

DISSERTAÇÃO DE MESTRADO Nº 1109

**ANÁLISE DE DESEMPENHO DE SED VIA AUTÔMATOS MODULARES
ESTOCÁSTICOS: ABORDAGEM VIA SIMULAÇÃO COMPUTACIONAL E
MÉTODO ANALÍTICO**

Lisley Mara Guimarães de Souza

DATA DA DEFESA: 20/02/2019

Universidade Federal de Minas Gerais

Escola de Engenharia

Programa de Pós-Graduação em Engenharia Elétrica

**ANÁLISE DE DESEMPENHO DE SED VIA AUTÔMATOS
MODULARES ESTOCÁSTICOS: ABORDAGEM VIA SIMULAÇÃO
COMPUTACIONAL E MÉTODO ANALÍTICO**

Lisley Mara Guimarães de Souza

Dissertação de Mestrado submetida à Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Escola de Engenharia da Universidade Federal de Minas Gerais, como requisito para obtenção do Título de Mestre em Engenharia Elétrica.

Orientador: Prof. Carlos Andrey Maia

Belo Horizonte - MG

Fevereiro de 2019

S729a

Souza, Lisley Mara Guimarães de.

Análise de desempenho de SED via autômatos modulares estocásticos [manuscrito]: abordagem via simulação computacional e método analítico / Lisley Mara Guimarães de Souza. – 2019.
75 f., enc.: il.

Orientador: Carlos Andrey Maia.

Dissertação (mestrado) Universidade Federal de Minas Gerais, Escola de Engenharia.

Bibliografia: f. 74 - 75.

1. Engenharia elétrica - Teses. 2. Modelagem de sistemas – Teses.
3. Robôs – Teses. 4. Sistemas a Eventos Discretos (SED) – Teses.
I. Maia, Carlos Andrey. II. Universidade Federal de Minas Gerais. Escola de Engenharia. III. Título.

CDU: 621.3(043)

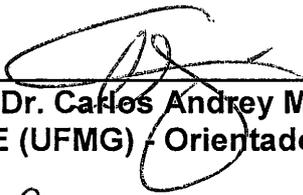
"Análise de Desempenho de Sed Via Autômatos Modulares Estocásticos: Abordagem Via Simulação Computacional e Método Analítico"

Lisley Mara Guimarães de Souza

Dissertação de Mestrado submetida à Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Escola de Engenharia da Universidade Federal de Minas Gerais, como requisito para obtenção do grau de Mestre em Engenharia Elétrica.

Aprovada em 20 de fevereiro de 2019.

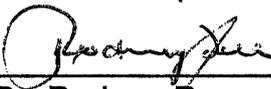
Por:



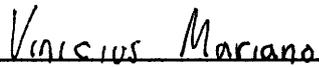
Prof. Dr. Carlos Andrey Maia
DEE (UFMG) - Orientador



Profa. Dr. Patricia Nascimento Pena
DELT (UFMG)



Prof. Dr. Rodney Rezende Saldanha
DEE (UFMG)



Prof. Dr. Vinicius Mariano Gonçalves
DEE (UFMG)



Dr. Roberto Gomes Ribeiro
ITV (VALE)

*Aos meus pais, Adair e Claudirene
Por vocês e para vocês. Sempre!*

Agradecimentos

Agradeço primeiramente a Deus e à Nossa Senhora por me guiar e proteger meus passos, sempre abençoando e iluminando minhas decisões.

Aos meus pais, Adair e Claudirene, por sempre me apoiarem e lutarem para que os meus sonhos fossem realizados; sem vocês nada disso seria possível.

Ao meu noivo Matheus, por todo amor e incentivo durante a caminhada; você foi essencial para que eu chegasse até aqui.

Ao meu orientador Professor Doutor Carlos Andrey Maia por todos os ensinamentos e conhecimentos transmitidos, por toda paciência e confiança em mim depositada para a realização deste trabalho.

Ao programa de pós-graduação em Engenharia Elétrica da UFMG, a todos os professores que contribuíram para o meu crescimento profissional e pessoal, em especial à Professora Doutora Patrícia Nascimento Pena por ter me apoiado e incentivado desde o início do mestrado. Ao CNPQ, pelo auxílio financeiro concedido durante o desenvolvimento deste trabalho.

A todos os amigos que fiz durante essa caminhada, em especial ao Marcelo que iniciou junto comigo, sempre companheiro nos momentos difíceis. Aos amigos do LACSED e da sala do mestrado, por todo o conhecimento e experiências compartilhadas.

A todos os amigos do GOU FAFICH, meu refúgio de paz e tranquilidade durante o mestrado. Às amigas de república que me acolheram e me fizeram sentir em casa: Tammy, Thais, Ana e Júlia; vocês foram a minha família.

Aos meus amigos e familiares, e a todos que torceram e me apoiaram. Serei eternamente grata!

"Não é sobre chegar no topo do mundo e saber que venceu. É sobre escalar e sentir que o caminho te fortaleceu. É sobre ser abrigo e também ter morada em outros corações. E assim ter amigos contigo em todas as situações..."

Trem Bala - Ana Vilela

"Só um sentido de invenção e uma necessidade intensa de criar levam o homem a revoltar-se, a descobrir e a descobrir-se com lucidez. "(Pablo Picasso)

Resumo

A tarefa de modelagem de sistemas, na área da engenharia, algumas vezes apresenta falhas por serem sistemas grandes e com diversas funcionalidades. Devido ao grande número de estados possíveis, torna-se complexa a elaboração de um modelo único que represente o comportamento do sistema como um todo. Por isso, técnicas com um caráter modular tornam-se uma alternativa atraente para a modelagem de sistemas. No contexto de Sistemas a Eventos Discretos (SED), é possível modelar sistemas utilizando autômatos, rede de Petri, álgebra Max-plus, entre outros; e trabalhar com sistemas de filas, cliente - servidor, manufatura, tráfego, e diversos outros. Este trabalho propõe uma metodologia que aborda a modelagem, a implementação e a análise dos resultados, para modelar e analisar sistemas de forma modular por meio de autômatos estocásticos. Além disso, são utilizadas propriedades da cadeia de Markov e da teoria de filas que tornam possível a análise de sistemas com grande complexidade computacional. Dessa forma, a mesma metodologia de modelagem é utilizada para análise por método analítico e por simulação. Um dos principais resultados deste trabalho é a demonstração da eficiência dos autômatos modulares estocásticos, devido à facilidade e à redução dos erros na etapa da modelagem. A metodologia foi aplicada a um sistema de veículos guiados automatizados e também a um sistema cliente-servidor com impaciência.

Palavras-chave: Autômatos, Avaliação de desempenho, Modelagem de sistemas, Sistemas a Eventos Discretos.

Abstract

The modeling of systems, in the engineering area, sometimes presents failures because they are large systems with many functionalities. Due to a large number of possible states, the development of a single model that represents the behavior of the system as a whole becomes complex. Therefore, techniques with a modular characteristic become an attractive alternative for modeling systems. In the context of Discrete Event Systems (DES), it is possible to model systems using automata, Petri Nets, max-plus algebra, among others; and to work with a queueing system, client-server, manufacturing, traffic, and many others. This work proposes a methodology that deals with the modeling, implementation, and analysis of the results in order to model and analyze systems in a modular way, by means of stochastic automata. In addition, properties of the Markov chain and queueing theory are used, which makes possible the analysis of systems with large computational complexity. Thus, the same modeling methodology is used for analysis through analytical and simulation methods. One of the main results of this work is the demonstration of the efficiency of stochastic modular automata due to the ease and reduction of errors in the modeling step. The methodology was applied to an automated guided vehicle system as well as a client-server system with impatience.

Keywords: Automata, Discrete Event Systems, Performance evaluation, System modeling.

Lista de ilustrações

Figura 2.1 – Sistema de filas	20
Figura 2.2 – Autômato finito	23
Figura 2.3 – Autômato para fila de capacidade 2	24
Figura 2.4 – Sistema de filas de forma modular	26
Figura 2.5 – Autômato para o sistema de fila de capacidade 2	26
Figura 2.6 – Autômato da composição paralela da Fila Servidor	27
Figura 2.7 – Cadeia de Markov do sistema da fila simples	31
Figura 2.8 – Sistema de filas simples	34
Figura 2.9 – Lei de Little	37
Figura 2.10–Sistema de filas $M/M/m$	38
Figura 2.11–Sistema de clientes - servidor com abandono	42
Figura 3.1 – Diagrama de blocos da metodologia	47
Figura 3.2 – Cadeia de Markov do sistema da fila simples	50
Figura 3.3 – Cadeia de Markov do sistema da fila simples sem os eventos instantâneos	50
Figura 3.4 – Autômatos para o sistema de fila de capacidade 2	52
Figura 3.5 – Autômatos para o sistema de fila de capacidade 2 com projeção inversa	53
Figura 4.1 – Trilho e estações de trabalho	55
Figura 4.2 – Autômato da esteira de entrada	57
Figura 4.3 – Autômatos das estações	58
Figura 4.4 – Autômatos das filas das estações	59
Figura 4.5 – Autômatos das falhas das estações	59
Figura 4.6 – Autômato esteira de entrada sem os eventos instantâneos	60
Figura 4.7 – Autômatos das filas das estações sem os eventos instantâneos	61
Figura 4.8 – Trilho para as posições das estações	61
Figura 4.9 – Variando Estação A	62
Figura 4.10–Variando Estação B	63
Figura 4.11–Variando Estação C	63
Figura 4.12–Produtividade do sistema	64

Figura 4.13–Cliente tipo 1	67
Figura 4.14–Cliente tipo 2	67
Figura 4.15–Servidores	67
Figura 4.16–Desempenho em função da taxa de serviço	70
Figura 4.17–Desempenho do sistema em função da taxa de chegada	70
Figura 4.18–Desempenho em função da taxa de chegada	71

Lista de tabelas

Tabela 2.1 – Estados \times colunas da matriz M_q	31
Tabela 2.2 – Notação de fila $A/B/m/K/Z$	35
Tabela 2.3 – Resumo dos resultados gerais para sistema $G/G/m$ de fila	38
Tabela 3.1 – Tabela de transição do sistema de fila de capacidade 2	52
Tabela 3.2 – Tabela de transição do sistema de fila de capacidade 2 com projeção inversa	53
Tabela 3.3 – Calendário de eventos	53
Tabela 4.1 – Tempo de processamento, falha e reparo de cada estação de trabalho .	56
Tabela 4.2 – Posições das estações	64
Tabela 4.3 – Síntese dos resultados dos AGVs	65
Tabela 4.4 – Síntese dos resultados Cliente - Servidor	69
Tabela 4.5 – Alguns resultados do desempenho em função da taxa de serviço	71

Lista de siglas

AGVs Veículos Guiados Automatizados

FIFO First In, First Out

GSMP Processos Semi-Markovianos Generalizados

IID Independente e Identicamente Distribuído

LD Linearmente Dependente

MTTF Mean Time To Failure

MTTR Mean Time To Repair

SED Sistemas a Eventos Discretos

Lista de símbolos

λ	Taxa média de chegada dos clientes
μ	Taxa média de serviço dos servidores
Γ	Função de eventos factíveis
γ	Taxa de abandono
π	Probabilidade estacionária do sistema
ρ	Intensidade de tráfego
E	Conjunto finito de eventos
G	Distribuição Geral
L	Número médio de clientes no sistema
L_q	Número médio de clientes esperando para serem atendidos
l	Capacidade da fila dos clientes do tipo 1
m	Número de servidores no sistema
M	Distribuição Exponencial (Markoviana)
M_q	Matriz de probabilidade de transição
n	Capacidade da fila dos clientes do tipo 2
p	Probabilidade de chegada de um cliente
q_{ij}	Taxa de transição do estado i para j
r	Número esperado de clientes em serviço
T_p	Taxa de produção
X	Conjunto finito de estados
W	Tempo total de espera
W_q	Tempo de espera na fila

Sumário

1	INTRODUÇÃO	17
1.1	Apresentação	17
1.2	Motivação	18
1.3	Objetivos	18
1.3.1	Objetivo Geral	18
1.3.2	Objetivos Específicos	18
1.4	Organização da Dissertação	18
2	CONCEITOS PRELIMINARES	20
2.1	Introdução	20
2.2	Sistemas a Eventos Discretos	20
2.3	Teoria de Linguagens e autômatos	21
2.3.1	Teoria de Linguagens	21
2.3.2	Autômatos	22
2.4	Cadeia de Markov	29
2.4.1	Autômato estocástico visto como uma cadeia de Markov	30
2.5	Teoria de filas	33
2.5.1	Desempenho de um sistema de Fila	35
2.5.2	Modelo Markoviano de fila ($M/M/m$)	38
2.6	Fila com clientes impacientes	40
2.6.1	<i>Balking</i>	41
2.6.2	<i>Reneging</i>	41
2.6.3	Sistemas do tipo cliente - servidor	42
2.7	Análise de desempenho de sistemas	42
2.7.1	Modelagem Analítica	43
2.7.2	Simulação	44

3	METODOLOGIA	46
3.1	Introdução	46
3.2	Modelagem dos problemas	48
3.2.1	Modelagem modular	48
3.3	Método analítico	49
3.3.1	Eliminação de eventos instantâneos	49
3.4	Método via simulação	51
4	PROBLEMAS E RESULTADOS	55
4.1	Apresentação	55
4.2	Problema dos veículos guiados automatizados	55
4.2.1	Modelagem modular do problema por autômatos	56
4.2.2	Modelagem do problema sem os eventos instantâneos	60
4.2.3	Analisando as posições de cada estação	61
4.2.4	Resultados	64
4.3	Aplicação em sistemas do tipo cliente - servidor	65
4.3.1	Modelagem modular do problema por autômatos	66
4.3.2	Resultados	68
5	CONCLUSÕES	72
5.1	Trabalhos Futuros	73
	REFERÊNCIAS	74

Introdução

1.1 Apresentação

Sistemas são estudados e trabalhados em todas as áreas do conhecimento. Existem sistemas físicos, biológicos e naturais. CASSANDRAS e LAFORTUNE (2008) definem sistema como uma combinação de componentes que agem em conjunto para executar uma função.

Nesta dissertação são tratados sistemas que são considerados físicos, de forma específica: os Sistemas a Eventos Discretos (SED), sistemas que captam a mudança de estado através da ocorrência de eventos, podendo essa alteração ser o fim de uma tarefa, a chegada de peças, ou o acionamento de um botão, não dependendo necessariamente da passagem do tempo para que exista uma evolução.

A análise de desempenho de sistemas é de extrema importância para verificar possíveis otimizações, ajustes e detectar possíveis falhas. Para um bom diagnóstico é necessário um modelo que represente bem o comportamento do sistema que será observado.

Ao finalizar a modelagem do sistema é necessário um método para análise das suas características, podendo essa análise ser de diversas formas, dentre as quais: analítica e simulação. Com isso surgiu o interesse de elaborar uma metodologia capaz de modelar e analisar o modelo de forma eficiente. Então foi desenvolvida uma metodologia utilizando uma modelagem modular por meio dos autômatos estocásticos e, através da operação de composição paralela a obtenção da dinâmica do sistema.

De posse do modelo completo, obtido pela modelagem modular, foi elaborado um algoritmo para analisar de forma específica as informações para solucionar os problemas, como por exemplo: calcular a produtividade máxima de uma manufatura, tempo médio de espera de um cliente, taxa de abandono, entre outras informações.

1.2 Motivação

A motivação deste trabalho surgiu com a necessidade de modelar um sistema de manufatura na qual seria fundamental uma metodologia para avaliar o sistema como um todo, assim como analisar as posições de cada estação de trabalho. Então seria necessário uma metodologia desde a etapa inicial de modelagem até a etapa final de análise de resultados, que fosse possível otimizar e resolver problemas de sistemas como melhorias na produtividade, taxa de abandono e impaciência em sistemas do tipo cliente - servidor.

Geralmente os sistemas são grandes e complexos, e sua modelagem se torna difícil de ser realizada. Com isso, uma das motivações foi a utilização da modelagem modular para permitir e facilitar a obtenção do comportamento de sistemas com milhares de estados.

Com a crescente utilização de métodos de simulação, uma das motivações foi a utilização de métodos analíticos, como a Cadeia de Markov e teoria de filas para análise de sistemas e comparação com métodos de simulação, tanto em questão dos resultados quanto do custo computacional. Como os resultados analíticos são bem definidos, eles servem como validação da simulação que possuem os resultados acompanhados por medidas estatísticas.

1.3 Objetivos

1.3.1 Objetivo Geral

O objetivo desta dissertação é desenvolver uma metodologia completa para análise de SED temporizados por meio de autômatos estocásticos, via método analítico e de simulação.

1.3.2 Objetivos Específicos

- ❑ Elaborar uma metodologia para modelar sistemas por meio dos autômatos estocásticos;
- ❑ Analisar o desempenho e buscar melhorias, por meio das propriedades da cadeia de Markov e pelo método de escalonamento de eventos utilizando a mesma modelagem para ambas as metodologias;
- ❑ Validar a metodologia, aplicando em problemas conhecidos na literatura.

1.4 Organização da Dissertação

Os capítulos seguintes são organizados da seguinte forma:

-
- ❑ **Conceitos preliminares:** Conceitua os principais fundamentos teóricos e conhecimentos necessários para a realização do trabalho de pesquisa;
 - ❑ **Metodologia:** Demonstra a metodologia elaborada para analisar o desempenho de sistemas;
 - ❑ **Problemas e resultados:** Explica os problemas utilizados para validação da metodologia elaborada e apresenta os resultados encontrados;
 - ❑ **Conclusões:** Apresenta as conclusões deste trabalho e define as sugestões para trabalhos futuros.

Conceitos Preliminares

2.1 Introdução

Neste capítulo serão apresentados os conceitos principais para a construção e elaboração do projeto. Tendo em vista, apenas as definições principais de cada conceito, para mais detalhes ver CASSANDRAS e LAFORTUNE (2008) e demais referências.

2.2 Sistemas a Eventos Discretos

Um sistema é caracterizado com uma parte que limita o universo que interage com o mundo externo através das fronteiras que o delimitam. CASSANDRAS e LAFORTUNE (2008) definem que um sistema consiste em componentes que interagem e, está associado a uma função destinada a ser executada. Pode-se exemplificar um sistema através de uma fila finita de atendimento em geral, em que existe a chegada de clientes, uma fila a qual esse cliente espera, um servidor para atendimento e a saída do mesmo, representado na Figura 2.1.



Figura 2.1 – Sistema de filas

Os sistemas que são tratados neste trabalho são sistemas que percebem a ocorrência externa através de estímulos denominados eventos, esses sistemas são chamados de Sistema a Eventos Discretos (SED). Esses eventos podem ser o início ou término de uma tarefa; a ocorrência desses estímulos causam uma mudança interna no sistema, não dependendo da passagem do tempo para evoluir como nos sistemas contínuos, (CASSANDRAS; LAFORTUNE, 2008).

CURY (2001) define SED como um sistema dinâmico que evolui de acordo com a ocorrência abrupta de eventos físicos, em intervalos de tempo em geral irregulares e desconhecidos. Os principais modelos utilizados para tratar de sistemas a eventos discretos são:

- Redes de Petri com e sem temporização;
- Redes de Petri Controladas com e sem temporização;
- Cadeias de Markov;
- Teoria das Filas;
- Processos Semi-Markovianos Generalizados (GSMP) e Simulação;
- Álgebra Max-Plus;
- Lógica Temporal e Lógica Temporal de Tempo Real;
- Teoria de Linguagens e Autômatos.

2.3 Teoria de Linguagens e autômatos

Como citado anteriormente, existem diversas formas para tratar e modelar o comportamento de um SED; neste trabalho foi utilizada a teoria de linguagens e autômatos estocásticos.

2.3.1 Teoria de Linguagens

Para definir o conceito de linguagens, primeiro explica-se o conceito de alfabeto. Alfabeto é o conjunto de eventos associados a um SED, representado por E . Considerando que E é um conjunto finito, uma cadeia ou palavra é definida como uma sequência de eventos. CASSANDRAS e LAFORTUNE (2008) definem uma linguagem como sendo um conjunto de cadeias de comprimento finito formado a partir de eventos que pertencem a E .

