum_{z=0}^{\infty} e^{-\nu D} \frac{(\nu D)^z}{z!} \sum_{n=0}^z P[\mathcal{N}(D) = n | \mathcal{W}(D) = z] FA_j(D, a | \mathcal{W}(D) = z, \mathcal{N}(D) = n) \quad (\text{V.32})$$

É sabido o fato que [28]:

$$P[\mathcal{N}(D) = n | \mathcal{W}(D) = z] = \binom{z}{n} \left(\frac{\gamma_d}{\gamma_d + \Lambda} \right)^n \left(\frac{\Lambda}{\gamma_d + \Lambda} \right)^{z-n}, \quad z \geq n \quad (\text{V.33})$$

Adicionalmente, como $\mathcal{W}(D)$ corresponde à composição dos processos $\mathcal{N}(D)$ e $\mathcal{M}(D)$, evidentemente:

$$FA_j(D, a | \mathcal{W}(D) = z, \mathcal{N}(D) = n) = FA_j(D, a | \mathcal{N}(D) = n, \mathcal{M}(D) = m) \quad (\text{V.34})$$

Com $m = z - n$. Para simplificar a notação, $FA_j(D, a | \mathcal{N}(D) = n, \mathcal{M}(D) = z - n)$ será escrito como $FA_j(D, L | n, m)$.

Ao substituir V.32, V.33 e V.34 em V.31, obtém-se:

$$FA_j(D, a) = \sum_{z=0}^{\infty} e^{-\nu D} \frac{(\nu D)^z}{z!} \sum_{n=0}^z \binom{z}{n} \left(\frac{\gamma_d}{\gamma_d + \Lambda} \right)^n \left(\frac{\Lambda}{\gamma_d + \Lambda} \right)^{z-n} FA_j(D, a | n, m) \quad (\text{V.35})$$

A disponibilidade do sistema é dada pelo tempo total em que o sistema encontra-se operacional num ciclo, dividido pelo tempo total do ciclo; portanto:

$$\begin{aligned} FA_j(D, a | n, m) &= P[A_j(D) \leq a] \\ &= P \left[\frac{D}{C^* + D + F_j^*(n, m)} \leq a \right] \end{aligned} \quad (\text{V.36})$$

Onde $F_j^*(n, m)$ corresponde ao tempo total gasto pelo sistema em operações de RR dado que: $\chi_1(C^*) = e_j$; nas operações de RR não podem acontecer falhas; $t_{rb}^* = 0$; $\mathcal{N}(D) = n$; $\mathcal{M}(D) = m$.

De V.36:

$$FA_j(D, a | n, m) = P \left[\frac{D}{a} - C^* - D \leq F_j^*(n, m) \right] \quad (\text{V.37})$$

Seja $L = \frac{D}{a} - C^* - D$. Como C^* , D e a são parâmetros de entrada do sistema; para um conjunto dado destes parâmetros L corresponde a um determinado valor constante. Portanto a equação V.37 pode ser reescrita da forma:

$$FA_j(D, a|n, m) = 1 - P[F_j^*(n, m, D) < L] \quad (\text{V.38})$$

Agora, usando a definição de $F_j^*(n, m)$, chega-se a que: temos que:

$$P[F_j^*(n, m, D) < L] = P\left[\sum_{l=1}^n r_j^{*l}(n, m) < L\right] \quad (\text{V.39})$$

Onde $r_j^{*l}(n, m)$ corresponde ao tempo gasto na l -ésima operação de RR dado que: $\chi_1(C^*) = e_j$, $\mathcal{N}(D) = n$, $\mathcal{M}(D) = m$ e nas operações de RR não podem acontecer falhas.

Seja w_q ($1 \leq q \leq n+1$) o intervalo $(t_{f_{q-1}}, t_{f_q}]$, (ver figura V.3). Seja $\mathcal{B}(v)$ a v.a. correspondente ao tempo acumulado pelo sistema atendendo transações num certo intervalo $v \in D$. Por definição de $r_j^{*l}(n, m)$, tem-se que (ver figura V.2):

$$r_j^{*l}(n, m) = \mathcal{B}(w_1) + \mathcal{B}(w_2) + \dots + \mathcal{B}(w_l) \quad (\text{V.40})$$

Sustituindo V.40 em V.39:

$$P[F_j^*(n, m) < L] = P[\{\mathcal{B}(w_1)\} + \{\mathcal{B}(w_1) + \mathcal{B}(w_2)\} + \dots + \{\mathcal{B}(w_1) + \dots + \mathcal{B}(w_n)\} < L] \quad (\text{V.41})$$

Logo,

$$P[F_j^*(n, m, D) < L] = P\left[\sum_{q=1}^n (n - q + 1)\mathcal{B}(w_q) < L\right] \quad (\text{V.42})$$

A equação V.42 indica que os segmentos de tempo $\in w_q$ ocupados atendendo transações, devem ser reprocessados $(n - q + 1)$ vezes, isto devido às falhas números $q, q+1, \dots, n$ (ver figura V.3).

Seja v_l ($1 \leq l \leq n+m+1$) o intervalo de tempo $\in D$ compreendido entre o $(l-1)$ -ésimo e o l -ésimo evento do processo $\mathcal{W}(D)$ (ver figura V.3). Seja k_q ($1 \leq q \leq n+1$) o número de eventos do processo $\mathcal{M}(D)$ ocorridos no intervalo w_q e seja $K_q = \sum_{l=1}^q k_q$. Em termos destas variáveis a equação V.42 pode ser reescrita da seguinte forma (ver figura V.3):

$$P[F_j^*(n, m, D) < L] = P\left[\sum_{q=1}^n (n - q + 1)\mathcal{B}\left(\sum_{l=K_{q-1}+1}^{K_q} v_l\right) < L\right] \quad (\text{V.43})$$

Como $v_l \cap v_h = \phi$, no caso em que $l \neq h$:

$$P[F_j^*(n, m, D) < L] = P \left[\sum_{q=1}^n (n - q + 1) \sum_{l=K_{q-1}+1}^{K_q} \mathcal{B}(v_l) < L \right] \quad (\text{V.44})$$

Por definição da v.a. $\mathcal{B}(v_l)$, cumpre-se que $\mathcal{B}(v_l) = 0$ no caso em que o sistema não atende usuários no intervalo $v_l \in D$; e $\mathcal{B}(v_l) = v_l$ em caso contrário, portanto:

$$\mathcal{B}(v_l) = v_l I[\chi_2(v_l) \in \mathcal{U}] \quad (\text{V.45})$$

Onde a função $I[x]$ vale 1 se o evento x for verdadeiro e vale 0 em caso contrário; e o evento $[\chi_2(v_l) \in \mathcal{U}]$ é verdadeiro se no segmento v_l o sistema está ocupado atendendo transações.

Para cada estado $e_k \in E$ atribue-se uma taxa de recompensa, dada pela seguinte equação:

$$\rho^{(n,m)}(e_k, v_l) = (n - q + 1) I[e_k \in \mathcal{U}], \quad \text{onde } v_l \in w_q \quad (\text{V.46})$$

$1 \leq l \leq n+m+1$

A taxa de recompensa atribuída aos estados do sistema segundo a equação V.46, corresponde a uma função de recompensa não homogênea, isto é: a recompensa atribuída a um certo estado varia no tempo, já que ela depende do intervalo v_l ($1 \leq l \leq n + m + 1$) em que o sistema permanece nesse estado. Outro aspecto interessante da equação V.46 é que o valor $\rho^{(n,m)}(e_k, v_l)$, é um inteiro $\in [0, n]$, pois o valor de q varia na equação V.46 entre 1 e $(n + 1)$. A discussão anterior implica que: as taxas de recompensas atribuídas aos estados $\in \mathcal{U}$ é da seguinte forma: No segmento w_1 os estados $\in \mathcal{V}$ ganham uma taxa igual a n , no seguinte segmento (w_2) ganham uma taxa $(n-1)$, ..., no penúltimo segmento (w_n) ganham uma taxa igual a 1, e no último segmento (w_{n+1}) não ganham recompensa. Por outro lado os estados $\in \mathcal{I}$ em nenhum caso ganham recompensa.

Segundo a equação V.44, as taxas de recompensas atribuídas aos estados são necessárias para calcular o tempo total ($F_j^*(n, m, D)$) usado em operações de recuperação no intervalo D e podem ser explicados em termos intuitivos da seguinte forma: Será visto, em primeiro lugar, o que acontece com as transações atendidas no intervalo $(0, t_{f1}]$. Estas transações necessariamente são parte de t_{b1} e portanto também devem ser parte de t_{b2}, \dots, t_{bm} (ver figura V.2), o que implica que tais transações devem ser reprocessada n vezes. Portanto: cada unidade de tempo que o sistema passa atendendo usuários no intervalo $[0, t_{f1}]$ deve ser reproduzido n vezes, nas operações de recuperação números $1 \dots n$. Outra forma de dizer o

anterior é que: para o cálculo do tempo total gasto em operações de recuperação cada unidade de tempo útil $\in [0, t_{f_1}]$ deve ser considerada como n unidades de tempo nas operações de recuperação (correspondentes às n diferentes operações de *recover*). Um raciocínio análogo demonstra que transações processadas no intervalo $(t_{f_{q-1}}, t_{f_q}]$ devem ser reprocessadas $(n - q + 1)$ vezes, devido as operações de recuperação correspondentes as falhas ocorridas depois do processamento original destas transações. Agora analisa-se o que acontece com um segmento de tempo útil no meio do qual acontece a q -ésima falha ($1 \leq q \leq n$), (como por exemplo no caso da n -ésima falha da figura V.3). Lembre-se que um segmento corresponde ao processamento de uma ou mais transações em forma consecutiva. Suponha que são conhecidos os tempos de processamento dessas transações (previamente estes tempos são selecionados aleatoriamente de uma v.a. exponencial de parâmetro μ), então um raciocínio análogo ao feito nos casos anteriores demonstra que a parte do segmento previa à falha deve ser reprocessada $(n - q + 1)$ vezes e a parte posterior à falha deve ser reprocessada $(n - q)$ vezes. Isto é verdade inclusive para a transação que estava sendo processada no instante de falha. Por último, os subintervalos de tempo $\in D$ nos quais o sistema não processa transações evidentemente não devem ser considerados nas operações de recuperação, e portanto recebem recompensa 0.

Voltando agora ao cálculo de $P[F_j^*(n, m, D) < L]$, substituímos V.45 e V.46 em equação V.44, de onde:

$$P[F_j^*(n, m, D) < L] = P \left[\sum_{q=1}^{n+1} \sum_{l=K_{q-1}+1}^{K_q} (n - q + 1) v_l I[\chi_2(l) \in \mathcal{U}] < L \right] \quad (V.47)$$

Conceitualmente a equação V.47 indica que $P[F_j^*(n, m, D) < L]$ corresponde à probabilidade que a recompensa total acumulada pelo sistema no intervalo D seja menor a L , dado que a taxa de recompensa atribuída a seus estados corresponde à estabelecida na equação V.46. Isto quer dizer que o problema de avaliar $P[F_j^*(n, m, D) < L]$ é um tipo da área de Desempenhabilidade [5]. É interessante ressaltar que a equação V.47 estabelece um nexso entre as áreas de CRR e Desempenhabilidade, e que o estabelecimento desta relação é uma das contribuições importantes deste trabalho. Para resolver a equação V.47 serão usados três métodos, já que eles foram empregados, anteriormente, com êxito, na área de desempenhoabilidade. O Método 1 foi usado em [5] para avaliar $F_i(t, r)$. O Método 2 corresponde ao Método A do capítulo II e é o método que foi usado em [9] para avaliar $F_i(t, r)$. O Método 3 é o método B do capítulo II, e é original desta tese, e, tal como ficará evidente mais adiante, é o mais conveniente (dentre os três citados) para resolver V.47.

V.6.1 Método 1

Este método é análogo ao aparecido em [5].

Seja $\vec{K}_{n,m}$ um vector de $n + 1$ componentes, da forma $\vec{K}_{n,m} = \langle k_n, k_{n-1}, \dots, k_0 \rangle$. Onde k_c ($0 \leq c \leq n$), corresponde ao número de segmentos v_l ($1 \leq l \leq n + m + 1$) nos quais o sistema ganha recompensa igual a c . De V.46, k_c corresponde ao número de segmentos v_l pertencentes ao intervalo w_{n-c+1} , nos quais o sistema está em estados $\in \mathcal{U}$. Como o número de transições de $\mathcal{W}(D)$ é $n + m$, cumpre-se que $|\vec{K}_{n,m}| = \sum_{c=0}^n k_c = n + m + 1$. Seja $\Omega_j[\vec{K}_{n,m}]$ a probabilidade que o sistema visite k_c ($0 \leq c \leq n$) estados com recompensa c , dadas as seguintes condições: $\mathcal{N}(D) = n$, $\mathcal{M}(D) = m$, e $\chi_1(C^*) = e_j$. Seja $\mathcal{K}_{n,m}$ o conjunto dos possíveis valores de $\vec{K}_{n,m}$. Condicionando (e descondicionando) no valor de $\Omega_j[\vec{K}_{n,m}]$ na equação V.47, obtém-se que:

$$P[F_j^*(n, m, D) < L] = \sum_{\vec{K}_{n,m} \in \mathcal{K}_{n,m}} P \left[\sum_{q=1}^{n+1} \sum_{l=K_{q-1}+1}^{K_q} (n - q + 1) v_l I[\chi_2(l) \in \mathcal{U}] < L | \vec{K}_{n,m} \right] \Omega_j[\vec{K}_{n,m}] \quad (\text{V.48})$$

Para resolver V.48 é necessário calcular $\Omega_j[\vec{K}_{n,m}]$. Isto será feito em forma recorrente, nos valores 'n' e 'm', porém, para fazê-lo, é preciso entender a forma em que o vector $\vec{K}_{n,m}$ pode ser obtido recorrentemente. Isto é o que será feito a seguir.

Considere o primeiro evento do processo de Poisson $\mathcal{W}(D)$. Este evento pode corresponder a uma falha, isto é pode ser um evento do processo $\mathcal{N}(D)$. Neste caso falamos que é um evento de tipo 1. A outra alternativa, é que o primeiro evento de $\mathcal{W}(D)$ corresponda a uma transição da cadeia de Markov randomizada, isto é: pode ser um evento do processo $\mathcal{M}(D)$. Neste caso dizemos que é um evento de tipo 2. Seja $U_{(1)}^1$ a primeira estatística de ordem do processo $\mathcal{N}(D)$ (tipo de processo 1), e seja $U_{(1)}^2$ a primeira estatística de ordem do processo $\mathcal{M}(D)$ (tipo de processo 2). Então, dado que acontecem n eventos do primeiro tipo e m eventos do segundo tipo, é fácil demonstrar que:

$$P[U_{(1)}^1 < U_{(1)}^2] = \frac{n}{n + m} \quad (\text{V.49})$$

Para calcular $\Omega_j[\vec{K}_{n,m}]$ condiciona-se no primeiro evento do processo de Poisson $\mathcal{W}(D)$ segundo V.49, a probabilidade de que este evento seja uma falha é $\frac{n}{n+m}$ e a probabilidade de que seja uma transição da cadeia de Markov randomizada é $\frac{m}{n+m}$.

Por outro lado, caso o primeiro evento de $\mathcal{W}(D) \in \mathcal{N}(D)$, $\vec{K}_{n,m}$ é obtido a partir de $\vec{K}_{n-1,m}$; e caso o primeiro evento de $\mathcal{W}(D) \in \mathcal{M}(D)$, $\vec{K}_{n,m}$ é obtido a partir de $\vec{K}_{n,m-1}$. Para entender melhor como o vector $\vec{K}_{n,m}$ é obtido em forma recorrente, a seguir indica-se como ele é obtido a partir de $\vec{K}_{n-1,m}$ ou $\vec{K}_{n,m-1}$, segundo seja o caso. Para simplificar a explicação, ela será feita em base as figuras V.4 ... V.11. Nelas aparece desenhado o intervalo D e os eventos dos processos $\mathcal{N}(D)$ e $\mathcal{M}(D)$. Os eventos $\in \mathcal{N}(D)$ estão desenhados como setas que

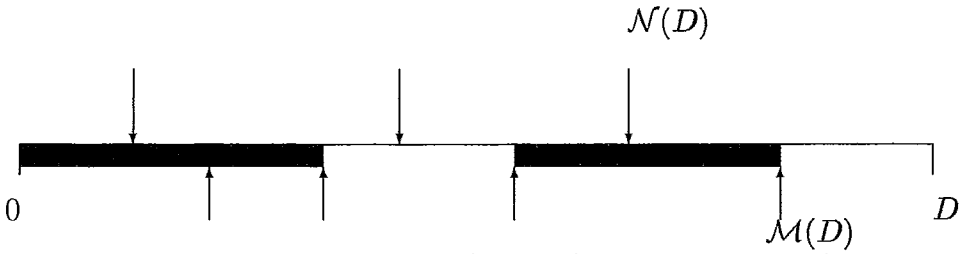


Figura V.4: Relação entre $\vec{K}_{n,m}$ e $\vec{K}_{n-1,m}$ (parte a): $\vec{K}_{3,4} = \langle 1, 2, 1, 4 \rangle$

apontam desde cima a algum instante de D . Analogamente, os eventos $\in \mathcal{M}(D)$ aparecem como setas que apontam desde baixo a algum instante $\in D$. Como antes, os segmentos $\in D$ nos quais o sistema está fazendo trabalho útil, aparecem desenhados com linha grossa, os demais segmentos estão desenhados com linha fina. No título de cada figura aparece o vetor $\vec{K}_{n,m}$ correspondente.

