



UNIVERSIDADE FEDERAL DE GOIÁS – REGIONAL CATALÃO
UNIDADE ACADÊMICA ESPECIAL DE MATEMÁTICA E TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM MODELAGEM E OTIMIZAÇÃO



Christiano Michel Fernandes Freitas

DIMENSIONAMENTO DE EQUIPES DE TRABALHO POR MEIO DE MODELOS PROBABILÍSTICOS

DISSERTAÇÃO DE MESTRADO

CATALÃO – GO, 2018

**TERMO DE CIÊNCIA E DE AUTORIZAÇÃO PARA DISPONIBILIZAR
VERSÕES ELETRÔNICAS DE TESES E DISSERTAÇÕES
NA BIBLIOTECA DIGITAL DA UFG**

Na qualidade de titular dos direitos de autor, autorizo a Universidade Federal de Goiás (UFG) a disponibilizar, gratuitamente, por meio da Biblioteca Digital de Teses e Dissertações (BDTD/UFG), regulamentada pela Resolução CEPEC nº 832/2007, sem ressarcimento dos direitos autorais, de acordo com a Lei nº 9610/98, o documento conforme permissões assinaladas abaixo, para fins de leitura, impressão e/ou *download*, a título de divulgação da produção científica brasileira, a partir desta data.

1. Identificação do material bibliográfico: ☒ **Dissertação** ☐ **Tese**

2. Identificação da Tese ou Dissertação:

Nome completo do autor: Christiano Michel Fernandes Freitas

Título do trabalho: Dimensionamento De Equipes De Trabalho Por Meio De Modelos Probabilísticos

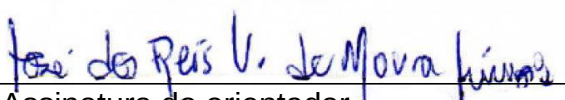
3. Informações de acesso ao documento:

Concorda com a liberação total do documento ☒ **SIM** ☐ **NÃO**

Havendo concordância com a disponibilização eletrônica, torna-se imprescindível o envio do(s) arquivo(s) em formato digital PDF da tese ou dissertação.


Assinatura do(a) autor(a)

Ciente e de acordo:


Assinatura do orientador

Data: 29 de Junho de 2018

CHRISTIANO MICHEL FERNANDES FREITAS

DIMENSIONAMENTO DE EQUIPES DE TRABALHO POR
MEIO DE MODELOS PROBABILÍSTICOS

Dissertação apresentada como requisito parcial para a obtenção do título de Mestre em Modelagem e Otimização pela Universidade Federal de Goiás – Regional Catalão.

Orientador:
José dos Reis Vieira de Moura Júnior

CATALÃO – GO

2018

Ficha de identificação da obra elaborada pelo autor, através do
Programa de Geração Automática do Sistema de Bibliotecas da UFG.

Freitas, Christiano

DIMENSIONAMENTO DE EQUIPES DE TRABALHO POR MEIO
DE MODELOS PROBABILÍSTICOS [manuscrito] / Christiano Freitas.
2018.

CXV, 125 f.: il.

Orientador: Prof. Dr. José dos Reis Vieira de Moura Júnior.

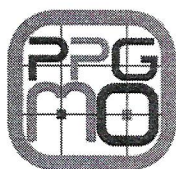
Dissertação (Mestrado) - Universidade Federal de Goiás, , Catalão,
Programa de Pós-Graduação em Modelagem e Otimização, Catalão, 2018.

Bibliografia. Anexos. Apêndice.

Inclui gráfico, tabelas, algoritmos, lista de figuras, lista de tabelas.

1. Sistemas de produção. 2. Processo de quase nascimento e
morte. 3. Demanda. 4. Estratégias organizacionais. I. Vieira de Moura
Júnior, José dos Reis , orient. II. Título.

CDU 517

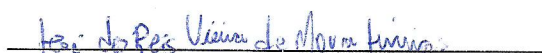


Defesa Nº 35

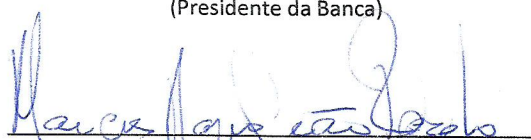
Ata de Defesa Pública – Dissertação de Mestrado

Aos 18 dias do mês de maio do ano de 2018, às 09h:00 min, reuniram-se os componentes da banca examinadora, professores(as) Dr. José dos Reis Vieira de Moura Júnior (presidente e Orientador), Dr. Júlio Cesar Valandro Soares e Dr. Marcos Napoleão Rabelo para, em sessão pública realizada na Sala de Reuniões (Centro Integrado de Pesquisa), da Regional Catalão (RC), da Universidade Federal de Goiás (UFG), procederem com a avaliação do trabalho intitulado: "Dimensionamento de Equipes de Trabalho por Meio de Modelos Probabilísticos", em nível de Mestrado, área de concentração *Modelagem e Otimização*, de autoria de **Christiano Michel Fernandes Freitas**, discente do Programa de Pós-Graduação em Modelagem e Otimização (PPGMO) da UFG/RC. A sessão foi aberta pelo presidente da banca, que fez a apresentação formal dos membros da banca. A seguir, a palavra foi concedida ao discente que, dentro do tempo regulamentar, procedeu a apresentação de seu trabalho. Terminada a apresentação, cada membro da banca arguiu o candidato, tendo-se adotado o sistema de diálogo sequencial. Terminada a fase de arguição, procedeu-se a avaliação do trabalho. Os membros da banca consideraram o trabalho final: (☒) **Aprovado** ou (☐) **Reprovado**. Cumpridas as formalidades de pauta, às 09 h: 00 min a presidência da mesa encerrou a sessão e para constar, eu José dos Reis Vieira de Moura Júnior, lavrei a presente Ata que, depois de lida e aprovada, segue assinada pelos membros da banca examinadora e pelo discente e, posteriormente, será homologada pelo Colegiado do PPGMO.

Catalão-GO, 18 de maio de 2018.


Prof. Dr.: José dos Reis Vieira de Moura Júnior
Programa de Pós-Graduação em Modelagem e
Otimização, UFG/RC.
(Presidente da Banca)


Prof. Dr.: Júlio Cesar Valandro Soares
Universidade Federal de Goiás


Prof. Dr.: Marcos Napoleão Rabelo
Universidade Federal de Goiás


Discente: Christiano Michel Fernandes Freitas
Programa de Pós-Graduação em Modelagem e
Otimização, UFG/RC.

Enfim, mais um ciclo se encerra. Termino este mestrado com duas convicções: a primeira é que depois de dois anos estudando, tenho menos conhecimento do que antes, e a segunda, é a de que ter um título não se trata de ser melhor que alguém, mas sim, de superar seus próprios limites. Diante disto, gostaria de dedicar essa dissertação primeiramente à minha Mãe Nilva (em memória), que por força maior não conseguiu terminar seu mestrado, e que mesmo não estando presente nesse momento, sem dúvidas, foi e sempre será meu maior exemplo na vida acadêmica e pessoal. Também dedico ao meu avô Otogamis, por ser a pessoa que mais torceu por mim e a minha vó Zica (em memória), por ter sido a melhor vó que alguém pode ter.

Agradecimentos

Agradeço primeiramente à minha família, em específico, ao meu Pai Lindolfo e as minhas irmãs Christielly e Camila, por toda a alicerce, cumplicidade e apoio.

Agradeço aos meus amigos de república Jonathan, Conrado e Glauber, pela parceria e amizade, e também, aos meus amigos da confraria, sobretudo, pelos exemplos e inspirações compartilhadas.

Agradeço aos colegas de mestrado por todo conhecimento compartilhado e pelas madrugadas nas quais pelejamos juntos para cumprir nossas atividades.

Agradeço aos parceiros de pesquisa, Bruno e Stanley, pela ajuda nos códigos e programações.

Agradeço às minhas tias Luciene e Ana Maria pelo apoio e torcida ao longo destes dois anos.

Agradeço ao professor Marcos Napoleão por ter acompanhado e colaborado com meu trabalho desde o início.

E por fim, agradeço ao meu orientador José Reis (Zé), primeiramente, pelo exemplo de pessoa que você é, onde, mesmo passando por mementos difíceis, nunca mudou a sua forma de tratar seus orientandos, sendo sempre um orientador disponível e de bom humor, além de disso, obrigado por sempre exigir o meu máximo ao longo do trabalho.

"E eu certamente não me converti em estátua nem me encontro postado como uma coluna, rígido, petrificado; gosto da carreira veloz. E embora haja na terra atoleiros e denso nevoeiro, aquele que tem os pés leve corre e dança por cima da lama como sobre gelo liso."

(Friedrich Nietzsche)

RESUMO

FREITAS, C. M. F. . *DIMENSIONAMENTO DE EQUIPES DE TRABALHO POR MEIO DE MODELOS PROBABILÍSTICOS*. 2018. 123 f. Dissertação (Mestrado em Modelagem e Otimização) – Unidade Acadêmica Especial de Matemática e Tecnologia, Universidade Federal de Goiás – Regional Catalão, Catalão – GO.

O presente trabalho tem como objetivos: modelar um sistema de produção com três unidades fabris, de modo a permitir o dimensionamento ideal de mantenedores, e a realização de uma análise de sensibilidade para avaliar a confiabilidade dos resultados obtidos pelo mesmo modelo. Um processo de quase nascimento e morte (*Quasi-Birth-and-Death - QBD*) é utilizado para modelar as unidades produtivas, e por meio dos geradores infinitesimais, são obtidas as probabilidades de entrada para o código desenvolvido. Geralmente as organizações definem suas equipes de mantenedores de forma empírica, fato que pode comprometer as estratégias organizacionais. Sendo assim, o código oferece auxílio na tomada de decisões destes profissionais. Deste modo, foram modeladas três unidades de produção X, Y e Z e realizado o dimensionamento mínimo de mantenedores que cada unidade deve ter. Observou-se que a unidade X com no mínimo dois mantenedores proporciona 70% de probabilidade de permanecer em funcionamento, a unidade Y com três, proporciona 76%, e a unidade Z, com um mantenedor, possibilita 80%. Por meio da análise de sensibilidade, notou-se que ao perturbar o gerador infinitesimal os valores de probabilidade de funcionamento tendem a se aproximar de 100% à medida que se acrescenta um mantenedor, no entanto, quando se acrescenta o quarto mantenedor, existe pouca variação no sistema. Já em relação ao tempo, quando se estressa o sistema por meio do crescimento da variável aleatória t , a confiabilidade dos resultados tende a diminuir, sendo que com um mantenedor, a probabilidade de funcionamento cai consideravelmente ao longo do tempo, e em contrapartida, com quatro mantenedores, a permanência de estado de funcionamento tende a ser duradoura.

Palavras-chaves: Sistemas de produção, Processo de quase nascimento e morte, Demanda, Estratégias organizacionais.

ABSTRACT

FREITAS, C. M. F. . *Monography Template. Master in Modeling and Optimization*. 2018. 123 f. Master Thesis in Modelling and Optimization – Unidade Acadêmica Especial de Matemática e Tecnologia, Universidade Federal de Goiás – Regional Catalão, Catalão – GO.

This work proposes the modeling of a production system with three manufacturing units, in order to allow the optimal dimensioning of maintainers and the accomplishment of a sensitivity analysis that allows to evaluate the reliability of the obtained results. A Quasi-Birth-and-Death (QBD) process is used to model the productive units, and through infinitesimal generators, the input probabilities for the developed code are obtained. Organizations usually define their supporter teams empirically, which can compromise organizational strategies. Thus, the code offers assistance in the decision making of these professionals. Thus, three production units X, Y and Z were modeled and the minimum dimensioning of maintainers that each unit had to be performed. Thus, the X unit with two maintainers provides a 70% probability of remaining in operation, the Y unit with three provides 76%, and finally, the Y unit with only one maintainer allows an 80% chance of remaining in operation. By means of the sensitivity analysis, it was noticed that when disturbing the infinitesimal generator the values of probability of operation tend to approximate to 100% whereas a maintainer is added, however, when the fourth maintainer is added, there is little variation in the system. However, when the system is stressed by the growth of the random variable t , the reliability of the results tends to decrease, whereas with a maintainer, the probability of functioning falls considerably over time, and in contrast, with four maintainers, the permanence of operating state tends to be distant.