Por exemplo, considere o conjunto $E = \{\alpha, \beta, \gamma\}$ pode-se definir uma linguagem como:

$$L_1 = \{\varepsilon, \alpha, \gamma, \alpha\alpha, \alpha\beta\gamma\}$$

Uma operação utilizada na construção de linguagem é conhecida como concatenação. A cadeia $\alpha\beta\gamma$ é a concatenação da cadeia $\alpha\beta$ com o evento γ ; e $\alpha\beta$ é a concatenação dos eventos α e β (CASSANDRAS; LAFORTUNE, 2008).

A cadeia ε é definida como cadeia vazia ou elemento identidade da concatenação. A concatenação de uma cadeia x com o ε é sempre a cadeia x .

Outra operação conhecida na teoria de linguagens é o fechamento de Kleene, definida como $*$. Por exemplo, supondo o alfabeto E citado acima, tem-se:

$$E^* = \{\varepsilon, \alpha, \beta, \gamma, \alpha\alpha, \beta\beta, \gamma\gamma, \alpha\alpha\alpha, \alpha\beta\gamma, \dots\}$$

CASSANDRAS e LAFORTUNE (2008) definem E^* como todas as possíveis cadeias finitas que podem ser formadas com os elementos do conjunto E , incluindo o ε . O comprimento da cadeia é dado pela quantidade de eventos que ela contém.

Outros tipos de operações executadas com cadeias são chamadas de projeção natural e projeção inversa. A projeção natural ou somente projeção é denota de P , é tipicamente utilizada para especificar e facilitar o uso de cadeias quando se tem vários conjuntos de alfabetos (CASSANDRAS; LAFORTUNE, 2008).

A operação P apaga os eventos de uma cadeia formada a partir de vários conjuntos, deixando apenas os eventos que pertencem ao conjunto que se quer analisar. Já a operação de projeção inversa, P^{-1} , acrescenta a uma cadeia todos os eventos que pertencem aos conjuntos.

Por exemplo, seja uma linguagem $L = \{c, ccb, abc, cacb, cabcbba\}$, e consideram-se os alfabetos $E_1 = \{a, b\}$ e $E_2 = \{b, c\}$. Podem-se calcular algumas projeções como:

- A projeção sobre a linguagem L , considerando o $E_1 : P_1(L) = \{\varepsilon, b, ab, abbba\}$;
- A projeção sobre a linguagem L , considerando o $E_2 : P_2(L) = \{c, ccb, bc, cbcbbc\}$;
- A projeção inversa sobre a projeção considerando o E_1 :
 $P_1^{-1}(P_1(L)) = \{c^*, c^*bc^*, c^*ac^*bc^*, c^*ac^*bc^*bc^*ac^*\}$;
- A projeção inversa sobre a projeção considerando o E_2 :
 $P_2^{-1}(P_2(L)) = \{a^*ca^*, a^*ca^*ca^*ba^*, a^*ba^*ca^*, a^*ca^*ba^*ca^*ba^*ba^*ca^*\}$.

CASSANDRAS e LAFORTUNE (2008) explicam que a projeção inversa gera uma linguagem que contém a original, mas não recupera a linguagem antes da projeção.

2.3.2 Autômatos

Os autômatos, também conhecidos como máquinas de estado, são uma ferramenta que podem representar a dinâmica de um sistema através de uma linguagem com regras bem definidas (CASSANDRAS; LAFORTUNE, 2008). Podendo ser classificado como determinístico ou estocástico, temporizado ou não temporizado, finito ou infinito.

CASSANDRAS e LAFORTUNE (2008) representam um autômato determinístico, finito e lógico, simbolizado por G , por uma sêxtupla $(X, E, f, \Gamma, x_o, X_m)$, em que:

- X é o conjunto finito de estados;

- E é conjunto finito de eventos associado ao autômato G ;
- $f : X \times E \rightarrow X$ é a função de transição: $f(x, e) = y$ que significa que se o autômato está no estado x e ocorre o evento e ($e \in E$), ocorre a transição para o estado y ;
- $\Gamma : X \rightarrow 2^E$ é a função de eventos factíveis. $\Gamma(x)$ representa todos os eventos e tal que $f(x, e)$ é definido e é chamado de conjunto de eventos ativos para G no estado x ;
- x_o é o estado inicial;
- $X_m \subseteq X$ é o conjunto de estados marcados.

A forma mais simples e usual para apresentar a definição de autômato é por meio da sua representação gráfica direcionada, em que os estados são representados por círculos e os eventos são representados por arcos direcionados. O estado inicial é indicado por uma seta adicional, e os estados marcados, que são os de interesse, são representados por dois círculos concêntricos. A Figura 2.2 representa graficamente um autômato finito com três estados.

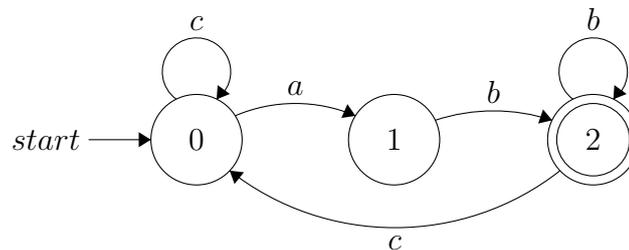


Figura 2.2 – Autômato finito

Através da definição de uma sêxtupla, o autômato pode ser representado como:

- $X = \{0, 1, 2\}$;
- $E = \{a, b, c\}$;
- $f : f(0, c) = 0; f(0, a) = 1; f(1, b) = 2; f(2, b) = 2; f(2, c) = 0$;
- $\Gamma : \Gamma(0) = \{a, c\}; \Gamma(1) = \{b\}; \Gamma(2) = \{b, c\}$
- $x_0 = 0$;
- $X_m = \{2\}$.

Com essas propriedades os autômatos podem representar sistemas de diversas áreas, como em logística, processo de manufaturas, sistemas de filas e sistema de tráfegos. Considerando o exemplo da fila, ilustrado na Figura 2.1, como um sistema de atendimento de

um supermercado, em que a fila tenha capacidade limitada a dois clientes e um servidor único. Define-se que os clientes chegam ao sistema através de um evento a , e ao finalizar o atendimento deixam o sistema através de um evento d . Os clientes só esperam caso tenha lugar na fila. O autômato que representa a dinâmica desse sistema está ilustrado na Figura 2.3.

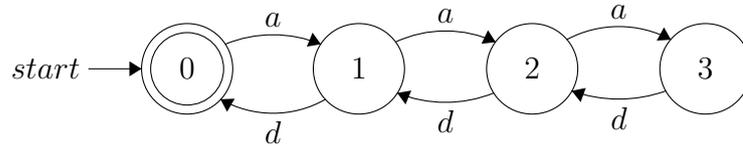


Figura 2.3 – Autômato para fila de capacidade 2

O autômato da Figura 2.3 indica o comportamento do sistema de fila de capacidade de dois clientes na fila e um em processo de atendimento, e pode ser definido como a sêxtupla:

- $X = \{0, 1, 2, 3\}$;
- $E = \{a, d\}$;
- $f : f(0, a) = 1; f(1, a) = 2; f(1, d) = 0; f(2, a) = 3; f(2, d) = 1; f(3, d) = 2$;
- $\Gamma : \Gamma(0) = \{a\}; \Gamma(1) = \{a, d\}; \Gamma(2) = \{a, d\}; \Gamma(3) = \{d\}$;
- $x_0 = 0$;
- $X_m = \{0\}$.

2.3.2.1 Linguagens representadas por autômatos

CASSANDRAS e LAFORTUNE (2008) associam duas linguagens a um autômato: a linguagem gerada ($L(G)$) e a linguagem marcada ($L_m(G)$). A linguagem $L(G)$ representa todas as cadeias formadas por um autômato partindo do estado inicial.

$$L(G) := \{s \in E^* : f(x_0, s) \text{ é factível} \}$$

Já $L_m(G)$ é a linguagem que representa todas as cadeias formadas, que começam no estado inicial e terminam em um dos estados marcados. A $L_m(G)$ é um subconjunto da $L(G)$.

$$L_m(G) := \{s \in L(G) : f(x_0, s) \in X_m\}$$

Todas as operações aplicadas à linguagem, que foram explicadas na seção 2.3.1, podem ser estendidas a autômatos, já que os autômatos são caracterizados pelas linguagens geradas e marcadas. A projeção natural em autômatos apaga os eventos que não estão no alfabeto considerado e estão presentes no autômato G . E a projeção inversa em autômatos é definida acrescentando auto-laços de todos os eventos do alfabeto que não estão no autômato G .

2.3.2.2 Modelagem Modular

Uma das formas para modelar sistemas grandes e complexos, é por meio da modelagem modular. Para isso, são necessários vários autômatos para representar o comportamento do sistema como um todo, então ao construir o sistema por parte são necessárias operações que modifiquem o seu diagrama de transição. CASSANDRAS e LAFORTUNE (2008) definem duas principais operações em autômatos: produto, indicado por \times , e a composição paralela, denotado por \parallel .

A operação por produto é mais restritiva, pois as transições dos autômatos devem estar sempre sincronizadas, um evento só ocorre se, e somente se, ocorrer em todos os autômatos (CASSANDRAS; LAFORTUNE, 2008). Então é mais usual utilizar a composição paralela para modelar sistemas modulares compostos por vários autômatos que interagem, com eventos privados que pertencem ao seu próprio comportamento e eventos em comum que são compartilhados com outros autômatos. CASSANDRAS e LAFORTUNE (2008) definem que a maneira padrão de construir modelos de sistemas inteiros a partir de modelos de componentes individuais do sistema é por composição paralela.

Como exemplo, $G_1 \parallel G_2$ é a composição paralela entre os autômatos G_1 e G_2 , que pode ser representada como uma sêxtupla (CASSANDRAS; LAFORTUNE, 2008):

$$(X_1 \times X_2, E_1 \cup E_2, f, \Gamma_{1 \parallel 2}, (x_{1_0}, x_{2_0}), X_{T_1} \times X_{T_2})$$

Em que:

- $X_1 \times X_2$ representa um conjunto de estados formado pelo produto cartesiano dos possíveis estados dos autômatos G_1 e G_2 ;
- $E_1 \cup E_2$ representa um conjunto de eventos formado pela união dos eventos que compõem os autômatos G_1 e G_2 ;
- $f : X \times E \rightarrow X$ é a função de transição, tal que:

$$f((x_1, x_2), e) := \begin{cases} (f_1(x_1, e), f_2(x_2, e)) & \text{se } e \in \Gamma_1(x_1) \cap \Gamma_2(x_2) \\ (f_1(x_1, e), x_2) & \text{se } e \in \Gamma_1(x_1) \setminus E_2 \\ (x_1, f_2(x_2, e)) & \text{se } e \in \Gamma_2(x_2) \setminus E_1 \\ \textit{indefinido} & \textit{caso contrário} \end{cases}$$

- $\Gamma_{1||2}$ é a função de eventos factíveis dado que o autômato se encontra em um dos estados provenientes do produto $X_1 \times X_2$;
- (x_{1_0}, x_{2_0}) formado por uma composição entre os estados iniciais de G_1 e G_2 ; é o estado inicial do sistema;
- $X_{T_1} \times X_{T_2}$ é um conjunto de estados terminais do sistema.

Na composição paralela, um evento que é comum só pode ser executado se todos os autômatos puderem executá-lo simultaneamente, dessa forma os autômatos são sincronizados nos eventos em comum. Já os eventos privados, podem ser disparados sempre que possíveis, sem nenhuma restrição. Ou seja, neste tipo de sistema, um autômato pode executar seus eventos individuais sem a participação do outro autômato; no entanto, um evento comum só pode acontecer se todos puderem dispará-lo (CASSANDRAS; LAFORTUNE, 2008). Como exemplo, de uma operação de composição paralela, retornando ao sistema da fila representado na Figura 2.1, pode-se adicionar um novo evento entre a fila e o servidor, que representa a admissão do cliente pelo servidor, como mostrado na Figura 2.4:

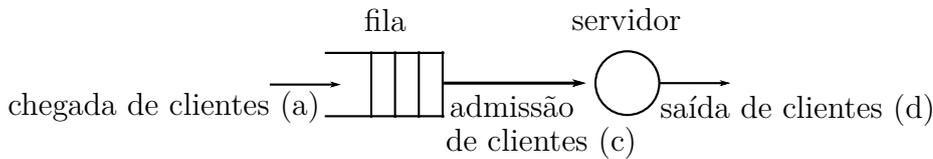


Figura 2.4 – Sistema de filas de forma modular

Modelando o sistema de forma individual através de autômatos, olhando apenas para o *buffer* (fila) de capacidade dois, depois para o servidor de capacidade única têm-se os autômatos da Figura 2.5:

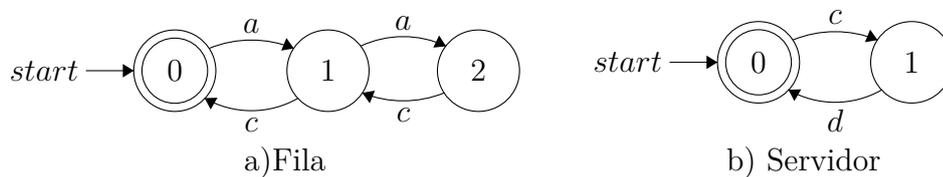


Figura 2.5 – Autômato para o sistema de fila de capacidade 2

Ao analisar a Figura 2.5, tem-se que o autômato a) ilustra uma fila de capacidade dois e o autômato b) mostra o comportamento do servidor. Os eventos a e d são eventos individuais da fila e do servidor, respectivamente, portanto podem ocorrer sem restrições, já o evento c, por ser um evento comum, só deve ser disparado se ambos os autômatos permitirem. Aplicando a composição paralela tem-se a sêxtupla que representa o comportamento completo do sistema $Fila||Servidor$:

- $X_1 \times X_2 = \{(0, 0), (0, 1), (1, 0), (1, 1), (2, 0), (2, 1)\}$;
- $E_1 \cup E_2 = \{a, c, d\}$;
- $f : f((0, 0), a) = (1, 0); f((0, 1), a) = (1, 1); f((0, 1), d) = (0, 0); f((1, 0), a) = (2, 0); f((1, 0), c) = (0, 1); f((1, 1), a) = (2, 1); f((1, 1), d) = (1, 0); f((2, 0), c) = (1, 1); f((2, 1), d) = (2, 0)$;
- $\Gamma_{1||2} : \Gamma_{1||2}(0, 0) = \{a\}; \Gamma_{1||2}(0, 1) = \{a, d\}; \Gamma_{1||2}(1, 0) = \{a, c\}; \Gamma_{1||2}(1, 1) = \{a, d\}; \Gamma_{1||2}(2, 0) = \{c\}; \Gamma_{1||2}(2, 1) = \{d\}$;
- $(x_{1_0}, x_{2_0}) = (0, 0)$;
- $X_{T_1} \times X_{T_2} = \{(0, 0)\}$.

O autômato que representa a composição paralela dos autômatos da fila e do servidor pode ser visto na Figura 2.6.

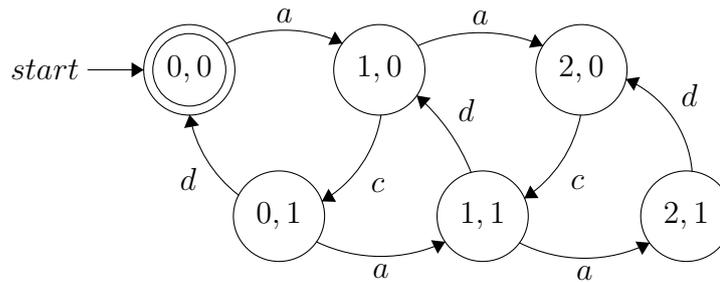


Figura 2.6 – Autômato da composição paralela da Fila||Servidor

Os estados da composição paralela representam o comportamento do sistema como um todo. Os estados da Figura 2.6 podem ser detalhados como:

- **Estado 0,0:** Sistema vazio, nenhum cliente na fila e nenhum cliente em atendimento;
- **Estado 1,0:** Sistema com um cliente em fila e nenhum cliente em atendimento;
- **Estado 0,1:** Sistema com um cliente em atendimento e nenhum cliente em fila;
- **Estado 1,1:** Sistema com dois clientes, um cliente em fila e um cliente em atendimento;
- **Estado 2,0:** Sistema com dois clientes em fila e nenhum cliente em atendimento;
- **Estado 2,1:** Sistema com três clientes, sendo dois clientes em fila e um cliente em atendimento.

2.3.2.3 Autômatos Temporizados

O autômato é utilizado para representar o comportamento de um sistema, cujos eventos são temporizados. Então, é necessária uma estrutura de temporização que defina uma sequência de tempo de vida para os eventos, podendo o tempo ser determinístico ou estocástico. Uma característica muito importante é que se o autômato for temporizado estocástico com uma distribuição exponencial, pode ser convertido facilmente em uma Cadeia de Markov.

Uma estrutura de temporização, também conhecida como estrutura de relógio, é definida como uma sequência de tempo de vida para os eventos. Para cada estado são verificados quais transições são factíveis, bem como os momentos de ocorrência (RIBEIRO, 2018). Uma estrutura de relógio (V) associado a um evento e pode ser definido como um conjunto (2.1):

$$V = \{v_i : i \in E\}, \quad (2.1)$$

em que v_i corresponde uma sequência de vida para cada evento i pertencente ao conjunto E . Portanto, a estrutura de relógio v_i é formada por momentos temporais definidos de tempos em tempos, até que o evento seja desabilitado. CASSANDRAS e LAFORTUNE (2008) explicaram três regras básicas para selecionar o próximo evento e garantir que a evolução do sistema aconteça de forma correta:

- O próximo evento é aquele que possui menor valor de relógio dentre todos aqueles que são factíveis;
- Um evento e é caracterizado como ativo quando:
 - e foi o último evento ocorrido e permaneceu factível após a transição de estados;
 - Um evento diferente ocorre enquanto e não é factível, fazendo com que ocorra uma transição para um novo estado onde e é factível;
- Um evento e é desativado quando outro evento ocorre causando uma transição para um estado em que e já não é mais factível.

Geralmente, os eventos ocorrem de forma estocástica, ou seja, existe uma incerteza em relação ao próximo evento. Segundo CASSANDRAS e LAFORTUNE (2008), um autômato temporizado estocástico é composto por uma estrutura de relógio estocástica, em que para cada evento aleatório, é associada uma função de distribuição que caracteriza o comportamento do mesmo. CASSANDRAS e LAFORTUNE (2008) definem um autômato temporizado estocástico por uma 6-tupla $(X, E, \Gamma, p, p_0, H)$, em que:

- X é o conjunto finito de estados;

- E é o conjunto finito de eventos associados ao autômato;
- $\Gamma : X \rightarrow 2^E$ é a função de eventos factíveis. $\Gamma(x)$ representa todos os eventos ativos dado que o autômato se encontra no estado x ;
- $p(x'; x, e)$ é a probabilidade de transição do estado x para x' , dada a ocorrência do evento e sendo que $p(x'; x, e) = 0 \forall e \notin \Gamma(x)$;
- p_0 é a probabilidade do estado inicial do sistema ser x ;
- H é uma estrutura de relógio estocástica.

Observando a definição de um autômato temporizado estocástico, pode-se destacar que a estrutura de relógio, as transições e o estado inicial são caracterizados por variáveis aleatórias, atribuindo um comportamento incerto ao sistema, sendo necessários alguns conceitos relacionados a processos estocásticos, (RIBEIRO, 2018).

2.4 Cadeia de Markov

O matemático russo A. Markov fundou um novo ramo da teoria da probabilidade, em janeiro de 1913, que ficou conhecido como cadeia de Markov (HAYES (2013)). Esta teoria é extremamente eficiente na solução de problemas práticos formados por informações de natureza incerta, (RIBEIRO, 2018).

WANG e LI (2011), definem a cadeia de Markov como um método de previsão de probabilidade e técnica para prever os próximos estados do sistema, estudando apenas o estado atual e sua tendência de mudança no sistema. Segundo BRENNER, FERNANDES e SALES (2002), as cadeias de Markov são empregadas para descrever e buscar soluções estacionárias para os mais diversos tipos de realidades.