Este vetor é obtido facilmente das figuras, da seguinte forma (ver definição de $\vec{K}_{n,m}$ e equação V.46): o c -ésimo componente ($1 \leq c \leq n-1$) corresponde ao número de segmentos desenhados com linha grossa que se encontram entre os eventos número c e $c+1$ de $\mathcal{N}(D)$ contando de direita a esquerda; $\vec{K}_{n,m}[n]$ corresponde ao número de segmentos com linha grossa à esquerda do primeiro evento de $\mathcal{N}(D)$ (o de mais a esquerda), por último $\vec{K}_{n,m}[0]$ corresponde ao número de segmentos desenhados com linha fina que estão no intervalo $[0, \mathcal{N}(D)]$ mais todos os segmentos à direita de $\mathcal{N}(D)$. A relação entre $\vec{K}_{n,m}$ e $\vec{K}_{n-1,m}$, $\vec{K}_{n,m-1}$, será estabelecida para os diferentes casos possíveis. Estos casos estão dados por duas situações independentes (que dão origem a 4 casos diferentes), que são: 1) se o primeiro evento de $\mathcal{W}(D)$ pertence a $\mathcal{N}(D)$ ou a $\mathcal{M}(D)$; e 2) se o estado inicial pertence a \mathcal{U} ou a \mathcal{I} .

Seja \oplus o operador concatenação entre dois vetores, isto é $\vec{v} = \vec{v}_1 \oplus \vec{v}_2$, indica que o vetor \vec{v} é obtido concatenando \vec{v}_2 ao final de \vec{v}_1 . Sejam as funções I_j e \bar{I}_j definidas como:

$$I_j = \begin{cases} 1, & \text{se } e_j \in \mathcal{U} \\ 0, & \text{se } e_j \in \mathcal{I}; \end{cases} \quad \bar{I}_j = 1 - I_j$$

Seja $\vec{1}_c^{(n)}$ o vetor de $n+1$ componentes (com índices $n \dots 0$), tal que todas elas são 0, excepto a c -ésima ($0 \leq c \leq n$) que vale 1.

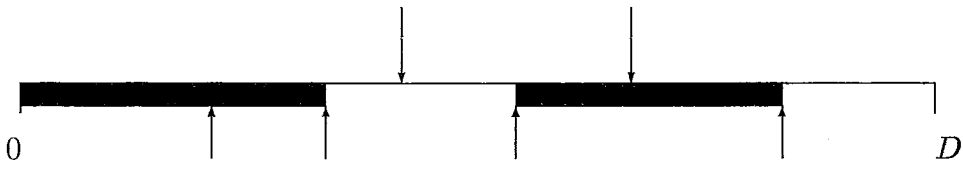


Figura V.5: Relação entre $\vec{K}_{n,m}$ e $\vec{K}_{n-1,m}$ (parte b): $\vec{K}_{2,4} = \langle 2, 1, 4 \rangle$

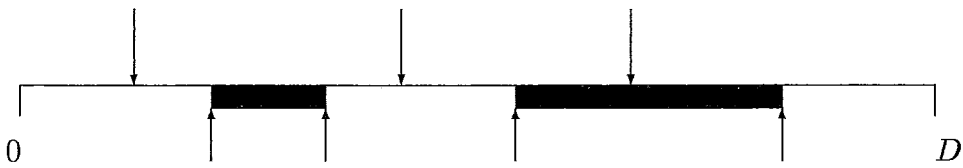


Figura V.6: Relação entre $\vec{K}_{n,m}$ e $\vec{K}_{n-1,m}$ (parte c): $\vec{K}_{3,4} = \langle 0, 1, 1, 6 \rangle$

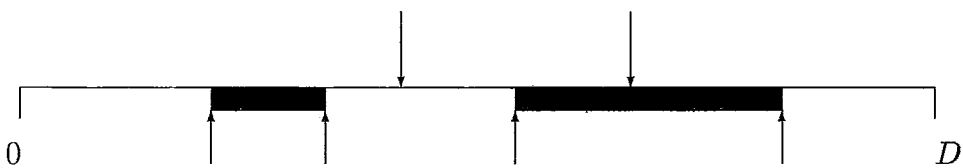


Figura V.7: Relação entre $\vec{K}_{n,m}$ e $\vec{K}_{n-1,m}$ (parte d): $\vec{K}_{2,4} = \langle 1, 1, 5 \rangle$

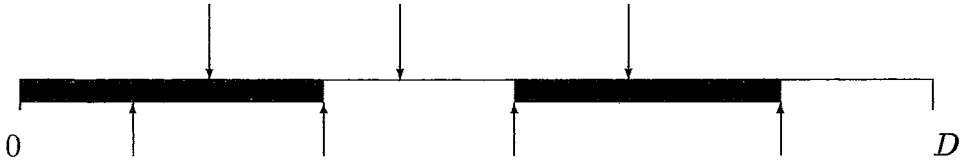


Figura V.8: Relação entre $\vec{K}_{n,m}$ e $\vec{K}_{n,m-1}$ (parte e): $\vec{K}_{3,4} = \langle 2, 1, 1, 4 \rangle$

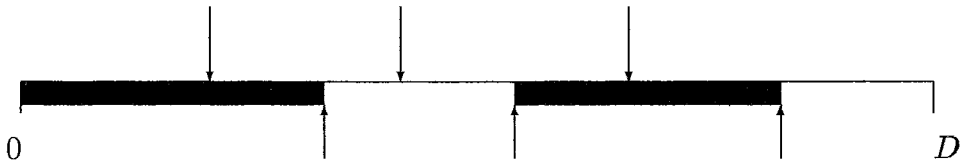


Figura V.9: Relação entre $\vec{K}_{n,m}$ e $\vec{K}_{n,m-1}$ (parte f): $\vec{K}_{3,3} = \langle 1, 1, 1, 4 \rangle$

A figura V.4 corresponde ao caso em que o primeiro evento de $\mathcal{W}(D)$ pertence a $\mathcal{N}(D)$, e o estado inicial pertence a \mathcal{U} , por outro lado, a figura V.5 corresponde à V.4, porém com o primeiro evento de $\mathcal{W}(D)$ eliminado. Ambas figuras ilustram a relação entre $\vec{K}_{n,m}$ e $\vec{K}_{n-1,m}$, que para este caso está dada por:

$$\vec{K}_{n,m} = \langle 1 \rangle \oplus \vec{K}_{n-1,m}, \quad \text{se } e_j \in \mathcal{U} \quad (\text{V.50})$$

Analogamente, as figuras V.6 e V.7 exemplificam o mesmo caso anterior, porém quando o primeiro estado pertence a \mathcal{I} . Neste caso:

$$\vec{K}_{n,m} = \langle 0 \rangle \oplus \left(\vec{K}_{n-1,m} + \vec{1}_0^{(n-1)} \right), \quad \text{se } e_j \in \mathcal{I} \quad (\text{V.51})$$

A figura V.8 corresponde ao caso em que o primeiro evento de $\mathcal{W}(D)$ pertence a $\mathcal{M}(D)$, e o estado inicial pertence a \mathcal{U} por outro lado, a figura V.9 é idêntica a V.8, exceto que foi tirado o primeiro evento de $\mathcal{W}(D)$. De ambas as figuras:

$$\vec{K}_{n,m} = \vec{K}_{n,m-1} + \vec{1}_n^{(n)}, \quad \text{se } e_j \in \mathcal{U} \quad (\text{V.52})$$

A figura V.10 corresponde ao caso em que o primeiro evento de $\mathcal{W}(D)$ pertence a $\mathcal{M}(D)$ e o estado inicial pertence a \mathcal{I} por outro lado, a figura V.11 é idêntica a V.10, excepto que foi tirado o primeiro evento de $\mathcal{W}(D)$. De ambas as figuras:

$$\vec{K}_{n,m} = \vec{K}_{n,m-1} + \vec{1}_0^{(n)}, \quad \text{se } e_j \in \mathcal{I} \quad (\text{V.53})$$

Seja I_p e \bar{I}_p as funções índices definidas como:

$$I_p = \begin{cases} 1, & \text{se o primeiro evento de } \mathcal{W}(D) \in \mathcal{N}(D) \\ 0 & \text{se o primeiro evento de } \mathcal{W}(D) \in \mathcal{M}(D) \end{cases}$$

$$\bar{I}_p = 1 - I_p$$



Figura V.10: Relação entre $\vec{K}_{n,m}$ e $\vec{K}_{n,m-1}$ (parte g): $\vec{K}_{3,4} = \langle 1, 1, 1, 5 \rangle$

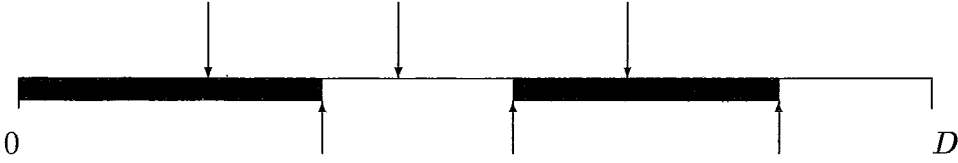


Figura V.11: Relação entre $\vec{K}_{n,m}$ e $\vec{K}_{n,m-1}$ (parte h): $\vec{K}_{3,3} = \langle 1, 1, 1, 4 \rangle$

As equações V.50 ... V.53 podem ser reescritas em forma mais resumida da seguinte forma:

$$\vec{K}_{n,m} = \begin{cases} I_p \left(\langle I_j \rangle \oplus \left(\vec{K}_{n-1,m} + \bar{I}_j \bar{I}_0^{(n-1)} \right) \right) \\ \bar{I}_j \bar{I}_p \left(\vec{K}_{n,m-1} + \bar{I}_0^{(n)} \right) + I_j \bar{I}_p \left(\vec{K}_{n,m-1} + \bar{I}_n^{(n)} \right) \end{cases} \quad (\text{V.54})$$

Voltando agora ao cálculo de $\Omega_j[\vec{K}_{n,m}]$, condiciona-se (e descondiciona-se) no tipo do primeiro evento de $\mathcal{W}(D)$, e usa-se V.54, de onde:

$$\begin{aligned} \Omega_j[\vec{K}_{n,m}] &= \frac{n}{n+m} \Omega_j[\vec{K}_{n-1,m}] \\ &+ \frac{m}{n+m} \sum_{e_k \in E} p_{jk} \left\{ \bar{I}_j \Omega_k[\vec{K}_{n,m} - \bar{I}_0^{(n)}] + I_j \Omega_k[\vec{K}_{n,m} - \bar{I}_n^{(n)}] \right\} \end{aligned} \quad (\text{V.55})$$

Note que os fatores $\frac{n}{n+m}$ e $\frac{m}{n+m}$ determinam os casos I_p e \bar{I}_p respectivamente. Isto será importante mais adiante na solução de V.48.

As condições iniciais de $\Omega_j[\vec{K}_{n,m}]$ estão dadas por:

$$\Omega_j[\vec{K}_{0,m}, 0, m] = I \left[\vec{K}_{0,m} = (m+1) \bar{I}_0^{(0)} \right], \quad \text{para todo } e_j \in E \quad (\text{V.56})$$

Voltando agora à solução da equação V.48; para calcular $P \left[\sum_{q=1}^{n+1} \sum_{K_{q-1}+1}^{K_q} I[\chi_2(l) \in \mathcal{U}] (n-q+1) v_l < L | \vec{K}_{n,m} \right]$ (segundo o Método 1) proceda-se analogamente a [5]. Para o qual nota-se que o termo entre chaves da equação V.48 corresponde ao lado direito da terceira igualdade da equação II.6, (no caso em que $c_l = (n-q+1) I[\chi_2(l) \in \mathcal{U}]$, $1 \leq q \leq n+1$). Isto quer dizer que este termo corresponde à combinação linear das estatísticas de ordem de $(n+m)$ v.a's uniformes em $[0, D]$. Então, para cada valor do vetor $\vec{K}_{n,m}$ (que indica o número de segmentos em cada recompensa), calcula-se uma combinação linear de $n+m$ v.a's Dirichlet (ou equivalentemente a CLEO de $n+m$ v.a's uniformes, ver equação II.6), da forma: $\vec{K}_{n,m}[n]$ variáveis com recompensa n , $\vec{K}_{n,m}[n-1]$ variáveis com recompensa $n-1$, ..., $\vec{K}_{n,m}[0]$ variáveis com recompensa 0. Portanto, análogamente a

[5] pode-se calcular esta combinação linear fazendo uso do resultado de [32] (outra alternativa seria usar, por exemplo, o resultado do capítulo II deste trabalho).

Embora seja simples calcular a combinação linear de estatísticas de ordem mencionada acima, o custo computacional associado ao Método 1 pode ser muito grande, devido ao número de termos $\Omega_j[\vec{K}_{n,m}]$ diferentes. Este número corresponde à cardinalidade do conjunto $\mathcal{K}_{n,m}$, que está dada por todos os valores diferentes possíveis que pode assumir o vetor $\vec{K}_{n,m}$, sujeito à restrição: $\sum_{c=0}^n k_c = n + m + 1$. Um breve raciocínio nos diz que $|\mathcal{K}_{n,m}| = O\left(\binom{n+m}{n}\right)$, isto é: a cardinalidade de $\mathcal{K}_{n,m}$ tem um custo combinacional, o que representa um custo muito alto. Entretanto como a probabilidade do número de falhas ser ‘grande’ (isto é maior que 2) no intervalo considerado é pequena, para uma determinada tolerância podemos truncar o valor de n para um valor baixo (por ex.: $n = 2, 3$), e portanto o custo total será da ordem

$$\binom{n+m}{n} = \frac{(n+m+1) \dots (m+1)}{n!} \approx \frac{m^n}{n!}$$

se $m \gg n$. Note que, devido à randomização, $m \approx \Lambda D$, onde Λ é a taxa de randomização da cadeia de Markov.

Devido ao alto custo computacional associado à solução entregue pelo Método 1, a seguir será usado o Método 2, com o objetivo de avaliar $P[F_j^*(n, m) < L]$ em forma mais eficiente.

V.6.2 Método 2

Este método corresponde ao Método A do capítulo II, e tal como foi indicado nesse capítulo, foi extraído de [9]. Com o objetivo de sintetizar a exposição serão usadas as explicações dadas no capítulo II.

Em primeiro lugar a equação V.48, no caso $n = m = 0$, implica que:

$$P[F_j^*(0, 0, D) < L] = P[\rho^{(0,0)}(e_j, v_1)v_1 < L] \quad (\text{V.57})$$

Porém neste caso $v_1 = D$, e portanto:

$$\begin{aligned} P[F_j^*(0, 0, D) < L] &= P[\rho^{(0,0)}(e_j, v_1)D < L] \\ &= u[L - \rho^{(0,0)}(e_j, v_1)D] \end{aligned} \quad (\text{V.58})$$

Para avaliar $P[F_j^*(n, m, D) < L]$ no caso em que $n+m \geq 1$, condiciona-se no instante τ de acontecimento do primeiro evento do processo $\mathcal{W}(D)$, com o qual:

$$P[F_j^*(n, m, D) < L] = \int_0^D \frac{(n+m)(D-\tau)^{n+m-1}}{D^{n+m}} \left\{ \frac{n}{n+m} P[F_j^*(n-1, m, D-\tau) < L - \tau\rho^{(n,m)}(e_j, v_1)] + \frac{m}{n+m} \sum_{e_k \in E} p_{jk} P[F_k^*(n, m-1, D-\tau) < L - \tau\rho^{(n,m)}(e_j, v_1)] \right\} d\tau \quad (\text{V.59})$$

Onde:

$\frac{(n+m)(D-\tau)^{n+m-1}}{D^{n+m}}$: função de densidade de probabilidade do tempo $\tau \in [0, D]$ do primeiro evento de $\mathcal{W}(D)$.

$\frac{n}{n+m}$: Probabilidade que o primeiro evento de $\mathcal{W}(D)$ seja uma falha.

$\frac{m}{n+m}$: Probabilidade que o primeiro evento de $\mathcal{W}(D)$ seja uma transição da CM randomizada.

p_{jk} : Probabilidade que a primeira transição da CM randomizada seja (e_j, e_k) , dado que o primeiro evento do processo $\mathcal{W}(D)$ foi uma transição da CM.