Keywords: Production systems, Quasi-Birth-and-Death process, Demand, Organizational strategies.

LISTA DE FIGURAS

Figura 2.1 – Elementos de uma fila.	28
Figura 2.2 – Localização das variáveis.	30
Figura 2.3 – Relação Poisson e exponencial.	32
Figura 2.4 – Linha de produção de motocicletas.	33
Figura 2.5 – Diagrama de transição.	35
Figura 2.6 – Diagrama de transição em um sistema com capacidade limitada.	36
Figura 2.7 – Sistema de produção.	41
Figura 2.8 – Para onde despachar?	44
Figura 2.9 – Considerar "roubo" de ciclo? Quando?	45
Figura 2.10 – Considerar "roubo" de ciclo? Quando?	45
Figura 2.11 – Rede de filas aberta.	46
Figura 2.12 – Rede de filas fechada.	46
Figura 2.13 – Capacidade ilimitada e capacidade limitada.	47
Figura 3.1 – Células de produção.	51
Figura 3.2 – Vetores de probabilidade.	54
Figura 3.3 – Modelagem sistema FIFO.	54
Figura 3.4 – Dimensões infinitas: Roubo de ciclo.	55
Figura 3.5 – Cadeia de Markov 1D.	56
Figura 3.6 – Estados agrupados do processo <i>QBD</i>	57
Figura 3.7 – Estrutura gerador infinitesimal.	58
Figura 3.8 – Máquina com peças reparáveis.	59
Figura 3.9 – Representação do processo <i>QBD</i>	59
Figura 4.1 – Fábrica de produtos alimentícios.	69
Figura 4.2 – Matriz de transição.	72
Figura 4.3 – Número mínimo de mantenedores - Unidade X.	72
Figura 4.4 – Número mínimo de mantenedores - Unidade Y.	73
Figura 4.5 – Número mínimo de mantenedores - Unidade Z.	73
Figura 4.6 – Probabilidade de funcionamento E_0	76
Figura 4.7 – Possibilidade de permanecer em funcionamento com mantenedores.	77

Figura 4.8 – Médias das probabilidades E_n .	78
Figura 4.9 – Variando t de forma crescente.	78
Figura A.1 – Gerador infinitesimal Q - Página 1	88
Figura A.2 – Gerador infinitesimal Q - Página 2	89
Figura A.3 – Gerador infinitesimal Q - Página 3	90
Figura A.4 – Gerador infinitesimal Q - Página 4	91
Figura A.5 – Gerador infinitesimal Q - Página 5	92
Figura A.6 – Gerador infinitesimal Q - Página 6	93
Figura B.1 – Gerador infinitesimal da unidade X - Página 1	96
Figura B.2 – Gerador infinitesimal da unidade X - Página 2	97
Figura B.3 – Gerador infinitesimal da unidade X - Página 3	98
Figura B.4 – Gerador infinitesimal da unidade X - Página 4	99
Figura B.5 – Gerador infinitesimal da unidade X - Página 5	100
Figura B.6 – Gerador infinitesimal da unidade X - Página 6	101
Figura C.1 – Gerador infinitesimal da unidade Y - Página 1	104
Figura C.2 – Gerador infinitesimal da unidade Y - Página 2	105
Figura C.3 – Gerador infinitesimal da unidade Y - Página 3	106
Figura C.4 – Gerador infinitesimal da unidade Y - Página 4	107
Figura C.5 – Gerador infinitesimal da unidade Y - Página 5	108
Figura C.6 – Gerador infinitesimal da unidade Y - Página 6	109
Figura D.1 – Gerador infinitesimal da unidade Z - Página 1	112
Figura D.2 – Gerador infinitesimal da unidade Z - Página 2	113
Figura D.3 – Gerador infinitesimal da unidade Z - Página 3	114
Figura D.4 – Gerador infinitesimal da unidade Z - Página 4	115
Figura D.5 – Gerador infinitesimal da unidade Z - Página 5	116
Figura D.6 – Gerador infinitesimal da unidade Z - Página 6	117

LISTA DE TABELAS

Tabela 2.1 – Resumo de relações entre a distribuição de Poisson e a Exponencial – Pesquisa Operacional.	33
Tabela 2.2 – Parâmetros e expressões analíticas M/M/1 – Modelagem e simulação de eventos discretos.	35
Tabela 2.3 – Parâmetros e expressões analíticas M/M/1/c – Modelagem e simulação de eventos discretos.	37
Tabela 2.4 – Parâmetros e expressões analíticas M/D/1 – Modelagem e simulação de eventos discretos.	38
Tabela 2.5 – Parâmetros e expressões analíticas M/M/s – Modelagem e simulação de eventos discretos.	40
Tabela 3.1 – Defeitos A, B, C.	52
Tabela 3.2 – Probabilidade de transição entre os estados.	52
Tabela 3.3 – Valor de E para que o sistema permaneça em funcionamento.	67

LISTA DE QUADROS

Quadro 2.1 – Variáveis randômicas fundamentais.	31
Quadro 2.2 – Diferenças entre <i>Jobshop</i> e <i>Flowshop</i>	41
Quadro 2.3 – Aplicações de modelagem probabilística.	47

LISTA DE CÓDIGOS

Código 3.1 – Código matriz de transição P	53
Código 3.2 – Código gerador infinitesimal.	64
Código 3.3 – Código de Huang-Ling-Guo.	66
Código 4.1 – Geradores infinitesimais.	71
Código 4.2 – Matriz pseudoaleatória S.	75
Código E.1 – Algoritmo de HUANG-LIANG-GUO - Parte 1.	121
Código E.2 – Algoritmo de HUANG-LIANG-GUO - Parte 2.	122
Código E.3 – Algoritmo de HUANG-LIANG-GUO - Parte 3.	123

SUMÁRIO

1	INTRODUÇÃO	25
1.1	Roteiro	26
2	TEORIA DAS FILAS - ABORDAGEM CLÁSSICA	27
2.1	Conceitos Fundamentais	27
2.2	Elementos de um Modelo de Filas	27
2.3	Variáveis Aleatórias Fundamentais	29
2.4	Fórmula de <i>Little</i>	31
2.4.1	Modelo de nascimento e morte	31
2.5	A notação de Kendall	33
2.6	Modelos tradicionais de filas	34
2.6.1	M/M/1	34
2.6.1.1	Exemplo de aplicação do modelo M/M/1	35
2.6.2	M/M/1/c	36
2.6.2.1	Exemplo de aplicação do modelo M/M/1/c	37
2.6.3	M/D/1	38
2.6.3.1	Exemplo de aplicação do modelo M/D/1	38
2.6.4	M/M/s	39
2.7	Sistemas de produção	40
2.7.1	Sistemas de produção sob a perspectiva das filas de espera	42
2.8	Múltiplos servidores	44
2.8.1	Redes de filas	46
3	MODELOS DE PROBABILIDADE	49
3.1	Modelos Markovianos	50
3.1.1	Exemplo matriz de probabilidade	52
3.2	Cadeia de Markov Multidimensional	54
3.2.1	Modelagem de roubo de ciclo por cadeias de Markov	55
3.3	Quasi-Birth-and-Death Process (QBD)	56
3.3.1	Exemplo QBD: Servidor definido por distribuição exponencial com <i>buffer</i> limitado, considerando avarias e reparação.	59
3.3.2	Exemplo numérico: Algoritmo de HUANG-LIANG-GUO.	61

3.3.3	Gerador infinitesimal	62
3.3.4	Algoritmo de <i>HUANG-LIANG-GUO</i>	65
4	DIMENSIONAMENTO DE EQUIPES DE MANTENEDORES	69
4.1	Análise de Sensibilidade do Algoritmo HUANG-LIANG-GUO	74
5	CONCLUSÕES E TRABALHOS FUTUROS	81
REFERÊNCIAS		83
APÊNDICE A	RESULTADO GERADOR INFINITESIMAL	88
APÊNDICE B	RESULTADO GERADOR X	96
APÊNDICE C	RESULTADO GERADOR Y	104
APÊNDICE D	RESULTADO GERADOR Z	112
APÊNDICE E	APLICAÇÃO DO ALGORITMO DE HUANG-LIANG-GUO . . .	120

Capítulo 1

INTRODUÇÃO

O desempenho de sistemas de produção se tornou uma questão significativa no que se refere à estratégia de atendimento à demanda. O êxito ou insucesso de uma operação fabril estará vinculado aos objetivos preestabelecidos pela organização. Logo, as técnicas para avaliar e prever o desempenho de um sistema produtivo ganham atenção desde os anos de 1900, sendo que tal período foi influenciado pela administração clássica e o crescimento da inovação nos setores de energia, siderurgia, indústria automobilística e outros (MARSUDI, 2010; POLIZELLI; RUIZ JÚNIOR, 2011).

Até os anos de 1970, as técnicas para análise de desempenho de sistemas estavam vinculadas à teoria das filas, sendo que a mesma forneceu uma gama de fórmulas que possibilitaram a avaliação de diversos modelos como o $M/M/1$, $M/D/1$ e $M/M/s$ (OSOGAMI, 2005). Entretanto, quando se analisa sistemas como: (i) flow shop flexível (ZHANG; CHEN; MAO, 2016), (ii) Job shop (SILVA; MORABITO, 2007) e (iii) máquinas com roubo de ciclo (OSOGAMI; HARCHOL-BALTER; SCHELLER-WOLF, 2005) se faz necessária a utilização de modelos de maior complexidade, dentre eles, os modelos de probabilidade.

Dentre os modelos de probabilidade, destacam-se os modelos markovianos, sendo os mesmos modelos estocásticos que representam um conjunto de transições aleatórias, nas quais o estado futuro não depende de todo o histórico percorrido, apenas do estado atual (*memoryless*) (KLEINROCK, 1975; TIER, 2017). Quando aplicados aos sistemas de produção, utiliza-se o termo cadeias de Markov, haja vista que tal cadeia representa as transições aleatórias em um conjunto de estados discretos.

Todavia, vale ressaltar que o uso de modelos de probabilidade não se restringe aos ambientes relacionados à manufatura, sendo que suas aplicações são diversas e podem estar associadas às áreas de comunicação, análise de sistemas de informática, análise física em nano escala com influência da observação e outros (KAMPEN, 2006; GALLAGER, 2013)

No que tange a modelagem de sistemas de produção por cadeias de Markov, acentua-se o processo de quase nascimento e morte (*Quasi-Birth-and-Death* - QBD). O QBD se trata

de uma extensão do processo de nascimento e morte, sendo que além das transições entre estados adjacentes, possibilita transições dentro do mesmo estado (OST, 2001).

Esta transição adicional possibilita maior representatividade de situações reais, como por exemplo, servidores em sequência com buffers finitos (BAUMANN; SANDMANN, 2017), produtos com operações distintas dentro de uma mesma planta (LEEUWAARDEN; WINANDS, 2006) e um modelo que estime o número de peças de reposição reparáveis em uma máquina (SALES; MORALES, 2011). Sendo assim, o processo QBD é conhecido pela sua versatilidade quanto à modelagem de sistemas e quanto à sua capacidade de análise numérica.

Deste modo, o presente trabalho tem como objetivos: modelar um sistema de produção com três unidades fabris, de modo a permitir o dimensionamento ideal de mantenedores, e a análise de sensibilidade para avaliar a confiabilidade dos resultados obtidos pelo mesmo modelo.

1.1 Roteiro

Esta dissertação é dividida em seis etapas, sendo elas:

- (i) Conceitos fundamentais: os conceitos fundamentais abordam os conhecimentos clássicos referentes aos estudos de Teoria das Fila. De forma geral, esta seção propicia os conhecimentos básicos relacionados aos componentes de uma fila e suas respectivas variáveis, assim como, exemplos de modelos clássicos;
- (ii) Da seção 2.6 a 2.8.1: é feita uma transição entre a teoria clássica e os modelos de probabilidade, dada a necessidade de representar sistemas de produção de maior complexidade;
- (iii) Modelos de probabilidade: (i) inicialmente é feita uma reflexão sobre a modelagem matemática de modelos probabilísticos, (ii) são apresentados conceitos fundamentais sobre modelos markovianos, (iii) é apresentado o processo de quase nascimento e morte e seu respectivo método de resolução e por fim (iv) foi desenvolvido um exemplo numérico como base do trabalho proposto.
- (iv) Resolução do problema de dimensionamento de mantenedores das três unidades fabris;
- (v) Análise de sensibilidade;
- (vi) Conclusões.