Por isso, a cadeia de Markov é utilizada para descrever e analisar propriedades de sistemas dinâmicos, cuja principal característica seja determinar a probabilidade do sistema estar em qualquer estado em qualquer instante de tempo (CASSANDRAS; LAFORTUNE, 2008).

CASSANDRAS e LAFORTUNE (2008) explicam que na cadeia de Markov o futuro é condicionalmente independente do passado, dado o estado atual. Toda a história passada é resumida no estado atual. Este fato é conhecido como a propriedade sem memória, uma vez que não é preciso nenhuma lembrança da história passada, apenas o presente, para prever probabilisticamente o futuro. Podem-se definir dois aspectos para a propriedade sem memória:

- Todas as informações de estado anteriores são irrelevantes;
- Quanto tempo o processo esteve no estado atual é irrelevante.

Uma cadeia de Markov cujo espaço amostral seja discreto pode ser definida como um processo Markoviano de tempo contínuo é matematicamente expresso pela equação:

$$\begin{aligned} P[X(t_{k+1}) \leq x_{k+1} | X(t_k) = x_k, X(t_{k-1}) = x_{k-1}, \dots, X(t_0) = x_0] \\ = P[X(t_{k+1}) \leq x_{k+1} | X(t_k) = x_k], \end{aligned} \quad (2.2)$$

a equação 2.2 implica que a probabilidade de transição do estado atual para o próximo não depende do passado, ou seja, todo passado está agregado no estado presente, portanto, está em função apenas do estado atual do sistema (CASSANDRAS; LAFORTUNE, 2008).

2.4.1 Autômato estocástico visto como uma cadeia de Markov

A composição paralela de vários autômatos estocásticos compostos por uma estrutura de relógio de acordo com a distribuição exponencial, pode ser reduzida a uma cadeia de Markov, pois a transição entre estados é também escrita em função de taxas de transferência (CASSANDRAS; LAFORTUNE, 2008). As taxas de transferência são os tempos de vida associado a cada evento.

RIBEIRO (2018) descreve que para determinar a probabilidade de cada estado, primeiramente é necessário a matriz de probabilidade de transição (M_q) composta por todas as probabilidades possíveis. Dada uma cadeia de Markov com L estados, a matriz (M_q) pode ser definida como:

$$M_q = \begin{pmatrix} q_{00} & q_{01} & q_{02} & \cdots & q_{0L} \\ q_{10} & \cdots & \cdots & \cdots & q_{1L} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ q_{L0} & \cdots & \cdots & \cdots & q_{LL} \end{pmatrix}, \quad (2.3)$$

em que q_{ij} é a taxa de transição do estado i para o estado j . A definição de q_{ij} é dada por:

$$q_{ij} = \begin{cases} \lambda_\alpha & \text{se } i \neq j \\ -\sum_{j=0}^L q_{ij} & \text{se } i = j \end{cases} \quad (2.4)$$

Em que λ_α é a taxa de ocorrência de um evento α , composto por uma distribuição exponencial, que é responsável pela mudança do sistema do estado i para o estado j . Assim, pode-se determinar a probabilidade do sistema estar em um estado em qualquer instante de tempo no regime permanente, como:

$$\pi M_q = 0, \quad (2.5)$$

onde π representa um vetor de probabilidade do sistema ser encontrado em cada estado factível, composto por cada π_i , $\forall x_i \in X$.

Voltando ao exemplo do sistema da fila modelado de forma modular mostrado na Figura 2.4, e utilizando o comportamento do sistema como um todo, que pode ser visto na composição paralela do sistema representado na Figura 2.6, podemos obter a cadeia de Markov:

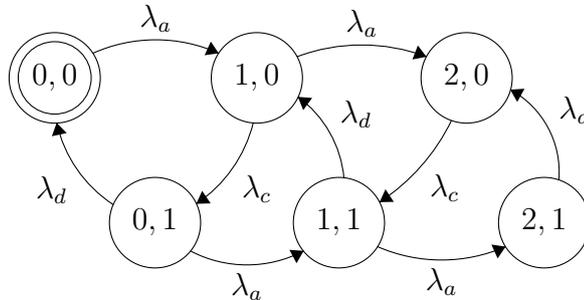


Figura 2.7 – Cadeia de Markov do sistema da fila simples

A matriz 2.3 que representa o sistema da fila ilustrado na Figura 2.7, é definida pela equação 2.6:

$$M_q = \begin{bmatrix} -\lambda_a & \lambda_a & 0 & 0 & 0 & 0 \\ 0 & -(\lambda_a + \lambda_c) & \lambda_c & 0 & \lambda_a & 0 \\ \lambda_d & 0 & -(\lambda_a + \lambda_d) & \lambda_a & 0 & 0 \\ 0 & \lambda_d & 0 & -(\lambda_a + \lambda_d) & 0 & \lambda_a \\ 0 & 0 & 0 & \lambda_c & -\lambda_c & 0 \\ 0 & 0 & 0 & 0 & \lambda_d & -\lambda_d \end{bmatrix}, \quad (2.6)$$

cada coluna da matriz 2.6, representa as taxas de transições de cada estado resultante da cadeia de Markov. A relação das colunas com os estados da cadeia de Markov são mostradas na Tabela 2.1.

Tabela 2.1 – Estados \times colunas da matriz M_q

Estado	Coluna
(0, 0)	1
(1, 0)	2
(0, 1)	3
(1, 1)	4
(2, 0)	5
(2, 1)	6

Definindo os valores de $\lambda_a = 1$, $\lambda_c = 10^3$, e $\lambda_d = 2$, e substituindo na equação 2.5,

tem-se 2.7:

$$\begin{bmatrix} \pi_{00} & \pi_{10} & \pi_{01} & \pi_{11} & \pi_{20} & \pi_{21} \end{bmatrix} \times \begin{bmatrix} -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & -(1+10^3) & 10^3 & 0 & 1 & 0 \\ 2 & 0 & -(1+2) & 1 & 0 & 0 \\ 0 & 2 & 0 & -(1+2) & 0 & 1 \\ 0 & 0 & 0 & 10^3 & -10^3 & 0 \\ 0 & 0 & 0 & 0 & 2 & -2 \end{bmatrix} = 0. \quad (2.7)$$

Ao resolver a multiplicação, obtém-se um sistema linear descrito em 2.8:

$$\begin{cases} -\pi_{00} + 2\pi_{01} = 0 \\ \pi_{00} - 1001\pi_{10} + 2\pi_{11} = 0 \\ 1000\pi_{10} - 3\pi_{01} = 0 \\ \pi_{01} - 3\pi_{11} + 1000\pi_{20} = 0 \\ \pi_{10} - 1000\pi_{20} + 2\pi_{21} = 0 \\ \pi_{11} - 2\pi_{21} = 0 \end{cases} \quad (2.8)$$

Como se trata de um sistema linear indefinido e, por se tratar de probabilidade, para calcular os valores de π_{ij} é preciso garantir que o $\sum \pi_{ij} = 1$, portanto, uma das equações é modificada para certificar essa propriedade. Qualquer uma das equações podem ser alteradas, pois se trata de um sistema Linearmente Dependente (LD). Alterando a última equação do sistema linear da equação 2.8, o novo sistema será 2.9:

$$\begin{cases} -\pi_{00} + 2\pi_{01} = 0 \\ \pi_{00} - 1001\pi_{10} + 2\pi_{11} = 0 \\ 1000\pi_{10} - 3\pi_{01} = 0 \\ \pi_{01} - 3\pi_{11} + 1000\pi_{20} = 0 \\ \pi_{10} - 1000\pi_{20} + 2\pi_{21} = 0 \\ \pi_{00} + \pi_{10} + \pi_{01} + \pi_{11} + \pi_{20} + \pi_{21} = 1 \end{cases} \quad (2.9)$$

Resolvendo o sistema de equações 2.9, encontram-se os resultados das probabilidades estacionária do sistema de fila ser encontrado em cada um dos estados em regime permanente, os resultados foram:

$$\begin{cases} \pi_{00} = 53.2516\% \\ \pi_{10} = 0.0799\% \\ \pi_{01} = 26.6258\% \\ \pi_{11} = 13.3528\% \\ \pi_{20} = 0.0134\% \\ \pi_{21} = 6.6764\% \end{cases} \quad (2.10)$$

Ao analisar o comportamento do sistema, pode-se perceber que devido à taxa de saída de clientes, λ_d , ser maior que a taxa de chegada de clientes, λ_a , o sistema tem maior probabilidade de ser encontrado no estado vazio, ou seja, estado $(0, 0)$, pois chegam poucos clientes e eles são atendidos rapidamente.

Alterando da taxa de saída, $\lambda_d = 0.5$, tem-se os seguintes resultados:

$$\begin{cases} \pi_{00} = 6.6556\% \\ \pi_{10} = 0.0200\% \\ \pi_{01} = 13.3111\% \\ \pi_{11} = 26.6622\% \\ \pi_{20} = 0.0267\% \\ \pi_{21} = 53.3244\% \end{cases} \quad (2.11)$$

Pode-se perceber que reduzindo a taxa de serviço, a probabilidade do sistema ser encontrado ocioso diminuiu, aumentando a probabilidade do sistema estar cheio, ou seja, no estado $(2, 1)$. Portanto, validando o funcionamento correto do sistema modelado.

2.5 Teoria de filas

A teoria de filas foi desenvolvida para fornecer modelos para prever o comportamento de sistemas que tentam fornecer serviço para demandas surgidas aleatoriamente, (GROSS et al., 2008). Tendo como o objetivo principal determinar o desempenho de um sistema em determinadas condições operacionais, e também desenvolver ferramentas descritivas para analisar e estudar o sistemas de filas, ao invés de utilizar ferramentas prescritivas para controlar seu comportamento incerto (CASSANDRAS; LAFORTUNE, 2008).

GROSS et al. (2008) cita que existem muitas aplicações valiosas da teoria, a maioria das quais tem sido bem documentadas na literatura de probabilidade, pesquisa operacional, ciência de gestão e engenharia industrial. Alguns exemplos são o fluxo de tráfego (veículos, aeronaves, pessoas, comunicações), agendamento (pacientes em hospitais, trabalhos em máquinas, programas em um computador) e projeto de instalações (bancos, agências dos correios, parques de diversões, restaurantes fast food). CASSANDRAS e LAFORTUNE (2008) enfatizam que os sistemas de filas realizaram algumas das mais importantes contribuições para a análise de SEDs estocásticos, especialmente para lidar com problemas de compartilhamento de recursos.

Um sistema de fila pode ser descrito como clientes que chegam para o serviço, clientes aguardando pelo serviço, que se não for imediato e tiverem que aguardar pelo serviço formam uma fila, e o cliente deixando o sistema após ser atendido. O termo cliente é usado em um sentido geral e não implica necessariamente um cliente humano. A Figura 2.8 ilustra um sistema de filas com espaço de armazenamento infinito e um único servidor.

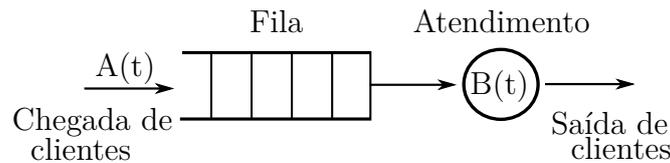


Figura 2.8 – Sistema de filas simples

GROSS et al. (2008) cita seis características básicas dos processos de teoria das filas: padrão de chegada de clientes, padrão de serviço dos servidores, disciplina de fila, capacidade do sistema, número de canais de atendimento e etapas do serviço. Para os sistemas de filas, existe uma notação consolidada na literatura através de uma série de símbolos e barras, que descreva resumidamente o modelo do sistema (GROSS et al., 2008), essa notação é a seguinte:

$$A/B/m/K/Z$$

Em que:

- A é a distribuição do tempo entre as chegadas;
- B é a distribuição do tempo de serviço;
- m é a quantidade de servidores do sistema;
- K é a capacidade do tamanho da fila;
- Z é a disciplina da fila.

Se $K = \infty$ e $Z = FIFO$, disciplina que trata de acordo com o primeiro a chegar, primeiro a sair (First In, First Out (**FIFO**)), pode - se omitir o K e Z sendo portando um sistema $A/B/m$. A Tabela 2.2 exemplifica alguns dos símbolos mais utilizados na teoria de filas.

Utilizando a Tabela 2.2 para construir alguns exemplos que podem ser encontrados para denotar um sistema de fila, tem-se:

- $M/M/1$ é um sistema de fila de servidor único com capacidade de armazenamento infinita, disciplina da fila FIFO. Os tempos entre chegadas e os tempos de serviço são markovianos, exponencialmente distribuídos, ou seja, $A(t) = 1 - e^{-\lambda t}$, $B(t) = 1 - e^{-\mu t}$, para λ e μ positivos.
- $M/M/2/10$ é um sistema de fila markoviana com dois servidores com capacidade de armazenamento igual a 10, incluindo espaço nos servidores. Os tempos entre as chegadas e os tempos de serviço são distribuídos exponencialmente, e a disciplina da fila é FIFO.

Tabela 2.2 – Notação de fila $A/B/m/K/Z$

Característica	Símbolo	Explicação
<i>Distribuição do tempo entre as chegadas (A)</i>	M	Exponencial (Markoviano)
	D	Determinístico
	E_k	Erlang tipo k ($k = 1, 2, \dots$)
<i>Distribuição do tempo de serviço (B)</i>	H_k	Hiper - exponencial (Combinação de k exponenciais)
	G	Geral
Quantidade de servidores do sistema (s)	$1, 2, \dots, \infty$	
Capacidade máxima do tamanho da fila (K)	$1, 2, \dots, \infty$	
Disciplina da fila (Z)	FIFO	Primeiro a chegar, primeiro a ser atendido
	LCFS	Último a chegar, primeiro a ser atendido
	RSS	Seleção aleatória para serviço
	PR	Prioritário
	GD	Disciplina Geral

Fonte: Modificada de GROSS et al. (2008)

- $D/G/m/K/LCFS$ é um sistema de fila com $m < \infty$ servidores, capacidade de armazenamento igual a $K < \infty$. Os tempos entre as chegadas são determinísticos, e os tempos de serviço seguem uma distribuição arbitrária (geral). E a disciplina de atendimento é: o último a chegar, primeiro a ser atendido.

2.5.1 Desempenho de um sistema de Fila

Na maioria das vezes os sistemas possuem os tempos de chegada e a duração do tempo de serviço de forma estocástica, ou seja, tem um comportamento incerto. Como foi definido anteriormente, o evento a representa uma chegada de um cliente. Seja $\{Y_1, Y_2, \dots\}$ em que Y_k é o k -ésimo tempo entre as chegadas ou seja, uma variável aleatória tal que:

$$Y_k = \text{tempo decorrido entre a } (k - 1) \text{ - ésimas e a } k \text{ - ésimas chegadas, } k = 1, 2, \dots$$

CASSANDRAS e LAFORTUNE (2008) explicam que na teoria de filas, assume-se que a sequência de tempo entre as chegadas são Independente e Identicamente Distribuído (IID), tendo uma única distribuição de probabilidade que descreve completamente essa sequência:

$$A(t) = P[Y \leq t], \quad (2.12)$$

em que Y , na equação 2.12, é a variável aleatória utilizada como tempo entre as chegadas exponenciais, não sendo necessário indexar a quantidade de chegadas ocorridas. A média da função de distribuição $A(t)$ é:

$$E[Y] = \frac{1}{\lambda}, \quad (2.13)$$

sendo λ a taxa média de chegada dos clientes. Da mesma forma, o tempo de serviço também é uma sequência IID, com as mesmas características dos intervalos entre os tempos de chegada, podendo então definir sua média por:

$$E[Z] = \frac{1}{\mu}, \quad (2.14)$$

na equação 2.14, Z é a variável aleatória utilizada como tempo de serviço de cada cliente e a taxa média de serviço do servidor é dada por μ . Tendo definido $E[Y]$ e $E[Z]$, pode-se também definir:

- A_k é o tempo da chegada do k -ésimo cliente;
- D_k é o tempo de saída do k -ésimo cliente;
- W_k é o tempo de espera do k -ésimo cliente, tempo entre o tempo de chegada até o início do atendimento;
- S_k é o tempo de sistema, considerando o tempo entre a chegada do cliente e sua saída.
- Z_k é o tempo de serviço do k -ésimo cliente;

É possível determinar que:

$$S_k = D_k - A_k \quad \text{ou} \quad S_k = W_k + Z_k. \quad (2.15)$$

CASSANDRAS e LAFORTUNE (2008) explicam que o ideal seria construir um sistema de filas que não houvesse espera pelos clientes, mas como se trata de um processo estocástico, não é possível disponibilizar infinitos servidores ou um super servidor para atender todos os clientes de modo que eles chegassem e esperassem o tempo zero. Assumindo que o sistema, ao ser executado por um período suficientemente longo de tempo, encontre o estado estacionário, é possível definir algumas medidas de desempenho tais como (CASSANDRAS; LAFORTUNE, 2008):

- $E[W]$ é o tempo médio de espera dos clientes no estado estacionário;
- $E[S]$ é o tempo médio do sistema de clientes no estado estacionário;
- $E[X]$ é o comprimento médio da fila no estado estacionário.

Em um sistema, ao atingir o estado estacionário, o fluxo de entrada e saída dos clientes deve ser equilibrado. Uma medida de análise para verificar o tráfego de clientes é definida como:

$$\rho = \frac{\lambda}{m\mu}, \quad (2.16)$$

na equação 2.16, ρ é chamado de intensidade de tráfego a qual é definida como taxa na qual o trabalho entra no sistema (KLEINROCK, 1975). GROSS et al. (2008) explica que o ideal é encontrar um número de servidores tal que ρ seja estritamente menor que 1 ($\rho < 1$), pois quando se tem $\rho > 1$ o número médio de chegadas ao sistema excede a taxa média máxima de serviço do sistema, conforme o tempo for passando e acontecer as chegadas de clientes, a fila fica cada vez maior. Se $\rho = 1$, a menos que as chegadas e o serviço sejam determinísticos, não existe estado estacionário, pois a aleatoriedade das chegadas e tempo de serviço impedirá que a fila se esvazie e permita que os usuários se atualizem, fazendo com que a fila cresça sem limite.

2.5.1.1 Lei de Little

Uma das relações mais poderosas na teoria das filas foi desenvolvida por John D. C. Little no início dos anos 1960 (GROSS et al., 2008). Little relacionou os tamanhos médios do sistema em estado estacionário aos tempos médios de espera do cliente. A Figura 2.9 ilustra a relação da lei de Little.

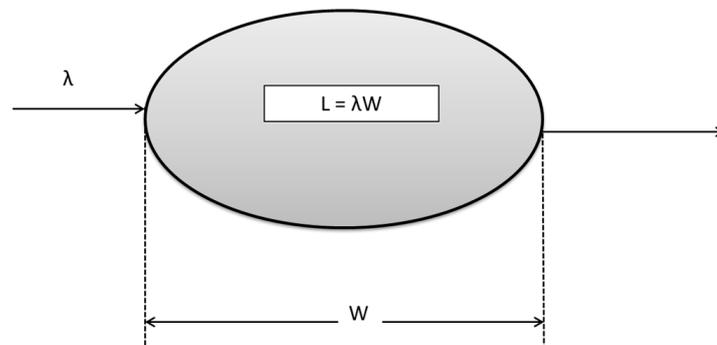


Figura 2.9 – Lei de Little

Considerando que W_q e W são o tempo médio de espera na fila e o tempo médio de permanência no sistema, as fórmulas de Little são:

$$L = \lambda W \quad e \quad L_q = \lambda W_q, \quad (2.17)$$

em que L e L_q são o número médio de clientes no sistema e o número médio de clientes esperando para serem atendidos, respectivamente. Um resultado interessante que pode

ser derivado das fórmulas de Little em relação ao tempo de espera é:

$$L - L_q = \lambda(W - W_q) = \lambda \left(\frac{1}{\mu} \right) = \left(\frac{\lambda}{\mu} \right). \quad (2.18)$$

A Tabela 2.3 resume os resultados que são encontrados para um sistema $G/G/m$ de fila:

Tabela 2.3 – Resumo dos resultados gerais para sistema $G/G/m$ de fila

$\rho = \lambda/m\mu$	Intensidade de tráfego
$L = \lambda W$	Número médio de clientes no sistema
$L_q = \lambda W_q$	Número médio de clientes esperados na fila
$W = W_q + 1/\mu$	Tempo total de espera
$p_b = \lambda/m\mu = \rho$	Probabilidade de servidor arbitrário estar ocupado
$r = \lambda/\mu$	Número esperado de clientes em serviço (Número de servidores ocupados)
$p_0 = 1 - \rho$	Probabilidade do sistema vazio

Fonte: Modificada de GROSS et al. (2008)

2.5.2 Modelo Markoviano de fila ($M/M/m$)

O sistema de fila tratado neste tópico é um sistema de m servidores com capacidade infinita e tempos entre chegada e de serviço distribuídos exponencialmente. As chegadas são exponenciais com taxa λ e cada servidor tem uma distribuição de tempo de serviço exponencial IID com taxa μ .