$\rho^{(n,m)}(e_j, v_1)$: taxa de recompensa associada ao segmento v_1 .

$\tau\rho^{(n,m)}(e_j, v_1)$: recompensa ganha pelo sistema no (primeiro) segmento v_1 , dado que o comprimento dele é τ .

Seja $\mathcal{F}_j(D, z|n, m)$ a transformada de Laplace Stieljes sobre a variável L de $P[F_j^*(n, m, D) < L]$, e $\mathcal{L}\mathcal{L}$ o seu operador. Aplicando esta transformada na equação V.58:

$$\mathcal{F}_j(D, z|0, 0) = e^{-\rho^{(0,0)}(e_j, v_1)Dz} \quad (\text{V.60})$$

Analogamente, aplicando a transformada em V.59 e usando as observações feitas no parágrafo anterior a II.10:

$$\mathcal{F}_j(D, z|n, m) = \frac{1}{D^{n+m}} \int_0^D e^{-\rho^{(n,m)}(e_j, v_1)\tau z} (D-\tau)^{n+m-1} \left\{ n\mathcal{F}_j(D-\tau, z|n-1, m) + m \sum_{e_k \in E} p_{jk} \mathcal{F}_k(D-\tau, z|n, m-1) \right\} d\tau \quad (\text{V.61})$$

Seja $\mathcal{H}_j(s, z|n, m)$ a transformada de Laplace sobre a variável D de $\mathcal{F}_j(D, z|n, m)$, e seja \mathcal{L} o operador dela. Aplicando esta transformada em V.60 e V.61:

$$\mathcal{H}_j(s, z|0, 0) = \frac{1}{s + \rho^{(0,0)}(e_j, v_1)z} \quad (\text{V.62})$$

$$\mathcal{H}_j(s, z|n, m) = \int_{b_1=s}^{\infty} \cdots \int_{b_{n+m}=b_{n+m-1}}^{\infty} \mathcal{L} \left[e^{-\rho^{(n,m)}(e_j, v_1)zD} \otimes D^{n+m-1} \right. \\ \left. \left\{ n\mathcal{F}_j(D, z|n-1, m) + m \sum_{e_k \in E} p_{jk} \mathcal{F}_k(D, z|n, m-1) \right\} \right] db_{n+m} \cdots db_1$$

Portanto:

$$\mathcal{H}_j(s, z|n, m) = \int_{b_1=s}^{\infty} \cdots \int_{b_{n+m}=b_{n+m-1}}^{\infty} \frac{1}{(b_{n+m} + \rho^{(n,m)}(e_j, v_1)z)} \\ (-1)^{n+m-1} \frac{d^{n+m-1}}{db_{n+m}^{n+m-1}} \left\{ n\mathcal{H}_j(b_{n+m}, z|n-1, m) \right. \\ \left. + m \sum_{e_k \in E} p_{jk} \mathcal{H}_k(b_{n+m}, z|n, m-1) \right\} db_{n+m} \cdots db_1 \quad (\text{V.63})$$

É importante notar que o produto dos dois primeiro termos de acima formam uma expressão semelhante ao lado direito de II.27, e que, o produto do primeiro com o terceiro termo de acima formam uma expressão semelhante à equação (18) de [9]. Portanto a solução de V.63, dada por V.64, é também semelhante à soma das soluções dadas por II.29 e (19) de [9]. A diferença consiste em que em V.63 a recorrência é feita em duas variáveis (n e m) interdependentes e, embora os passos da solução sejam os mesmos descritos anteriormente, devido a que aqui estamos resolvendo um problema maior, a solução final é bem mais complexa. Para explicitar os detalhes da solução e indicar as diferenças com respeito à solução anterior, a continuação estão todos os passos necessários do método.

Seja $\mathcal{R}_n = \{r_n, r_{n-1}, \dots, r_0\}$ o conjunto de valores diferentes das taxas de recompensas $\rho^{(n,m)}(e_j, v_l)$, ($1 \leq l \leq n+m+1$). Segundo já foi discutido, a equação V.46 estabelece que $\mathcal{R}_n = \{n, n-1, \dots, 0\}$, porém para manter a notação do capítulo II, continuar-se-á considerando o caso mais geral, no qual $\mathcal{R}_n = \{r_n, r_{n-1}, \dots, r_0\}$. A seguir demonstra-se por indução que:

$$\mathcal{H}_j(s, z|n, m) = \int_{b_1=s}^{\infty} \cdots \int_{b_{n+m}=b_{n+m-1}}^{\infty} (n+m)! \left\{ \frac{\alpha_j^{(n,m)}}{(b_{n+m} + r_0 z)^{n+m+1}} \right.$$

$$+ \left. \sum_{h=1}^{n+m} \frac{\beta_j^{(n,m)}(h)}{(b_{n+m} + r_0 z)^h} + \sum_{c=1}^n \sum_{h=1}^{m+1} \frac{\gamma_j^{(n,m)}(c,h)}{(b_{n+m} + r_c z)^h} \right\} db_{n+m} \dots db_1 \quad (\text{V.64})$$

Onde $\alpha_j^{(n,m)}$, $\beta_j^{(n,m)}(h)$ e $\gamma_j^{(n,m)}(c,h)$ são valores reais que não dependem das variáveis $z, s, b_1, \dots, b_{n+m}$ e são calculados recorrentemente da forma que será indicada mais adiante.

Demonstração de V.64

Base

$$\mathcal{H}_j(s, z|0, 0) = \frac{\alpha_j^{(0,0)}}{(s + r_0 z)} \quad (\text{V.65})$$

Onde foi usado o fato que quando $n = m = 0$, as integrais $\int_{b_1}^{\infty} \dots \int_{b_{n+m}}^{\infty}$ não existem.

Definindo:

$$\alpha_j^{(0,0)} \stackrel{\text{def}}{=} 1 \quad (\text{V.66})$$

a equação V.64 corresponde à equação V.62 tem-se que pois $r_0 = 0$, e de V.46: $\rho^{(0,0)}(e_j, v_1) = 0$ para todo e_j , pois $q = 1$ quando $n = m = 0$. Em conseqüência a base da indução é satisfeita.

Hipótese Indutiva. Supondo que a equação V.64 é válida no caso em que o número de eventos do processo $\mathcal{W}(D)$ é menor ou igual a $(n + m - 1)$, a derivada da equação V.63 pode ser calculada da seguinte forma:

$$\begin{aligned} & \frac{d^{n+m-1}}{db_{n+m}^{n+m-1}} \left\{ n \mathcal{H}_j(b_{n+m}, z|n-1, m) + m \sum_{e_k \in E} p_{jk} \mathcal{H}_k(b_{n+m}, z|n, m-1) \right\} = \\ & (n+m-1)! \frac{d^{n+m-1}}{db_{n+m}^{n+m-1}} \left\{ \int_{x_1=b_{n+m}}^{\infty} \dots \int_{x_{n+m-1}=x_{n+m-2}}^{\infty} \right. \\ & n \left[\frac{\alpha_j^{(n-1,m)}}{(x_{n+m-1} + r_0 z)^{n+m}} + \sum_{h=1}^{n+m-1} \frac{\beta_j^{(n-1,m)}(h)}{(x_{n+m-1} + r_0 z)^h} + \sum_{c=1}^{n-1} \sum_{h=1}^{m+1} \frac{\gamma_j^{(n-1,m)}(c,h)}{(x_{n+m-1} + r_c z)^h} \right] \\ & + m \sum_{e_k \in E} p_{jk} \left[\frac{\alpha_k^{(n,m-1)}}{(x_{n+m-1} + r_0 z)^{n+m}} + \sum_{h=1}^{n+m-1} \frac{\beta_k^{(n,m-1)}(h)}{(x_{n+m-1} + r_0 z)^h} \right. \\ & \left. \left. + \sum_{c=1}^n \sum_{h=1}^m \frac{\gamma_k^{(n,m-1)}(c,h)}{(x_{n+m-1} + r_c z)^h} \right] \right\} dx_{n+m-1} \dots dx_1 \\ & = (-1)^{n+m-1} (n+m-1)! \left\{ n \left[\frac{\alpha_j^{(n-1,m)}}{(b_{n+m} + r_0 z)^{n+m}} + \sum_{h=1}^{n+m-1} \frac{\beta_j^{(n-1,m)}(h)}{(b_{n+m} + r_0 z)^h} \right. \right. \end{aligned}$$

$$\begin{aligned}
& + \sum_{c=1}^{n-1} \sum_{h=1}^{m+1} \frac{\frac{\gamma_j^{(n-1,m)}(c,h)}{z^{n+m-h}}}{(b_{n+m} + r_c z)^h} \Bigg] \\
+ m \sum_{e_k \in E} p_{jk} & \left[\frac{\alpha_k^{(n,m-1)}}{(b_{n+m} + r_0 z)^{n+m}} + \sum_{h=1}^{n+m-1} \frac{\frac{\beta_k^{(n,m-1)}(h)}{z^{n+m-h}}}{(b_{n+m} + r_0 z)^h} \right. \\
& \left. + \sum_{c=1}^n \sum_{h=1}^m \frac{\frac{\gamma_k^{(n,m-1)}(c,h)}{z^{n+m-h}}}{(b_{n+m} + r_c z)^h} \right] \quad (V.67)
\end{aligned}$$

Por simplicidade da notação o termo $\rho^{(n,m)}(e_j, v_1)$ será denotado como r^* . Substituindo V.67 em V.63:

$$\begin{aligned}
\mathcal{H}_j(s, z|n, m) = & \int_{b_1=s}^{\infty} \cdots \int_{b_{n+m}=b_{n+m-1}}^{\infty} (n+m-1)! \left\{ n \left[\frac{\alpha_j^{(n-1,m)}}{(b_{n+m} + r^* z)(b_{n+m} + r_0 z)^{n+m}} \right. \right. \\
& + \sum_{h=1}^{n+m-1} \frac{\frac{\beta_j^{(n-1,m)}(h)}{z^{n+m-h}}}{(b_{n+m} + r^* z)(b_{n+m} + r_0 z)^h} + \sum_{c=1}^{n-1} \sum_{h=1}^{m+1} \frac{\frac{\gamma_j^{(n-1,m)}(c,h)}{z^{n+m-h}}}{(b_{n+m} + r^* z)(b_{n+m} + r_c z)^h} \Bigg] \\
& + m \sum_{e_k \in E} p_{jk} \left[\frac{\alpha_k^{(n,m-1)}}{(b_{n+m} + r^* z)(b_{n+m} + r_0 z)^{n+m}} + \sum_{h=1}^{n+m-1} \frac{\frac{\beta_k^{(n,m-1)}(h)}{z^{n+m-h}}}{(b_{n+m} + r^* z)(b_{n+m} + r_0 z)^h} \right. \\
& \left. \left. + \sum_{c=1}^n \sum_{h=1}^m \frac{\frac{\gamma_k^{(n,m-1)}(c,h)}{z^{n+m-h}}}{(b_{n+m} + r^* z)(b_{n+m} + r_c z)^h} \right] \right\} db_{n+m} \cdots db_1 \quad (V.68)
\end{aligned}$$

Por outro lado, a equação V.46 estabelece que:

$$r^* = r_n I_j + r_0 \bar{I}_j \quad (V.69)$$

Lembrando, que no problema particular que está sendo tratado, $r_n = n$ e $r_0 = 0$, conclue-se que $r^* = n$ se $e_j \in \mathcal{U}$ e $r^* = 0$ se $e_j \in \mathcal{I}$. Porém, tal como foi indicado anteriormente, em vez disto, será utilizada a equação V.69 com o objetivo de manter a discussão num plano mais geral.

Usando V.69 e separando os termos de acordo com o valor de I_j ou \bar{I}_j , a equação V.68 pode ser reescrita da forma:

$$\begin{aligned}
\mathcal{H}_j(s, z|n, m) &= \int_{b_1=s}^{\infty} \dots \int_{b_{n+m}=b_{n+m-1}}^{\infty} (n+m-1)! \\
&\left\{ n \left[\frac{\alpha_j^{(n-1,m)} \bar{I}_j}{(b_{n+m} + r_0 z)^{n+m+1}} + \frac{\alpha_j^{(n-1,m)} I_j}{(b_{n+m} + r_n z)(b_{n+m} + r_0 z)^{n+m}} \right. \right. \\
&\quad \left. \left. + \sum_{h=1}^{n+m-1} \left(\frac{\beta_j^{(n-1,m)}(h) \bar{I}_j}{z^{n+m-h}} + \frac{\beta_j^{(n-1,m)}(h) I_j}{z^{n+m-h}} \right) \right. \right. \\
&\quad \left. \left. + \sum_{c=1}^{n-1} \sum_{h=1}^{m+1} \left(\frac{\gamma_j^{(n-1,m)}(c,h) \bar{I}_j}{z^{n+m-h}} + \frac{\gamma_j^{(n-1,m)}(c,h) I_j}{z^{n+m-h}} \right) \right] \right. \\
&\quad \left. + m \sum_{e_k \in E} p_{jk} \left[\frac{\alpha_k^{(n,m-1)} \bar{I}_j}{(b_{n+m} + r_0 z)^{n+m+1}} + \frac{\alpha_k^{(n,m-1)} I_j}{(b_{n+m} + r_n z)(b_{n+m} + r_0 z)^{n+m}} \right. \right. \\
&\quad \left. \left. + \sum_{h=1}^{n+m-1} \left(\frac{\beta_k^{(n,m-1)}(h) \bar{I}_j}{z^{n+m-h}} + \frac{\beta_k^{(n,m-1)}(h) I_j}{z^{n+m-h}} \right) \right. \right. \\
&\quad \left. \left. + \sum_{c=1}^n \sum_{h=1}^m \left(\frac{\gamma_k^{(n,m-1)}(c,h) \bar{I}_j}{z^{n+m-h}} \right. \right. \right. \\
&\quad \left. \left. \left. + \frac{\gamma_k^{(n,m-1)}(c,h) I_j}{z^{n+m-h}} \right) \right] \right\} db_{n+m} \dots db_1 \quad (V.70)
\end{aligned}$$

Agora, usando II.34, os diferentes termos do lado direito da equação V.70 são expandidos em frações parciais. Para simplificar o acompanhamento, os termos de V.70 serão expandidos (nos casos que isto seja pertinente) em ordem, isto é: primeiro escrevem-se toda a expansão do primeiro termo de V.70, depois toda a expansão do segundo termo, etc. Portanto, usando a variável $h' = h + 1$:

$$\begin{aligned}
\mathcal{H}_j(s, z|n, m) &= \int_{b_1=s}^{\infty} \dots \int_{b_{n+m}=b_{n+m-1}}^{\infty} (n+m-1)! \\
&\left\{ n \left[\frac{\alpha_j^{(n-1,m)} \bar{I}_j}{(b_{n+m} + r_0 z)^{n+m+1}} + \frac{\alpha_j^{(n-1,m)} I_j}{z^{n+m} (r_0 - r_n)^{n+m}} \right. \right. \\
&- \sum_{h=1}^{n+m} \frac{\alpha_j^{(n-1,m)} I_j}{z^{n+m+1-h} (r_0 - r_n)^{n+m+1-h}} + \sum_{h'=2}^{n+m} \frac{\beta_j^{(n-1,m)} (h'-1) \bar{I}_j}{z^{n+m+1-h'}} \\
&+ \sum_{h=1}^{n+m-1} \frac{\beta_j^{(n-1,m)}(h) I_j}{z^{n+m} (r_0 - r_n)^h} - \sum_{h=1}^{n+m-1} \sum_{q=1}^h \frac{\beta_j^{(n-1,m)}(h) I_j}{z^{n+m+1-q} (r_0 - r_n)^{h+1-q}} \\
&+ \sum_{c=1}^{n-1} \sum_{h=1}^{m+1} \frac{\gamma_j^{(n-1,m)}(c, h) \bar{I}_j}{z^{n+m} (r_c - r_0)^h} - \sum_{c=1}^{n-1} \sum_{h=1}^m \sum_{q=1}^h \frac{\gamma_j^{(n-1,m)}(c, h) \bar{I}_j}{z^{n+m+1-q} (r_c - r_0)^{h+1-q}} \\
&\left. + \sum_{c=1}^{n-1} \sum_{h=1}^{m+1} \frac{\gamma_j^{(n-1,m)}(c, h) I_j}{z^{n+m} (r_c - r_n)^h} - \sum_{c=1}^{n-1} \sum_{h=1}^m \sum_{q=1}^h \frac{\gamma_j^{(n-1,m)}(c, h) I_j}{z^{n+m+1-q} (r_c - r_n)^{h+1-q}} \right] \\
&+ m \sum_{e_k \in E} p_{jk} \left[\frac{\alpha_k^{(n,m-1)} \bar{I}_j}{(b_{n+m} + r_0 z)^{n+m+1}} + \frac{\alpha_k^{(n,m-1)} I_j}{z^{n+m} (r_0 - r_n)^{n+m}} \right. \\
&- \sum_{h=1}^{n+m} \frac{\alpha_k^{(n,m-1)} I_j}{z^{n+m+1-h} (r_0 - r_n)^{n+m+1-h}} + \sum_{h'=2}^{n+m} \frac{\beta_k^{(n,m-1)} (h'-1) \bar{I}_j}{z^{n+m+1-h'}} \\
&+ \sum_{h=1}^{n+m-1} \frac{\beta_k^{(n,m-1)}(h) I_j}{z^{n+m} (r_0 - r_n)^h} - \sum_{h=1}^{n+m-1} \sum_{q=1}^h \frac{\beta_k^{(n,m-1)}(h) I_j}{z^{n+m+1-q} (r_0 - r_n)^{h+1-q}} \\
&+ \sum_{c=1}^n \sum_{h=1}^m \frac{\gamma_k^{(n,m-1)}(c, h) \bar{I}_j}{z^{n+m} (r_c - r_0)^h} - \sum_{c=1}^n \sum_{h=1}^{m-1} \sum_{q=1}^h \frac{\gamma_k^{(n,m-1)}(c, h) \bar{I}_j}{z^{n+m+1-q} (r_c - r_0)^{h+1-q}} \\
&+ \sum_{h'=2}^{m+1} \frac{\gamma_k^{(n,m-1)}(n, h'-1) I_j}{z^{n+m+1-h'}} + \sum_{c=1}^{n-1} \sum_{h=1}^m \frac{\gamma_k^{(n,m-1)}(c, h) I_j}{z^{n+m} (r_c - r_n)^h} \\
&\left. - \sum_{c=1}^{n-1} \sum_{h=1}^m \sum_{q=1}^h \frac{\gamma_k^{(n,m-1)}(c, h) I_j}{z^{n+m+1-q} (r_c - r_n)^{h+1-q}} \right] \Bigg\} db_{n+m} \dots db_1 \tag{V.71}
\end{aligned}$$