Capítulo 2

Teoria das Filas - Abordagem Clássica

2.1 Conceitos Fundamentais

As filas surgem em distintas situações, sejam em sistemas de produção ou de serviço. Dentre vários exemplos, pode-se citar: produtos em espera para serem processados em máquinas ou estações de trabalho; pessoas no aguardo para serem atendidas em hospitais; aviões em espera para pousar; navios em espera para descarregar em portos e solicitações de clientes remotos a serem processados em sistemas virtuais através de redes de computadores (ARENALES *et al.*, 2007; TAHA, 2008; PRADO, 2009).

Por meio da utilização de algumas medidas de desempenho, tais quais, o comprimento médio de uma fila, ritmo médio de chegadas e o intervalo médio entre chegadas, o estudo de filas possibilita a quantificação do evento de espera. O primeiro estudo de filas foi desenvolvido por A.K. Erlang em 1908, aplicado ao problema de redimensionamento de centrais telefônicas (ARENALES *et al.*, 2007; TAHA, 2008; PRADO, 2009).

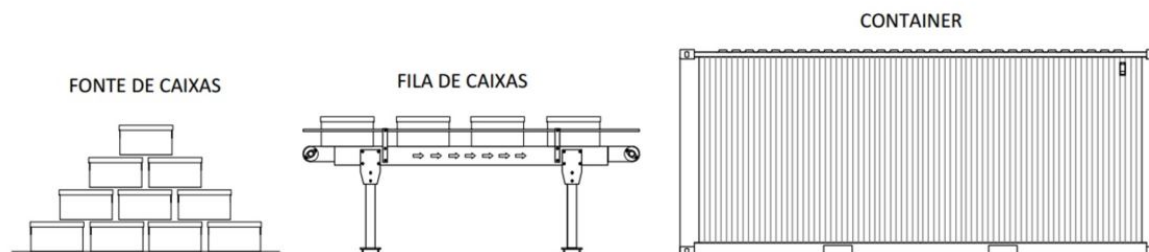
As filas existentes em sistemas modernos são dispendiosas para as organizações pois aumentam estoques, comprometem o capital de giro, diminuem os espaços de trabalho e aumentam o risco da má qualidade do produto ou atendimento. Geralmente elas se formam no momento em que a demanda excede a capacidade do sistema que fornece o serviço, e também, porque a taxa de chegadas de clientes não é constante. Vale salientar que o ideal seria a inexistência de filas, no entanto, esta situação pode se tornar inviável financeiramente (ARENALES *et al.*, 2007; TAHA, 2008; PRADO, 2009; MOREIRA, 2010). Uma alternativa à redução das filas seria o uso de atendimento agendado no caso de serviços ou o uso de Kanban em sistemas de produção.

2.2 Elementos de um Modelo de Filas

Os personagens principais em um sistema de filas são os clientes e os servidores. De acordo com a Figura 2.1, observa-se que as caixas são oriundas de uma população ou fonte,

que ao estarem disponíveis para o serviço de transporte (container), formam uma fila sobre a esteira. Como produto desta ação, tem-se uma situação de espera (ARENALES *et al.*, 2007; TAHA, 2008; PRADO, 2009).

Figura 2.1 – Elementos de uma fila.



Fonte: o autor.

O tamanho de uma população pode ser considerado infinito ou finito. Na população infinita, as chegadas são abundantes e independentes, como por exemplo: telefonemas que chegam em uma central telefônica e as chegadas de clientes à uma estação de metrô. No que se refere as populações finitas, suas chegadas são ditas dependentes e fornecem clientes de maneira limitada. Por exemplo, pode-se considerar, a população de três caminhões que buscam minério em uma mina (TAHA, 2008; PRADO, 2009).

Outro importante aspecto é a dependência. Levando em consideração dois eventos A e B , entende-se que A é dependente ou condicionado a B , caso a ocorrência de B torne conhecida a probabilidade de ocorrência de A (CHWIF; MEDINA, 2015). Por exemplo, considere um espaço amostral S , composto por 4 peças, duas perfeitas e duas defeituosas. Dado que ocorreu a chegada de uma peça defeituosa na estação de trabalho, a probabilidade de chegar outra peça defeituosa é de aproximadamente 33%. Já a probabilidade de chegar uma peça perfeita é maior, sendo de aproximadamente 67%.

O processo de chegada de peças é representado pelo ritmo médio de chegadas (exemplo: 10 peças por minuto) e o intervalo médio entre chegadas (exemplo: 5 segundos entre chegadas). Vale destacar que tanto o ritmo médio de chegadas quanto o intervalo médio entre chegadas são valores médios que representam apenas o comportamento geral do sistema, o que implica considerar variações ao longo do tempo (TAHA, 2008; PRADO, 2009).

Já o processo de atendimento é representado pelo ritmo médio de atendimento (exemplo: 5 peças processadas por minuto) e pelo tempo ou duração média do atendimento ou serviço (exemplo: 15 segundos por peça). Assim como no processo de chegada, o processo de atendimento raramente é regular (TAHA, 2008; PRADO, 2009).

Em um sistema de filas, os clientes ou peças podem ser atendidos ou processados conforme alguns critérios, estes critérios são categorizados de acordo com a disciplina da

fila, entre as principais, destacam-se:

- (i) Primeiro a chegar, primeiro a ser atendido (*FIFO – First In First Out*).
- (ii) Último a chegar, primeiro a ser atendido (*LIFO – Last In First Out*).
- (iii) Serviço em ordem aleatória (*SIRO – Service In Random Order*).
- (iv) Disciplina genérica de atendimento (*GD – Generic discipline*).

Apesar das disciplinas preestabelecidas em algumas situações, podem ocorrer eventos com base em prioridades, como por exemplo: serviços de urgência em uma oficina para um cliente diferenciado (TAHA, 2008; PRADO, 2009; CHWIF; MEDINA, 2015).

Já em relação ao tamanho da fila, frisa-se algumas medidas de desempenho como: tamanho médio da fila; tamanho máximo da fila (exemplo: a quantidade de cadeiras em uma barbearia) e; tempo médio de espera (Somatória dos tempos de atendimento dos clientes à frente). Assim como as populações, as filas também podem ser finitas, como um *buffer* limitado entre processos de produção, ou infinitas, como em serviços de mala direta (TAHA, 2008; PRADO, 2009).

2.3 Variáveis Aleatórias Fundamentais

Dentre as particularidades pertencentes ao estudo de filas, ressalta-se as características de suas variáveis, sendo elas de cunho aleatório. Ao se associar a probabilidade de ocorrência de um evento a um espaço amostral S , assume-se que uma variável aleatória X pode, probabilisticamente, assumir valores distintos. No entanto, por maior que seja a probabilidade de ocorrência do evento, não é possível garantir o acontecimento do mesmo, fato que não impede a formação de um valor médio (TAHA, 2008; DOWNING; CLARK, 2011; CHWIF; MEDINA, 2015).

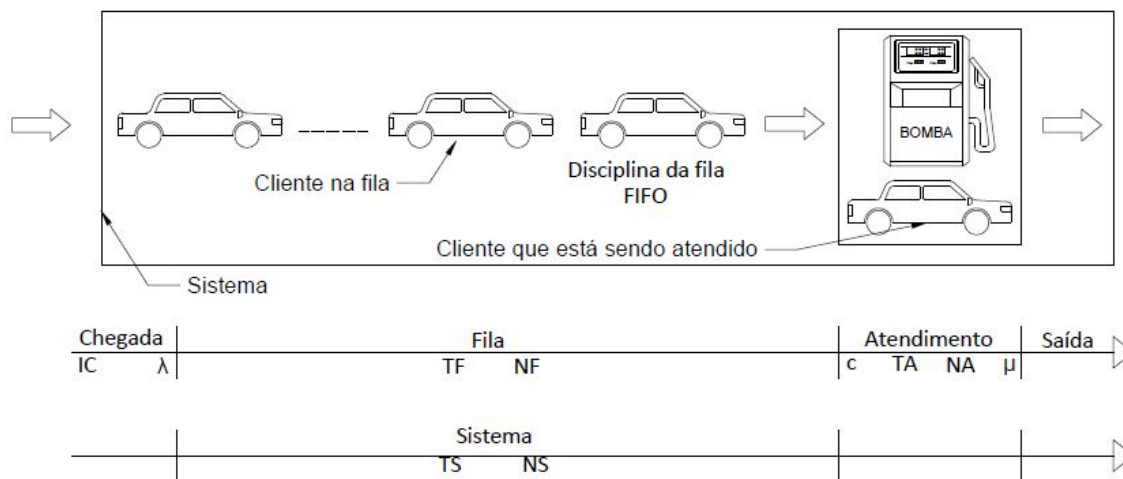
Sendo a média conhecida, os eventos aleatórios formam um comportamento específico em torno da mesma, o que caracteriza uma distribuição de probabilidade de X , que por sua vez, demonstra como suas probabilidades de ocorrência se distribuem em torno do intervalo $[0, 1]$ (PRADO, 2009; DOWNING; CLARK, 2011; CHWIF; MEDINA, 2015).

As variáveis aleatórias podem ser classificadas de duas maneiras: (i) variável aleatória discreta – todos os possíveis valores da variável aleatória X são finitos, ou seja, seus valores esgotam o espaço amostral S e (ii) variável aleatória contínua – a variável aleatória X pode assumir qualquer valor de um intervalo, havendo infinitas possibilidades. De forma geral, a variável aleatória discreta pode ser representada por um histograma, já a variável aleatória contínua, por uma função densidade de probabilidade (MOREIRA, 2010; CHWIF; MEDINA, 2015).

Dada as características das variáveis, existe um conjunto de fórmulas matemáticas que permitem calcular algumas das respostas quistas por um pesquisador ou proprietário de um ambiente constituído por filas. Porém, é necessário que o sistema estudado seja estável, ou seja, que haja estabilidade tanto no fluxo de chegada quanto no processo de atendimento. Caso não haja estabilidade é preciso recorrer à simulação de sistemas. Já que o processo analítico se torna complexo (FREITAS FILHO, 2001; PRADO, 2009)

Para exemplificar a aplicação das variáveis fundamentais, considere a Figura 2.2, onde é possível observar um sistema estável, no qual clientes chegam e entram em fila, havendo um servidor c para atendê-los. O ritmo médio de chegada e o ritmo médio de atendimento são respectivamente representados pelas letras gregas λ e μ . As demais variáveis da Figura 2.2 são definidas no Quadro 2.1.

Figura 2.2 – Localização das variáveis.



Fonte: o autor.

As demais variáveis representadas na Figura 2.2, são ilustradas e definidas no Quadro 2.1.

Variáveis	Definição
Variáveis referentes ao sistema	
TS	Tempo médio de permanência no sistema
NS	Número médio de clientes no sistema
Variáveis referentes ao processo de chegada	
λ	Ritmo médio de chegada
IC	Intervalo médio entre chegadas
Por definição: $IC = \frac{1}{\lambda}$	
Variáveis referentes à fila	
TF	Tempo médio de permanência na fila
NF	Número médio de clientes na fila
Variáveis referentes ao processo de atendimento	
TA	Tempo médio de atendimento ou serviço
c	Capacidade de atendimento ou quantidade de serviço
NA	Número médio de clientes que estão sendo atendidos
μ	Ritmo médio de atendimento de cada atendente
Por definição: $TA = \frac{1}{\mu}$	

Fonte: Adaptado de (PRADO, 2009).

Quadro 2.1 – Variáveis randômicas fundamentais.