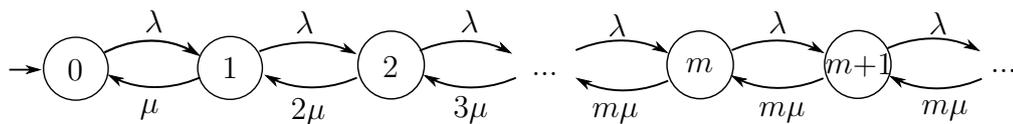


Figura 2.10 – Sistema de filas $M/M/m$

A Figura 2.10 ilustra um sistema de fila Markoviana com m servidores disponíveis. GROSS et al. (2008) explica que, como a taxa de chegada, ou nascimento, é constante para todo n , independente do número de clientes no sistema, pode-se definir $\lambda_n = \lambda$. O que difere da taxa de saída, ou morte, que dependem do número de clientes no sistema, se houver m ou mais clientes no sistema, todos os servidores m devem estar ocupados. Como cada servidor processa clientes com taxa μ , a taxa de conclusão de serviço combinada para o sistema é $m\mu$. Quando há menos de m clientes no sistema, $n < m$, apenas n dos m

servidores estão ocupados e a taxa de conclusão de serviço combinada para o sistema é $n\mu$. O que pode ser escrita como:

$$\mu_n = \begin{cases} n\mu & (1 \leq n < m) \\ m\mu & (n \geq m) \end{cases} \quad (2.19)$$

Com essa definição juntamente com a equação $\pi M_q = 0$, (2.5), definida na seção 2.4.1, pode-se derivar medidas de eficácia para o modelo $M/M/m$ utilizando as probabilidades de estado estacionário (GROSS et al., 2008). Primeiramente, pode-se determinar o tamanho esperado da fila L_q como:

$$L_q = \left(\frac{r^m \rho}{m!(1-\rho)^2} \right) \pi_0, \quad (2.20)$$

em que π_0 na equação 2.20 é a probabilidade do sistema ser encontrado no estado 0 e r é o número esperado de clientes em serviço, definido como $r = \lambda/\mu$. Através da lei de Little, utilizando as equações dadas na Tabela 2.3, pode-se definir W_q , tempo de espera na fila:

$$W_q = \frac{L_q}{\lambda} = \left(\frac{r^m}{m!(m\mu)(1-\rho)^2} \right) \pi_0. \quad (2.21)$$

Aplicando a equação 2.21, é possível encontrar agora o tempo total de sistema, W :

$$W = W_q + \frac{1}{\mu} = \frac{1}{\mu} + \left(\frac{r^m}{m!(m\mu)(1-\rho)^2} \right) \pi_0. \quad (2.22)$$

Finalmente, empregando novamente a fórmula de Little, é possível obter o número médio de clientes no sistema, L :

$$L = \lambda W = r + \left(\frac{r^m \rho}{m!(1-\rho)^2} \right) \pi_0. \quad (2.23)$$

2.5.2.1 Teorema de Burke

O teorema de Burke é extremamente útil na análise de filas, em que a taxa de entrada e o serviço em todos os servidores são exponenciais, e não há restrição no tamanho da fila entre as estações. GROSS et al. (2008) explica que em 1956, Burke provou que a distribuição de saída é idêntica à distribuição de entrada.

Seja uma fila $M/M/m$ no estado estacionário, com taxa de chegada λ , pode-se provar que os tempos de saída no sistema também são exponenciais com taxa λ . $N(t)$ é o número de clientes no sistema no momento t após a última saída de um cliente, tem-se (GROSS et al., 2008):

$$Pr\{N(t) = n\} = p_n, \quad (2.24)$$

em que t representa uma variável aleatória de tempo entre as saídas do sistema. Portanto, $F_n(t)$ é a probabilidade conjunta de encontrar n clientes no sistema em um tempo t após a última partida.

$$F_n(t) = Pr\{N(t) = n \text{ e } T > t\}. \quad (2.25)$$

A distribuição cumulativa da variável aleatória T , $C(t)$ é dado por:

$$C(t) \equiv Pr\{T \leq t\} = 1 - \sum_{n=0}^{\infty} F_n(t), \quad (2.26)$$

desde que:

$$\sum_{n=0}^{\infty} F_n(t) = Pr\{T > t\}. \quad (2.27)$$

Para encontrar $C(t)$, primeiro é definido $F_n(t)$ através das equações diferenciais:

$$\begin{aligned} \frac{dF_n(t)}{dt} &= -(\lambda + m\mu)F_n(t) + \lambda F_{n-1}(t) \quad (m \leq n), \\ \frac{dF_n(t)}{dt} &= -(\lambda + n\mu)F_n(t) + \lambda F_{n-1}(t) \quad (1 \leq n \leq m), \\ \frac{dF_0(t)}{dt} &= -\lambda F_0(t). \end{aligned} \quad (2.28)$$

Utilizando a condição de limite, a partir de 2.25 :

$$F_n(0) \equiv Pr\{N(0) = n \text{ e } T > 0\} = Pr\{N(0) = n\} = p_n, \quad (2.29)$$

a solução de 2.28:

$$F_n(t) = p_n e^{(-\lambda t)}. \quad (2.30)$$

Para obter $C(t)$ (2.26), a distribuição cumulativa dos tempos entre as saídas:

$$C(t) = 1 - \sum_{n=0}^{\infty} p_n e^{-\lambda t} = 1 - e^{-\lambda t} \sum_{n=0}^{\infty} p_n = 1 - e^{-\lambda t}. \quad (2.31)$$

A equação 2.31 mostra que os tempos de saída também são exponenciais idênticos aos tempos de chegada.

2.6 Fila com clientes impacientes

Uma característica importante na modelagem de sistemas que possuem fila é a presença de clientes impacientes. GROSS et al. (2008) explica que os clientes que são impacientes tendem a entrar na fila apenas quando há uma espera curta e permanecem na fila se a espera tiver sido pequena. A impaciência resultante de uma espera excessiva é tão importante no sistema de fila quanto as chegadas e partidas. Algumas pesquisas ignoram

o fator de impaciência durante o estudo do sistema de filas comum, mas os atos impacientes dos clientes devem estar envolvidos no estudo do sistema de filas para modelar um sistema de forma real (WANG; LI; JIANG, 2010). Pode-se dividir a paciência em duas principais formas: *balking* e *reneging*.

Na literatura os trabalhos mais voltados à impaciência de clientes são direcionados aos *call centers*, podem-se citar os artigos: Roubos e Jouini (2013), WHITT (2004), YUE, W. e ZHAO (2014), AL-SEEDY et al. (2009), ABOU-EL-ATA e HARIRI (1992), entre outros que buscam modelar o comportamento de sistemas de call centers com impaciência.

2.6.1 *Balking*

GROSS et al. (2008) explica que na realidade, muitas vezes os clientes sentem-se desencorajados quando a fila é longa e não desejam esperar. De acordo com Wang, Li e Jiang (2010), o *balking* é quando o cliente toma decisões de acordo com o tamanho da fila. Os clientes possuem um valor N do comprimento da fila antes que ele receba o serviço, se o observado for menor que N , ele se junta a fila, caso contrário abandona o sistema; o cliente possui duas opções: esperar (entrar na fila) ou se desligar (abandonar o sistema).

A forma de abandono tipo *balking*, contém duas regras convencionais definidas no artigo do Wang, Li e Jiang (2010), a do tipo I descreve que os clientes conservam um valor limite N do comprimento da fila, antes que ele receba o serviço e então, decide se espera o serviço ou se abandona. Já o tipo II lida com probabilidade p_n em que N é o número de clientes no momento da chegada.

2.6.2 *Reneging*

Diferentemente do *balking*, no abandono dos clientes através do *reneging*, os clientes que tendem a ser impacientes nem sempre são desencorajados pelo tamanho excessivo da fila, mas podem entrar na fila para ver quanto tempo a sua espera pode estender, permanecendo o tempo todo ou abandonando se a estimativa de sua espera total for maior que sua tolerância (GROSS et al., 2008).

Wang, Li e Jiang (2010) mostram que quando um indivíduo decide se continua esperando pelo serviço na fila ou não, as regras que ele obedece são assumidas variadamente na literatura. Seja T o tempo máximo de espera que um cliente tem disponível, o caso mais simples é se esse tempo de espera for fixo, definido como regra do tipo III.

A regra do tipo IV define que o tempo de espera T segue uma distribuição exponencial com parâmetro i , em que o i é o número de clientes no sistema. Considerando agora, que o tempo de serviço segue uma distribuição lognormal, e os tempos de abandono (tempo de espera) não são exponenciais, sendo esta regra tipo V.

Por último, a regra tipo VI estabelece que os clientes podem abandonar se a taxa de serviço estiver lenta e houver muitas falhas no serviço. Wang, Li e Jiang (2010) apontam

que há mais algumas regras de *reneging*, em que os clientes abandonam o sistema por outras diversas razões, mas ele resume nas que mais ocorrem.

2.6.3 Sistemas do tipo cliente - servidor

Sistemas do tipo cliente - servidor são caracterizados de forma geral como sistemas em que há presença de clientes em fila aguardando atendimento, seja um call center, fila de banco ou supermercado, atendimento em um restaurante, e diversos outros. Como existe a espera dos clientes pelo serviço, há também o abandono do mesmo, e um dos desafios para modelagem desse tipo de sistemas é a paciência dos clientes.

O estudo da impaciência dos clientes é uma área que está desenvolvendo bastante para melhorias principalmente em call centers, uma vez que a maioria das empresas estão utilizando esse tipo de serviço para resoluções de problemas, atendimentos dos clientes, marketing e os mais diversos serviços. Para isso é necessário estimar a impaciência dos clientes para uma melhor eficiência.

Roubos e Jouini (2013) propõem no seu artigo, modelar a paciência dos clientes através da distribuição hiperexponencial. Roubos e Jouini (2013) explicam que a distribuição hiperexponencial é uma mistura de duas distribuições exponenciais tais que com probabilidade p com taxa γ_1 e com probabilidade $(1 - p)$ com taxa γ_2 .

Um sistema do tipo cliente - servidor com abandono pode ser visto na Figura 2.11, em que existe a chegada de dois tipos de cliente, por meio dos eventos a_1 e a_2 , a admissão do cliente pelo servidor via eventos c_1 e c_2 , o evento d representa a saída dos clientes do sistema após o atendimento e também o abandono dos clientes através dos eventos b_1 e b_2 .

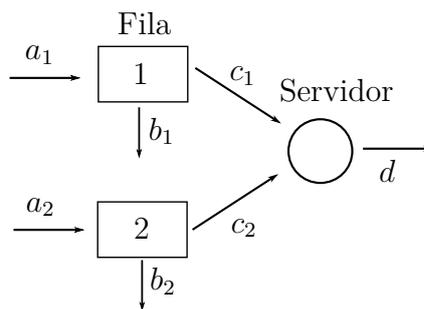


Figura 2.11 – Sistema de clientes - servidor com abandono

2.7 Análise de desempenho de sistemas

A análise do desempenho dos sistemas é de extrema relevância para seu bom funcionamento e desenvolvimento, a qual auxilia na otimização, pode prevenir as falhas ou

detectar gargalos.

A maioria dos sistemas possuem seu comportamento de natureza estocástica, ou seja, são provenientes de procedimentos geralmente incertos. RIBEIRO (2018) explica que para interpretação de um sistema assim denominado é preciso utilizar um banco de dados que demonstre o seu comportamento, possibilitando assim, a obtenção de uma aproximação dos seus resultados.

Ao trabalhar com sistemas de natureza estocástica, é preciso estimar o valor esperado das medidas de desempenho de interesse, podendo ser a taxa de produção, o tempo médio do sistema, o tamanho médio da fila e diversas outras características do sistema observado. Esse valor esperado pode ser definido como a equação 2.32.

$$E[J(\theta)] = \lim_{k \rightarrow \infty} \frac{\sum_{i=1}^k J(\theta, \omega_i)}{k}, \quad (2.32)$$

sendo k o número de amostras coletadas e $J(\theta, \omega_i)$ é uma medida de desempenho que depende de um parâmetro θ e da sequência amostral (ω_i) , (RIBEIRO, 2018). Lembrando que na prática k deve ser finito.

Para SEDs, existem duas principais formas de avaliar o seu desempenho: a modelagem analítica e a simulação. Neste trabalho, por meio dos autômatos estocásticos são aplicadas as propriedades da Cadeia de Markov e teoria de filas para o primeiro método, e o algoritmo de escalonamento de eventos, para a segunda forma.

2.7.1 Modelagem Analítica

CASSANDRAS e LAFORTUNE (2008) explicam que a modelagem analítica consiste em um modelo de sistema de equações matemáticas que descrevem o comportamento de um sistema. Esse método constitui-se em projetar um sistema real através das relações matemáticas existentes no funcionamento do mesmo. Por meio dessas relações pode-se descrever o sistema como um conjunto de estados possíveis e definir transições entre esses estados (SANTOS, 2002).

Os métodos analíticos possuem a grande vantagem de ter um custo computacional menor que a simulação, já que não é necessária uma amostra para representar o comportamento estocástico do sistema, e também ao fato de não ser necessário gerar números pseudo-aleatórios. É possível descrever seu comportamento por meio de um modelo utilizando as propriedades da Cadeia de Markov, quando os tempos de vida dos eventos são distribuídos exponencialmente, que resultam em um sistema de equações lineares.

Mediante a solução do sistema de equações lineares, têm-se as probabilidades estacionárias do sistema, como visto na seção 2.4, podendo obter o valor esperado dos parâmetros de interesse do sistema observado.

Quando os tempos de vida dos eventos não seguem uma distribuição exponencial, uma aproximação analítica pode ser feita por meio de distribuições de fase. RIBEIRO (2018)

explica que neste caso, cada distribuição geral é modelada por uma rede de processos pressupostos com tempos de vida dos eventos distribuídos exponencialmente, na qual, sua combinação descreve a distribuição geral que a representa.

Com isso, os tempos de vida entre eventos podem seguir distribuições diversas e aproximadas em distribuições exponenciais e, conseqüentemente, encontrar uma cadeia de Markov equivalente. Na literatura, pode-se citar o artigo do WHITT (2004) que desenvolveu um algoritmo baseado em uma aproximação Markoviana para um sistema de fila $M/G/s/r + G$.

CASSANDRAS e LAFORTUNE (2008) informam no seu livro que geralmente os sistemas do dia a dia são complexos para serem representados em soluções analíticas. E também, o modelo matemático pode ser válido, mas pode não existir ferramentas para resolver as equações que compõem esse modelo. Outra desvantagem desse método, é que o número de estados cresce de forma exponencial, o que dificulta calcular as probabilidades estacionárias, dependendo da capacidade computacional acessível, podendo ser até mesmo impossível em sistemas complexos.

2.7.2 Simulação

O método de análise de desempenho de sistema através da simulação é definido como um processo de projetar um modelo computacional a partir um sistema real e gerir experimentos com esse modelo, com o objetivo de entender e avaliar estratégias para o seu comportamento e possíveis melhorias (PEGDEN, 1990).

CASSANDRAS e LAFORTUNE (2008) definem a simulação como um processo pelo qual um modelo de sistema é avaliado numericamente e os dados desse processo são usados para estimar várias quantidades de interesse. Já MEDINA e CHWIF (2010) explicam que simular é analisar um novo sistema antes da sua implantação ou melhorar a operação e o funcionamento de sistemas existentes, podendo confrontar resultados e medir eficiências.

BANKS e CARSON (2005) citam no seu livro que o extraordinário avanço da computação, aliado às vantagens provenientes ao uso de simulação e o aperfeiçoamento da sua metodologia, fazem da simulação uma das ferramentas mais utilizadas e aceitas no âmbito de pesquisa operacional e análise de sistema.

No caso do SEDs, a simulação é amplamente utilizada em diversas aplicações como projetar sistemas de manufatura e avaliar seu desempenho, projetar redes de comunicação e testar vários protocolos para lidar com mensagens de recursos de rede e projetar aeroportos ou rodoviárias para lidar com cargas de tráfego projetadas. Sistemas que são considerados estocásticos é de difícil implementação prática e de alto custo para fazer experimentos e verificar melhorias e otimizações.

Para simular SEDs é necessário um número finito de amostras para estimar o valor esperado do parâmetro de interesse, por isso é necessário o uso de técnicas estatísticas para se avaliar a variância em função do número de amostras disponíveis (RIBEIRO, 2018).

De acordo com MEDINA e CHWIF (2010) uma das técnicas que pode ser utilizada para determinar o número de amostras é em função da precisão e da confiança estatística desejada. A precisão considera o tamanho do intervalo e, a confiança estatística é a probabilidade de que o intervalo de confiança contenha a média estimada.

Em técnicas de simulações, são feitas rodadas para coletar vários valores de amostras para os parâmetros de interesse. A rodada é uma repetição da simulação do modelo, com a mesma configuração, a mesma duração e com os mesmos parâmetros de entrada, mas com uma semente de geração dos números aleatórios diferentes (MEDINA; CHWIF, 2010).

Para simular a dinâmica dos sistemas também é considerado o regime transitório e o regime permanente. MEDINA e CHWIF (2010) explica que durante o regime transitório ou tempo de aquecimento, o comportamento do sistema varia muito devido o modelo não está aquecido, e nem todas as partes estão funcionando corretamente, portanto durante esse período as medidas de desempenho fornecidas não são precisas. Já o regime permanente ou estado estacionário mostra o funcionamento do sistema após aquecido, em que as medidas de desempenho já são mais precisas e podem ser consideradas. Portanto, o tempo de aquecimento do sistema pode ser eliminado no momento de análise dos resultados.

Para calcular o intervalo de confiança, é necessário definir o valor esperado da média e da variância do parâmetro observado. O valor esperado da média de um parâmetro θ pode ser calculado como:

$$\hat{\theta}_n = \frac{1}{n} \sum_{i=1}^n \theta_i, \quad (2.33)$$

em que θ_i são os valores das amostras observadas em n rodadas. Para determinar o valor esperado para variância dessas amostras, é utilizada a seguinte equação:

$$E[S_n^2] = \sigma^2 = \frac{1}{n-1} \sum_{i=1}^n (X_i - \hat{\theta}_n)^2. \quad (2.34)$$

Determinando o valor esperado da média e da variância pode-se determinar o intervalo de confiança por meio do teorema do limite central, utilizando os valores de α definidos na tabela *t-Student* (MEDINA; CHWIF, 2010).

$$P \left[\hat{\theta}_n - z_{\frac{\alpha}{2}} \sqrt{\frac{\sigma^2}{n}} \leq \theta \leq \hat{\theta}_n + z_{\frac{\alpha}{2}} \sqrt{\frac{\sigma^2}{n}} \right] \cong 1 - \alpha, \quad (2.35)$$

sendo o α definido de acordo com a probabilidade de que o intervalo de confiança contenha a média. Se for um intervalo com 95% de confiança, o α é de 5% ou 0,05.

RIBEIRO (2018) explica que quanto maior o valor desta probabilidade, maior a confiança estatística de que a média da população encontra-se dentro do intervalo construído. Com isso, a desvantagem da simulação é que para uma maior precisão, é necessário um maior número de amostras e, por consequência, um maior custo computacional.