Tal como em CLEO, para demonstrar a identidade das equações V.63 e V.71 deve-se fazer com que as respectivas frações sejam iguais. Isto acontece somente se os termos α 's, β 's e γ 's de ambas equações satisfazem as relações que se estabelecem a seguir. Além do mais, para simplificar alguns termos será usada a identidade $\sum_{h=1}^L \sum_{q=1}^h = \sum_{q=1}^L \sum_{h=q}^L$, e trocada a variável 'q' pela 'h' (com o qual o duplo somatório fica $\sum_{h=1}^L \sum_{q=h}^L$).

Para que as frações das equações V.63 e V.71 que tem denominador $(b_{n+m} + r_0 z)^{n+m+1}$ sejam iguais, (os termos 1 e 11 da equação V.71), a igualdade abaixo deve ser satisfeita:

$$(n+m)\alpha_j^{(n,m)} = n\alpha_j^{(n-1,m)}\bar{I}_j + \sum_{e_k \in E} mp_{jk}\alpha_k^{(n,m-1)}\bar{I}_j \quad (\text{V.72})$$

O fator \bar{I}_j , implica que $\alpha_j^{(n,m)} \neq 0$ somente se $e_j \in \mathcal{I}$. isto mesmo implica que $\alpha_k^{(n-1,m)} \neq 0$ somente se $e_k \in \mathcal{I}$. Portanto nas equações seguintes os termos da forma $\alpha_k^{(n,m)}\bar{I}_k$ serão considerados iguais a zero para todo $e_k \in E$. Em particular, V.72 pode ser reescrita da forma:

$$\alpha_j^{(n,m)} = \bar{I}_j \left[\frac{n}{n+m}\alpha_j^{(n-1,m)} + \sum_{e_k \in \mathcal{I}} \frac{m}{n+m}p_{jk}\alpha_k^{(n,m-1)} \right] \quad (\text{V.73})$$

Dos termos 3, 4, 13 e 14 de V.71 tem-se que, para $(n+m) \geq 2$:

$$\begin{aligned} (n+m)\beta_j^{(n,m)}(n+m) &= \frac{-n\alpha_j^{(n-1,m)}I_j}{(r_0 - r_n)} + n\beta_j^{(n-1,m)}(n+m-1)\bar{I}_j \\ &\quad - \sum_{e_k \in E} \frac{mp_{jk}\alpha_k^{(n,m-1)}I_j}{(r_0 - r_n)} + \sum_{e_k \in E} mp_{jk}\beta_k^{(n,m-1)}(n+m-1)\bar{I}_j \end{aligned} \quad (\text{V.74})$$

Lembrando que $I_j\alpha_j^{(n,m)} = 0$, ve-se que o primeiro termo de acima é igual a zero e primeiro somatório e é significativo somente para $e_k \in \mathcal{I}$, portanto:

$$\begin{aligned} \beta_j^{(n,m)}(n+m) &= \\ &\quad \bar{I}_j \left[\frac{n}{n+m}\beta_j^{(n-1,m)}(n+m-1) + \frac{m}{n+m} \sum_{e_k \in \mathcal{I}} p_{jk}\beta_k^{(n,m-1)}(n+m-1) \right] \\ &\quad - \frac{m}{(n+m)} \frac{I_j}{(r_0 - r_n)} \sum_{e_k \in \mathcal{I}} p_{jk}\alpha_k^{(n,m-1)} \end{aligned} \quad (\text{V.75})$$

Dos termos 3, 4, 6, 13, 14 e 16 da equação V.71:

$$\begin{aligned}
(n+m)\beta_j^{(n,m)}(h) &= \frac{-n\alpha_j^{(n-1,m)}I_j}{(r_0-r_n)^{n+m+1-h}} + n\beta_j^{(n-1,m)}(h-1)\bar{I}_j \\
&- \sum_{q=h}^{n+m-1} \frac{n\beta_j^{(n-1,m)}(q)I_j}{(r_0-r_n)^{q+1-h}} - \sum_{e_k \in E} \frac{mp_{jk}\alpha_k^{(n,m-1)}I_j}{(r_0-r_n)^{n+m+1-h}} \\
&+ \sum_{e_k \in E} mp_{jk}\beta_k^{(n,m-1)}(h-1)\bar{I}_j - \sum_{e_k \in E} \sum_{q=h}^{n+m-1} \frac{mp_{jk}\beta_k^{(n,m-1)}(q)I_j}{(r_0-r_n)^{q+1-h}}
\end{aligned} \tag{V.76}$$

Eliminando os termos iguais a zero, isolando o primeiro termo dos somatórios sob a variável q e aproveitando que $I_j\bar{I}_j = 0$ (para somar termos convenientes iguais a zero):

$$\begin{aligned}
\beta_j^{(n,m)}(h) &= \frac{I_j}{n+m} \left[n\beta_j^{(n-1,m)}(h)\bar{I}_j - \sum_{q=h+1}^{n+m-1} \frac{n\beta_j^{(n-1,m)}(q)I_j}{(r_0-r_n)^{q+1-h}} \right. \\
&+ \left. \sum_{e_k \in E} mp_{jk}\beta_k^{(n,m-1)}(h-1)\bar{I}_j - \sum_{e_k \in E} \sum_{q=h+1}^{n+m-1} \frac{mp_{jk}\beta_k^{(n,m-1)}(q)I_j}{(r_0-r_n)^{q+1-h}} \right] \\
&+ \frac{n}{n+m}\beta_j^{(n-1,m)}(h-1)\bar{I}_j - \frac{n}{n+m}\frac{\beta_j^{(n-1,m)}(h)I_j}{(r_0-r_n)} \\
&+ \frac{m}{n+m}\sum_{e_k \in E} p_{jk}\beta_k^{(n,m-1)}(h-1)\bar{I}_j - \frac{m}{n+m}\sum_{e_k \in E} \frac{p_{jk}\beta_k^{(n,m-1)}(h)I_j}{(r_0-r_n)}
\end{aligned} \tag{V.77}$$

Note-se que os termos entre '[]' desta equação são iguais a $(n+m)\beta_j^{(n,m)}(h+1)$ em V.76, pois $\alpha_j^{(n,m)}I_j = 0$ e $I_j\bar{I}_j = 0$, portanto:

$$\begin{aligned}
\beta_j^{(n,m)}(h) &= I_j\beta_j^{(n,m)}(h+1) \\
&+ \bar{I}_j \left[\frac{n}{n+m}\beta_j^{(n-1,m)}(h-1) + \frac{m}{n+m}\sum_{e_k \in E} p_{jk}\beta_k^{(n,m-1)}(h-1) \right] \\
&- \frac{I_j}{(r_0-r_n)} \left[\frac{n}{n+m}\beta_j^{(n-1,m)}(h) + \frac{m}{n+m}\sum_{e_k \in E} p_{jk}\beta_k^{(n,m-1)}(h) \right]
\end{aligned} \tag{V.78}$$

Do termo número 19 da equação V.71:

$$(n+m)\gamma_j^{(n,m)}(n,m+1) = I_j \sum_{e_k \in E} mp_{jk}\gamma_k^{(n,m-1)}(n,m) \tag{V.79}$$

Portanto:

$$\gamma_j^{(n,m)}(n, m+1) = \frac{m}{n+m} I_j \sum_{e_k \in E} p_{jk} \gamma_k^{(n,m-1)}(n, m) \quad (\text{V.80})$$

Dos termos 18 e 19 de V.71 para o caso $c = n$, e trocando a variável q pela h em V.71:

$$(n+m) \gamma_j^{(n,m)}(n, h) = - \sum_{e_k \in E} \sum_{q=h}^m \frac{m p_{jk} \gamma_k^{(n,m-1)}(n, q) \bar{I}_j}{(r_n - r_0)^{q+1-h}} + \sum_{e_k \in E} m p_{jk} \gamma_k^{(n,m-1)}(n, h-1) I_j \quad (\text{V.81})$$

Aproveitando que $I_j \bar{I}_j = 0$, e isolando o primeiro termo do somatório sob a variável q de acima:

$$\begin{aligned} \gamma_j^{(n,m)}(n, h) &= \frac{m}{n+m} \sum_{e_k \in E} p_{jk} \left[\gamma_k^{(n,m-1)}(n, h-1) I_j - \frac{\gamma_k^{(n,m-1)}(n, h) \bar{I}_j}{(r_n - r_0)} \right] \\ &+ \frac{\bar{I}_j}{(r_n - r_0)} \sum_{e_k \in E} p_{jk} \frac{m}{(n+m)} \left\{ \sum_{q=h+1}^m \frac{\gamma_k^{(n,m-1)}(n, q) \bar{I}_j}{(r_n - r_0)^{q+1-(h+1)}} + \gamma_k^{(n,m-1)}(n, h) I_j \right\} \end{aligned}$$

O último somatório de acima é igual a $\gamma_j^{(n,m)}(n, h+1)$ (ver equação V.81), então:

$$\begin{aligned} \gamma_j^{(n,m)}(n, h) &= \bar{I}_j \frac{\gamma_j^{(n,m)}(n, h+1)}{(r_n - r_0)} + \frac{m I_j}{n+m} \sum_{e_k \in E} p_{jk} \gamma_k^{(n,m-1)}(n, h-1) \\ &- \frac{m \bar{I}_j}{n+m} \sum_{e_k \in E} \frac{p_{jk} \gamma_k^{(n,m-1)}(n, h)}{(r_n - r_0)} \quad (\text{V.82}) \end{aligned}$$

Dos termos 8 e 10 da equação V.71:

$$\gamma_j^{n-1,m}(c, m+1) = - \frac{n}{n+m} \left[\frac{\bar{I}_j \gamma_j^{(n-1,m)}(c, m+1)}{(r_c - r_0)} + \frac{I_j \gamma_j^{(n-1,m)}(c, m+1)}{(r_c - r_n)} \right] \quad (\text{V.83})$$

Dos termos 8, 10, 18 e 21 da equação V.71:

$$\begin{aligned}
& \begin{matrix} (n+m)\gamma_j^{(n,m)}(c,h) \\ 1 \leq c < n \\ 1 \leq h \leq m \end{matrix} = - \sum_{q=h}^{m+1} \frac{n\gamma_j^{(n-1,m)}(c,q)\bar{I}_j}{(r_c - r_0)^{q+1-h}} - \sum_{q=h}^{m+1} \frac{n\gamma_j^{(n-1,m)}(c,q)I_j}{(r_c - r_n)^{q+1-h}} \\
& - \sum_{e_k \in E} \sum_{q=h}^m \frac{mp_{jk}\gamma_k^{(n,m-1)}(c,q)\bar{I}_j}{(r_c - r_0)^{q+1-h}} - \sum_{e_k \in E} \sum_{q=h}^m \frac{mp_{jk}\gamma_k^{(n,m-1)}(c,q)I_j}{(r_c - r_n)^{q+1-h}} \quad (\text{V.84})
\end{aligned}$$

Evidenciando os termos para os quais $q = h$, V.84 pode ser reescrita da forma:

$$\begin{aligned}
& \begin{matrix} (n+m)\gamma_j^{(n,m)}(c,h) \\ 1 \leq h < m \\ 1 \leq h \leq m \end{matrix} = - \frac{\bar{I}_j}{(r_c - r_0)} \left[n\gamma_j^{(n-1,m)}(c,h) + \sum_{e_k \in E} mp_{jk}\gamma_k^{(n,m-1)}(c,h) \right] \\
& - \frac{I_j}{(r_c - r_n)} \left[n\gamma_j^{(n-1,m)}(c,h) + \sum_{e_k \in E} mp_{jk}\gamma_k^{(n,m-1)}(c,h) \right] \\
& + \frac{1}{(r_c - r_0)} \left[- \sum_{q=h+1}^{m+1} \frac{n\gamma_j^{(n-1,m)}(c,q)\bar{I}_j}{(r_c - r_0)^{q+1-(h+1)}} - \sum_{e_k \in E} \sum_{q=h+1}^m \frac{mp_{jk}\gamma_k^{(n,m-1)}(c,q)\bar{I}_j}{(r_c - r_0)^{q+1-(h+1)}} \right] \\
& + \frac{1}{(r_c - r_n)} \left[- \sum_{q=h+1}^{m+1} \frac{n\gamma_j^{(n-1,m)}(c,q)I_j}{(r_c - r_n)^{q+1-(h+1)}} - \sum_{e_k \in E} \sum_{q=h+1}^m \frac{mp_{jk}\gamma_k^{(n,m-1)}(c,q)I_j}{(r_c - r_n)^{q+1-(h+1)}} \right]
\end{aligned}$$

Como $I_j\bar{I}_j = 0$, a equação de acima pode ser reescrita da forma:

$$\begin{aligned}
& \begin{matrix} (n+m)\gamma_j^{(n,m)}(c,h) \\ 1 \leq c < n \\ 1 \leq h \leq m \end{matrix} = - \frac{\bar{I}_j}{(r_c - r_0)} \left[n\gamma_j^{(n-1,m)}(c,h) + \sum_{e_k \in E} mp_{jk}\gamma_k^{(n,m-1)}(c,h) \right] \\
& - \frac{I_j}{(r_c - r_n)} \left[n\gamma_j^{(n-1,m)}(c,h) + \sum_{e_k \in E} mp_{jk}\gamma_k^{(n,m-1)}(c,h) \right] \\
& + \left[\frac{\bar{I}_j}{(r_c - r_0)} + \frac{I_j}{(r_c - r_n)} \right] \left[- \sum_{q=h+1}^{m+1} \frac{n\gamma_j^{(n-1,m)}(c,q)\bar{I}_j}{(r_c - r_0)^{q+1-(h+1)}} \right. \\
& - \sum_{e_k \in E} \sum_{q=h+1}^m \frac{mp_{jk}\gamma_k^{(n,m-1)}(c,q)\bar{I}_j}{(r_c - r_0)^{q+1-(h+1)}} - \sum_{q=h+1}^{m+1} \frac{n\gamma_j^{(n-1,m)}(c,q)I_j}{(r_c - r_n)^{q+1-(h+1)}} \\
& \left. - \sum_{e_k \in E} \sum_{q=h+1}^m \frac{mp_{jk}\gamma_k^{(n,m-1)}(c,q)I_j}{(r_c - r_n)^{q+1-(h+1)}} \right]
\end{aligned}$$