2.4 Fórmula de Little

De acordo com Chwif e Medina (2015), a fórmula de Little pode ser aplicada em diversos sistemas de filas em estado estacionário. Chwif e Medina (2015) afirma que a fórmula de Little pode ser definida de acordo com a Equação 2.1.

$$NS = \lambda \cdot TS \quad (2.1)$$

Sendo que equação 2.1 pode estar em função do número esperado de clientes na fila e do tempo médio de espera, como apresentado na Equação 2.2

$$NF = \lambda \cdot TF \quad (2.2)$$

Ou em função do número médio de clientes em atendimento e do tempo médio de atendimento, de acordo com a Equação 2.3:

$$NA = \lambda \cdot TA \quad (2.3)$$

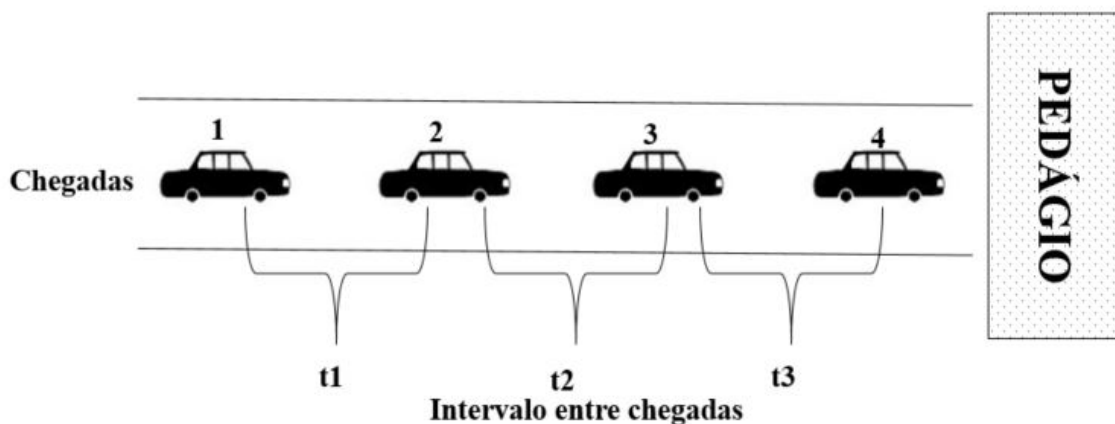
2.4.1 Modelo de nascimento e morte

O modelo de nascimento e morte pode ser tratado sob duas perspectivas, a saber: (i) nascimento puro: onde somente chegadas são permitidas (exemplo: Emissão de certidões

de nascimento) e (ii) morte pura: onde só podem ocorrer partidas (exemplo: A retirada aleatória de uma peça do estoque, sem reposição). Diante deste fato, a distribuição exponencial é utilizada para descrever o intervalo de tempo entre chegadas (em modelo de nascimento puro) e o intervalo de tempo entre partidas (em modelos de morte pura) (ARENALES *et al.*, 2007; TAHA, 2008).

Existe uma inerente relação entre a distribuição exponencial e a distribuição de Poisson, na qual assume-se que uma distribuição define a outra automaticamente. Enquanto a distribuição exponencial é contínua, a distribuição de Poisson é caracterizada como discreta. Dito isto, diante de um mesmo experimento, existem duas concepções diferentes, não obstante, complementares (ARENALES *et al.*, 2007; TAHA, 2008; PRADO, 2009). Conforme segue a Figura 2.3.

Figura 2.3 – Relação Poisson e exponencial.



Fonte: o autor.

A Figura 2.3 descreve a chegada de carros em uma área de pedágios. As chegadas formam o ritmo e à frequência pela qual carros chegam no sistema ao longo de um tempo τ . Dado um evento discreto, esta situação é representada pela distribuição de Poisson. Todavia, percebe-se que por conta das chegadas, formam-se intervalos de tempo entre as mesmas (τ_1 , τ_2 e τ_3), que por sua vez, são compreendidos pela distribuição exponencial (ARENALES *et al.*, 2007; TAHA, 2008; PRADO, 2009).

Na Tabela 2.1, expõe-se de forma resumida, a relação da distribuição exponencial com a distribuição de Poisson.

Tabela 2.1 – Resumo de relações entre a distribuição de Poisson e a Exponencial – Pesquisa Operacional.

	Exponencial	Poisson
Variável aleatória	Tempo entre chegadas	Número de chegadas, n , durante um período especificado T
Faixa	$t \geq 0$	$n = 0, 1, 2, \dots$
Função densidade probabilidade	$f(t) = \lambda \cdot e^{-\lambda t}, t \geq 0$	$P_n(T) = \frac{(\lambda T)^n \cdot e^{-\lambda T}}{n!}, n = 0, 1, 2, \dots$
Valor médio	$\frac{1}{\lambda}$ unidades de tempo	λT chegadas durante T

Fonte: o autor.

2.5 A notação de Kendall

Arenales et al. (2007) afirmam que os sistemas de fila podem variar de acordo com as especificidades do processo, dentre algumas variações, observa-se:

- Fila única e um servidor.
- Filas única e múltiplos servidores em paralelo.
- Múltiplas filas e múltiplos servidores em paralelo.
- Fila única e múltiplos servidores em série.

Ainda de acordo com Arenales et al. (2007), uma diferença entre as realidades supracitadas é fato de ter ou não, múltiplos estágios. No sistema de pedágios (Figura 2.3) se percebe que existe apenas um estágio. Portanto o usuário passa por apenas um posto de atendimento. Já em uma linha de produção de motocicletas (equipamentos em série), conforme exposto na Figura 2.4, os produtos cumprem um fluxo com várias estações de trabalho (E1–E5) até deixarem o sistema, o que caracteriza múltiplos estágios.

Figura 2.4 – Linha de produção de motocicletas.



Fonte: o autor.

Para Arenales et al. (2007) e Chwif e Medina (2015), todas estas variações de sistemas de filas correspondem à um conjunto de modelos de filas, que em casos de menor complexidade, podem ser representados analiticamente. Para facilitar esta análise, a caracterização

de um sistema de filas, pode ser dada pela notação de Kendall-Lee, que se equivale à uma sigla com até seis variáveis: $(a/b/c):(d/e/f)$, onde:

- a : representa o processo de chegada dos elementos aos postos de atendimento, determinado pelo intervalo entre chegadas. Dentre as principais abreviações, destacam-se:
 - (i) M : o intervalo de tempo entre chegadas é constituído por variáveis aleatórias independentes, com distribuição exponencial.
 - (ii) D : os intervalos de tempo entre chegadas são determinísticos.
 - (iii) E_k : os intervalos de tempo entre chegadas são variáveis aleatórias com distribuição de Erlang, que assim como a distribuição exponencial, é contínua. Pode ser utilizada por exemplo, para analisar o número de chamadas telefônicas que uma rede de comutação recebe simultaneamente.
 - (iv) G : os intervalos de tempo entre chegadas são variáveis aleatórias representadas pela distribuição genérica.
- b : indica a distribuição de probabilidade do processo de atendimento de cada estação de serviço. Abreviações: M , D , E_k e G .
- c : número de servidores em paralelo.
- d : disciplina de atendimento da fila.
- e : número máximo de usuários no sistema
- f : tamanho da população ou da fonte de usuários.

2.6 Modelos tradicionais de filas

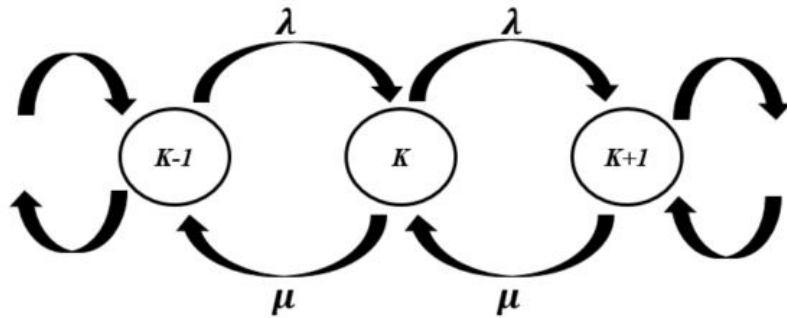
2.6.1 M/M/1

Dentre os modelos de filas clássicos, destaca-se o modelo $M/M/1$. Kleinrock (1975) afirma que tal modelo pode ser descrito através do nascimento e morte dos seguintes coeficientes descritos pela Equação 2.4.

$$\begin{aligned} \lambda_k & \quad k = 0, 1, 2, \dots \\ \mu_k & \quad k = 1, 2, 3, \dots \end{aligned} \tag{2.4}$$

Como característica, salienta-se o fato de que as variáveis aleatórias (i) intervalo entre chegadas e (ii) intervalo de atendimento são distribuídas exponencialmente. Além disso, existe apenas um único servidor (KLEINROCK, 1975). O Comportamento desse sistema pode ser representado pelo diagrama de taxas de transição conforme Figura 2.5.

Figura 2.5 – Diagrama de transição.



Fonte: o autor.

No que se refere aos parâmetros e expressões analíticas do modelo $M/M/1$, pode-se observar a Tabela 2.2.

Tabela 2.2 – Parâmetros e expressões analíticas M/M/1 – Modelagem e simulação de eventos discretos.

Parâmetro	Expressão
Índice de ocupação do sistema	$\rho = \frac{\lambda}{\mu}$
Probabilidade de o sistema estar livre	$p_0 = 1 - \mu$
Probabilidade de j elementos no sistema	$P_j = (1 - \mu) \cdot p^j, \quad j = 1, 2, \dots$
Probabilidade de mais do que K elementos no sistema	$P[\geq k \text{ no sistema}] = \rho^k$
Média de elementos em fila	$L_q = \frac{\rho^2}{1 - \rho} = \frac{\lambda^2}{\mu(\mu - \lambda)}$
Média de elementos em atendimento	$L_s = \rho$
Média de atendimento no sistema	$L = \frac{\rho}{1 - \rho} = \frac{\lambda}{\mu - \lambda}$

Fonte: (CHWIF; MEDINA, 2015).

A resolução analítica deste modelo é possível graças à sua estabilidade, o que significar dizer que a fila não cresce indefinidamente ao longo do tempo. Matematicamente, esta condição é atendida quando o índice de congestionamento ρ for menor do que 1 (CHWIF; MEDINA, 2015).

2.6.1.1 Exemplo de aplicação do modelo M/M/1

Em uma oficina mecânica de reparos leves que possui apenas um reparador, os clientes chegam a uma taxa de 10 clientes por hora, sendo que o reparador atende em média 15 clientes por hora. Considera-se a taxa de chegada λ e a taxa de atendimento μ variáveis aleatórias distribuídas exponencialmente. Em relação aos indicadores básicos de desempenho, observa-se: as relações representadas pelas Equações 2.5, 2.6, 2.7, 2.8 e 2.9

$$\rho = \frac{\lambda}{\mu} = \frac{10}{15} = 0.666 \text{ ou } 66\% (\text{sendo } \rho < 1, \text{ o sistema é estável}) \quad (2.5)$$

$$P_0 = 1 - \rho = 1 - \frac{10}{15} = 33\% \quad (2.6)$$

$$L = \frac{\lambda}{\mu - \lambda} = \frac{10}{15 - 10} = 2 \text{ Clientes} \quad (2.7)$$

$$L_q = \frac{\rho^2}{1 - \rho} = \frac{(\frac{10}{15})^2}{1 - \frac{10}{15}} = 1.3 \text{ Clientes} \quad (2.8)$$

$$L_s = \rho = 0.66 \text{ Clientes} \quad (2.9)$$

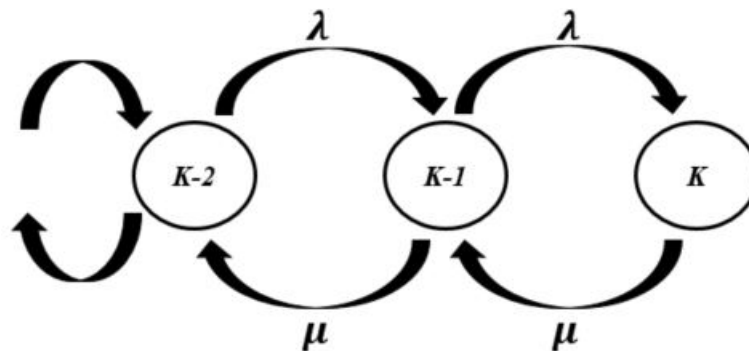
Conforme o sistema descrito, foi possível obter a probabilidade de 33% de o sistema estar ocioso; a média de atendimento no sistema de 2 clientes; a média de clientes na fila de 1.3 e; a média de clientes em atendimento de 0.66 cliente. Além disso, por meio do cálculo de ρ , tornou-se possível dizer que o sistema é estável, já que o valor obtido é menor que 1.