Metodologia

3.1 Introdução

Neste capítulo, serão apresentados os métodos utilizados para a construção e elaboração do projeto, explicando as metodologias e instrumentos usados para a modelagem e elaboração da metodologia desenvolvida.

De forma geral, como pode-se observar no diagrama de blocos na Figura 3.1, a primeira parte da metodologia, após a definição do sistema a ser explorado, é a elaboração de um modelo que represente a dinâmica do sistema por meio de autômatos estocásticos. Se o sistema que será analisado possuir comportamento cíclico, sem abandonos, um segundo modelo é implementado sem os eventos, que são considerados instantâneos. Todos os modelos são implementados na biblioteca *UltraDES* (LACSED, 2016) e, para cada método de análise de desempenho são realizadas algumas operações.

Após a implementação do modelo de forma modular, para o método analítico é obtido o autômato global que representa o comportamento do sistema como um todo, por meio da operação da composição paralela, considerando os dois modelos: o modelo completo e o modelo sem os eventos instantâneos. Esses são eventos que possuem um tempo de vida muito pequeno. E então, é atribuído a cada um dos estados do autômato global um índice, para a construção de todas as transições possíveis do sistema.

De posse de todas as transições, com auxílio do *software Matlab*, é feita a construção da matriz de probabilidade de transição (M_q). E por meio dessa matriz, é definido o sistema linear que representa as probabilidades de cada estado do sistema. Através das probabilidades, é realizada a análise dos parâmetros de interesse.

Já para o método via simulação, após a implementação do modelo modular na biblioteca *UltraDES*, é utilizada a operação de projeção inversa para que todos os autômatos reconheçam todos os eventos e, então são definidas as matrizes de transição de cada autômato estocástico. Dispondo das matrizes de transição, é implementada a técnica de escalonamento de eventos no *software Matlab*. E, por meio dessa técnica, é realizada a análise de desempenho dos parâmetros de interesse.

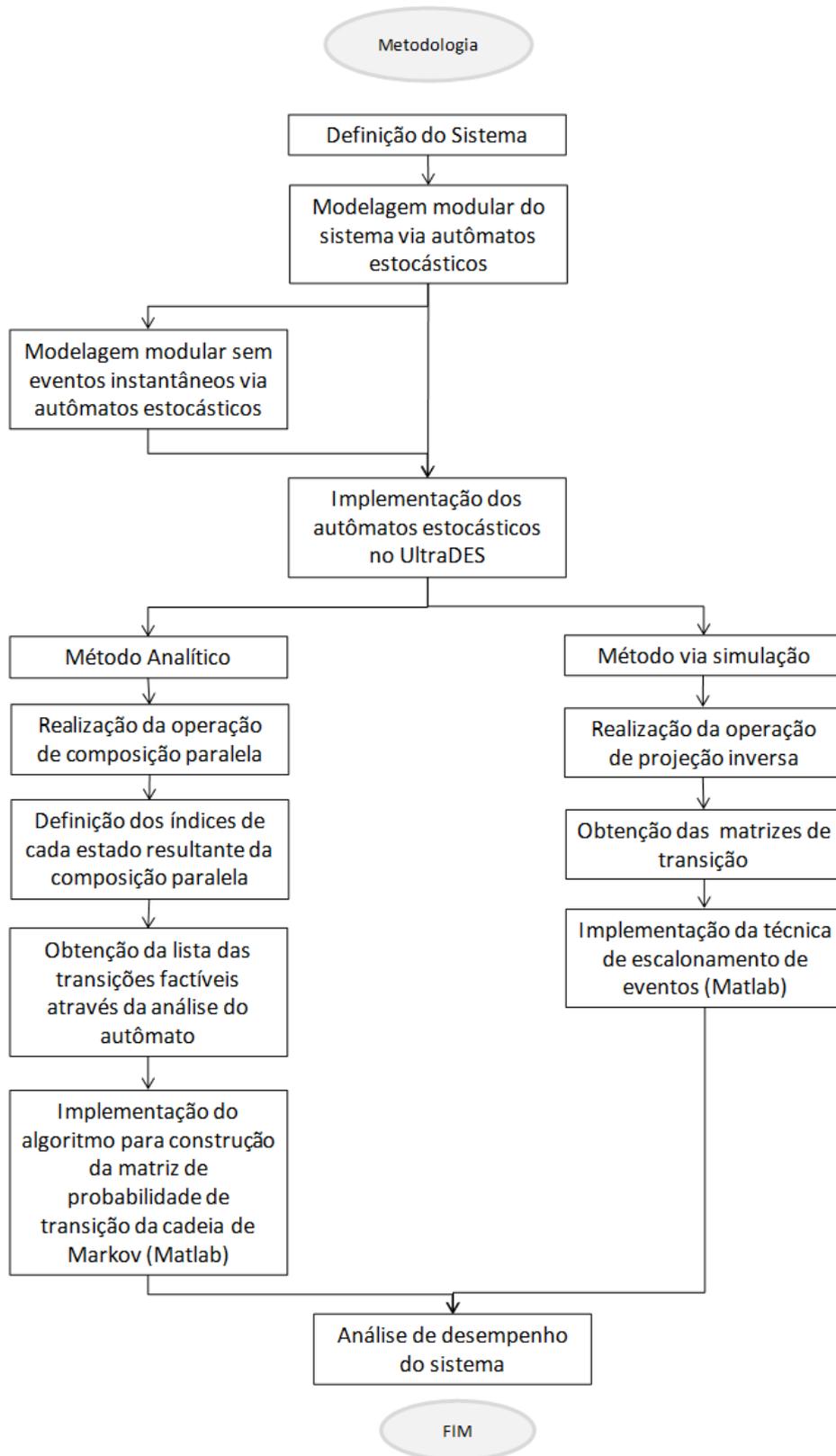


Figura 3.1 – Diagrama de blocos da metodologia

Todo trabalho computacional foi implementado em um notebook com sistema operacional Windows 10, processador Intel Core i7, 8 GB de memória RAM e sistema operacional de 64 bits, utilizando o *software Matlab* e com auxílio do *Visual Studio* para executar a biblioteca *UltraDES* (LACSED (2016)).

3.2 Modelagem dos problemas

O primeiro tópico do trabalho é a obtenção do modelo do sistema proposto para validação, modelo esse que descreva e represente o comportamento do sistema. Ao definir o modelo a ser analisado, a modelagem do sistema requer muito cuidado, pois para os resultados apresentarem confiabilidade é necessário um modelo que represente bem o sistema. Por isso, é preciso analisar todas as partes do sistema de forma específica e modular, evitando erros nesta etapa.

3.2.1 Modelagem modular

Para obter o comportamento dos sistemas como um todo, é necessário observar todas as possíveis partes que o sistema pode estar, para que o modelo construído represente bem o sistema e os resultados sejam coerentes. A modelagem de forma modular consiste em separar o sistema por partes, como visto na seção 2.3.2.2, e através da composição paralela obter o comportamento do sistema como um todo.

Essa modelagem permite representar sistemas com grande quantidade de estados facilmente, pois ao dividir em sub-partes se torna mais visível cada função. Podendo analisar cada parte do sistema de forma individual.

Ao observar cada parte do sistema se torna possível representar cada particularidade e assim perceber como ele evolui. Com isso, a construção do modelo torna-se mais clara e eficiente e, reduz os erros na etapa de modelagem.

Além disso, ao elaborar o modelo de forma modular por meio dos autômatos é possível obter facilmente a cadeia de Markov resultante e representar o sistema por meio de matrizes de transições para análise via simulação, não necessitando do autômato global para simular o comportamento do sistema.

Como exemplo, pode-se observar a modelagem do problema dos Veículos Guiados Automatizados (AGVs), seção 4.1, que de forma individual é simples e fácil de modelar e de entender o seu funcionamento. Já o comportamento completo do sistema resultou em 25.920 estados, o que seria impossível modelar com o autômato único.

Por isso, a modelagem modular juntamente com a operação de composição paralela torna-se eficiente e prática para a representação de sistemas gerais, especificamente para sistemas com grande número de estados.

3.3 Método analítico

O método analítico utilizou as propriedades da cadeia de Markov para obter os resultados. O comportamento do sistema completo, foi alcançado ao realizar a operação da composição paralela dos autômatos, obtidos na modelagem modular do sistema.

A operação de composição paralela foi realizada na biblioteca *UltraDES* (LACSED, 2016). É importante ressaltar a importância da utilização dessa biblioteca, pois em sistemas com grande número de estados torna-se complicado e complexa a representação e obtenção do sistema completo.

Por meio do autômato resultante da composição paralela, é observada cada transição e, através dela, é construída a matriz de probabilidade de transição (M_q). Essas transições são geradas por meio da análise de cada estado possível do sistema. Com o auxílio do *UltraDES*, foi possível atribuir rapidamente um índice para cada um desses estados, facilitando assim a obtenção das transições.

Através dos índices, a biblioteca *UltraDES* escaneia o autômato e grava em um arquivo, no formato ".txt", todas as transições do autômato, as transições são gravadas em três colunas. A primeira coluna representa o índice do estado que o sistema se encontra, a segunda coluna mostra o estado em que o sistema irá, com o evento representado na terceira coluna. Essa análise é realizada de forma a garantir que a soma das taxas das transições de chegada ao estado são iguais às taxas das transições de saída.

Por meio dessa análise das transições do arquivo, é obtido a matriz M_q , sendo possível construir o sistema linear que representa o comportamento completo do sistema. Por se tratar de probabilidade é necessário garantir que o somatório das probabilidades seja igual a 1, por isso, uma das equações é modificada para garantir essa propriedade.

Após isso, foi utilizada a função *mldivide* do *software Matlab* para resolução do sistema linear. Através dos resultados encontrados, pela solução do sistema linear, pode ser realizada a análise de desempenho de acordo com os parâmetros de interesse do problema avaliado.

Uma desvantagem desse método é que o número de estados cresce de forma exponencial, em sistemas grandes e complexos, resultando em sistemas lineares com grande quantidade de variáveis, tendo um alto custo computacional para a resolução.

A seção 2.4.1 explica e exemplifica todos os passos para derivar o modelo analítico por meio da modelagem modular via autômatos estocásticos.

3.3.1 Eliminação de eventos instantâneos

Para o método analítico foram elaborados dois modelos, o primeiro considerando todos os eventos e o segundo eliminando os eventos instantâneos. Os eventos instantâneos são aqueles cujo tempos de vida são muito pequenos, ou seja, assim que estiverem factíveis

são disparados. A probabilidade do sistema ser encontrado em estados, que possuem esses eventos como evento de saída, são infinitesimais, e podem ser desconsideradas.

Esses eventos servem para organizar a sequência que o sistema evolui. Eles são adicionados para facilitar na modelagem modular dos sistemas. Retornando ao exemplo da fila simples e, utilizando a cadeia de Markov obtida pelo resultado da composição paralela do sistema, mostrada novamente na Figura 3.2.

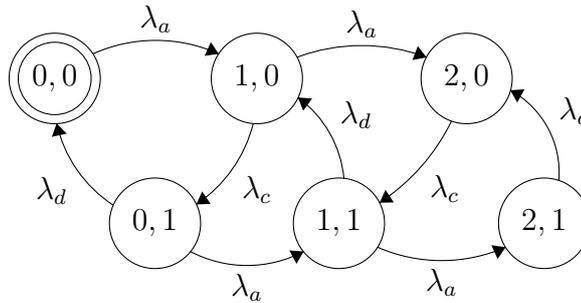


Figura 3.2 – Cadeia de Markov do sistema da fila simples

Lembrando que, $\lambda_a = 1$, $\lambda_c = 10^3$, e $\lambda_d = 2$. Os resultados das probabilidades estacionária do sistema de fila ser encontrado em cada um dos estados em regime permanente foram:

$$\begin{cases} \pi_{00} = 53.2516\% \\ \pi_{10} = 0.0799\% \\ \pi_{01} = 26.6258\% \\ \pi_{11} = 13.3528\% \\ \pi_{20} = 0.0134\% \\ \pi_{21} = 6.6764\% \end{cases} \quad (3.1)$$

Ao retirar os eventos instantâneos, nesse caso λ_c , o sistema funcionaria de acordo com a Figura 3.3.

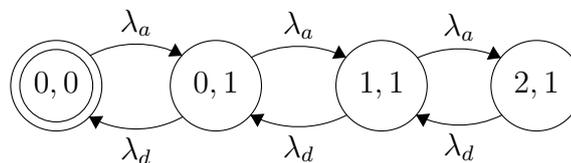


Figura 3.3 – Cadeia de Markov do sistema da fila simples sem os eventos instantâneos

Uma explicação matemática para a eliminação dos eventos instantâneos é que o somatório das taxas de chegadas ao estado são iguais ao somatório das taxas de saídas, para garantir que o sistema esteja em equilíbrio. Considerando o estado (1,0), da Figura

3.2, em que há um evento considerado instantâneo factível, pode-se elaborar a equação de probabilidade do estado como:

$$\textit{Estado} (1, 0) : \pi_{1,0}(\lambda_a + \lambda_c) = \lambda_a\pi_{0,0} + \lambda_d\pi_{1,1}. \quad (3.2)$$

Como a taxa do evento c é muito grande podemos desconsiderar a taxa do evento a , então é possível descrever que evento c equivale aos dois eventos de chegada ao estado $(1, 0)$, como mostrado na equação 3.3.

$$\textit{Estado} (1, 0) : \lambda_c\pi_{1,0} = \lambda_a\pi_{0,0} + \lambda_d\pi_{1,1}. \quad (3.3)$$

Portanto é possível excluir o estado $(1, 0)$, e as duas transições de chegada a esse estado são modificadas para o estado $(0, 1)$ que seria o estado que o evento c levaria o sistema. Da mesma forma pode-se fazer essa mesma análise para o estado $(2, 0)$, da Figura 3.2, que existe o evento c factível.

$$\textit{Estado} (2, 0) : \lambda_c\pi_{2,0} = \lambda_a\pi_{1,0} + \lambda_d\pi_{2,1}. \quad (3.4)$$

A retirada dos eventos instantâneos reduz a quantidade de estados resultantes no comportamento do sistema como um todo, pode-se verificar que no exemplo utilizado o sistema passou de seis estados para quatro estados, representando o mesmo sistema, com um menor número de estados e com ganho computacional.

A eliminação dos eventos instantâneos, em sistemas com grande número de estados, é realizada de forma a elaborar um segundo modelo através do primeiro, em que é observado como o sistema evolui com os eventos instantâneos e assim modelado a dinâmica do sistema sem os eventos instantâneos. Esse fato ocorre devido não ser possível analisar cada estado do autômato resultante da composição paralela com grande número de estados. Portanto, o modelo elaborado sem os eventos instantâneos tem como base a evolução do comportamento do sistema com os eventos instantâneos, que foi modelado primeiro.

Para verificar a veracidade da eliminação dos eventos em sistemas maiores, foi realizado um estudo no problema dos veículos guiados automatizados, sistema que funciona de forma cíclica. Essa eliminação só é válida para o método analítico, pois no método de simulação é necessário que o sistema evolua passo a passo, sendo necessária a ocorrência dos eventos instantâneos. O modelo sem os eventos instantâneos foi verificado no método via simulação, não funcionando de forma aceitável.

3.4 Método via simulação

Para análise de desempenho do sistema por meio da simulação, foi utilizada uma técnica padrão descrita no livro do CASSANDRAS e LAFORTUNE (2008), conhecida como esquema de escalonamento de eventos.

Para utilização da técnica é necessário construir a tabela de transição que descreve o que acontece em cada estado quando um evento é disparado. Essa tabela foi construída com o auxílio da biblioteca *UltraDES*, que gera essas matrizes através da análise dos autômatos implementados.

Considerando como exemplo, o sistema de filas do capítulo 2, ilustrado novamente na Figura 3.4, para construir a tabela de transição que é utilizada na técnica.

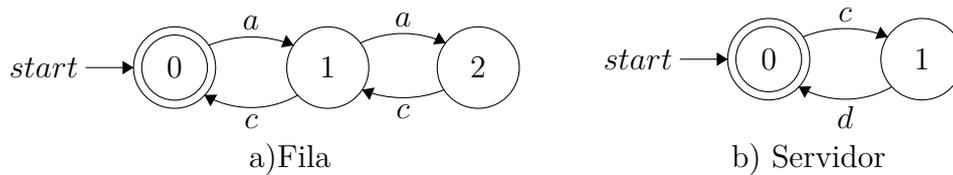


Figura 3.4 – Autômatos para o sistema de fila de capacidade 2

A tabela é construída de acordo com os eventos factíveis em cada estado, por exemplo: Considerando apenas o autômato da fila (3.4 - a)), observa-se que no estado 0 se ocorrer um evento a o sistema irá para o estado 1, assim no estado 1 se ocorrer um evento a o sistema irá para o estado 2, ocorrendo o evento c retorna ao estado 0. Essa análise é realizada para todos os eventos e para todos os estados de cada autômato de forma individual. A Tabela 3.1 exemplifica as tabelas de transições dos autômatos do sistema de fila de capacidade dois.

Tabela 3.1 – Tabela de transição do sistema de fila de capacidade 2

(a) Tabela de transição - fila

$E \setminus X$	0	1	2
a	1	2	-1
c	-1	0	1

(b) Tabela de transição - servidor

$E \setminus X$	0	1
c	1	-1
d	-1	0

O valor -1 significa que o evento não é factível naquele estado. É importante lembrar que, como mencionado seção 2.3.2, X é o conjunto finito de estados e E é o conjunto finito de eventos associado ao autômato.

Para facilitar na construção das matrizes dos problemas propostos, cada autômato passou a reconhecer todos os eventos através da propriedade da operação de projeção inversa, para que assim não fosse necessário identificar em todas as transições, qual autômato reconhecia aquele evento.

Com essa modificação, por meio da operação de projeção inversa, os autômatos que representam o sistema resultariam nos autômatos da Figura 3.5.

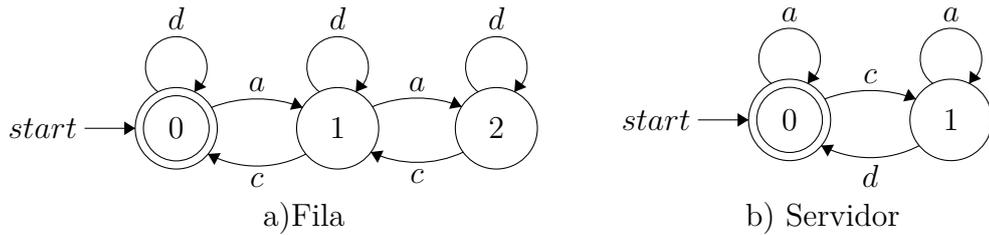


Figura 3.5 – Autômatos para o sistema de fila de capacidade 2 com projeção inversa

Assim, as tabelas de transições mudariam para as seguintes:

Tabela 3.2 – Tabela de transição do sistema de fila de capacidade 2 com projeção inversa

(a) Tabela de transição - fila

$E \setminus X$	0	1	2
a	1	2	-1
c	-1	0	1
d	0	1	2

(b) Tabela de transição - servidor

$E \setminus X$	0	1
a	0	1
c	1	-1
d	-1	0

A operação de projeção inversa é utilizada para que em sistemas maiores e com grande quantidade de eventos não seja necessário, a cada transição, verificar qual autômato reconhece aquele evento, reduzindo assim, o custo computacional.

Definindo a matriz de transição, o próximo passo é a construção do calendário de eventos, que é definido de acordo com os tempos de vida de cada evento, no estado atual do sistema.

Considerando novamente o exemplo do sistema de fila, pode-se construir o calendário de eventos para esse sistema, considerando que os tempos de vida dos eventos a, c e d são respectivamente, $1, 1; 0, 01$ e 2 . Por meio da Tabela 3.3 pode ser observado o calendário dos eventos do sistema de fila usado para exemplo.