A última expressão entre '[]' de acima é igual a $(n+m)\gamma_j^{(n,m)}(c,h+1)$ em V.84, portanto:

$$\begin{aligned}
& \gamma_j^{(n,m)}(c, h) \\
& \substack{1 \leq c < n \\ 1 \leq h \leq m} \\
& = -\frac{\bar{I}_j}{(r_c - r_0)} \left[\frac{n}{n+m} \gamma_j^{(n-1,m)}(c, h) + \frac{m}{n+m} \sum_{e_k \in E} p_{jk} \gamma_k^{(n,m-1)}(c, h) \right] \\
& \quad - \frac{I_j}{(r_c - r_n)} \left[\frac{n}{n+m} \gamma_j^{(n-1,m)}(c, h) + \frac{m}{n+m} \sum_{e_k \in E} p_{jk} \gamma_k^{(n,m-1)}(c, h) \right] \\
& \quad + \left[\frac{\bar{I}_j}{(r_c - r_0)} + \frac{I_j}{(r_c - r_n)} \right] \gamma_j^{(n,m)}(c, h+1) \tag{V.85}
\end{aligned}$$

Dos termos 3, 6, 7, 13, 16 e 17 da equação V.71:

$$\begin{aligned}
(n+m)\beta_j^{(n,m)}(1) & = -\frac{n\alpha_j^{(n-1,m)} I_j}{(r_0 - r_n)^{(n+m+1)-1}} - \sum_{q=1}^{n+m-1} \frac{n\beta_j^{(n-1,m)}(q) I_j}{(r_0 - r_n)^{(q+1)-1}} \\
& \quad + \sum_{c=1}^{n-1} \sum_{q=1}^{m+1} \frac{n\gamma_j^{(n-1,m)}(c, q) \bar{I}_j}{(r_c - r_0)^q} - \sum_{e_k \in E} \frac{mp_{jk} \alpha_k^{(n,m-1)} I_j}{(r_0 - r_n)^{(n+m+1)-1}} \\
& \quad - \sum_{e_k \in E} \sum_{q=1}^{n+m-1} \frac{mp_{jk} \beta_k^{(n,m-1)}(q) I_j}{(r_0 - r_n)^{(q+1)-1}} + \sum_{e_k \in E} \sum_{c=1}^n \sum_{q=1}^m \frac{mp_{jk} \gamma_k^{(n,m-1)}(c, q) \bar{I}_j}{(r_c - r_0)^q} \\
& = \frac{I_j}{(r_0 - r_n)} \left[\frac{-n\alpha_j^{(n-1,m)} I_j}{(r_0 - r_n)^{(n+m+1)-2}} + n\beta_j^{(n-1,m)}(1) \bar{I}_j \right. \\
& \quad - \sum_{q=2}^{n+m-1} \frac{n\beta_j^{(n-1,m)}(q) I_j}{(r_0 - r_n)^{(q+1)-2}} - \sum_{e_k \in E} \frac{mp_{jk} \alpha_k^{(n,m-1)} I_j}{(r_0 - r_n)^{(n+m+1)-2}} \\
& \quad \left. + \sum_{e_k \in E} mp_{jk} \beta_k^{(n,m-1)}(1) \bar{I}_j - \sum_{e_k \in E} \sum_{q=2}^{n+m-1} \frac{mp_{jk} \beta_k^{(n,m-1)}(q) I_j}{(r_0 - r_n)^{(q+1)-2}} \right] \\
& \quad - \frac{I_j}{(r_0 - r_n)} \left[n\beta_j^{(n-1,m)}(1) + \sum_{e_k \in E} mp_{jk} \beta_k^{(n,m-1)}(1) \right] \\
& \quad - \bar{I}_j \sum_{c=1}^{n-1} \left[-\sum_{q=1}^{m+1} \frac{n\gamma_j^{(n-1,m)}(c, q) \bar{I}_j}{(r_c - r_0)^{(q+1)-1}} - \sum_{q=1}^{m+1} \frac{n\gamma_j^{(n-1,m)}(c, q) I_j}{(r_c - r_n)^{(q+1)-1}} \right. \\
& \quad \left. - \sum_{e_k \in E} \sum_{q=1}^m \frac{mp_{jk} \gamma_k^{(n,m-1)}(c, q) \bar{I}_j}{(r_c - r_0)^{(q+1)-1}} - \sum_{e_k \in E} \sum_{q=1}^m \frac{mp_{jk} \gamma_k^{(n,m-1)}(c, q) I_j}{(r_c - r_n)^{(q+1)-1}} \right] \tag{V.86}
\end{aligned}$$

A primeira expressão entre '[]' é igual a $(n+m)\beta_j^{(n,m)}(2)$ em V.76 e a última expressão entre '[]' é igual a $(n+m)\gamma_j^{(n,m)}(c, 1)$ em V.84, portanto:

$$\beta_j^{(n,m)}(1) = \frac{I_j}{(r_0 - r_n)} \left[\beta_j^{(n,m)}(2) - \frac{n}{n+m} \beta_j^{(n-1,m)}(1) - \frac{m}{n+m} \sum_{e_k \in E} p_{jk} \beta_k^{(n,m-1)}(1) \right]$$

$$-\bar{I}_j \sum_{c=1}^{n-1} \gamma_j^{(n,m)}(c, 1) \quad (\text{V.87})$$

Dos termos 2,5,12,15,9,20, e 18 da equação V.71:

$$\begin{aligned} (n+m)\gamma_j^{(n,m)}(n, 1) &= \frac{n\alpha_j^{(n-1,m)}I_j}{(r_0-r_n)^{n+m}} + \sum_{q=1}^{n+m-1} \frac{n\beta_j^{(n-1,m)}(q)I_j}{(r_0-r_n)^q} \\ &+ \sum_{e_k \in E} \frac{mp_{jk}\alpha_k^{(n,m-1)}I_j}{(r_0-r_n)^{n+m}} + \sum_{e_k \in E} \sum_{q=1}^{n+m-1} \frac{mp_{jk}\beta_k^{(n,m-1)}(q)I_j}{(r_0-r_n)^q} \\ &+ \sum_{c=1}^{n-1} \sum_{q=1}^{m+1} \frac{n\gamma_j^{(n-1,m)}(c, q)I_j}{(r_c-r_n)^q} + \sum_{e_k \in E} \sum_{c=1}^{n-1} \sum_{q=1}^m \frac{mp_{jk}\gamma_k^{(n,m-1)}(c, q)I_j}{(r_c-r_n)^q} \\ &- \sum_{e_k \in E} \sum_{q=1}^m \frac{mp_{jk}\gamma_k^{(n,m-1)}(n, q)\bar{I}_j}{(r_n-r_0)^q} \end{aligned} \quad (\text{V.88})$$

Devido a que $I_j\bar{I}_j = 0$, a equação de acima pode ser reescrita da forma:

$$\begin{aligned} (n+m)\gamma_j^{(n,m)}(n, 1) &= -I_j \left[\frac{-n\alpha_j^{(n-1,m)}I_j}{(r_0-r_n)^{(n+m+1)-1}} + n\beta_j^{(n-1,m)}(1)\bar{I}_j \right. \\ &- \sum_{q=2}^{n+m-1} \frac{n\beta_j^{(n-1,m)}(q)I_j}{(r_0-r_n)^{(q+1)-1}} - \sum_{e_k \in E} \frac{mp_{jk}\alpha_k^{(n,m-1)}I_j}{(r_0-r_n)^{(n+m+1)-1}} \\ &+ \sum_{e_k \in E} mp_{jk}\beta_k^{(n,m-1)}(1)\bar{I}_j - \sum_{e_k \in E} \sum_{q=2}^{n+m-1} \frac{mp_{jk}\beta_k^{(n,m-1)}(q)I_j}{(r_0-r_n)^{(q+1)-1}} \left. \right] \\ &- \frac{n\beta_j^{(n-1,m)}(1)I_j}{(r_0-r_n)} - \sum_{e_k \in E} \frac{mp_{jk}\beta_k^{(n,m-1)}(1)I_j}{(r_0-r_n)} \\ &- \sum_{c=1}^{n-1} I_j \left[- \sum_{q=1}^{m+1} \frac{n\gamma_j^{(n-1,m)}(c, q)\bar{I}_j}{(r_c-r_0)^{(q+1)-1}} - \sum_{q=1}^{m+1} \frac{n\gamma_j^{(n-1,m)}(c, q)I_j}{(r_c-r_n)^{(q+1)-1}} \right. \\ &- \sum_{e_k \in E} \sum_{q=1}^m \frac{mp_{jk}\gamma_k^{(n,m-1)}(c, q)\bar{I}_j}{(r_c-r_0)^{(q+1)-1}} - \sum_{e_k \in E} \sum_{q=1}^m \frac{mp_{jk}\gamma_k^{(n,m-1)}(c, q)I_j}{(r_c-r_n)^{(q+1)-1}} \left. \right] \\ &+ \frac{\bar{I}_j}{(r_n-r_0)} \left[- \sum_{e_k \in E} \sum_{q=2}^m \frac{mp_{jk}\gamma_k^{(n,m-1)}(n, q)\bar{I}_j}{(r_n-r_0)^{(q+1)-2}} \right. \\ &+ \sum_{e_k \in E} mp_{jk}\gamma_k^{(n,m-1)}(n, 1)I_j \left. \right] - \sum_{e_k \in E} \frac{mp_{jk}\gamma_k^{(n,m-1)}(n, 1)\bar{I}_j}{(r_n-r_0)} \quad (\text{V.89}) \end{aligned}$$

A primeira expressão entre '[']' de acima é igual a $(n+m)\beta_j^{(n,m)}(1)$ (ver a segunda igualdade de V.86). A segunda expressão entre '[']' de acima é

igual a $(n+m)\gamma_j^{(n,m)}(c,1)$ (ver V.84). A terceira expressão entre '[]' de acima é igual a $(n+m)\gamma_j^{(n,m)}(n,2)$ (ver V.81), portanto:

$$\begin{aligned} \gamma_j^{(n,m)}(n,1) &= -I_j \left\{ \beta_j^{(n,m)}(1) + \sum_{c=1}^{n-1} \gamma_j^{(n,m)}(c,1) \right. \\ &\quad \left. + \frac{n}{n+m} \frac{\beta_j^{(n-1,m)}(1)}{(r_0 - r_n)} + \sum_{e_k \in E} \frac{m}{n+m} \frac{p_{jk} \beta_k^{(n,m-1)}(1)}{(r_0 - r_n)} \right\} \\ &\quad + \frac{\bar{I}_j}{(r_n - r_0)} \left\{ \gamma_j^{(n,m)}(n,2) - \frac{m}{n+m} \sum_{e_k \in E} p_{jk} \gamma_k^{(n,m-1)}(n,1) \right\} \quad (\text{V.90}) \end{aligned}$$

Com o qual fica demonstrada a validade da equação V.64.

Para completar a análise falta inverter a equação V.64. Para inverter sob a variável 's', usam-se as observações da seção II.3.3 do capítulo II, de onde:

$$\begin{aligned} \mathcal{F}_j(D, z|n, m) &= \frac{(n+m)!}{D^{n+m}} \mathcal{L}^{-1} \left[\frac{\alpha_j^{(n,m)}}{(s+r_0z)^{n+m+1}} \right. \\ &\quad \left. + \sum_{h=1}^{n+m} \frac{\beta_j^{(n,m)}(h)}{z^{n+m+1-h}} + \sum_{c=1}^n \sum_{h=1}^{m+1} \frac{\gamma_j^{(n,m)}(c,h)}{z^{n+m+1-h}} \frac{1}{(s+r_cz)^h} \right] \\ &= \frac{(n+m)!}{D^{n+m}} \left[\frac{\alpha_j^{(n,m)} D^{n+m} e^{-r_0zD}}{(n+m)!} + \sum_{h=1}^{n+m} \frac{\beta_j^{(n,m)}(h) D^{h-1} e^{-r_0zD}}{(h-1)! z^{n+m+1-h}} \right. \\ &\quad \left. + \sum_{c=1}^n \sum_{h=1}^{m+1} \frac{\gamma_j^{(n,m)}(c,h) D^{h-1} e^{-r_czD}}{(h-1)! z^{n+m+1-h}} \right] \quad (\text{V.91}) \end{aligned}$$

Agora, invertendo sobre a variável 'z':

$$\begin{aligned} fa_j^*(D, y|n, m) &= \alpha_j^{(n,m)} \delta[y - r_0D] \\ (n+m)! \sum_{h=1}^{n+m} \frac{\beta_j^{(n,m)}(h)}{D^{n+m+1-h} (h-1)!} &\underbrace{\int_{-\infty}^y \dots \int_{-\infty}^y}_{n+m+1-h \text{ vezes}} \delta[x - r_0D] (dx)^{n+m+1-h} \end{aligned} \quad (\text{V.92})$$

Onde $fa_j^*(D, y|n, m)$ corresponde à função de densidade de probabilidade da v.a $P[F_j^*(n, m, D) < L]$, e $\delta(x)$ corresponde a derivada da função $u[x]$ definida no capítulo II.

Efetuando as integrais chega-se a que:

$$\begin{aligned}
fa_j^*(D, y|n, m) &= \alpha_j^{(n,m)} \delta[y - r_0 D] \\
&+ (n+m)! \sum_{h=1}^{n+m} \frac{\beta_j^{(n,m)}(h) (y - r_0 D)^{n+m-h} u[y - r_0 D]}{D^{n+m+1-h} (h-1)! (n+m-h)!} \\
&+ (n+m)! \sum_{c=1}^n \sum_{h=1}^{m+1} \frac{\gamma_j^{(n,m)}(c, h) (y - r_c D)^{n+m-h} u[y - r_c D]}{D^{n+m+1-h} (h-1)! (n+m-h)!}
\end{aligned} \tag{V.93}$$

Por último, integrando $fa_j^*(D, y|n, m)$, sobre a variável y , conclue-se que:

$$\begin{aligned}
P[F_j^*(n, m, D) < L] &= \alpha_j^{(n,m)} u[L - r_0 D] \\
&+ \sum_{h=1}^{n+m} \binom{n+m}{h-1} \beta_j^{(n,m)}(h) \left(\frac{L - r_0 D}{D}\right)^{n+m+1-h} u[L - r_0 D] \\
&+ \sum_{c=1}^n \sum_{h=1}^{m+1} \binom{n+m}{h-1} \gamma_j^{(n,m)}(c, h) \left(\frac{L - r_c D}{D}\right)^{n+m+1-h} u[L - r_c D]
\end{aligned} \tag{V.94}$$

Onde os termos α 's, β ' e γ 's são calculados recorrentemente segundo as equações V.72 ... V.90.

Por último usando V.33, V.37, V.38 e V.94 em V.30 obtemos a expressão final para a função de distribuição da disponibilidade.

V.6.3 Método 3

Tal como foi indicado no capítulo II, o Método A usado na seção V.6.2 para obter $P[F_j^*(n, m, D) < L]$ implica muitos detalhes matemáticos cansativos e pouco elucidantes, porém é possível obter o mesmo resultado do Método A (no caso as equações V.72 ... V.90, V.94) usando o método B do capítulo II. Tal como foi comentado naquele capítulo, este último método têm as vantagens de dar uma interpretação para os diferentes termos do resultado (termos α 's, β 's e γ 's no presente caso) e a dedução da expressão final é bastante simples.