2.6.2 M/M/1/c

O Modelo $M/M/1/c$ possui praticamente as mesmas características do modelo $M/M/1$. Entretanto, a capacidade (c) do sistema neste caso, é limitada. Se o sistema está operando em sua capacidade máxima, novos clientes não podem entrar em um processo de espera, fato que levam aos mesmos optarem por desistirem do serviço. A chegada de clientes ainda ocorrerá segundo a distribuição de Poisson, mas, só serão atendidos aqueles que encontrarem um número de clientes menor que c no sistema (KLEINROCK, 1975).

O comportamento do modelo $M/M/1/c$ pode ser representado conforme Figura 2.6 (considere $c = k$). Uma empresa de telefonia por exemplo, possui um número limitado c de atendentes, que ao estarem todos ocupados, as chamadas excedentes são bloqueadas (KLEINROCK, 1975).

Figura 2.6 – Diagrama de transição em um sistema com capacidade limitada.



Fonte: o autor.

No que se refere aos parâmetros e expressões analíticas do modelo $M/M/1/c$, pode-se observar a Tabela 2.3.

Tabela 2.3 – Parâmetros e expressões analíticas $M/M/1/c$ – Modelagem e simulação de eventos discretos.

Parâmetro	Expressão
Taxa efetiva de entrada no sistema	$\lambda_c = \lambda(1 - P_c)$
Índice de ocupação do sistema	$\rho = \frac{\lambda}{\mu}; \lambda \neq \mu$
Probabilidade de o sistema estar livre	$P_0 = \frac{1-\rho}{1-\rho^{c+1}}; \lambda = \mu$
Probabilidade de j elementos no sistema	$P_j = \frac{1}{1+c} \quad j = 0, 1, 2, \dots, c; \lambda = \mu$
Probabilidade de mais do que K elementos no sistema	$P[\geq k \text{ no sistema}] = \rho^k$
Média de elementos em fila	$L_q = L - L_s$
Média de elementos em atendimento	$L_s = 1 - P_0$
Média de atendimento no sistema	$L = \frac{\rho[1-(c+1)\rho^c + c\rho^{c+1}]}{(1-\rho^{c+1})(1-\rho)} \quad \lambda \neq \mu$

Fonte: (CHWIF; MEDINA, 2015).

2.6.2.1 Exemplo de aplicação do modelo $M/M/1/c$

A oficina de reparos leves do Exemplo 2.6.2.1 tem capacidade máxima de 4 clientes, ou seja, enquanto 1 está em atendimento, 3 estão em fila. Diante deste fato, o próximo cliente desistirá do serviço por limitações de capacidade. Em relação aos indicadores básicos de desempenho, é possível obter as propriedades expressas pelas equações 2.10 - 2.15, considerando $\lambda \neq \mu$.

$$\rho = \frac{\lambda}{\mu} = \frac{10}{15} = 0.666 \text{ ou } 66\% \quad (2.10)$$

$$P_0 = \frac{1-\rho}{1-\rho^{c+1}} = \frac{1-\frac{10}{15}}{1-\frac{10^{4+1}}{15}} = 0.383 \text{ ou } 38.3\% \quad (2.11)$$

$$L_s = 1 - \rho_0 = 1 - 0.383 = 0.61 \text{ Clientes} \quad (2.12)$$

$$L = L = \frac{\rho[1-(c+1)\rho^c + c\rho^{c+1}]}{(1-\rho^{c+1})(1-\rho)} = \frac{\frac{10}{15}[1-(4+1)\frac{10^4}{15} + 4\frac{10^{4+1}}{15}]}{(1-\frac{10^{4+1}}{15})(1-\frac{10}{15})} = 4 \text{ Clientes} \quad (2.13)$$

$$L_q = L - L_s = 3.39 \text{ Clientes} \quad (2.14)$$

A probabilidade de o sistema estar cheio (1 em atendimento e 3 em fila):

$$\rho^c \cdot P_0 = \left(\frac{10}{15}\right)^4 \cdot \frac{1-\frac{10}{15}}{1-\left(\frac{10}{15}\right)^{4+1}} = 0.075 \text{ ou } \cong 8\% \quad (2.15)$$

Logo, as resoluções das equações possibilitaram calcular o índice de ocupação do sistema de 66%; a probabilidade de 38,3% de o sistema estar livre; a média de 0.61 clientes em atendimento; a média de atendimento de 4 clientes no sistema; o tamanho médio da fila de 3.39 clientes e; a probabilidade de 8% de o sistema estar cheio.

2.6.3 M/D/1

O Modelo $M/D/1$ possui as seguintes características, a saber:

- Intervalos entre chegadas distribuídos exponencialmente.
- Tempo de serviço fixo.
- Apenas um servidor (PRASAD; B.USHA, 2015).

No que se refere aos parâmetros e expressões analíticas do modelo $M/D/1$, pode-se observar a Tabela 2.4.

Tabela 2.4 – Parâmetros e expressões analíticas M/D/1 – Modelagem e simulação de eventos discretos.

Parâmetro	Expressão
Índice de ocupação do sistema	$\rho = \frac{\lambda}{\mu}$
Probabilidade de o sistema estar livre	$P_0 = 1 - \rho$
Probabilidade de j elementos no sistema	$P_j = (1 - \rho)\rho^j, \quad j = 1, 2, 3, \dots$
Probabilidade de mais do que K elementos no sistema	$P[\geq k \text{ no sistema}] = \rho^k$
Média de elementos em fila	$L_q = \frac{\rho^2}{2(1-\rho)}$
Média de elementos em atendimento	$L_s = \rho$
Média de atendimento no sistema	$L = \rho + \frac{\rho^2}{2(1-\rho)}$

Fonte: (CHWIF; MEDINA, 2015).

2.6.3.1 Exemplo de aplicação do modelo M/D/1

Suponha que o tempo de serviço da oficina descrita anteriormente seja fixo, os indicadores de desempenho para o sistema seriam expressos pelos resultados das Equações 2.16 - 2.20.

$$\rho = \frac{\lambda}{\mu} = \frac{10}{15} = 0.666 \text{ ou } 66\% \quad (2.16)$$

$$P_0 = 1 - \rho = 1 - \frac{10}{15} = 0.333 \text{ ou } 33.3\% \quad (2.17)$$

$$L_s = \rho = 0.66 \text{ Clientes} \quad (2.18)$$

$$L_q = \frac{\rho^2}{2(1-\rho)} = \frac{(\frac{10}{15})^2}{2(1-\frac{10}{15})} = 0.66 \text{ Clientes} \quad (2.19)$$

$$L = \rho + \frac{\rho^2}{2(1-\rho)} = \frac{10}{15} + 0.66 = 1.33 \text{ Clientes} \quad (2.20)$$

Portanto, por meio dos devidos cálculos obteve-se como taxa de ocupação do sistema um índice de 66%; uma probabilidade de ociosidade de 38.3%; uma média de clientes em atendimento de 0.61; a média de atendimento no sistema de 4 cliente; um tamanho médio da fila de 3.39 clientes e por fim; a probabilidade de o sistema estar cheio foi de aproximadamente 8%.

Vale ressaltar que em grande parte das situações reais, os intervalos de tempos dos serviços não são determinísticos, sobretudo, os que possuem maior customização. Isto posto, [Baek et al. \(2016\)](#) afirma que os estudos relacionados ao modelo $M/D/1$ são escassos se comparados aos modelos com tempos de serviço variados.

2.6.4 M/M/s

O modelo de fila $M/M/s$ também possui seus intervalos entre chegadas e de atendimentos distribuídos exponencialmente. Entretanto, ao contrário dos modelos anteriores, o mesmo é caracterizado por apresentar um sistema com s servidores em paralelo. O modelo com múltiplos servidores, geralmente é mais aplicável no mundo real em detrimento de modelos de servidor único ([KLEINROCK, 1975](#); [KIM; LIM, 2016](#)).

Outra situação, é o fato de que o tempo de espera em um sistema $M/M/s$ é menor do que o tempo de espera em um sistema $M/M/1$ para qualquer $s > 1$, haja vista que as taxas de chegada e de serviço são idênticas ([KIM; LIM, 2016](#)). Portanto, filas com múltiplos servidores possibilitam a modelagem e análise de sistemas como: filas de serviços integradas, alocação da produção, congestionamento em tráfego, arranjo físico de produção com máquinas em paralelo etc. ([LEE; LI, 1992](#)).

A Tabela 2.5 apresenta os parâmetros e expressões analíticas referentes ao modelo $M/M/s$.

Tabela 2.5 – Parâmetros e expressões analíticas $M/M/s$ – Modelagem e simulação de eventos discretos.

Parâmetro	Expressão
Índice de ocupação do sistema	$\rho = \frac{\lambda}{s \cdot \mu}$
Probabilidade de o sistema estar livre	$P_0 = \left[\sum_{n=0}^{s-1} \frac{1}{n!} \left(\frac{\lambda}{\mu} \right)^n + \frac{1}{s!} \left(\frac{\lambda}{\mu} \right)^s \cdot \left(\frac{s\mu}{s\mu - \lambda} \right) \right]^{-1}$
Probabilidade de j elementos no sistema	$P_j = \frac{(s\rho)^j}{j!} \cdot P_0, \quad j = 1, 2, 3, \dots, s-1$
Probabilidade de existir s ou mais elementos no sistema	$P_j = \frac{(s\rho)^j}{s! \cdot s^{j-s}} \cdot P_0, \quad j = s, s+1, s+2, \dots$
	$P(j \geq s) = \frac{(s\rho)^s}{s!(1-\rho)} \cdot P_0$
Média de elementos em fila	$L_q = \frac{P(j \geq s)\rho}{1-\rho} \cdot P_0$
Média de elementos em atendimento	$L_s = \frac{\lambda}{\mu}$
Média de atendimento no sistema	$L = \frac{\lambda}{\rho} + \frac{P(j \geq s)\rho}{1-\rho}$

Fonte: (CHWIF; MEDINA, 2015).

No que se refere ao modelo $M/M/s$, destacam-se alguns trabalhos, como Rothkopf e Rech (1987), Al-Seedy *et al.* (2009) e A.Gómez-Corral e García (2014).

2.7 Sistemas de produção

Desde de o princípio da civilização o ser humano se organiza para produzir bens de consumo. Dentre as principais fases de evolução dos sistemas de produção, destacam-se: **i.** Produção antiga: registro de estoques e construção de pirâmides; **ii.** Sistema feudal: caracterizava-se pela produção doméstica; **iii.** Europeu: iniciou-se no renascentismo no século *XIII*, mas sofreu as maiores transformações no início do século *XVII* com a revolução industrial e **iv.** Americano: criação do torno de usinagem (FERNANDES; GODINHO FILHO, 2010).

Um Sistema de Produção (SP) pode ser definido como um conjunto de elementos humanos, físicos e gerenciais que ao trabalharem em conjunto, propiciam produtos finais com valor agregado. De outro modo, conforme a Figura 2.7, sistema de produção é um sistema que transforma entradas em saídas com valor intrínseco (FERNANDES; GODINHO FILHO, 2010).

Figura 2.7 – Sistema de produção.



Fonte: o autor.

No que diz respeito às diversas classificações de sistemas de produção, destacam-se as que categorizam os mesmos em *jobshop* e *flowshop*. Pode-se observar no Quadro 2.2, algumas diferenças entre ambos (FERNANDES; GODINHO FILHO, 2010).

Jobshop	Flowshop
- Trabalha com lotes. - Varia a produção através do tamanho e frequência dos lotes. - Geralmente possui custos maiores de setup. - A operação de produção é centralizada por processo. Tende a ter filas maiores em centros de produção específicos. - Utilização flexível de equipamentos	- Opera em fluxo de materiais. - Varia a produção conforme a taxa de produção.). - Tende a ter custo menores de setup. - Operações distintas são sequenciadas para manter o fluxo. Filas são pequenas e relativas às taxas de produção. - Uso dedicado de equipamentos

Fonte: Adaptado de (FERNANDES; GODINHO FILHO, 2010).

Quadro 2.2 – Diferenças entre *Jobshop* e *Flowshop*.