Tabela 3.3 – Calendário de eventos

$X_0 = (0, 0)$		$X_1 = (1, 0)$		$X_2 = (0, 1)$		$X_3 = (1, 1)$		$X_4 = (1, 0)$	
$t_0 = 0$		$t_1 = 1.10$		$t_2 = 1.11$		$t_3 = 2.20$		$t_4 = 3.11$	
$\Gamma(X_0) = \{a\}$		$\Gamma(X_1) = \{a, c\}$		$\Gamma(X_2) = \{a, d\}$		$\Gamma(X_3) = \{a, d\}$		$\Gamma(X_4) = \{a, c\}$	
e	t	e	t	e	t	e	t	e	t
a	1, 10	a	2, 20	a	2, 20	a	3, 30	a	3, 30
		c	1, 11	d	3, 11	d	3, 11	c	3, 12
$X_1 = f(X_0, a)$		$X_2 = f(X_1, c)$		$X_3 = f(X_2, a)$		$X_4 = f(X_3, d)$		$X_5 = f(X_4, c)$	

Em que X_n são os estados do sistema, t_n é o tempo, $\Gamma(X_n)$ é o conjunto de eventos factíveis naquele estado. De forma geral, o método de escalonamento eventos consiste em, inicializar o sistema a partir de um tempo, $t = 0$, considerando que o sistema se encontra no estado inicial $x_0 \in X$, verifica - se os eventos factíveis no estado atual e com

base nesses eventos associa-se um tempo de vida para cada evento, ocorrendo o evento de menor tempo. Seja y o vetor dos tempos associados a cada evento i , o evento e que irá ocorrer a seguir naquele estado é:

$$e = \underset{i \in \Gamma(x)}{\operatorname{arg\,min}} \{y_i\}. \quad (3.5)$$

Com isso é conhecido o próximo evento que ocorrerá no sistema. O passo seguinte é atualizar o estado do sistema, através da função de transição, $f(x, e) = x'$. O tempo gasto para a ocorrência do evento é:

$$y^* = \underset{i \in \Gamma(x)}{\operatorname{min}} \{y_i\}. \quad (3.6)$$

Por meio da definição de y^* , na equação 3.6, é possível atualizar o tempo do sistema:

$$t' = t + y^*. \quad (3.7)$$

Ao atualizar o estado e o tempo da simulação, verificam-se novamente os eventos factíveis para o novo estado, e repetem os passos descritos anteriormente. Os eventos que estavam habilitados no estado anterior e continuam habilitados no novo estado mantêm o mesmo tempo de vida, os eventos que forem desabilitados saem do sistema de escalonamento.

Resumindo, essa técnica consiste em realizar para cada ocorrência de um evento os seguintes passos (CASSANDRAS; LAFORTUNE, 2008):

1. Remover a primeira linha do calendário;
2. Atualizar o tempo com o valor de t_1 ;
3. Atualizar o estado de acordo com $x' = f(x, e_1)$;
4. Apagar no calendário as linhas correspondentes a eventos infactíveis no novo estado;
5. Acrescentar ao calendário os eventos factíveis que ainda não estejam escalonados;
6. Reordenar o calendário por ordem crescente do valor do tempo em que os eventos foram escalonados.

Vale ressaltar que, para o método via simulação, devido à modelagem modular, não é necessário realizar a operação da composição paralela para a obtenção do autômato global que representa o comportamento do sistema como um todo. Portanto, tem-se uma economia no uso de memória e também no custo computacional.

Problemas e Resultados

4.1 Apresentação

Este capítulo exemplifica os problemas utilizados para validação da metodologia elaborada, desde a modelagem até a parte da análise de desempenho dos sistemas. Foram utilizados um problema adaptado do livro de simulação de eventos discretos do BANKS e CARSON (2005) e, um sistema do tipo cliente - servidor com clientes impacientes.

4.2 Problema dos veículos guiados automatizados

O problema dos **AGVs** consiste em determinar a máxima produtividade de uma manufatura que realiza o transporte de peças em três estações de trabalho em série posicionadas ao redor de um trilho, como ilustrado na Figura 4.1.

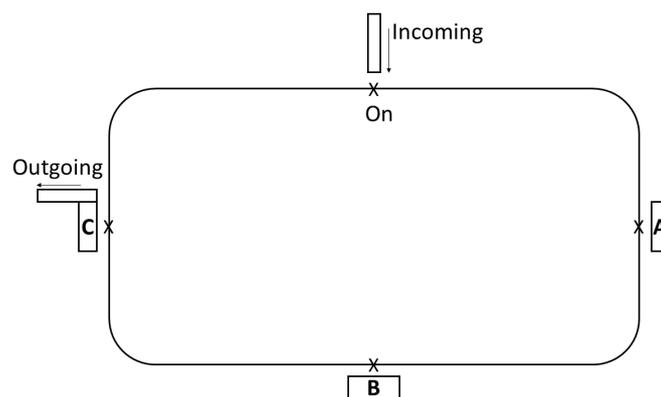


Figura 4.1 – Trilho e estações de trabalho
Fonte: Adaptado de (BANKS; CARSON, 2005)

Os AGVs percorrem o trilho no sentido horário, inicialmente vazios, partindo do ponto de entrada *On*. As peças são colocadas na esteira de entrada e recolhidas para o AGV, uma de cada vez. Após a peça ser colocada no AGV, ele a transporta até a primeira estação de trabalho para o processamento. Ao finalizar o processo o AGV leva a peça

para a estação seguinte, e assim até completar o ciclo e ser automaticamente descarregada após o processamento da máquina C e o AGV segue para a fila de AGVs vazios no ponto de entrada.

O tempo gasto para retirar uma peça da esteira *Incoming* e inserir no AGV para transporte é de 30 segundos. Quando a peça concluir o processo nas três estações de trabalho, a operação de descarga leva 45 segundos.

As posições das máquinas de trabalho são mostradas de forma aproximada no trilho, podendo variar de acordo com a capacidade da fila da máquina. A restrição é que cada estação deve ser colocada na seção do trilho na qual ela está mostrada na Figura 4.1.

Para cada estação de trabalho é associado um tempo de processamento por peça, tempo de falha, (Mean Time To Failure (MTTF)) e tempo de reparo, (Mean Time To Repair (MTTR)), mostrados na Tabela 4.1.

Tabela 4.1 – Tempo de processamento, falha e reparo de cada estação de trabalho

Estação	Processamento (s)	MTTF (min.)	MTTR (min.)
A	120	90	3,1
B	100	82	4,0
C	45	20	5,0

O tempo de falha e de reparo são distribuídos exponencialmente em torno das médias especificadas em MTTF e MTTR para cada estação de trabalho. O tempo de falha não depende do funcionamento da máquina, portanto a falha ocorre com a estação estando em operação ou ociosa.

4.2.1 Modelagem modular do problema por autômatos

A modelagem do problema resultou em dez autômatos, em que são divididos em quatro autômatos gerais: esteira de entrada, estação de trabalho, fila e quebra. De acordo com SOUZA e MAIA (2018), os estados dos autômatos foram nomeados como:

1. (V_x): indicando estação x vazia, $x = Incoming, A, B$ e C ;
2. (P_x): indicando estação x processando, $x = A, B$ e C ;
3. (F_x): indicando estação x pronta, mas com o AGV parado, $x = Incoming, A, B$ e C ;
4. (O_x): indicando estação x ocupada, $x = Incoming$;
5. (iF_x): indicando o tamanho da fila da estação x, $i =$ variando de acordo com a capacidade da fila e $x = A, B, C$;
6. (x_T): indicando estação trabalhando, $x = A, B, C$;

7. (x_F): indicando estação com falha, $x = A, B, C$.

E para os eventos foram utilizados os nomes (SOUZA e MAIA (2018)):

1. (x_{in}): chegada do AGV na estação x , $x = Incoming, A, B$ e C ;
2. (x_{end}): finalizou processo na estação x , $x = A, B$ e C ;
3. (x_{out}): saída do AGV da estação x , $x = Incoming, A, B$ e C ;
4. (f_x): falha da estação x , $x = A, B$ e C ;
5. (r_x): reparo da estação x , $x = A, B$ e C .

4.2.1.1 Autômato da esteira de entrada

A esteira de entrada foi modelada com dois estados, em que o estado inicial (V_{in}) representa esteira vazia, ao ocorrer o evento In_{in} que simboliza a inserção da peça no AGV, ocorre a evolução do autômato para o estado ocupado (O_{in}). Se houver espaço na fila da estação A, ocorre o evento In_{out} , e o autômato retorna ao estado inicial, permitindo que aconteça uma nova retirada de peça por um próximo AGV. O autômato que ilustra a esteira de entrada pode ser visto na Figura 4.2.

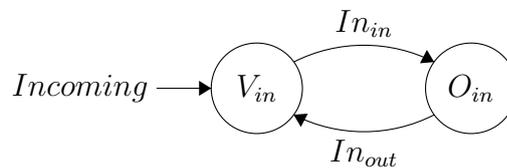


Figura 4.2 – Autômato da esteira de entrada

O tempo de vida do evento In_{in} é de 30 segundos e o evento In_{out} é considerado instantâneo, ou seja, assim que factível pode ser disparado.

4.2.1.2 Autômatos das estações

As três estações de trabalho foram modeladas da mesma forma, como pode ser visto na Figura 4.3. Cada estação de trabalho contém três estados, considerando a estação A para a explicação, ilustrada na Figura 4.3 a), o estado inicial (V_A) representa a estação ociosa e vazia. Ao ocorrer o evento de admissão de uma peça (A_{in}) o sistema evolui para o estado de processamento P_A e ao finalizar o trabalho na peça, ocorre o disparo do evento A_{end} para que o sistema passe ao estado F_A que significa que a máquina está ociosa, mas com uma peça aguardando para sair.

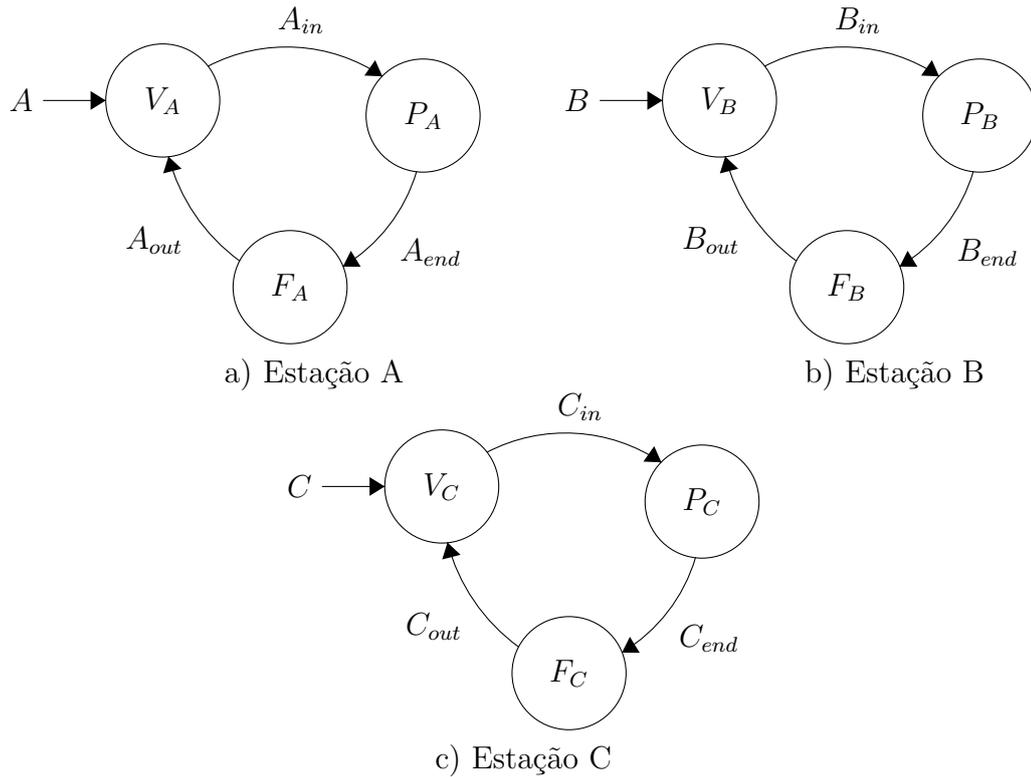


Figura 4.3 – Autômatos das estações

Se houver um espaço entre a estação que se encontra no estado F_A e a fila da estação seguinte, o AGV retira a peça instantaneamente para que assim a máquina possa iniciar um novo processo. Os eventos x_{in} e x_{out} de cada estação, são considerados eventos instantâneos. Já os eventos x_{end} seguem o tempo de processamento de cada estação, representados na Tabela 4.1.

4.2.1.3 Autômatos das filas de cada estação

A posição das estações no trilho define a quantidade de AGVs enfileirados que a fila de cada estação suporta. Foi realizado um estudo para verificar a melhor posição das estações de trabalho que resultasse a maior produtividade, que será explicado na seção 4.2.3.

De acordo com o resultado da análise, as estações ficaram posicionadas de forma que a estação A tinha capacidade de 2 AGVs na fila, a estação B e C, 3 e 4 AGVs em fila respectivamente.

Os autômatos da Figura 4.4, representam os modelos das filas das estações. Por exemplo, de acordo com a Figura 4.4 b), a fila da estação B, tem-se o estado inicial é $0F_B$, a fila aumenta de acordo com a saída de peça da estação anterior, no caso estação A, representada pelo evento A_{out} . O decremento da fila se dá pela ocorrência do evento B_{in} que significa a admissão da peça pela estação B, o que faz a retirada de uma das peças que estão na fila aguardando o serviço.

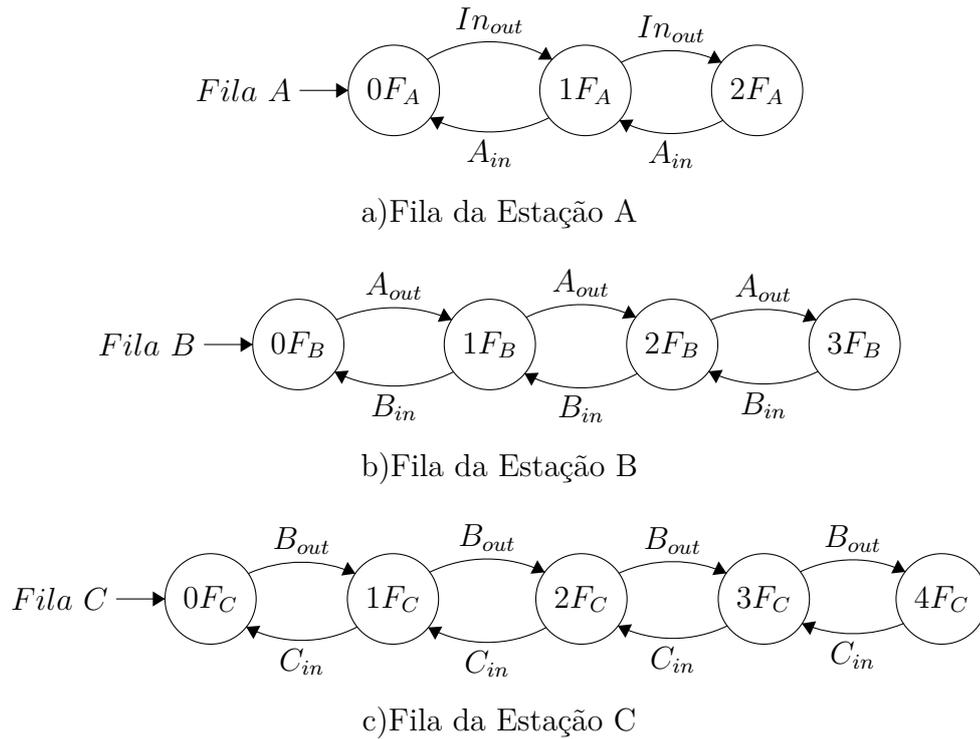


Figura 4.4 – Autômatos das filas das estações

O estado inicial ($0F_B$) representa fila vazia, o estado $1F_B$ simboliza um AGV na fila e assim sucessivamente até alcançar a capacidade total da fila.

4.2.1.4 Autômatos da falha e reparo de cada estação

Cada estação da manufatura está sujeita a quebras independentemente se a máquina estiver operando ou não. A Figura 4.5 ilustra o modelo das falhas de cada estação.

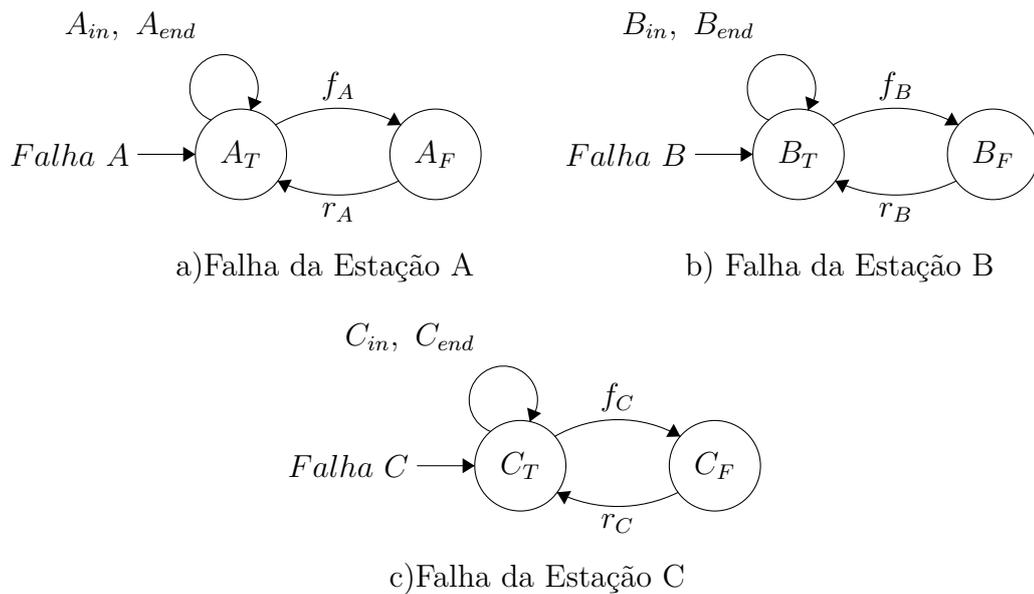


Figura 4.5 – Autômatos das falhas das estações

Utilizando a estação C como exemplo, representado pela Figura 4.5 c), o estado inicial (C_T) representa a estação funcionando normalmente, estado em que a máquina está apta a receber uma peça ou processar uma peça através dos eventos C_{in} e C_{end} . Quando ocorre uma quebra através do evento f_C o sistema passa ao estado C_F , proibindo que ocorra uma chegada ou processamento da peça até que ocorra o evento de reparo, r_C . A saída de peça através do evento C_{out} é factível nos dois estados. Os tempos de vida dos eventos de falha e reparo são mostrados na Tabela 4.1.

4.2.2 Modelagem do problema sem os eventos instantâneos

Os eventos instantâneos, como mostrado na seção 3.3.1, podem ser eliminados para a análise através do método analítico. Essa eliminação reduz a quantidade de estados e transições que representam o sistema como um todo, ajudando assim na redução do custo computacional.

A modelagem do problema dos AGVs passou de dez autômatos para sete autômatos, em que apenas os autômatos das falhas e reparos permanecem os mesmos, apenas eliminando o evento x_{in} .

Lembrando que, nos sistemas com grande número de estados, é elaborado um segundo modelo com base no modelo com os eventos instantâneos, pois não é possível analisar cada estado resultante do sistema completo. Mas, é possível analisar como seria a evolução do modelo sem os eventos instantâneos e remodelar a dinâmica do sistema.

4.2.2.1 Autômato da esteira de entrada

O autômato que representa o comportamento da esteira de entrada sem os eventos instantâneos, Figura 4.6, tem a mesma quantidade de estados, mas agora o que retorna ao estado inicial é o disparo do evento A_{end} .

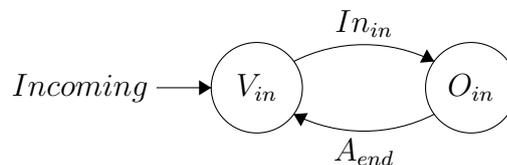


Figura 4.6 – Autômato esteira de entrada sem os eventos instantâneos

4.2.2.2 Autômatos das estações e das filas

Com a retirada dos eventos instantâneos os autômatos das estações se uniram com os eventos das filas, como pode ser visto na Figura 4.7.

Em que pode-se observar que o incremento é ocasionado pelo evento de processamento da estação que está sendo observada a fila, e o decremento é o evento de processamento da próxima máquina.