Nesta seção apresenta-se o cálculo de $P[F_j^*(n, m, D) < L]$ segundo o método B do capítulo II. Para isto, em primeiro lugar considera-se novamente a equação V.48; reescrita abaixo:

$$P[F_j^*(n, m, D) < L] = \sum_{\vec{K}_{n,m} \in \mathcal{K}_{n,m}} \Omega_j[\vec{K}_{n,m}] P \left[\sum_{q=1}^{n+1} \sum_{l=K_{q-1}+1}^{K_q} (n-q+1) v_l I[\chi_2(l) \in \mathcal{U}] < L | \vec{K}_{n,m} \right] \tag{V.95}$$

Tal com já foi comentado, $P\left[\sum_{q=1}^{n+1} \sum_{l=K_{q-1}+1}^{K_q} (n-q+1)I[\chi_2(l) \in \mathcal{U}]v_l < L | \vec{K}_{n,m}\right]$ corresponde à combinação linear das estatísticas de ordem de $(n+m)$, v.a's uniformes em $[0, D]$ (ver seção V.6.1), onde $\vec{K}_{n,m}$ é o vetor cujo c -ésimo componente $k_c = \vec{K}_{n,m}[c]$ indica o número de segmentos com recompensa r_c . Portanto, usando a equação II.46, chega-se a que:

$$P\left[\sum_{q=1}^{n+1} \sum_{l=K_{q-1}+1}^{K_q} (n-q+1)v_l I[\chi_2(l) \in \mathcal{U}] < L | \vec{K}_{n,m}\right] = \sum_{c=0}^n \sum_{h=1}^{\vec{K}_{n,m}[c]} \binom{n+m}{h-1} g(c, h, \vec{K}_{n,m}) \left(\frac{L-r_c D}{D}\right)^{n+m+1-h} u[L-r_c D] \quad (\text{V.96})$$

Sustituindo V.96 em V.95:

$$P[F_j^*(n, m, D) < L] = \sum_{\vec{K}_{n,m} \in \mathcal{K}_{n,m}} \Omega_j[\vec{K}_{n,m}] \sum_{c=0}^n \sum_{h=1}^{\vec{K}_{n,m}[c]} \binom{n+m}{h-1} g(c, h, \vec{K}_{n,m}) \left(\frac{L-r_c D}{D}\right)^{n+m+1-h} u[L-r_c D] \quad (\text{V.97})$$

A equação de acima tem a mesma estrutura que a equação II.75, portanto a seguir procedemos em forma analoga à seção II.5. Invertendo a ordem dos somatórios:

$$P[F_j^*(n, m, D) < L] = \sum_{c=0}^n \sum_{h=1}^{\vec{K}_{n,m}[c]} \sum_{\substack{\vec{K}_{n,m} \in \mathcal{K}_{n,m} \\ K_{n,m}[c] \geq h}} \binom{n+m}{h-1} \Omega_j[\vec{K}_{n,m}] g(c, h, \vec{K}_{n,m}) \left(\frac{L-r_c D}{D}\right)^{n+m+1-h} u[L-r_c D] \quad (\text{V.98})$$

Por outro lado, a equação V.46 e a definição de $\mathcal{K}_{n,m}$, implicam que:

$$1 \leq \vec{K}_{n,m}[0] \leq n+m+1 \quad ; \quad 0 \leq \vec{K}_{n,m}[c] \leq m+1 \quad , \quad 1 \leq c \leq m \quad (\text{V.99})$$

Portanto, isolando as expressões para $\{c=0, \vec{K}_{n,m} = (n+m+1)\vec{1}_0^{(n)}\}$ e $\{c=0, \vec{K}_{n,m} \ll (n+m+1)\vec{1}_0^{(n)}\}$, V.99 permite reescrever V.98 da forma:

$$\begin{aligned}
P[F_j^*(n, m, D) < L] = & \\
& \Omega_j[(n+m+1)\vec{1}_0^{(n)}] g(0, n+m+1, (n+m+1)\vec{1}_0^{(n)}) u[L-r_0D] \\
& + \sum_{h=1}^{n+m} \binom{n+m}{h-1} \sum_{\substack{\vec{K}_{n,m} \in \mathcal{K}_{n,m} \\ (n+m) \geq K_{n,m}[0] \geq h}} \Omega_j[\vec{K}_{n,m}] g(0, h, \vec{K}_{n,m}) \left(\frac{L-r_0D}{D}\right)^{n+m+1-h} u[L-r_0D] \\
& + \sum_{c=1}^n \sum_{h=1}^{m+1} \binom{n+m}{h-1} \sum_{\substack{\vec{K}_{n,m} \in \mathcal{K}_{n,m} \\ K_{n,m}[c] \geq h}} \Omega_j[\vec{K}_{n,m}] g(c, h, \vec{K}_{n,m}) \left(\frac{L-r_cD}{D}\right)^{n+m+1-h} u[r-r_cD]
\end{aligned} \tag{V.100}$$

Da mesma maneira que na seção II.5.2, o fato de $\Omega_j[\vec{K}_{n,m}]$ e $g(c, \vec{K}_{n,m})$ serem funções recorrentes em $\vec{K}_{n,m}$, permite achar uma expressão recorrente para o valor dos somatórios sob $\vec{K}_{n,m}$ em V.100. Mais ainda, também da mesma maneira que no capítulo II, é possível definir **grupos** de termos de $\Omega_j[\vec{K}_{n,m}]$ e $g(c, \vec{K}_{n,m})$ de forma tal que estes grupos sejam recorrentes nos valores de n e m . A seguir definem-se algumas agrupações de termos que permitem avaliar os somatórios de V.100 recorrentemente.

$$\alpha_j^{(n,m)} = \Omega_j[(n+m+1)\vec{1}_0^{(n)}] g(0, n+m+1, (n+m+1)\vec{1}_0^{(n)}) \tag{V.101}$$

$$\beta_j^{(n,m)}(h) = \sum_{\substack{\vec{K}_{n,m} \in \mathcal{K}_{n,m} \\ n+m \geq K_{n,m}[0] \geq h \geq 1}} \Omega_j[\vec{K}_{n,m}] g(0, h, \vec{K}_{n,m}) \tag{V.102}$$

$$\gamma_j^{(n,m)}(c, h) = \sum_{\substack{\vec{K}_{n,m} \in \mathcal{K}_{n,m} \\ m+1 \geq K_{n,m}[c] \geq h \geq 1}} \Omega_j[\vec{K}_{n,m}] g(c, h, \vec{K}_{n,m}) \tag{V.103}$$

Note-se que a definição de $\alpha_j^{(n,m)}$ implica que os $(n+m+1)$ estados de $\vec{K}_{n,m}$ tenham associada recompensa r_0

Voltando ao cálculo de $P[F_j^*(n, m, D) < L]$, os termos definidos em V.101 ... V.103, permitem reescrever V.100 da forma:

$$P[F_j^*(n, m, D) < L] = \alpha_j^{(n,m)} u[L-r_0D]$$

$$\begin{aligned}
& + \sum_{h=1}^{n+m} \binom{n+m}{h-1} \beta_j^{(n,m)}(h) \left(\frac{L-r_0D}{D} \right)^{n+m+1-h} u[L-r_0D] \\
& + \sum_{c=1}^n \sum_{h=1}^{m+1} \binom{n+m}{h-1} \gamma_j^{(n,m)}(c,h) \left(\frac{L-r_cD}{D} \right)^{n+m+1-h} u[L-r_cD]
\end{aligned} \tag{V.104}$$

A equação V.104 é a mesma que a V.64, o único diferente é que, nesta seção, ela foi obtida em forma muito simples, diretamente de argumentos probabilísticos, sem utilizar do recurso da transformada de Laplace. Uma consequência do fato que V.104 é igual a V.94, é que evidentemente deve acontecer que as expressões para os diferentes termos α 's, β 's e γ 's de V.104 devem ser as mesmas que as das equações V.73 ... V.90. Mais ainda, a expressão para $P[F_j^*(n, m, D) < L]$ (obtida pelo Método 3) deve ser igual à da equação V.94. Porém, apesar desta observação e com o objetivo de explicitar mais o Método 3, a seguir obtém-se as equações que permitem calcular em forma recorrente alguns dos diferentes termos $\alpha_j^{(n,m)}$, $\beta_j^{(n,m)}(h)$, $\gamma_j^{(n,m)}(c, h)$, o que será feito analisando os diferentes casos possíveis. Para simplificar a leitura, em primeiro lugar reproduzem-se, em forma apropriada ao presente caso, as equações que permitem avaliar recorrentemente alguns dos diferentes termos $\Omega_j[\vec{K}_{n,m}]$ e $g(c, h, \vec{K}_{n,m})$. Para isto é necessário considerar que as equações recorrentes dos termos $g(c, h, \vec{K}_n)$ do capítulo II dependem da recompensa do primeiro segmento. Além do mais, no capítulo II, dados \vec{K}_n e a recompensa do primeiro segmento, fica evidente qual é o vetor \vec{K}_{n-1} das equações recorrentes. Porém no presente caso podem acontecer diversas situações, que se excluem umas a outras. Estas situações acontecem dependendo da recompensa do primeiro segmento (neste caso, a variável I_j , identifica o tipo de situação) e de se o tipo do primeiro evento de \mathcal{W}_D pertence a $\mathcal{N}(D)$ ou a $\mathcal{M}(D)$ (situação identificada pela variável I_p). Portanto, considerando os diferentes casos possíveis, as equações II.70 ... II.74 ficam da seguinte forma:

$$g(\psi(e_j), 1, \vec{K}_{0,0}) = I[\vec{K}_{0,0} = \vec{1}_0^{(0)}] \tag{V.105}$$

$$\begin{aligned}
g(\psi(e_j), h, \vec{K}_{n,m}) & = I_p g(\psi(e_j), h-1, \vec{K}_{n-1,m}) + \bar{I}_p \bar{I}_j g(\psi(e_j), h-1, [\vec{K}_{n,m} - \vec{1}_0^{(n)}]) \\
& \quad 2 \leq h \leq \vec{K}_{n,m}[\psi(e_j)] \\
& + \bar{I}_p I_j g(\psi(e_j), h-1, [\vec{K}_{n,m} - \vec{1}_n^{(n)}])
\end{aligned} \tag{V.106}$$

$$\begin{aligned}
g(\psi(e_j), 1, \vec{K}_{n,m}) & = - \sum_{c=1}^{|\mathcal{R}|} g(c, 1, \vec{K}_{n,m}) \\
& \quad n+m \geq 1 \\
& \quad c \neq \psi(e_j); \vec{K}_{n,m}[c] > 0
\end{aligned} \tag{V.107}$$

$$\begin{aligned}
g(c, \vec{K}_{n,m}[c], \vec{K}_{n,m}) &= -I_p \frac{g(c, \vec{K}_{n-1,m}[c], \vec{K}_{n-1,m})}{(r_c - r_{\psi(e_j)})} \\
&\quad \substack{c \neq \psi(e_j) \\ n+m \geq 1} \\
&\quad - \bar{I}_p \bar{I}_j \frac{g(c, \vec{K}_{n,m}[c], [\vec{K}_{n,m} - \bar{1}_0^{(n)}])}{(r_c - r_{\psi(e_j)})} \\
&\quad - \bar{I}_p I_j \frac{g(c, \vec{K}_{n,m}[c], [\vec{K}_{n,m} - \bar{1}_n^{(n)}])}{(r_c - r_{\psi(e_j)})} \quad (V.108)
\end{aligned}$$

Na equação de acima, devido a que $c \neq \psi(e_j)$, cumpre-se que $\vec{K}_{n,m}[c] = \vec{K}_{n-1,m}[c] = \vec{K}'[c]$, onde $\vec{K}' = (\vec{K}_{n,m} - \bar{1}_{\psi(e_j)}^{(n)})$.

$$\begin{aligned}
g(c, h, \vec{K}_{n,m}) &= \frac{g(c, h+1, \vec{K}_{n,m})}{(r_c - r_{\psi(e_j)})} - I_p \frac{g(c, h, \vec{K}_{n-1,m})}{(r_c - r_{\psi(e_j)})} \\
&\quad \substack{c \neq \psi(e_j) \\ 1 \leq h < \vec{K}_{n,m}[c] \\ n+m \geq 1} \\
&\quad - \bar{I}_p \bar{I}_j \frac{g(c, h, [\vec{K}_{n,m} - \bar{1}_0^{(n)}])}{(r_c - r_{\psi(e_j)})} - \bar{I}_p I_j \frac{g(c, h, [\vec{K}_{n,m} - \bar{1}_n^{(n)}])}{(r_c - r_{\psi(e_j)})} \quad (V.109)
\end{aligned}$$

$$\Omega_j[\vec{K}_{0,m}] = I[\vec{K}_{0,m} = (m+1)\bar{1}_0^{(0)}], \quad \text{para todo } e_j \in E \quad (V.110)$$

$$\Omega_j[\vec{K}_{n,m}] = \frac{n}{n+m} \Omega_j[\vec{K}_{n-1,m}] + \frac{m}{n+m} \sum_{e_k \in E} p_{jk} \left\{ \bar{I}_j \Omega_k[\vec{K}_{n,m} - \bar{1}_0^{(n)}] + I_j \Omega_k[\vec{K}_{n,m} - \bar{1}_n^{(n)}] \right\} \quad (V.111)$$

Onde $\vec{K}_{n,m}$ é obtido recorrentemente da forma:

$$\vec{K}_{n,m} = \begin{cases} I_p \left(\langle I_j \rangle \oplus \left(\vec{K}_{n-1,m} + \bar{I}_j \bar{1}_0^{(n-1)} \right) \right) \\ \bar{I}_j \bar{I}_p \left(\vec{K}_{n,m-1} + \bar{1}_0^{(n)} \right) + I_j \bar{I}_p \left(\vec{K}_{n,m-1} + \bar{1}_n^{(n)} \right) \end{cases} \quad (V.112)$$

Avaliação de $\alpha_j^{(n,m)}$, $\beta_j^{(n,m)}(h)$ e $\gamma_j^{(n,m)}(c, h)$.

A seguir são obtidas expressões recorrentes para todos os termos possíveis deste tipo. Estas expressões são obtidas a partir das equações V.101 ... V.112. Analisando os diferentes casos possíveis, tem-se que:

Caso $\alpha_j^{(0,0)}$: Por definição cumpre-se que: $\alpha_j^{(0,0)} = \Omega_j [1_0^{(0)}] g(0, 1, 1_0^{(0)})$, portanto, usando V.105 e V.110 conclue-se que:

$$\alpha_j^{(0,0)} = 1 \quad (\text{V.113})$$

Caso $\alpha_j^{(n,m)}$, $n + m > 1$: Usando V.106 e V.111 em V.101:

$$\alpha_j^{(n,m)} = \left\{ \frac{n}{n+m} \Omega_j [(n+m) \vec{1}_0^{(n-1)}] + \bar{I}_j \frac{m}{n+m} \sum_{e_k \in E} p_{jk} \Omega_k [(n+m) \vec{1}_0^{(n)}] \right\} \\ \left\{ I_p \bar{I}_j g(0, n+m, (n+m) \vec{1}_0^{(n-1)}) + \bar{I}_p \bar{I}_j g(0, n+m, (n+m) \vec{1}_0^{(n)}) \right\} \quad (\text{V.114})$$

Na equação de acima não aparecem os termos de V.106 e V.110 que incluem o vetor $[\vec{K}_{n,m} - \vec{1}_n^{(n)}]$, pois ele tem um componente negativo, o que é impossível de acontecer. Tal como foi comentado na equação V.55 (que corresponde à primeira expressão entre $\{ \}$ de acima), o termo que contém o fator $\frac{n}{n+m}$ corresponde ao caso I_p , e o termo que tem o fator $\frac{m}{n+m}$ corresponde ao caso \bar{I}_p , portanto:

$$\alpha_j^{(n,m)} = \bar{I}_j \frac{n}{n+m} \Omega_j [(n+m) \vec{1}_0^{(n-1)}] g(0, n+m, (n+m) \vec{1}_0^{(n-1)}) \\ + \bar{I}_j \frac{m}{n+m} \sum_{e_k \in E} p_{jk} \Omega_k [(n+m) \vec{1}_0^{(n)}] g(0, n+m, (n+m) \vec{1}_0^{(n)}) \quad (\text{V.115})$$

Novamente, como em V.55, o termo com fator $\frac{n}{n+m}$ corresponde ao caso I_p e o termo com o fator $\frac{m}{n+m}$ corresponde ao caso \bar{I}_p . Agora, reconhecendo os termos de acima em V.101 :

$$\alpha_j^{(n,m)} = \bar{I}_j \left[\frac{n}{n+m} \alpha_j^{(n-1,m)} + \frac{m}{n+m} \sum_{e_k \in \mathcal{I}} p_{jk} \alpha_k^{(n,m-1)} \right] \quad (\text{V.116})$$

Onde foi usado o fato que em V.115 a única forma que $\vec{K}_{n,m-1} = (n+m) \vec{1}_0^{(n)}$, é que $e_k \in \mathcal{I}$. Isto é: se todos os $(n+m)$ segmentos estão em estados com recompensa r_0 , então o estado do primeiro segmento *deve* ter recompensa r_0 . Por outro lado, V.116 implica que $I_j \alpha_j^{(n,m)} = 0$, pois V.115 requer que os $(n+m+1)$ segmentos estejam em estados com recompensa r_0 .