Pode-se dizer que o sistema *jobshop* possibilita uma variedade maior na família de produtos, haja vista que, conforme o Quadro 2.2, seus fluxos de produção e equipamentos possuem maior flexibilidade quando comparados ao sistema *flowshop*. No entanto, um sistema *jobshop* tende a ser mais complexo, já que situações como filas e custos maiores de tempos de preparação são inerentes ao processo. Vale ressaltar que a escolha de um sistema de produção está vinculada à estratégia que a organização adota quanto ao atendimento da demanda, e não ao fato de ser melhor ou pior Slack, Chambers e Johnston (2010).

Outro aspecto relevante nos sistemas de produção é o arranjo físico. Slack, Chambers e Johnston (2010) afirmam que existem cinco tipos básicos de arranjo físico:

1. Arranjo físico posicional: Recursos dedicados que se movimentam em torno do produto. Por exemplo: Construção de navios em estaleiros e prédios.

2. Arranjo físico por processo: Máquinas e equipamentos agrupados de acordo com a afinidade e características em comum. Produtos menores que apesar de terem necessidades específicas, compartilham recursos. Por exemplo: Indústria de confecção.
3. Arranjo físico celular: Também conhecido como tecnologia de grupo, pois, as máquinas são divididas em grupos e os componentes em famílias. Sendo assim, todos os elementos de uma família podem ser processados em somente um grupo de máquinas. Por exemplo: o abastecimento de bancos em uma linha de produção por uma estação de trabalho celular.
4. Arranjo físico por produto: Caracterizado por produzir grandes volumes e baixas variáveis. Por exemplo: Linha de produção automotiva.
5. Arranjos físicos mistos: De acordo com necessidades específicas, combina-se as propriedades supracitadas.

2.7.1 Sistemas de produção sob a perspectiva das filas de espera

Com a variação e diversificação da demanda, os sistemas de produção tornaram-se complexos, o que implica na busca por desenvolvimento contínuo de novas estratégias de produção. Diante de tal conjuntura, o gerente de produção deve agir de maneira sistêmica, levando em consideração aspectos como qualidade, velocidade, confiabilidade, flexibilidade e custo (SLACK; CHAMBERS; JOHNSTON, 2010).

Evidencia-se que além da crescente complexidade dos sistemas de produção, momentos de instabilidade econômica fazem com que os recursos financeiros sejam constantemente reavaliados. Nesta perspectiva, destaca-se alguns princípios que colaboram para a avaliação destes recursos, como é o caso da produção enxuta, que tem como base o Sistema Toyota de Produção, desenvolvido a partir da década de 1950 com Taiichi Ohno, que foi sobretudo, influenciado após visitar os Estados Unidos em 1956 (SALGADO, 2009; VOTTO; FERNANDES, 2013).

A produção enxuta é composta por princípios fundamentais, como: especificar valor para os clientes; identificar o fluxo de valor; realizar as atividades em fluxo contínuo (sempre que possível); onde o fluxo contínuo não for possível, implementar a produção puxada e; trabalhar com a filosofia de melhoria contínua (SALGADO, 2009; VOTTO; FERNANDES, 2013).

Diante da mentalidade da produção enxuta, a produção contínua é tratada como uma forma eficiente de produzir e tem como resultado a redução do tempo de entrega de produção. Para isso é necessário: (a) efetuar o balanceamento das estações de trabalho; (b) redução do tamanho de lote de transferência; (c) flexibilizar o arranjo físico de produção, com a utilização de linhas ou células de produção e (d) através da otimização dos tempos

de preparação, reduzir o tamanho dos lotes em processamento (SALGADO, 2009; VOTTO; FERNANDES, 2013).

O pensamento base da produção enxuta é a eliminação de desperdícios dentro das empresas. Existem sete desperdícios segundo o Sistema Toyota de Produção (STP), a saber:

- (i) Superprodução.
- (ii) Espera.
- (iii) Transporte excessivo.
- (iv) Processos inadequados.
- (v) Inventário desnecessário.
- (vi) Movimentação desnecessária.
- (vii) Produtos defeituosos (SALGADO, 2009; VOTTO; FERNANDES, 2013).

Diante dos sete desperdícios, salienta-se que os mesmos podem ser aportados sob diferentes percepções, inclusive sob a de filas de espera, tendo em vista que certos problemas como o de inventário desnecessário, de movimentação excessiva e da própria espera podem ser solucionados através de modelagem estocástica (ROY, 2015).

No entanto, em vez de utilizar modelos estocásticos em questões táticas e estratégicas como a escolha do equipamento, planejamento da capacidade e realocação de trabalhadores Bitran e Dasu (1992), afirmam que grande parte das soluções propostas em face a problemas fabris eram relacionados a táticas de sequenciamento da produção.

No que corresponde aos últimos dez anos (período compreendido entre 2007 a 2017), percebe-se que existe uma grande variedade de trabalhos relacionados à filas em sistemas de manufatura. Dentre outros, pode-se citar:

- Smith (2008): Análise de performance em um ambiente com múltiplos servidores, *buffers* finitos entre estações e tempos de espera limitados.
- Sarkar, Mukhopadhyay e Ghosh (2015): Otimização de tempos ociosos através da alocação de operadores em grupos de máquinas.
- Ghezavati e Saidi-Mehrabad (2011) e Fardis, Zandi e Ghezavati (2013): Planeamento e análise da capacidade em células de produção.
- Marsudi e Shafeek (2014): Análise de performance de uma linha de produção com múltiplos estágios e múltiplos produtos.

- [Buyukkaramikli, Bertrand e VanOoijen \(2011\)](#): Análise do tempo de entrega de um sistema de produção com restrições de prazos.
- [Raman, Nagalingam e Gurd \(2009\)](#): Utilização de algoritmos genéticos e teoria das filas com o objetivo de facilitar o desenvolvimento de um arranjo físico.
- [Perlman, Elalouf e Bodinger \(2013\)](#): Prioridade de reparo dinâmico para uma linha de transferência com um *buffer* finito. Modelagem do ambiente fabril através de cadeias de Markov.
- [Heragu e Srinivasan \(2011\)](#): Análise de sistemas de manufatura via redes de filas semi-abertas;

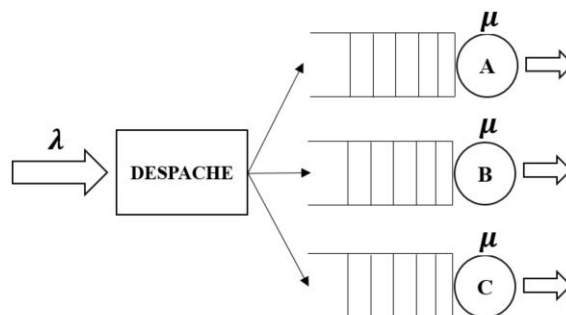
Embora as fórmulas de Erlang e a teoria de Jackson sejam definidas para sistemas com múltiplos servidores, a maior parte da teoria das filas foi desenvolvida para sistemas com um único servidor. Antes de 1970, a teoria das filas forneceu uma gama de fórmulas simples que possibilitaram a avaliação satisfatória de modelos como $M/M/1$, $M/D/1$ e $M/M/c$. No entanto, diante dos trabalhos referenciados, percebe-se que foi necessário um avanço no estado da arte, sobretudo, quando se questiona a capacidade de representação da realidade de alguns modelos ([OSOGAMI, 2005](#)).

2.8 Múltiplos servidores

Conforme introduzido na seção [2.6.4](#) o modelo com múltiplos servidores permite a representação realista de sistemas reais, como é o caso de servidores bancários, caixas de supermercado e atendentes de *fast-food*. Nada obstante, [Osogami \(2005\)](#) ressalta que a análise de performance desses sistemas se torna difícil à medida que surgem questões como:

- (i) Para qual estação (*A*, *B* ou *C*) despachar a tarefa? (Figura [2.8](#)).

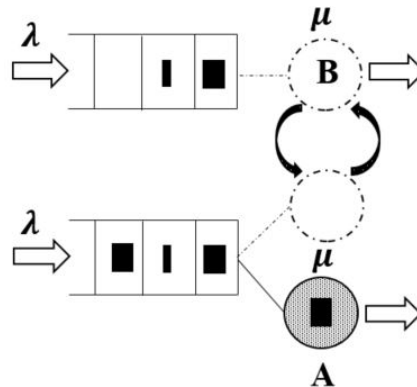
Figura 2.8 – Para onde despachar?



Fonte: Próprio Autor

- (ii) Em uma estação de trabalho, a flexibilidade pode agilizar a resposta à demanda, sendo assim, quando a capacidade da máquina *B* é excedida e a máquina *A* está ociosa (Figura 2.9), uma tarefa pode quebrar o fluxo e ir para máquina *A*? Esta situação pode ser caracterizada como “roubo” de ciclo, que por sua vez, foi implementado em vários sistemas de gestão de redes de estação de trabalho, tais como [Osogami](#), [Harchol-Balter e Scheller-Wolf \(2005\)](#) e [Semchedine e Bouallouche-Medjkoune \(2011\)](#).

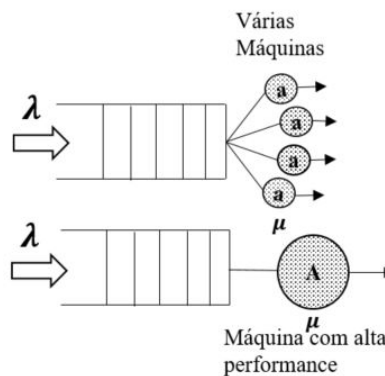
Figura 2.9 – Considerar “roubo” de ciclo? Quando?



Fonte: o autor.

- (iii) Qual situação é melhor? Utilizar uma máquina de alta performance ou utilizar várias máquinas de performance menor, Figura 2.10.

Figura 2.10 – Considerar “roubo” de ciclo? Quando?



Fonte: o autor.

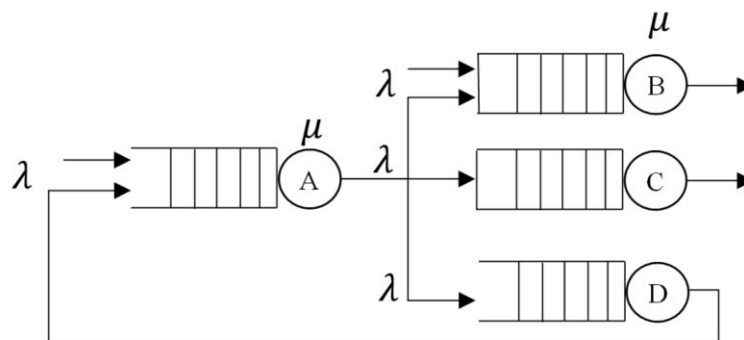
Além da dificuldade proveniente das questões (i), (ii) e (iii), à medida que se aumenta o dinamismo de um ambiente fabril, é necessário lidar com o conceito de redes de filas, haja vista que em um sistema de produção, podem existir várias estações de trabalho com fluxos distintos, como é o caso de um ambiente *jobshop* ([SHANTHIKUMAR; BUZACOTT, 1981](#))

2.8.1 Redes de filas

Segundo Silva e Morabito (2007), as redes de filas são compostas por vários sistemas de filas conectados entre si, no qual os usuários ou produtos se movimentam com o intuito de adquirir serviço ou serem processados. Quando relacionadas à manufatura, faz-se a seguinte analogia: (a) nós da rede: estações de trabalho (*shops*); (b) usuários: produtos (*jobs*) e (c) arcos conectados aos nós: fluxo dos produtos ou rotas tecnológicas. As redes podem ser classificadas como abertas e fechadas:

- (i) Abertas: o número de *jobs* na rede pode variar ao longo do tempo (Figura 2.11);

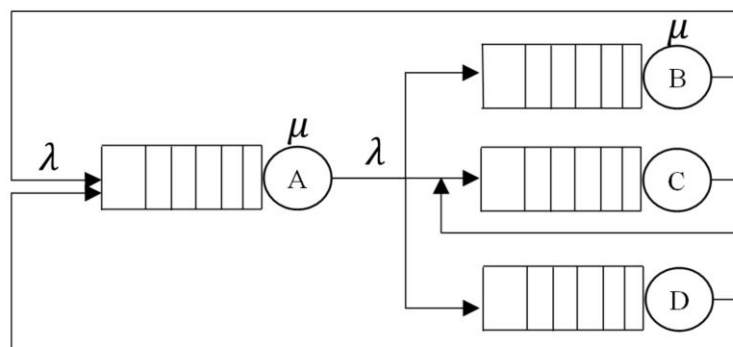
Figura 2.11 – Rede de filas aberta.