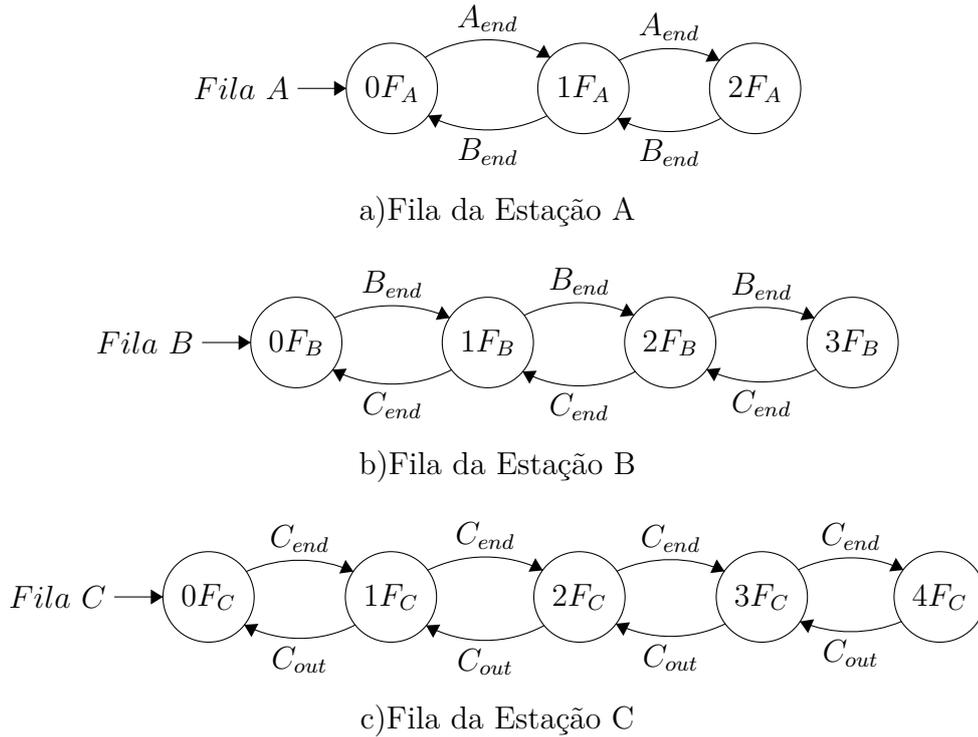


Figura 4.7 – Autômatos das filas das estações sem os eventos instantâneos

4.2.3 Analisando as posições de cada estação

As estações poderiam variar sua posição no trilho, respeitando a regra de permanecer na mesma seção que originalmente foram colocadas. Podendo conter de 1 a 3 AGVs em fila. Foi realizado um estudo variando a posição de cada estação com o objetivo de encontrar a melhor posição das estações para gerar a maior produtividade com uma menor variância.

SOUZA e MAIA (2018) explicaram que a estação A poderia ocupar a posição de 1 a 3, a estação B poderia ficar na posição 4 a 6, já a estação C ocuparia a posição de 7 a 9, como mostrado na Figura 4.8.

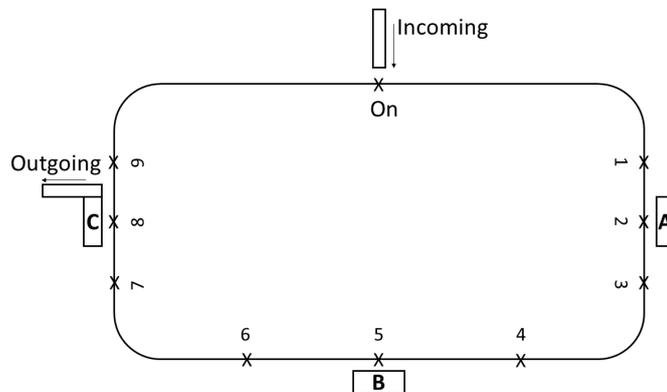


Figura 4.8 – Trilho para as posições das estações
 Fonte: Adaptado de (BANKS; CARSON, 2005)

Cada estação foi analisada separadamente, a primeira a ser analisada foi a estação A, em que a estação B e C foram fixadas nas posições 6 e 9 respectivamente. Com isso, quando a estação A estivesse na posição 1, portando apenas de 1 AGV na fila, a estação B estava com capacidade de 5 AGVs em fila, já a estação C ficava com a quantidade fixa de 3 AGVs. Assim quando a estação A fosse para a posição 2, podendo ter até 2 AGVs na sua fila a estação B passaria a ter capacidade de 4 AGVs, e assim progressivamente. A imagem 4.9 mostra o resultado da produção da manufatura ao variar a posição da estação A.

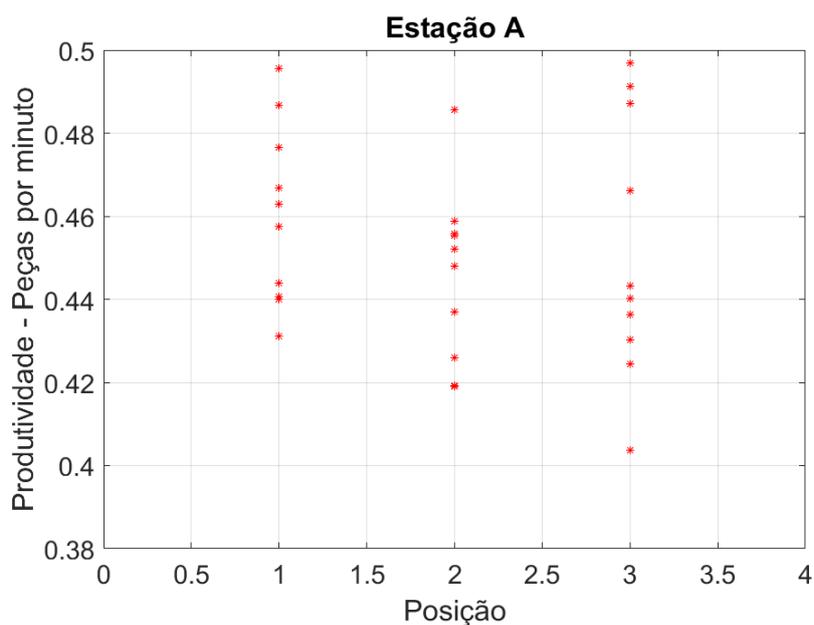


Figura 4.9 – Variando Estação A

A segunda estação a ser analisada, foi a estação B. Em que a estação A foi fixada com capacidade de 3 AGVs na fila, posição 3 e a estação C ficaria na posição 9. Assim, quando a estação B ocupar a posição 4, com um AGV na fila, a estação C teria capacidade máxima de 5 AGVs. O gráfico da Figura 4.10 mostra o resultado da produtividade ao variar a estação B com 1 a 3 AGVs em fila.

Ao variar a estação C, as estações A e B foram fixadas com 3 AGVs em sua fila, sendo colocadas nas posições 3 e 6 respectivamente. A Figura 4.11 mostra o gráfico resultante da análise da estação C em suas 3 posições.

Analisando o resultado das três estações, a Figura 4.9 mostra que a estação A teve uma menor variância quando ficou posicionada na posição 2, em que teria 2 AGVs em fila. Já a estação B, pela Figura 4.10, pode-se perceber uma menor variância na posição 5, fazendo com que a estação B tenha capacidade de 3 AGVs em sua fila.

A Figura 4.11 mostra que a estação C teve a maior produtividade e menor variância estando na posição 9. Pode-se observar também que quanto maior a sua capacidade de fila, maior era a sua produtividade. Portanto, a estação C, ocupou a posição 9, podendo conter 4 AGVs em sua fila. A Tabela 4.2 mostra as posições definidas para as três estações.

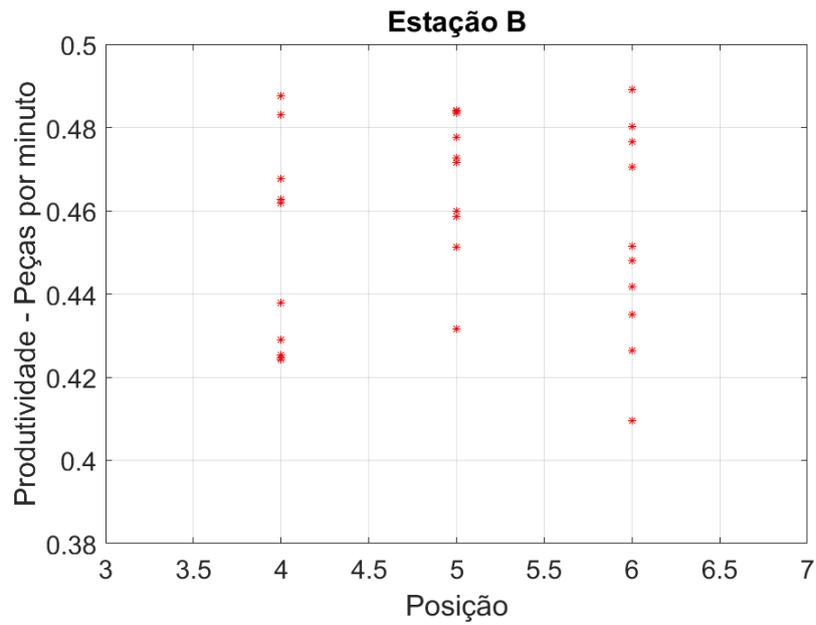


Figura 4.10 – Variando Estação B

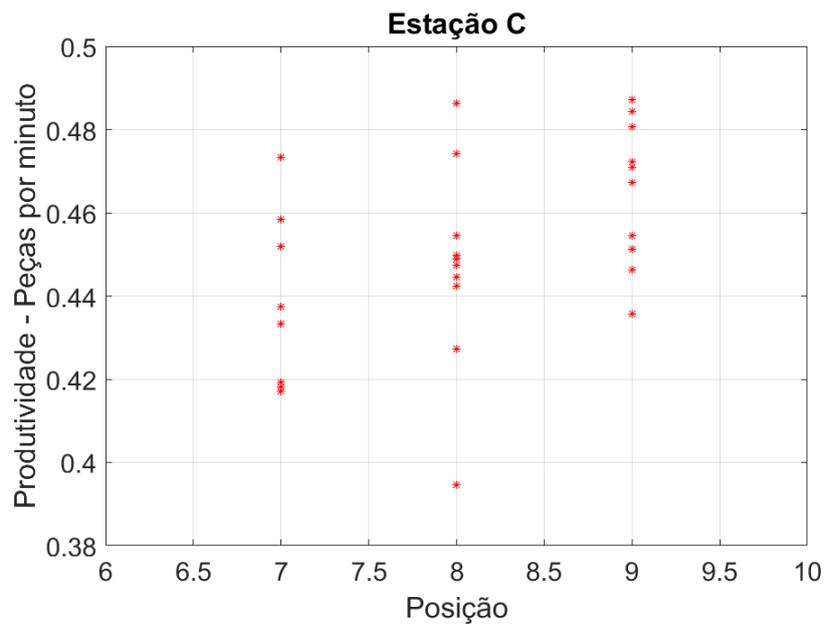


Figura 4.11 – Variando Estação C

Outra consideração para definir as posições das estações no trilho, foi o tempo de processamento de cada estação, definidos na Tabela 4.1, em que quanto maior o tempo de processamento da máquina, menor seria a capacidade de fila necessária. Por exemplo, a estação A possui o maior tempo de processamento, portanto, a capacidade da sua fila poderia ser menor. Já a estação C, por ter um tempo de processamento menor, seria necessária uma fila com maior capacidade de AGVs.

Tabela 4.2 – Posições das estações

Estação	Posição	Capacidade da Fila
A	2	2
B	5	3
C	9	4

Cada posição foi analisada com simulações de 10 rodadas com tempo de aquecimento de 10.000 segundos e com tempo total de 100.000 segundos. A variância da produtividade do sistema se dá em função das incertezas associadas às falhas e aos reparos.

As posições escolhidas para cada estação, Tabela 4.2, foram utilizadas para simular a máxima produtividade do sistema.

4.2.4 Resultados

O objetivo do problema dos AGVs era encontrar a máxima produtividade do sistema. Para isso, o método de simulação através do escalonamento de eventos calculou a produtividade através da quantidade de saída de peças, representada pelo evento C_{out} . Por se tratar de um método de simulação, não apresenta exatidão nas respostas, portanto os resultados são representados por medidas de confiabilidade.

A produtividade máxima foi encontrada no intervalo de confiança de 95% de 0,387931 a 0,405143, com uma média de 0,403517 peças por minutos, em uma simulação de 100.000 segundos, em que os primeiros 10.000 segundo são considerados como tempo de aquecimento, em 10 rodadas. O gráfico da Figura 4.12 mostra a variância da produtividade.

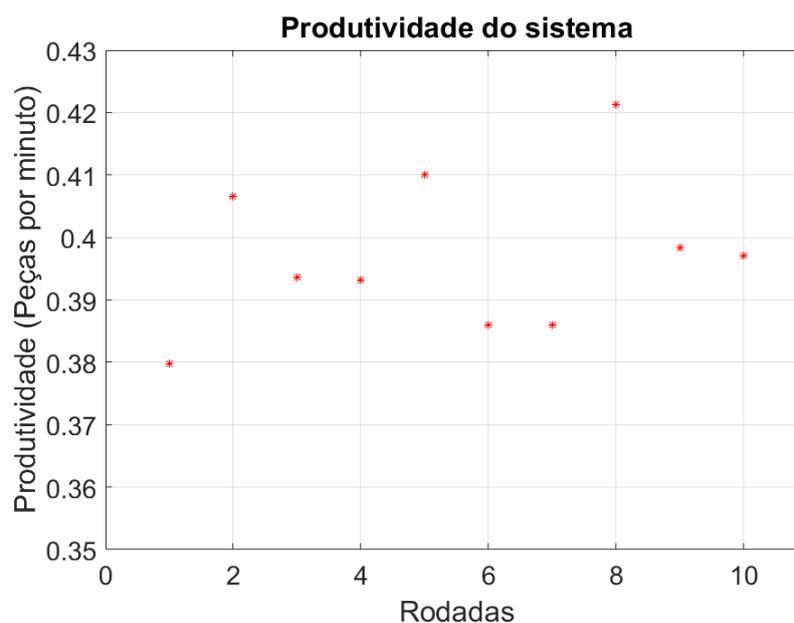


Figura 4.12 – Produtividade do sistema

Por meio do método analítico a máxima produtividade encontrada foi de 0,412263, utilizando o sistema completo e ao retirar os eventos instantâneos foi de 0,416461. A grande diferença entre o sistema completo e sem os eventos instantâneos foi o custo computacional, pois devido à grande quantidade de estados, com os eventos instantâneos, resultou em uma matriz de probabilidade de transição com tamanho de 25.920×25.920 . A maior parte do tempo gasto computacionalmente se deve à resolução dessa matriz, mas com isso se tem a probabilidade do sistema ser encontrado em todos os estados possíveis, não apenas a produtividade como no método por simulação.

Para a modelagem analítica, a produtividade foi calculada através do teorema de Burke, em que a taxa de saída é igual a taxa de entrada, ou seja, tudo que entra no sistema vai ser processado, pois nenhuma peça sai do trilho sem concluir o serviço. Com isso, a produtividade foi encontrada através da soma de todas as probabilidades do sistema estar processando uma peça na estação A multiplicado pela taxa de transição de saída da máquina A, A_{end} .

A Tabela 4.3 mostra uma síntese dos resultados, para facilitar a comparação entre os métodos. Vale ressaltar que o método analítico é um método bem definido e apresenta grande número de informações, facilitando, assim, a observação de cada parte, podendo computar todos indicadores de resultados dos equipamentos do processo, visando determinar a eficiência dos mesmos (SOUZA; MAIA, 2018).

Tabela 4.3 – Síntese dos resultados dos AGVs

Método		Número de estados	Produtividade (peças/min)	Tempo Computacional (s)
Analítico	Markov - Completo	25.920	0,412263	67,9456
	Markov - Sem os eventos instantâneos	1.920	0,416461	0,5988
Simulação	Escalonamento de eventos	25.920	0,387931 a 0,405143	1,1907

4.3 Aplicação em sistemas do tipo cliente - servidor

Sistemas do tipo cliente - servidor são caracterizados de forma geral como sistemas em que há presença de clientes em fila aguardando atendimento. Como existe a espera dos clientes pelo serviço, há também o abandono do mesmo. Através das propriedades da teoria de fila, apresentadas na seção 2.5, juntamente com a modelagem do sistema através da distribuição hiperexponencial, é possível modelar e analisar o desempenho de sistemas que possuem abandono.

A taxa de produção do sistema, também considerada como nível de serviço, no método analítico foi calculada de acordo com a probabilidade dos servidores serem encontrados

processando de $1, \dots, m$ clientes como mostrado na equação 4.1.

$$T_p = \pi_{xx1} \times \lambda_d + \pi_{xx2} \times 2\lambda_d + \dots + \pi_{xxm} \times m\lambda_d. \quad (4.1)$$

Ressaltando que são analisadas todas as probabilidades do sistema ser encontrado nos estados de $1, \dots, m$ clientes em atendimento. Para o método de simulação, a taxa de produção foi calculada através da quantidade de ocorrência dos eventos d , dividido pelo tempo total de simulação.

Para utilizar as equações da teoria de filas foi necessário encontrar a taxa de chegada efetiva ($\hat{\lambda}$) dos clientes no sistema, ou seja, a quantidade de clientes que realmente permanecem no sistema aguardando até o atendimento. De acordo com o teorema de Burke, seção 2.5.2.1, a taxa de saída é igual à taxa de entrada, ou seja, a taxa de produção define a quantidade de clientes que aguardam o atendimento, portanto tem-se que:

$$\hat{\lambda} = T_p. \quad (4.2)$$

Então, a taxa de abandono do sistema pode ser calculada como:

$$T_{abandono} = \lambda - T_p. \quad (4.3)$$

De posse da taxa de chegada efetiva no sistema é possível calcular várias medidas de desempenho, como: tempo total de espera, tempo de espera na fila, número médio de clientes no sistema, etc., utilizando as fórmulas mostradas na Tabela 2.2.

4.3.1 Modelagem modular do problema por autômatos

A modelagem para os sistemas cliente - servidor foi baseada em três autômatos: na chegada de dois tipos de clientes, os pacientes e os impacientes, e o atendimento dos servidores.

Para a modelagem da impaciência dos clientes, considerando a definições apresentadas na seção 2.6, foi modelada de acordo com o *balking* do tipo I, em que os clientes observam o comprimento da fila e abandonam ou não de acordo com um comprimento "N".

Para a impaciência do tipo *reneging* foi considerada a regra do tipo IV, em que os clientes decidem esperar ou não de acordo com o tempo de espera T que segue uma distribuição exponencial com parâmetro i , em que i é o número de clientes no sistema.

A Figura 4.13 mostra o comportamento da chegada do cliente do tipo 1, a chegada desse tipo de cliente é de acordo com uma probabilidade p multiplicado pela taxa de chegada λ , representado pelo evento a_1 . A fila tem capacidade l e o decremento da fila se dá pela admissão do cliente pelo servidor, através do evento c_1 ou pelo abandono do cliente por meio do evento b_1 , uma taxa $i\gamma_1$, em que i representa o número de clientes na fila.

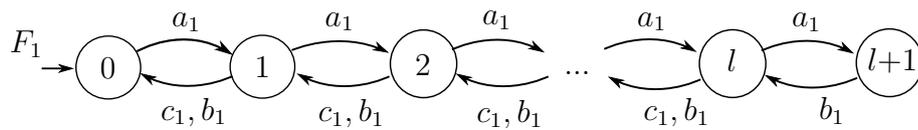


Figura 4.13 – Cliente tipo 1

Assim como os clientes do tipo 1, os clientes do tipo 2 chegam ao sistema mediante o evento a_2 com uma probabilidade de $(1 - p)$ multiplicado pela taxa de chegada λ . A fila dos clientes do tipo 2 tem capacidade n e o decremento da fila, assim como para os clientes do tipo 1, acontece com admissão do cliente pelo servidor através do evento c_2 ou pelo abandono do cliente por meio do evento b_2 por uma taxa $i\gamma_2$. A Figura 4.14 mostra o autômato que representa o comportamento da fila do cliente do tipo 2.