Caso $\beta_j^{(n,m)}(n+m)$, $n+m \geq 2$. De V.102 ve-se que os únicos vetores que satisfazem a condição do somatório, são aqueles do tipo $\vec{K}_{n,m} = (n+m)\vec{I}_0^{(n)} + \vec{I}_c^{(n)}$, com $c \neq 0$. Mais ainda, caso $e_j \in \mathcal{U}$, cumpre-se que $\vec{K}_{n,m} = (n+m)\vec{I}_0^{(n)} + \vec{I}_n^{(n)}$; e caso $e_j \in \mathcal{I}$, acontece que $\vec{K}_{n,m} = (n+m)\vec{I}_0^{(n)} + \vec{I}_c^{(n)}$ com $c \neq 0, n$. Portanto, usando V.102, V.106, V.108 e V.111:

$$\begin{aligned}
\beta_j^{(n,m)}(n+m) &= \sum_{\substack{(n+m)\vec{I}_0^{(n)} + \vec{I}_c^{(n)} \\ \neq 0}} \left\{ I_j \frac{n}{n+m} \Omega_j \left[(n+m)\vec{I}_0^{(n-1)} + \vec{I}_n^{(n)} - \vec{I}_n^{(n)} \right] \right. \\
&+ \bar{I}_j \frac{n}{n+m} \Omega_j \left[(n+m-1)\vec{I}_0^{(n-1)} + \vec{I}_c^{(n-1)} \right] \\
&+ \bar{I}_j \frac{m}{n+m} \sum_{e_k \in E} p_{jk} \Omega_k \left[(n+m-1)\vec{I}_0^{(n)} + \vec{I}_c^{(n)} \right] \\
&+ \left. I_j \frac{m}{n+m} \sum_{e_k \in E} p_{jk} \Omega_k \left[(n+m)\vec{I}_0^{(n)} + \vec{I}_n^{(n)} - \vec{I}_n^{(n)} \right] \right\} \\
&\left\{ \bar{I}_j \left[I_p g \left(0, n+m-1, (n+m-1)\vec{I}_0^{(n-1)} + \vec{I}_c^{(n-1)} \right) \right. \right. \\
&\quad \left. \left. + \bar{I}_p g \left(0, n+m-1, (n+m-1)\vec{I}_0^{(n)} + \vec{I}_c^{(n)} \right) \right] \right. \\
&\left. - \frac{I_j}{(r_0 - r_n)} \left[I_p g \left(0, n+m-1, (n+m)\vec{I}_0^{(n-1)} + \vec{I}_n^{(n)} - \vec{I}_n^{(n)} \right) \right. \right. \\
&\quad \left. \left. + \bar{I}_p g \left(0, n+m-1, (n+m)\vec{I}_0^{(n)} + \vec{I}_n^{(n)} - \vec{I}_n^{(n)} \right) \right] \right\} \tag{V.117}
\end{aligned}$$

Efetuando a multiplicação de acima e considerando que não podem acontecer vetores com algum componente negativo, conclue-se que:

$$\begin{aligned}
\beta_j^{(n,m)}(n+m) &= \sum_{\substack{(n+m)\vec{I}_0^{(n)} + \vec{I}_c^{(n)} \\ \neq 0}} \left\{ -\frac{n}{n+m} \frac{I_j}{(r_0 - r_n)} \Omega_j \left[(n+m)\vec{I}_0^{(n-1)} \right] g \left(0, n+m-1, (n+m)\vec{I}_0^{(n-1)} \right) \right. \\
&+ \frac{n}{n+m} \bar{I}_j \Omega_j \left[(n+m-1)\vec{I}_0^{(n-1)} + \vec{I}_c^{(n-1)} \right] g \left(0, n+m-1, (n+m-1)\vec{I}_0^{(n-1)} + \vec{I}_c^{(n-1)} \right) \\
&+ \frac{m}{n+m} \bar{I}_j \sum_{e_k \in E} p_{jk} \Omega_k \left[(n+m-1)\vec{I}_0^{(n)} + \vec{I}_c^{(n)} \right] g \left(0, n+m-1, (n+m-1)\vec{I}_0^{(n)} + \vec{I}_c^{(n)} \right) \\
&\left. - \frac{m}{n+m} \frac{I_j}{(r_0 - r_n)} \sum_{e_k \in E} p_{jk} \Omega_k \left[(n+m)\vec{I}_0^{(n)} \right] g \left(0, n+m-1, (n+m)\vec{I}_0^{(n)} \right) \right\} \tag{V.118}
\end{aligned}$$

Agora, reconhecendo termos em V.101 e V.102 temos:

$$\beta_j^{(n,m)}(n+m) = \frac{I_j}{(r_0 - r_n)} \frac{-n}{n+m} \alpha_j^{(n-1,m)} + \bar{I}_j \frac{n}{n+m} \beta_j^{(n-1,m)}(n+m-1)$$

$$+\bar{I}_j \frac{m}{n+m} \sum_{e_k \in E} p_{jk} \beta_k^{(n,m-1)}(n+m-1) - \frac{I_j}{(r_0 - r_n)} \frac{m}{n+m} \sum_{e_k \in E} p_{jk} \alpha_k^{(n,m-1)} \quad (\text{V.119})$$

Lembrando que $I_j \alpha_j^{(n,m)} = 0$, ve-se que o primeiro termo de acima é igual a zero e o último somatório tem termos significativos somente para $e_k \in \mathcal{I}$, portanto:

$$\begin{aligned} \beta_j^{(n,m)}(n+m) &= \\ n+m \geq 2 & \\ \bar{I}_j & \left[\frac{n}{n+m} \beta_j^{(n-1,m)}(n+m-1) + \frac{m}{n+m} \sum_{e_k \in E} p_{jk} \beta_k^{(n,m-1)}(n+m-1) \right] \\ - \frac{m}{(n+m)} \frac{I_j}{(r_0 - r_n)} & \sum_{e_k \in E} p_{jk} \alpha_k^{(n,m-1)} \end{aligned} \quad (\text{V.120})$$

Caso $\gamma_j^{(n,m)}(n, m+1)$; $m \geq 1$: De V.103 ve-se que o único vetor que satisfaz a condição do somatório é $\vec{K}_{n,m} = (m+1)\vec{1}_n^{(n)} + n\vec{1}_0^{(n)}$, o que implica que: $\vec{K}_{n-1,m} = \Phi$ e $(\vec{K}_{n,m} - \vec{1}_n^{(n)}) = m\vec{1}_n^{(n)} + n\vec{1}_0^{(n)}$. Portanto, de V.103, V.106 e V.111:

$$\begin{aligned} \gamma_j^{(n,m)}(n, m+1) &= \frac{m}{n+m} I_j \bar{I}_p \sum_{e_k \in E} p_{jk} \Omega_k \left[\vec{K}_{n,m} - \vec{1}_n^{(n)} \right] g(n, m, \vec{K}_{n,m} - \vec{1}_n^{(n)}) \\ m \geq 1 & \\ & \frac{m}{n+m} I_j \sum_{e_k \in E} p_{jk} \Omega_k \left[\vec{K}_{n,m} - \vec{1}_n^{(n)} \right] g(n, m, \vec{K}_{n,m} - \vec{1}_n^{(n)}) \end{aligned} \quad (\text{V.121})$$

Agora, reconhecendo termos em V.103:

$$\gamma_j^{(n,m)}(n, m+1)_{m \geq 1} = I_j \frac{m}{n+m} \sum_{e_k \in E} p_{jk} \gamma_j^{(n,m-1)}(n, m) \quad (\text{V.122})$$

Os demais termos $\beta_j^{(n,m)}(h)$, $\gamma_j^{(n,m)}(c, h)$ se encontram em forma análoga aos já calculados, só que este trabalho é tedioso, cansativo e não fornece novos antecedentes, já que, tal como foi comentado anteriormente, as expressões recorrentes para **todos** os termos α 's, β 's e γ 's já foram encontrados e aparecem nas equações V.73 ... V.90. Adicionalmente, o fato que as equações V.104 e V.94 sejam iguais, implica que a expressão para $P[F_j^*(n, m, D) < L]$ encontrada pelos métodos 2 e 3 também são iguais, e é a que aparece na equação V.94.

Por último, para completar a análise feita com os métodos 2 e 3, a seguir analisa-se o custo computacional associado com a avaliação numérica de $FA_i(D, a)$ para o caso em que é usado o resultado comum destes métodos.

V.6.4 Complexidade

Custo de memória.

O maior custo de memória necessário para calcular a função de distribuição da disponibilidade é o que se precisa para armazenar os diferentes termos $\alpha_j^{(n)}$, $\beta_j^{(n,m)}(h)$, $\gamma_j^{(n,m)}(c, h)$. De V.95 ve-se que valor deste custo é dominado pela quantidade de memória necessária para armazenar os diferentes termos $\gamma_j^{(n,m)}(c, h)$ que é $O(|E|N_{max}(N_{max} + M_{max}))$, onde N_{max} e M_{max} são os máximos valores de n e m respectivamente, e estes estão dados pela técnica de reandomização.

Número de operações.

Nas equações V.74 ... V.91 ve-se que no passo (n, m) do algoritmo, para calcular os termos $\alpha_j^{(n)}$, $\beta_j^{(n,m)}(h)$ e $\gamma_j^{(n,m)}(c, h)$, são necessárias $O(|E| + n)$ operações de multiplicação ou divisão. Portanto considerando que $1 \leq n \leq N_{max}$, o maior número destas operações empregadas na avaliação de $P[F_j^*(n, m, D) < L]$ é $O(\{|E| + N_{max}^2\}\{|E|N_{max}(N_{max} + M_{max})\})$. Por outro lado, para calcular $FA_i(D, a)$, além de avaliar $P[F_j^*(n, m, D) < L]$, é preciso avaliar $P[\chi_1(C^*) = e_j | \chi_1(0) = e_i]$. De V.26 tem-se que o cálculo desta probabilidade requer de $O(|E|dN_{2max})$ operações de multiplicação, onde d é o máximo número de transições fora de um estado da CM que modela o sistema (em geral $d \ll E$, pois a CM normalmente é esparsa) e N_{2max} é o máximo número de falhas (para uma determinada tolerância no erro) que podem acontecer no intervalo $[0, C^*]$. Resumindo: o número de operações de multiplicação ou divisão necessárias para avaliar $FA_i(D, a)$ é $O(|E|dN_{2max} + \{|E| + N_{max}^2\}\{|E|N_{max}(N_{max} + M_{max})\})$.

Capítulo VI

Conclusões e Trabalhos Futuros

VI.1 Conclusões

Nesta tese foi encontrado um novo método para o cálculo da função de distribuição da CLEO de um conjunto de v.a.'s uniformes. Com base neste resultado foi obtido, usando argumentos probabilísticos, o resultado de [9] para $F_i(t, r)$ que é o resultado mais eficiente encontrado até agora para esta medida.

Também foi encontrada uma nova expressão para a função de desempenho $G_i(t, r)$. Esta expressão é muito mais eficiente que as anteriormente publicadas na literatura especializada.

Aproveitando que o cálculo de $G_i(t, r)$ é mais eficiente de realizar computacionalmente do que calcular $F_i(t, r)$, foi proposto um método para aproximar $F_i(t, r)$ a partir de: $G_i(t, r)$ e uma atribuição de recompensas apropriada às transições da CM que modela o sistema. A aproximação proposta permite calcular $F_i(t, r)$ com um custo computacional menor do que o requerido quando se usa o resultado de [9]. Além do mais, o erro de aproximação pode ser especificado a priori e portanto ser tão pequeno quanto desejarmos. Mais ainda, existe um compromisso entre a precisão obtida e os requisitos computacionais necessários para chegar a esta precisão.

Também foi analisada a técnica de gravar periodicamente o estado de um sistema de Banco de Dados. Esta técnica é normalmente usada para manter a integridade dos dados. Um resultado importante deste estudo foi demonstrar que o cálculo da disponibilidade pode ser identificado como um problema da área de desempenho. Outro resultado foi calcular o valor médio e a função de distribuição da disponibilidade. Neste caso o modelo usado para representar o sistema é mais realista que aqueles tratados analiticamente na literatura; a diferença mais importante com respeito aos modelos da literatura é que os tempos de reparo do Banco de Dados são representados em forma exata, o qual é muito importante segundo foi indicado em [26].

A seguir são indicadas algumas possíveis extensões do trabalho realizado.

VI.2 Trabalhos Futuros

A função de distribuição das estatísticas de ordem de n v.a's uniformes no intervalo $[0, t]$ (ou seja $F(\vec{a}_n, t, r)$), é uma medida que aparece repetidas vezes na área de Desempenhabilidade. Por exemplo, nesta tese foi usada para o calcular a função de distribuição da desempenhabilidade, na qual existe uma taxa de recompensas associada aos estados da CM que modela o sistema; também ela foi usada para calcular a função de distribuição da disponibilidade de um sistema de banco de dados submetido a *checkpoints* periódicos. Como resultado desta dependência, o custo computacional necessário para avaliar estas medidas aumenta com o custo associado ao cálculo da CLEO de n v.a's uniformes em $[0, t]$. Portanto, encontrar novos métodos que permitam avaliar mais eficientemente a citada combinação linear, permitirá atingir uma maior eficiência no cálculo das mencionadas medidas de desempenhabilidade, e possivelmente de outro tipo de medidas desta área. Uma forma de atingir o desejado aumento de eficiência, seria obter uma expressão analítica para $F(\vec{a}_n, t, r)$ empregando o mesmo método da seção II.4.2, isto é: resolvendo as expressões recorrentes obtidas no capítulo II para a CLEO de n v.a's uniformes em $[0, t]$.

Outra alternativa para conseguir mais eficiência no cálculo de medidas de desempenhabilidade, é encontrar métodos aproximados para avaliar a função de distribuição da CLEO de n v.a's uniformes em $[0, t]$. Para que isto seja conveniente, os métodos devem ser mais eficientes que os métodos exatos ou aproximados usados atualmente. Por exemplo em [20] aparece uma aproximação da CLEO de n v.a's uniformes em $[0, t]$ usando a v.a Normal. Esta aproximação tem a vantagem de que entrega um limite quanto ao erro obtido e pode permitir calcular medidas de desempenhabilidade em forma aproximada, porém com limite em quanto ao erro obtido.

Outra possível extensão da tese seria calcular o valor médio (e possivelmente a função de distribuição) do tempo de resposta às transações de um sistema de Banco de Dados que opera sob a técnica de CRR. A forma de resolver este problema seria mapeá-lo num problema equivalente da área de desempenhabilidade. O mapeamento deve ser feito em forma análoga ao método empregado para calcular a FDP da disponibilidade do modelo, isto é: cada unidade de tempo que as transações estão no sistema, deve corresponder a uma determinada recompensa associada ao estado em que este se encontra. A dificuldade de encontrar estas recompensas consiste em representar corretamente o efeito das operações de reparo.

A técnica de *Checkpointing* também tem sido empregada em aplicações diferentes à de CRR [2, 1, 10],[17],[29, 11]. Portanto outra possível extensão dos resultados do capítulo V, seria usar a metodologia desse capítulo para calcular

medidas que sejam pertinentes às diferentes aplicações onde é usada a técnica de *Checkpointing*.

Referências Bibliográficas

- [1] N. Bowen and D. Pradham. Virtual checkpoints: architecture and performance. *IEEE Transactions on Computers*, C-41(5), May 1992.
- [2] S. D. Carson and S. Setia. Analysis of the periodic update write policy for disk cache. *IEEE Transactions on Software Engineering*, 18(1), January 1992.
- [3] E. Çinlar. *Introduction to Stochastic Process*. Prentice Hall, 1975.
- [4] B. Cinciani and V. Grassi. Performability evaluation of fault-tolerant satellite systems. *IEEE Transactions on Communications*, COM-35(4):403–409, April 1987.
- [5] E. de Souza e Silva and R. Gail. Calculating availability and performability measures of reparable computer systems using randomization. *Journal of the ACM*, 36(1):171–193, January 1989.
- [6] E. de Souza e Silva and R. Gail. Calculating cumulative operational time distributions of reparable computer systems. *IEEE Transactions on Computer*, C-35(4):322–332, April 1986.
- [7] E. de Souza e Silva and R. Gail. *Performability Analysis of Computer Systems: from Models Specification to Solution*. Technical Report RC 16961 (# 75239), IBM, 1991.
- [8] A.P. Dempster and M. Kleyle. Distributions determined by cutting a simplex with hiperplanes. *The Annals of Mathematical Statistics*, 39(5):1473–1478, 1968.
- [9] L. Donatiello and V. Grassi. On evaluating the cumulative performance distribution of fault-tolerant computer systems. *IEEE Transactions on Computer*, 40(11):1301–1307, November 1991.
- [10] E. Elnozahy and W. Zwaenepoel. Manetho: transparente roll-back-recovery with low overhead, limited rollback and fast output commit. *IEEE Transactions on Computers*, C-41(5), May 1992.
- [11] J. Gait. A checkpointing page store for write-once optical disk. *IEEE Transactions on Computers*, 39(1), January 1990.

- [12] E. Gelembé. On the optimum checkpoint interval. *Journal of the ACM*, 26(2), April 1979.
- [13] E. Gelembé and D. Derochette. Performance of rollback recovery systems under intermittent failures. *Commun. ACM*, 21(6), 1978.
- [14] E. Gelembé and M. Hernandez. Optimum checkpoints with age dependent failures. *Acta Informática*, 27, 1990.
- [15] V. Grassi, L. Donatiello, and S. Tucci. Availability distribution on roll-back recovery systems. In O. J. Boxma G. Iazeolla, P. J. Courtois, editor, *Computer Performance and Reliability*, Elsevier Science Publishers, 1988.
- [16] V. Grassi, L. Donatiello, and S. Tucci. On the optimal checkpointing of critical tasks and transaction-oriented systems. *IEEE Transactions on Software Engineering*, 18(1), 1992.
- [17] E. G. Coffman Jr. and E. N. Gilbert. Optimal strategies for scheduling checkpoints and preventive maintenance. *IEEE Transactions on Reliability*, 39(1), April 1990.
- [18] S. Karlin and H.M. Taylor. *A Second Course in Stochastic Processes*. Academic Press, 1981.
- [19] L. Kleinrock. *Queueing Systems. Volume I: Theory*. John Wiley and Sons, Inc., 1975.
- [20] T. Matsunawa. The exact and approximate distributions of linear combinations of selected order statistics from a uniform distribution. *Ann. Ins. Statist. Math*, 37:1–16, 1985.
- [21] J. F. Meyer. On evaluating the performability of degradable computing systems. *IEEE Transactions on Computers*, C-29(8):720–731, 1980.
- [22] J. F. Meyer. Performability modeling of distributed real-time systems. In *Mathematical Computer Performance and Reliability*, 1984.
- [23] J. F. Meyer, K.H. Muraldihar, and W.H. Sanders. Performability of a token bus network under transient fault conditions. In *Proc. 19th Int. Symp. on Fault-tolerant Computing*, July 1989.
- [24] V. Nicola. A single server queue with mixed types of interruptions. *Acta Inform.*, 23:465–486, 1986.
- [25] V. Nicola and F.J. Kylstra. A markovian model, with state-dependent parameters of a transactional system supported by checkpointing and recovery strategies. In K.M. Schultz P.J. Kuhn, editor, *Messung, Modellierung und Bewertung Von Rechensystemen*, Springer-Verlag, 1983.
- [26] V. Nicola and J. Van Spanje. Comparative analysis of different models of checkpointing and recovery. *IEEE Transactions on Software Engineering*, 16(8), August 1990.