Fonte: o autor.

- (ii) Fechadas: número de *jobs* constantes ao longo do tempo (Figura 2.12)

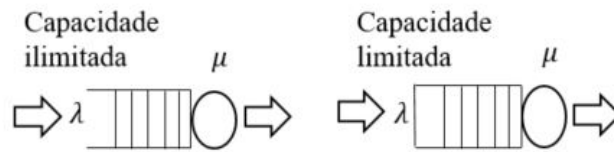
Figura 2.12 – Rede de filas fechada.



Fonte: o autor.

No que se refere a capacidade de atendimento de um servidor, a Figura 2.13 apresenta a diferença entre capacidade ilimitada e capacidade limitada.

Figura 2.13 – Capacidade ilimitada e capacidade limitada.



Fonte: o autor.

Para se avaliar a performance em redes de filas, além de definir quais são os parâmetros de desempenho, deve-se levar em consideração fatores como o desenho do layout e melhoria contínua da produtividade, haja vista que são diretamente relacionados. Sendo assim, vários trabalhos foram desenvolvidas por meio de modelos de probabilidade para avaliar performance em filas, como apresenta o Quadro 2.3.

Citação	Tema	Método	Revista	País
(ARBOS; SANTOS; SANCHEZ, 2011)	Reengenharia	Ferramenta de simulação (OT-Chart)	Computers & Industrial Engineering	Espanha
(FARDIS; ZANDI; GHEZAVATI, 2013)	Projeto de células de manufatura	Programação de números inteiros misto não-linear - Cplex de GAMS	Journal of Industrial Engineering International	Iran
(GHEZAVATI; SAIDI-MEHRABAD, 2011)	Projeto de células de manufatura	Algoritmo genético (GA) e simulated annealing (SA)	Expert Systems With Applications	Iran
(MARSUDI, 2010)	Análise de capacidade	Processos de Markov	International Journal of Integrated Engineering	Malásia
(WU; MCGINNIS, 2012)	Avaliação de Performance	Modelo de decomposição	European Journal of Operational Research	Singapura
(GANDHI <i>et al.</i> , 2014)	M/M/k/setup class of Markov chains	Recursive renewal reward	Queueing Systems	EUA
(YANG; CHIANG; TSOU, 2013)	Modelagem e análise de capacidade	Método recursivo	Journal of Manufacturing Systems	Taiwan

Fonte: o autor.

Quadro 2.3 – Aplicações de modelagem probabilística.

O Quadro 2.3 demonstra diferentes trabalhos de modelos de probabilidade desenvolvidos em sistemas de produção, que variam desde projeto de otimização de um layout até a utilização de simulação de sistemas no auxílio ao processo de reengenharia. Sendo assim, dada a diversidade supracitada, o presente trabalho foca nos modelos de probabilidade aplicados à sistemas de produção, cujo os quais, podem ser representados por um processo markoviano. Especificamente, opta-se pelo processo de quase nascimento e morte (*quasi-birth-and-death*) haja vista a facilidade que o mesmo tem em modelar situações como o estado de serviço de uma máquinas e tempos de preparação (KAMPEN, 2006; GALLAGER, 2013).

Antes de definir o processo de *quasi-birth-and-death* (QBD) na seção 3.3, o capítulo 3 busca dissertar sobre os conceitos fundamentais de modelos de probabilidade, onde, dentre eles, tem-se os modelos com propriedade markoviana (seção 3.1). Em seguida, busca-se nas seções 3.1.1, 3.2 e 3.2.1 discutir sobre as características e estrutura destes modelos, além de demonstrar algumas possibilidades de aplicações por meio de exemplos. Finalmente,

define-se o processo QBD e é apresentado o desenvolvimento de um exemplo por meio do código adaptado de [Sales e Morales \(2011\)](#).

Capítulo 3

Modelos de Probabilidade

Um modelo de probabilidade é tratado como um dos conceitos centrais da Teoria da Probabilidade. Geralmente é composto por três componentes:

- (i) Espaço amostral: Chamado de pontos de amostra ou resultados.
- (ii) Classe de eventos: Considerado como a classe de todos os subconjuntos do espaço amostral.
- (iii) Medida de probabilidade: Trata-se da atribuição de números não negativos a cada resultado. Esses números devem somar “1”, sobre o espaço amostral. Logo, a probabilidade de um evento é a soma das probabilidades dos resultados que compreendem este evento ([GALLAGER, 2013](#)).

Os modelos de probabilidade desempenham um duplo papel, sendo que mesmo não havendo grande similaridade com o mundo real, o mesmo pode servir como bloco de construção ou como um exemplo útil da teoria e também, é usado para representar uma experiência qualquer envolvendo eventos aleatórios ([GALLAGER, 2013](#)).

Para se criar modelos estocásticos realistas é necessário considerar o fato de que o mundo é composto por situações de dependência. Por exemplo: as compras realizadas em uma loja de roupas no próximo mês, depende do nível de satisfação das compras feitas até o momento; o monitoramento dos índices pluviométricos de uma região, dependerá das leituras prévias e; o número de peças em uma célula de produção dependerá do número prévio de peças em períodos anteriores ([KLEINROCK, 1975](#); [RESNICK, 1992](#)).

O fator de dependência torna os modelos estocásticos realistas. Em contrapartida, seus respectivos cálculos se tornam complexos. Já os Modelos com eventos independentes possibilitam cálculos explícitos, porém, comprometem o fator realidade. Imagine se um modelo de probabilidade de um reator nuclear considerasse que as falhas dos seus complexos

componentes ocorressem de forma independente a gestão da manutenção seria comprometida. A premissa de independência viabiliza os cálculos, mas, torna o modelo utópico (KLEINROCK, 1975; RESNICK, 1992).

Diante destes fatos como, modelos realistas, dependência de eventos e complexidade nos cálculos, faz-se necessário buscar um equilíbrio no que se refere o nível de representação da realidade. Assim, modelos de Markov podem balancear estas questões, já que os mesmos possuem a capacidade de levar em consideração um nível parcial de dependência entre eventos, conforme seção 3.1 (KLEINROCK, 1975; RESNICK, 1992).

3.1 Modelos Markovianos

Modelos de Markov são modelos estocásticos que são utilizados para representar sistemas que são formados por transições aleatórias (KLEINROCK, 1975; TIER, 2017). Um modelo é dito de Markov, a partir do momento que o mesmo possui propriedade Markoviana, isto significa dizer que as probabilidades dos eventos futuros não dependem de toda a história percorrida, somente do estado atual (*memoryless*). Logo, as dependências tornam-se gerenciáveis, tendo em vista que os eventos futuros são condicionalmente independentes do passado (KLEINROCK, 1975; RESNICK, 1992).

Para Kleinrock (1975) e Resnick (1992), um processo markoviano pode ser definido como: X_{k+1} (evento futuro) depende somente de X_k (evento corrente), e não do conjunto de eventos $X_0, X_1, \dots, X_k$. Sendo que $X_0, X_1, \dots$ é um processo estocástico em que X_i assume valores discretos em um espaço amostral S . Um processo estocástico tem propriedade de Markov se respeitar Equação 3.1.

$$P(X_{n+1} = i | X_n, X_{n-1}, \dots, X_0) = P(X_{n+1} = i | X_n) \quad (3.1)$$

$$\forall n \geq 0 \text{ e } i \in S$$

O modelo de Markov pode ser representado em termos de suas probabilidades de transições, conforme as equações 3.2 e 3.3:

$$P_{ij} = P(X_{n+1} = j | X_n = i) \quad \forall i, j \in S \quad (3.2)$$

$$\therefore \sum_{j \in S} P_{ij} = 1 \quad \forall i \in S \quad (3.3)$$

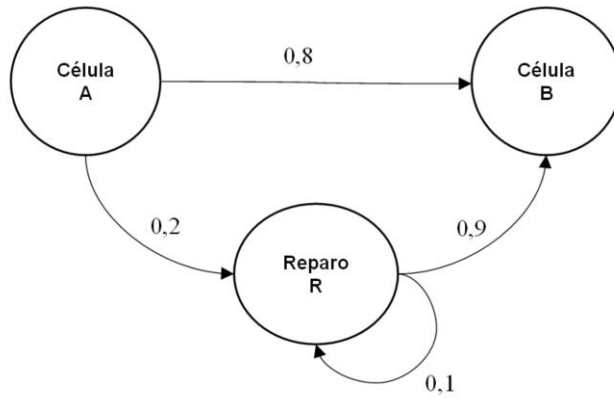
Uma cadeia de Markov pode ser considerada de tempo discreto ou contínuo. No que se refere ao tempo discreto - *Markov Chain Discret Time (MCDT)*, as transições ocorrem em um horizonte de tempo discreto onde a probabilidade de transição $P\{X(K+1) = X_{k+1} | X(K) = X_k\}$ representa a probabilidade do estado $X(K+1)$ se X_{k+1} no tempo $K+1 \quad \forall t = 1, 2, 3, \dots, n$.

As probabilidades de transição em uma *MCDT* podem ser representadas de forma matricial, por meio da matriz de probabilidades de transição P , expressa na Equação 3.4.

$$P = \begin{bmatrix} P_{11} & \cdots & P_{m1} \\ \vdots & \ddots & \vdots \\ P_{1m} & \cdots & P_{mm} \end{bmatrix} \quad (3.4)$$

Para exemplificar, considere a seguinte situação: Uma fábrica de componentes mecânicos é composta por duas células de produção e uma estação intermediária de reparo (A , B e R respectivamente). Caso o produto processado na célula A esteja livre de avarias, o mesmo cumpre o fluxo e vai para a célula B (probabilidade de ocorrência de 80%). No entanto, se a inspeção detectar alguma irregularidade, o produto quebra o fluxo e vai para a estação de reparo (probabilidade de ocorrência de 20%). Na estação R , caso o reparo seja bem-sucedido, o produto volta para o fluxo normal e vai para a célula B (probabilidade de ocorrência de 90%), caso contrário, o mesmo permanece na estação de reparo até fique apto aos olhos dos inspetores (probabilidade de ocorrência de 10%). As possibilidades, assim como as respectivas probabilidades, podem ser vistas na Figura 3.1.

Figura 3.1 – Células de produção.



Fonte: o autor.

As probabilidades representadas no modelo estão dispostas na matriz de probabilidades de transição P .

$$P = \begin{bmatrix} 0 & 0.8 & 0.2 \\ 0 & 0 & 0 \\ 0 & 0.9 & 0.1 \end{bmatrix} \quad (3.5)$$

A matriz P é composta pelos estados, que são referidos como i e j . Logo $PA(i, j)$ significa a probabilidade de transição de i para j .

Dado um estado inicial representado por um vetor de probabilidade π^0 , é possível calcular as probabilidades futuras para $\pi^1, \pi^2, \pi^3, \dots, \pi^n$. Para isso, é necessário multiplicar o vetor do estado zero pela matriz de transição P elevada ao estado futuro desejado. Considere o exemplo 3.1.1 desenvolvido em Matlab.

3.1.1 Exemplo matriz de probabilidade

Em uma área de recebimento de peças, podem ser detectados defeitos do tipo A , B e C . De acordo com a Tabela 3.1, observa-se o estado do mês de janeiro.

Tabela 3.1 – Defeitos A, B, C.

Defeito	Probabilidade
A	0.50
B	0.15
C	0.35

Fonte: o autor.

Os valores da Tabela 3.1 podem ser alocados em um vetor de probabilidade inicial π . Logo $\pi^0 = [0.5 \ 0.15 \ 0.35]$.

Levando em consideração o intervalo de um mês, têm-se as seguintes probabilidades de transição na Tabela 3.2

Tabela 3.2 – Probabilidade de transição entre os estados.

	Para A	Para B	Para C
De A	0.8	0.1	0.1
De B	0.2	0.3	0.5
De C	0.4	0.3	0.3

Fonte: o autor.

A partir do estado inicial π^0 e da matriz de probabilidade P , pode-se calcular a probabilidade de estado π para os sucessivos períodos $\pi^1, \pi^2, \pi^3, \dots, \pi^n$, ou seja, é possível inferir sobre o comportamento futuro do sistema e tomar decisões baseadas nas probabilidades de ocorrência de cada defeito. No Código 3.1, observa-se o desenvolvimento numérico em matlab.