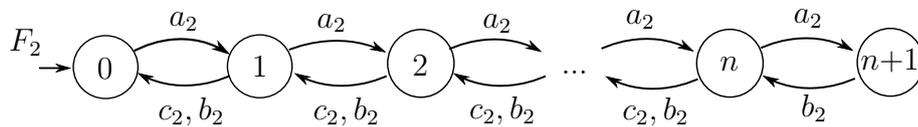


Figura 4.14 – Cliente tipo 2

A diferença do comportamento das filas dos clientes do tipo 1 e 2 está na taxa de ocorrência do abandono e, também na probabilidade de chegada de cada tipo de cliente, de acordo com o valor p . Observando as Figuras 4.13 e 4.14, pode-se perceber que a partir dos estados l e n os clientes já não aguardam atendimento e abandonam o sistema com os eventos b_1 e b_2 , seguindo a regra *balking*. Também é possível verificar a regra *reneging* em que as taxas de abandono γ aumentam de acordo com a quantidade de clientes no sistema.

A Figura 4.15 ilustra o comportamento dos servidores, que através dos eventos c_1 ou c_2 admitem um cliente para atendimento, e finalizam o serviço com o evento d . O tempo de vida do evento d é calculado de acordo com a quantidade de clientes em atendimento com o limite máximo de s servidores.

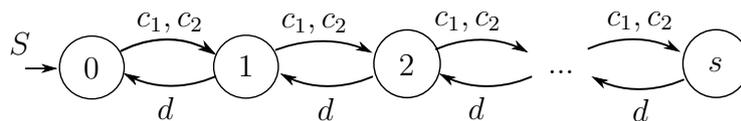


Figura 4.15 – Servidores

A equação 4.4 representa o autômato geral obtido para o comportamento do sistema completo. Portanto, através desses três autômatos é possível realizar o estudo para a

impaciência dos clientes em sistemas que possuem fila de espera e com s servidores.

$$G = F1||F2||S. \quad (4.4)$$

O número máximo de estados resultantes no autômato G é calculado de acordo com a seguinte expressão:

$$N_{estados} = (l + 2) \times (n + 2) \times (s + 1), \quad (4.5)$$

em que l e n , na equação 4.5, representam a capacidade das filas dos clientes do tipo 1 e tipo 2 respectivamente, somado a dois estados, que representam o estado inicial e o estado de abandono de acordo com a regra *balking*. E s é o número de servidores disponíveis para atendimento somado ao estado inicial.

4.3.2 Resultados

As simulações tiveram como objetivo encontrar a probabilidade do sistema estar ocioso, a taxa de produção do sistema e o tempo médio de sistema. Os resultados encontrados foram simulados com os parâmetros $l = 4$, $n = 6$, $m = 3$, $p = 0,1$, $\gamma_1 = 2$ e $\gamma_2 = 1$ para uma fila $M/M/3$. Em que l é a capacidade máxima da fila dos clientes do tipo 1, n é a capacidade máxima da fila dos clientes do tipo 2, m é o número de servidores, p é a probabilidade de chegar um determinado tipo de cliente, γ_1 e γ_2 são as taxas de abandonos dos clientes tipo 1 e tipo 2 respectivamente. Os valores dos parâmetros foram retirados do artigo do Roubos e Jouini (2013).

A intensidade de tráfego dos clientes, como descrito na Tabela 2.3, para esse sistema é de:

$$\rho = \frac{\hat{\lambda}}{m\mu} = \frac{1,774025}{3 \times 1} = 0,5913. \quad (4.6)$$

De posse da intensidade de tráfego estritamente menor que 1, foi possível calcular as medidas de desempenho para a análise do desempenho do sistema cliente - servidor.

Tabela 4.4 – Síntese dos resultados Cliente - Servidor

Resultados	Método	
	Analítico	Simulação
Probabilidade do sistema vazio	13,6035 %	13,5884 % a 14,1326 %
Taxa de produção	1,774025	1,771889 a 1,776238
Tempo médio de sistema	1,2527	1,2555
Tempo médio de fila	0,2527	0,2555
Tempo Computacional (s)	0,0237	4,852544
Número de estados	192	192

O sistema completo, encontrado por meio da equação 4.4, que representa a composição paralela dos três autômatos, resultou em um sistema de 192 estados. Como pode ser visto na Tabela 4.4, foi analisada a probabilidade do sistema ser encontrado vazio, a taxa de produção e o tempo médio de fila e de sistema.

O custo computacional por ambos os métodos foi satisfatório. Mas como previsto o método analítico teve um custo menor que o método por simulação. Os intervalos de 95% de confiança da simulação foram obtidos por meio de simulações de 10.000 segundos, com tempo de aquecimento de 1.000 segundos, em 10 rodadas. Juntamente com os resultados da Tabela 4.4 foi observado o comportamento do sistema ao variar alguns parâmetros.

No gráfico 4.16 é possível observar que com o aumento da taxa d , ou seja, um atendimento mais rápido, a taxa de produção aumenta e diminui o número de abandonos, assim aumentando o nível de serviço.

A Figura 4.17 mostra o tempo médio de sistema em função do desempenho da taxa de chegada, em que pode-se observar que na medida que aumenta a taxa de chegada há um maior tempo de sistema.

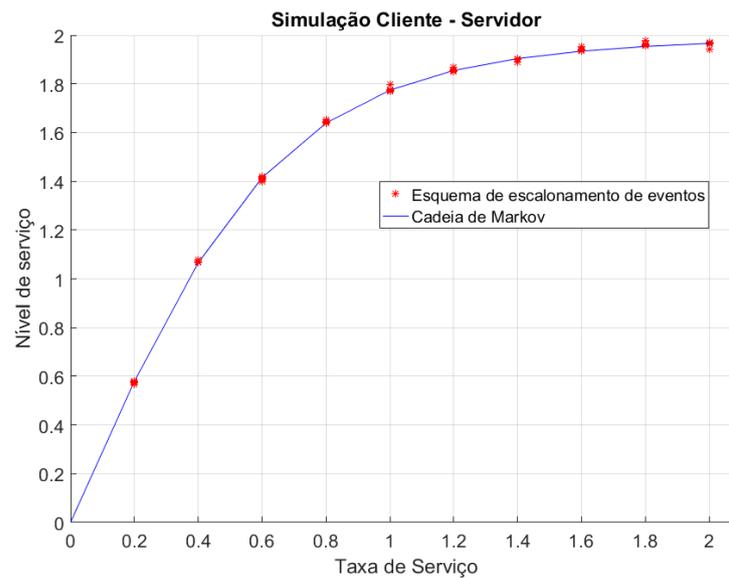


Figura 4.16 – Desempenho em função da taxa de serviço

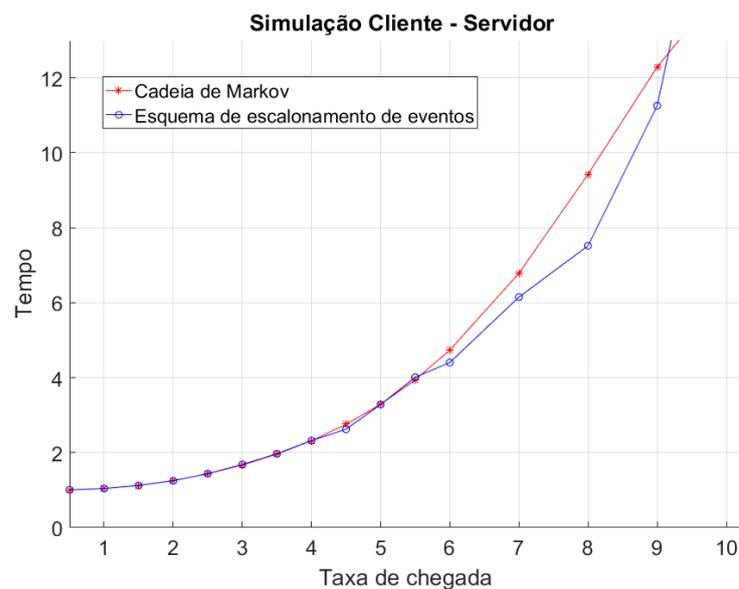
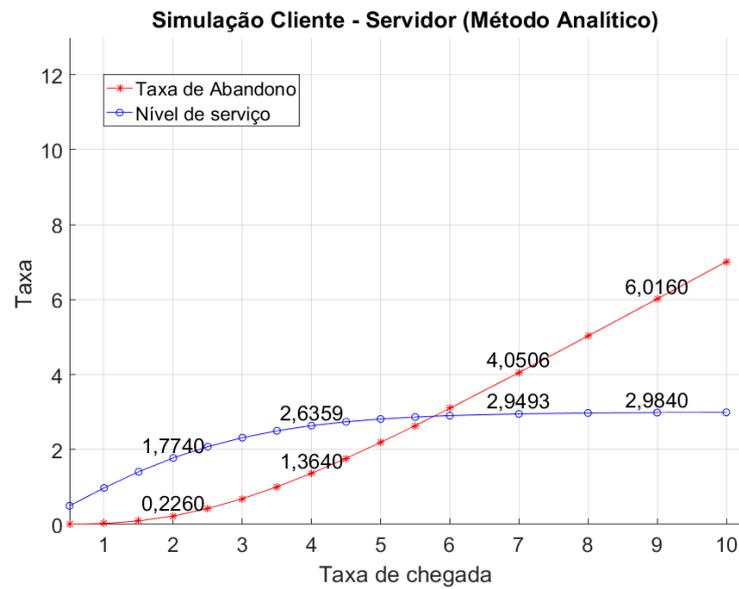


Figura 4.17 – Desempenho do sistema em função da taxa de chegada

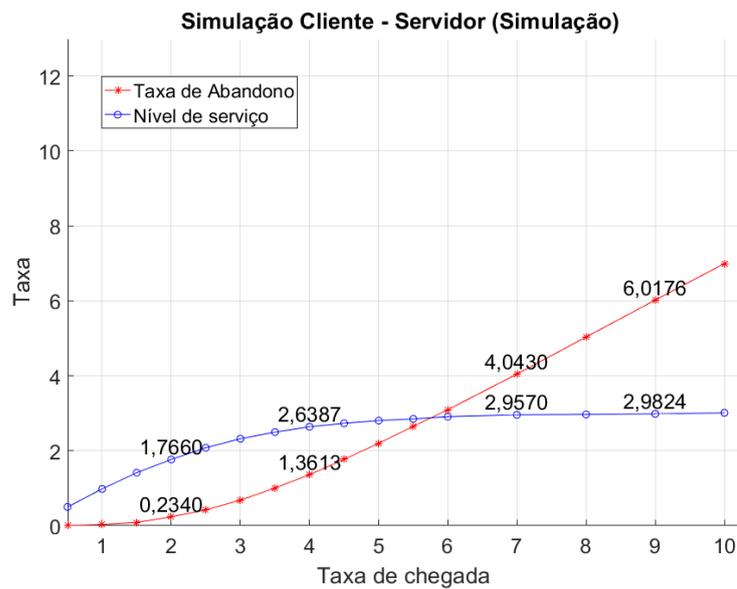
Também foi alterada a taxa de chegada para análise da eficiência do sistema com três servidores. Através do gráfico da Figura 4.18 pode-se verificar que o sistema atende com nível de serviço satisfatório até uma taxa de chegada de $\lambda = 3$, em ambas as metodologias.

Outro parâmetro que pode ser observado ao variar a taxa de chegada, nos gráficos da Figura 4.18, é a taxa de abandono nas duas metodologias, em que pode-se observar um grande aumento na taxa de abandono quando há uma maior taxa de chegada.

Os gráficos da Figura 4.18 são bem semelhantes devido ao fato dos resultados diferirem nas casas decimais. A Tabela 4.5 mostra alguns resultados encontrados para o nível de serviço e a taxa de abandono para ambas metodologias.



a) Método Analítico



b) Método via simulação

Figura 4.18 – Desempenho em função da taxa de chegada

Tabela 4.5 – Alguns resultados do desempenho em função da taxa de serviço

λ	Método analítico		Método via Simulação	
	Nível de serviço	Taxa de Abandono	Nível de serviço	Taxa de Abandono
0,5	0,4974	0,0026	0,4990	0,0010
1,5	1,4055	0,0945	1,4126	0,0874
3,0	2,3143	0,6857	2,3194	0,6806
5,0	2,8120	2,1879	2,8033	2,1967
10,0	2,9900	7,0099	3,0090	6,9910

Conclusões

Ao finalizar esse trabalho pode-se concluir que a metodologia elaborada foi satisfatória e ambos os métodos foram eficientes para a análise de desempenho dos sistemas aplicados.

Destaca-se a eficiência da modelagem por meio de autômatos estocásticos modulares, pois foi possível representar sistemas muito complexos, com milhares de estados, facilitando e reduzindo os erros na etapa de modelagem e também possibilitando a simulação computacional e a obtenção da cadeia de Markov.

Outra importância foi a utilização da biblioteca UltraDES (LACSED (2016)), pois com o auxílio da biblioteca foi possível realizar a composição paralela dos diversos autômatos para a obtenção do comportamento do sistema como um todo; e também adicionar a cada um dos estados resultante da composição paralela um índice, o que facilitou a construção das matrizes de probabilidade de transição, utilizada no método da cadeia de Markov. E também na matriz de transição necessária para o método de escalonamento de eventos.

Também foi possível ressaltar a importância da eliminação dos eventos instantâneos para redução do número de estados da Cadeia de Markov fazendo que assim houvesse uma redução do custo computacional. Evidenciando que essa eliminação é válida apenas para sistemas fechados, em que não há abandonos, em que tudo que chega ao sistema só deixa-o ao receber todos os serviços a ele preestabelecidos.

Pode-se enfatizar que ambos os métodos foram eficientes e satisfatórios, mas o método analítico apresenta uma maior eficácia por ter um menor custo computacional e uma maior quantidade de informações, além de convergir a resultados bem definidos. A cadeia de Markov retorna todas as probabilidades de encontrar os sistemas em todos os estados possíveis facilitando, assim, a observação de cada parte do sistema de forma individual o que facilita a obtenção de melhorias e detecção de possíveis falhas e gargalos contribuindo para um melhor funcionamento e produtividade do sistema.

A metodologia por escalonamento de eventos, também se mostrou eficiente quanto ao custo computacional, mesmo em sistemas com grande quantidade de estados, mas devido à sua necessidade de seguir uma sequência para evolução do sistema, não foi possível a retirada dos eventos instantâneos. Outra importância para o método foi a utilização da

propriedade da projeção inversa nos autômatos, devido à grande quantidade de eventos, o processo de verificar qual autômato reconhecia o evento a ser disparado teria um grande custo computacional.

Destaca-se também que no método via simulação não é necessário obter o autômato global, que representa o comportamento do sistema como um todo, o que facilita a representação de sistemas grandes através da modelagem modular via autômatos estocásticos. Outro aspecto é que por se tratar de um método de simulação seus resultados são baseados em medidas estatísticas, seguindo intervalos de confiança.

Concluindo então que o objetivo do trabalho foi alcançado e foi possível elaborar uma metodologia completa desde a modelagem até a parte da análise de desempenho dos sistemas. Portanto, foi possível modelar o comportamento de sistemas de forma modular por meio dos autômatos estocásticos e, através da biblioteca *UltraDES*, foi plausível a operação da composição paralela para a obtenção da cadeia de Markov em conjunto com as matrizes de transição dos autômatos. Por fim, a análise de desempenho dos parâmetros de interesse foi realizada com sucesso por meio das propriedades dos sistemas juntamente com a teoria de filas.

5.1 Trabalhos Futuros

Algumas ideias e discussões foram iniciadas e podem ser realizadas em pesquisas futuras, dentre elas:

- ❑ Desenvolver um programa para a utilização conjunta do *UltraDES* e *Matlab*, para facilitar o uso da metodologia em que será mais prático a utilização da mesma;
- ❑ Utilizar distribuições de probabilidades gerais, tendo como base a distribuição exponencial, para que outras distribuições possam ser estudadas e analisadas através das propriedades da Cadeia de Markov;
- ❑ Desenvolver a modelagem em redes de Petri estocásticas, para comparar a eficiência da mesma com a modelagem via autômatos estocásticos;
- ❑ Aplicar a metodologia em um problema real, para validar os resultados estimados na análise de desempenho via simulação computacional e método analítico.

Referências

- ABOU-EL-ATA, M.; HARIRI, A. The M/M/c/N queue with balking and reneging. **Computers e Operations Research**, v. 19, p. 713 – 716, 1992.
- AL-SEEDY, R. et al. Transient solution of the M/M/c queue with balking and reneging. **Computers and Mathematics with Applications**, v. 57, p. 1280 – 1285, 2009.
- BANKS, J.; CARSON, J. **Discrete-Event Systems Simulation**. University of Illinois, Urbana-Champaign: Prentice Hall, 2005. v. 4.
- BRENNER, L.; FERNANDES, P.; SALES, A. Redes de autômatos estocásticos: Um formalismo para avaliação de desempenho e confiabilidade de sistemas. FACULDADE DE INFORMÁTICA PUCRS - Brasil. 2002.
- CASSANDRAS, C. G.; LAFORTUNE, S. **Introduction to Discrete Event Systems**. Second. USA: Springer Science Business Media, 2008. v. 11.
- CURY, J. E. R. Teoria de controle supervisorio de sistemas a eventos discretos. V Simpósio Brasileiro de Automação Inteligente. 2001.
- GROSS, D. et al. **Fundamentals of Queueing Theory**. 4. ed. [S.l.]: Wiley-Interscience, 2008. (Wiley Series in Probability and Statistics). ISBN 047179127X,9780471791270.
- HAYES, B. First links in the Markov chain. v. 101, n. 2, p. 92, 2013. American Scientist.
- KLEINROCK, L. **Queueing Systems: Theory**. 1. ed. [S.l.]: Wiley-Interscience, 1975. Volume 1. ISBN 0471491101,9780471491101.
- LACSED. **UltraDES**. 2016. www.github.com/lacsed/UltraDES. Desenvolvido pelo Laboratório de Análise e Controle de Sistemas a Eventos Discretos - UFMG (Accessed: 2018-02-20).
- MEDINA, A. C.; CHWIF, L. **Modelagem e Simulação de Eventos Discretos: Teoria e Aplicações**. 3. ed. São Paulo: [s.n.], 2010.
- PEGDEN, C. e. a. **Introduction to simulation using SIMAN**. Second. New York, NY, USA: McGraw-Hill, Inc., 1990.
- RIBEIRO, R. G. **Performance Evaluation of Stochastic DES Through Analytical Models and Simulation: An Open-Pit Mine Study**. Tese (Doutorado) — Universidade Federal de Minas Gerais (UFMG), 2018.

ROUBOS, A.; JOUINI, O. Call centers with hyperexponential patience modeling. p. 307 – 315, 2013.

SANTOS, T. C. W. **Estudo Comparativo de Técnicas de Modelagem em Redes de Autômatos Estocásticos**. Dissertação (Mestrado) — PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL, 2002.

SOUZA, L. M. G.; MAIA, C. A. Análise de desempenho de Sistemas a Eventos Discretos via autômatos estocásticos modulares. p. 1 – 8, 2018. ISSN 2525-8311. XXII Congresso Brasileiro de Automática.

WANG, K.; LI, N.; JIANG, Z. Queueing system with impatient customers: A review. Proceedings of 2010 IEEE International Conference on Service Operations and Logistics, and Informatics, SOLI 2010, p. 82 – 87, 2010.

WANG, Z.; LI, S. Some application of Markov chain to market occupation rate and promotion strategy. p. 467 – 470, April 2011. ISSN 2160-7443. IEEE Fourth International Conference on Information and Computing.

WHITT, W. Engineering solution of a basic call-center model. Department of Industrial Engineering and Operations Research - Columbia University, New York. 2004.

YUE, D.; W., Y.; ZHAO, G. Analysis of an M/M/c queueing system with impatient customers and synchronous vacations. **Journal of Applied Mathematics**, v. 2014, p. 1 – 12, 2014. Department of Intelligence and Informatics, Konan University.