- [27] T. Ramalingam. Symbolic computing the exact distributions of l-statistics from a uniform distribution. *Ann. Inst. Statist. Math*, 41(4):677–681, 1989.
- [28] S. M. Ross. *Introduction to Probability Models*. Academic Press, Inc., 1989.
- [29] K. Shin, T. Ling, and Y. Lee. Optimal checkpointing of real time tasks. *IEEE Transactions on Computers*, C-36(11), Nov. 1987.
- [30] T. Takano, T. Yamada, H. Kato, T. Tanaka, H. Ihara, N. Kanekawa, and H. Maejima. Longlife dependable computers for spacecrafts. In A. Avizienis and J.C. Laprie, editors, *Dependable Computing for Critical Applications*, Springer–Verlang Wien New York, 1991.
- [31] P. Traverse. Dependability of digital computers on board airplanes. In A. Avizienis and J.C. Laprie, editors, *Dependable Computing for Critical Applications*, Springer–Verlang Wien New York, 1991.
- [32] H. Weisberg. The distribution of linear combinations of order statistics from the uniform distribution. *The Annals of Mathematical Statistics*, 42(2):704–709, 1971.

Apêndice A

Glossário de Termos

FDP: função de distribuição de probabilidade.

fdp: função de densidade de probabilidade.

v.a.: variável aleatória.

CM: cadeia de Markov.

$\chi_1 = \{X_1(t), t \geq 0\}$: Processo homogêneo de Markov de tempo contínuo que descreve o comportamento do sistema.

$E = \{e_k, k = 1, 2, \dots, |E|\}$: Espaço finito de estado associado ao modelo do sistema.

$|E|$: número total de estados do sistema.

$T \subseteq E \times E$: conjunto de transições entre estados da CM que modela o sistema.

$t_{ij} \in T$: transição desde o estado $e_i \in E$ ao estado $e_j \in E$.

$|\vec{v}|$: soma dos componentes do vetor v .

$\vec{v}[c]$: c -ésimo componente do vetor v .

$M = [m_{ij}]$: forma de notação de uma matriz.

Q: matriz correspondente às taxas de transição do processo de Markov χ_1 .

$\rho(e_k)$: taxa de recompensa associada ao estado e_k .

$R = \{r_1, r_2, \dots, r_{|R|}\}$: conjunto de taxas de recompensas associadas aos diferentes estados de E .

$|R|$: número de taxas de recompensas diferentes associadas aos estados de E .
($|R| \leq |E|$).

$RTA_i(t)$: recompensa total ganha pelo sistema no intervalo $[0, t]$, quando este inicia sua operação no estado $e_i \in E$ e a atribuição de recompensas está associada aos estados de E .

$F_i(t, r)$: função de distribuição de probabilidade de $RTA_i(t)$. Em termos formais: $F_i(t, r) = P[RTA_i(t) \leq r]$.

$O_i(t)$: corresponde a $RTA_i(t)$ no caso particular em que $\rho(e_k) \in \{0, 1\}$, para todo $e_k \in E$.

$A_i(t)$: disponibilidade do sistema. Corresponde à fração do tempo de observação em que o sistema está em estados 'operacionais', em termos formais: $A_i(t) = \frac{O_i(t)}{t}$.

$\rho(t_{ij})$: recompensa (fixa) atribuída à transição $t_{ij} \in T$.

$\Theta = \{\theta_k, k = 1, 2, \dots, |\Theta|\}$: conjunto de recompensas diferentes associadas às transições de T .

$|\Theta|$: número de transições diferentes associadas as transições de T . Em geral cumpre-se que: $|\Theta| \leq |E|^2$.

$RTAD_i(t)$: recompensa total ganha pelo sistema no intervalo $[0, t]$, quando este inicia sua operação no estado $e_i \in E$ e atribuição de recompensas está associada às transições $\in T$.

$G_i(t, r) (= P[RTAD_i(t) \leq r])$: função de distribuição de probabilidade de $RTAD_i(t)$, em termos formais: $G_i(t, r) = P[RTAD_i(t) \leq r]$

Λ : valor (arbitrário) maior ou igual à máxima taxa de saída dos estados de Q , isto é: $\Lambda \geq \min_i \{q_{i|i}\}$.

P : matriz de transição de probabilidades, obtida a partir de Q da forma: $P = \frac{Q}{\Lambda} + I$, onde I corresponde à matriz identidade.

$N(= N(t))$: Processo de Poisson com parâmetro Λt . N é independente do processo χ_2 definido a seguir.

χ_2 : processo estocástico de tempo discreto, obtido a partir de χ_1 de forma tal que suas transições acontecem somente nos eventos do processo de Poisson N .

Randomização: teorema que estabelece que o processo estocástico χ_2 , obtido a partir de χ_1 da forma indicada acima, é equivalente ao processo estocástico χ_1 . Em termos formais: $Prob[\chi_1(t) = e_k] = Prob[\chi_2(N(t)) = e_k]$.

$\{U_1, U_2, \dots, U_n\}$: conjunto de variáveis aleatórias independentes, uniformemente distribuídas no intervalo $[0, t]$.

$\{U_{(1)}^m(t), U_{(2)}^m(t), \dots, U_{(n)}^m(t)\}$: conjunto de v.a's obtidas permutando os elementos do conjunto $\{U_i, 1 \leq i \leq n\}$, de forma de se obter uma ordenação crescente.

- $U_{(i)}^n(t)$: i -ésima estatística de ordem do conjunto $\{U_1, U_2, \dots, U_n\}$. Esta v.a. corresponde ao i -ésimo menor valor obtido na execução dos experimentos U_1, \dots, U_n .
- $U_{(i)}$: forma simplificada de escrever $U_{(i)}^n(t)$.
- $U_{(0)}$: valor constante igual a 0.
- $U_{(n+1)}$: valor constante igual a t .
- $\vec{a}_n = \langle a_1, a_2, \dots, a_{n+1} \rangle$: onde $\{a_k, k = 1, 3, \dots, n + 1\}$ é um conjunto de valores (constantes) pertencentes aos reais.
- $CLEO(\vec{a}_n, t)$: v.a. definida como $CLEO(\vec{a}_n, t) = \sum_{k=1}^{n+1} a_k U_{(k)}$, isto é: $CLEO(\vec{a}_n, t)$ corresponde a uma combinação linear do conjunto de v.a's $\{U_1, U_2, \dots, U_n\}$.
- $F(\vec{a}_n, t, r)$: FDP da v.a. $CLEO(\vec{a}_n, t)$, isto é: $F(\vec{a}_n, t, r) = P[CLEO(\vec{a}_n, t) \leq r]$.
- $V_{(k)}^n(t), 1 \leq k \leq n + 1$: Conjunto de v.a's definidas da forma $V_{(k)}^n(t) = U_{(k)}^n(t) - U_{(k-1)}^n(t)$.
- $V_{(k)}$: forma simplificada de escrever $V_{(k)}^n(t)$.
- $c_k, 1 \leq k \leq n + 1$: conjunto de valores (constantes) pertencentes aos reais, definidos como $c_k = \sum_{j=k}^{n+1} a_j$.
- $\vec{c}_n = \langle c_1, c_2, \dots, c_{n+1} \rangle$
- $u(x)$: função definida da forma: $u(x) = \begin{cases} 0, & \text{se } x < 0 \\ 1, & \text{se } x \geq 0 \end{cases}$
- $\delta(x)$: derivada de $u(x)$.
- $I(e)$: função definida da forma: $I(e) = \begin{cases} 1, & \text{se o evento } e \text{ for verdadeiro} \\ 0, & \text{em caso contrário} \end{cases}$
- \vec{K}_n : vetor cujo c -ésimo componente, isto é $\vec{K}_n[c]$, corresponde ao número de componentes do vetor \vec{c}_n que são iguais a $r_c \in R$. Em termos formais: $\vec{K}_n[c] = \sum_{l=1}^{n+1} I[\vec{c}_n[l] = r_c], 1 \leq c \leq |R|$. Note-se que da definição cumpre-se que $|\vec{K}_n| = n + 1$.
- $\vec{1}_c$: vetor de dimensão $|R|$, composto somente de zeros, exceto no c -ésimo componente, onde vale 1.
- Φ : função que indica a qual elemento do conjunto R corresponde cada um dos elementos do vetor \vec{c}_n , ou seja, $\Phi(c_i) = j$ se somente se $c_i = r_j$.
- $\Gamma_i[\vec{K}_n]$: probabilidade agregada de todos os caminhos da tamanho $(n + 1)$ que partem do estado e_i e que transitam por $\vec{K}_n[1]$ estados com recompensa r_1 , $\vec{K}_n[2]$ estados com recompensa $r_2, \dots, \vec{K}_n[|R|]$ estados com recompensa $r_{|R|}$.
- $\Psi(e_i)$: índice no conjunto R da recompensa associada ao estado e_i , isto é: $\Psi(e_i) = c$ se somente se $\rho(e_i) = r_c$.

\mathcal{R}_i^n : conjunto de todas as recompensas diferentes que pode acumular o sistema no caso em que: o estado inicial é e_i , $N(t) = n$ e o ganho de recompensas acontece somente quando o sistema muda de estado (recompensa associada às transições de estado).

$\mathcal{r}_i^n \subset \mathcal{R}_i^n$: subconjunto de \mathcal{R}_i^n composto por todas as recompensas $w \in \mathcal{R}_i^n$ tal que $w \leq r$.

$r_i(n)$: recompensa atribuída a cada visita que o sistema faz ao estado e_i , no caso em que $N(t) = n$. Esta recompensa é usada para calcular em forma aproximada $F_i(t, r)$, é tal que $r_i(n) = \frac{r_i t}{n+1}$.

$\Delta t = \frac{t}{n+1}$: tamanho médio da v.a. $V_{(k)}^n(t)$.

OBS: As variáveis definidas a seguir referem-se aum sistema de Banco de Dados submetido a *checkpoints* periódicos.

C: v.a. correspondente ao tempo de duração do *checkpoint* de um banco de dados.

D: período de tempo em que o sistema de banco de dados está disponível para atender transações num ciclo de operação.

RR: operações de retorno (*roll-back*) e recuperação (*recover*) feitas num banco de dados submetidos a *checkpoints* periódicos. Estas operações são feitas para reparar uma falha que acontece durante o período D.

τ_{fl} : v.a. correspondente ao instante de acontecimento da l-ésima falha acontecida durante uma operação de *checkpoint*. Este tempo é medido a partir do inicio da operação de *checkpoint*.

t_{fl} : v.a. correspondente ao acontecimento da l-ésima falha que ocorre durante o tempo disponível. Este tempo é medido a partir do término da operação de *checkpoint*.

v^* variável v associada à operação de um sistema de banco de dados submetido a *checkpoints* periódicos. O superíndice $*$ indica que para o cálculo da variável v , supõe-se que não acontecem falhas.

r_l : v.a. correspondente ao tempo necessário para efetuar a l-ésima operação de RR.

t_{rb}^* : tempo necessário para efetuar uma operação de *retorno*, dado que nesta operação não acontecem falhas. No modelo do capítulo V este tempo supõe-se constante.

t_{bl} : tempo necessário para efetuar a l-ésima operação de *recuperação* (*recover*).

x_l : tamanho do l-ésimo intervalo de tempo ocupado (*busy*) que acontece durante o intervalo D.

- $A_i(D)$: Disponibilidade do sistema, no caso em que ao início do ciclo ele está no estado e_i e o intervalo de tempo disponível no ciclo é constante, de valor D .
- γ_c : taxa constante de acontecimento de falhas durante uma operação de *checkpoint*.
- γ_d : taxa constante de acontecimento de falhas durante o intervalo de tempo disponível.
- $E[v]$: valor médio da v.a. v .
- $F_i^*(D)$: v.a. correspondente ao tempo total acumulado em operações de RR, dado que: o estado do sistema ao início do ciclo é e_i e o tamanho do tempo disponível é constante, igual a D .
- $F_{i,k}^n(D)$: tempo acumulado pelo sistema atendendo transações no intervalo $[0, U_{(k)}^n(D)]$ (medido a partir do final do *checkpoint*), no caso em que o estado ao início do ciclo é e_i e $U_{(k)}^n(D)$ é a k -ésima estatística de ordem de n v.a.'s independentes, uniformes em $[0, D]$.
- $\Omega_i[m, l]$: probabilidade que a CM randomizada transite por l estados em que o sistema está atendendo transações, no caso em que acontecem m transições e o estado inicial é o estado e_i .
- p_{ij}^l : probabilidade que a CM randomizada transite do estado $e_i \in E$ ao estado $e_j \in E$, em exatamente l passos.
- t_{fl} : instante pertencente ao período D em que acontece a l -ésima transição da CM randomizada.
- $w_l; 1 \leq l \leq n + 1$: v.a. correspondente ao intervalo D ; w_l corresponde ao tempo disponível transcorrido entre a $(l - 1)$ -ésima e a l -ésima falha acontecida no período D . Em termos formais: $w_l = [t_{fl-1}, t_{fl})$.
- $\mathcal{N}(D)$: processo de Poisson correspondente ao número de falhas do sistema durante o período de disponibilidade.
- $\mathcal{M}(D)$: processo de Poisson correspondente ao número de transições da CM randomizada que acontecem no período de disponibilidade.
- $\mathcal{W}(D)$ processo de Poisson correspondente à composição dos processos $\mathcal{M}(D)$ e $\mathcal{N}(D)$.
- $FA_i(D, a)$: função de distribuição da disponibilidade para um ciclo de operação, no caso em que o estado ao início do ciclo é e_i e o intervalo de tempo disponível é constante, igual a D .
- $r_j^{*l}(n, m)$: tempo gasto na l -ésima operação de RR, dado que $\chi_1(C^*) = e_j$, $\mathcal{N}(D) = n$, $\mathcal{M}(D) = m$ e nas operações de RR não podem acontecer falhas.
- $\mathcal{B}(v)$: v.a. correspondente ao tempo acumulado pelo sistema atendendo transações num certo intervalo $v \in D$.

$\rho^{(n,m)}(l_k, v_l)$: taxa de recompensa atribuída ao estado e_k , no caso em que o sistema está neste estado no segmento $v_l \in D$; com $v_l = [U_{(l-1)}^{n+m}(D), U_{(l)}^{n+m}(D)]$
 $1 \leq l \leq n + m + 1$.

$\vec{K}_{n,m}$: vetor de $(n + 1)$ componentes, da forma $\vec{K}_{n,m} = \langle k_n, k_{n-1}, \dots, k_0 \rangle$. Onde k_c ($0 \leq c \leq n$), corresponde ao número de segmentos v_l ($1 \leq l \leq n + m + 1$) nos quais o sistema ganha recompensa c .

$\mathcal{K}_{n,m}$: conjunto de todos os valores possíveis de $\vec{K}_{n,m}$.

$\Omega_j[\vec{K}_{n,m}]$: probabilidade que o sistema visite k_c ($0 \leq c \leq n$) estados com recompensa c , dadas as seguintes condições: $\mathcal{N}(D) = n$, $\mathcal{M}(D) = m$, e $\chi_1(C^*) = e_j$.

$U_{(1)}^1$: primeira estatística de ordem do processo $\mathcal{N}(D)$.

$U_{(1)}^2$: primeira estatística de ordem do processo $\mathcal{M}(D)$.

\oplus : operador de concatenação de dois vetores.

I_p : função definida como: $I_p = \begin{cases} 1, & \text{se o primeiro evento de } \mathcal{W}(D) \in \mathcal{N}(D) \\ 0, & \text{caso contrário} \end{cases}$

\bar{I}_p : função definida como: $\bar{I}_p = 1 - I_p$.