Código 3.1 – Código matriz de transição P.

```

i=5; % Janeiro a junho
for n=1:i
    Ei = [0.30, 0.20, 0.50]; % Estado inicial
    T = [0.8, 0.1, 0.1; 0.1, 0.7, 0.2; 0.0, 0.1, 0.9]; % Matriz de
        transicao
    x = Ei*T^(n); % n = periodo
    disp(x)
end

```

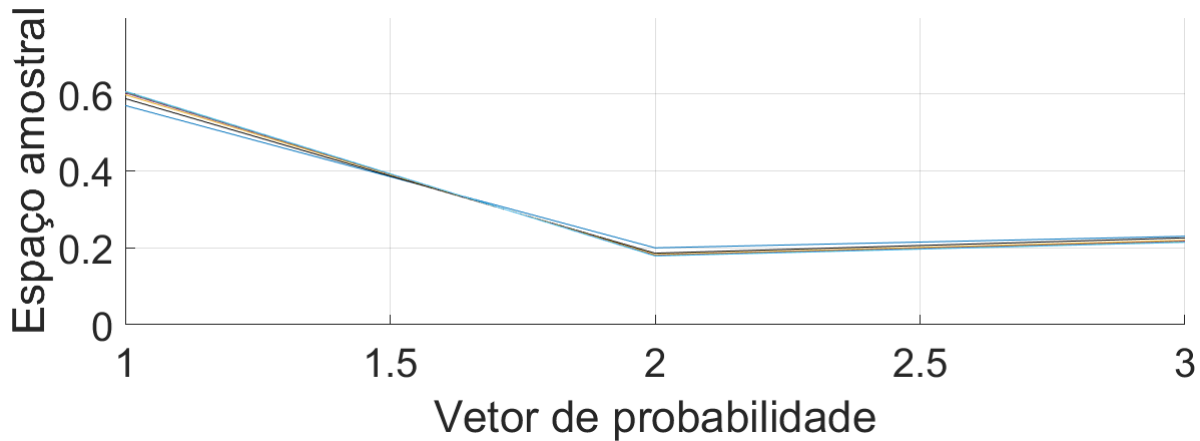
Fonte: o autor.

Por meio do Código 3.1 é possível analisar o comportamento das probabilidades das falhas, tendo como estado inicial, o vetor π^0 . Sendo assim, obtém-se os vetores de probabilidade correspondentes aos meses de janeiro a junho, o que possibilita analisar o comportamento das falhas em um mês específico, como pode ser visto:

- Janeiro - π^0 : (0.5 0.15 0.35).
- Fevereiro - π^2 : (0.57 0.2 0.23).
- Março - π^3 : (0.58 0.18 0.22).
- Abril - π^4 : (0.59 0.18 0.21).
- Maio - π^5 : (0.60 0.18 0.22).
- Junho - π^6 : (0.60 0.17 0.21).

A partir do momento em que a quantidade de observações (n) aumenta, menor será a variância dos valores em relação a média. Em outras palavras, [Downing e Clark \(2011\)](#) afirma que quanto maior o tamanho da amostra, maior será a confiança no fato de que a variável aleatória estará próxima ao valor da média. Esse resultado é conhecido como lei dos grandes números. Sendo assim, quando o código gera 1000 amostras, os vetores convergem para um valor médio que é (0.60 0.18 0.21), sendo que, a medida que as iterações aumentam, os vetores dependem cada vez menos do estado inicial π^0 , conforme observado na Figura 3.2.

Figura 3.2 – Vetores de probabilidade.



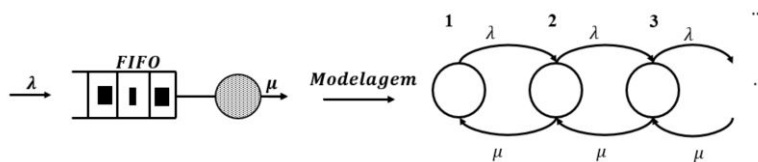
Fonte: o autor.

Por meio da Figura 3.2, torna-se possível perceber o comportamento dos valores de $A(1)$, $B(2)$ e $C(3)$ representados nos vetores de estado. Os mesmos convergem para o valor médio (0.60 0.18 0.21) ao longo do tempo. Logo, ao conhecer a média, a assertividade na tomada de decisão relacionada à falha tende a aumentar, já que este valor é o responsável por representar o comportamento do sistema.

3.2 Cadeia de Markov Multidimensional

As cadeias de Markov possibilitam a modelagem de problemas de diferentes complexidades. Por exemplo, um processo de chegada de peças em uma máquina única e uma situação em que haja roubo de ciclo. A primeira situação pode ser representada pela Figura 3.3.

Figura 3.3 – Modelagem sistema FIFO.



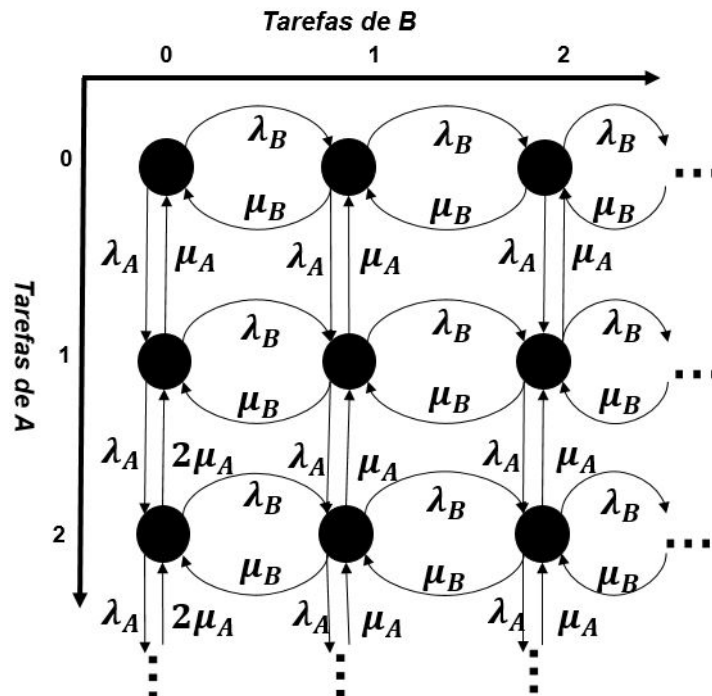
Fonte: o autor.

Conforme a Figura 3.3, percebe-se que há um único servidor, chegadas infinitas e uma política *FIFO* de atendimento. Por conta das infinitas chegadas, obtém-se uma dimensão infinita na horizontal. Já a situação de roubo de ciclo, pode ser entendida pelo exemplo 3.2.1.

3.2.1 Modelagem de roubo de ciclo por cadeias de Markov

Considere duas máquinas A e B . As taxas de chegada de tarefas em ambas as máquinas ocorrem por processo de Poisson, com taxa λ_A e λ_B respectivamente. Os tempos de serviços são distribuídos exponencialmente. As tarefas de A e de B são processadas a uma taxa de μ_A e μ_B . Quando a máquina B está ociosa, e a máquina A está ocupada e com mais uma tarefa na fila, B “rouba” ciclo de A para colaborar com o processo. Quando há o roubo de ciclo, a taxa de processamento de A passa de μ_A para $2\mu_A$. A Figura 3.4 representa a modelagem por cadeia de Markov.

Figura 3.4 – Dimensões infinitas: Roubo de ciclo.

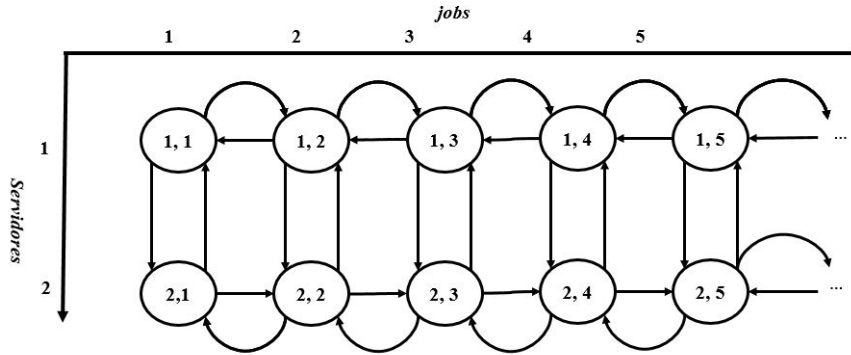


Fonte: o autor.

De acordo com a Figura 3.4, é possível perceber que a cadeia é infinita em duas dimensões, tendo em vista que chegam infinitas tarefas em ambas as máquinas. No que se refere ao roubo de ciclo, trata-se do momento em que B está com zero tarefas e A está com duas ou mais, logo a máquina B se comporta como uma doadora de ciclo e a máquina A como beneficiária. Dentre outros fatores atenuantes de complexidade, pode-se citar: recursos compartilhados; priorização de tarefas e diferentes classes de processamento (OSOGAMI, 2005).

A maioria das cadeias de Markov multidimensionais não podem ser corretamente analisadas por métodos matriciais, logo a versatilidade quanto à aplicação destes métodos se restringe à uma estrutura geral, conforme exposta na Figura 3.5

Figura 3.5 – Cadeia de Markov 1D.



Fonte: o autor.

Nota-se na Figura 3.5 que a cadeia é finita na direção vertical e infinita na horizontal. Essa cadeia é denominada 1D – finita em uma direção. Esta estrutura pode representar, por exemplo, a chegada infinita de tarefas com um número fixo de dois servidores em paralelo. Dentre outras possibilidades relacionadas à estrutura 1D, destaca-se o processo de Quase-Nascimento-e-Morte.

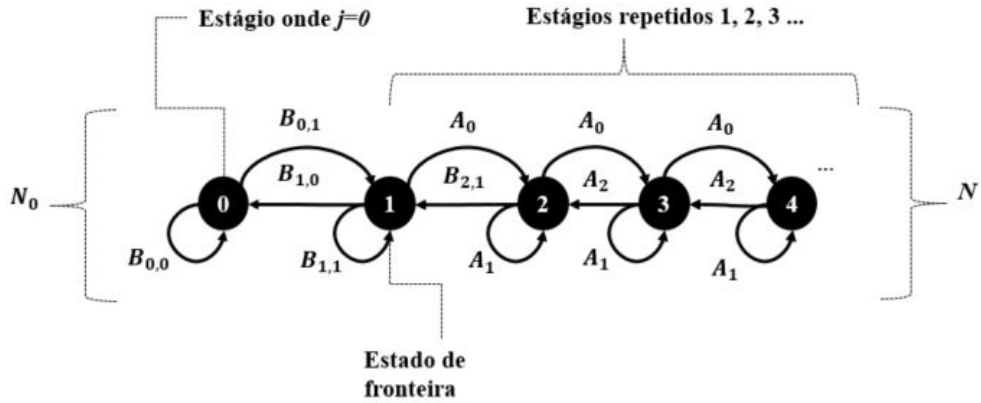
3.3 Quasi-Birth-and-Death Process (QBD)

O processo *QBD* (Quase Nascimento e Morte) se trata de um tipo de processo estocástico que possui grande expressividade na modelagem e capacidade de análise numérica. O mesmo representa uma classe especial de cadeias de Markov que possui como ideia central o fato de representar uma generalização dos sistemas de fila $M/M/1$, fornecendo uma dimensão adicional para explicar o estado aleatório de um sistema (OST, 2001).

O processo *QBD* pode representar por exemplo: um sistema de produção complexo com estações duplicadas de trabalho; o estado de serviço de uma máquina; a simulação de tempos de preparação e; um sistema que estime do número mínimo de peças de reposição reparáveis conforme trabalho proposto por (SALES; MORALES, 2011).

Os estados individuais da cadeia de Markov são identificados por pares (i, j) , $\forall j \in (0, 1, 2, \dots)$, sendo j um conjunto infinito de valores e i um conjunto finito. Logo, o processo *QBD* pode ser representado por uma tira semi infinita, conforme Figura 3.6.

Figura 3.6 – Estados agrupados do processo QBD.



Fonte: o autor.

De acordo com a Figura 3.6, percebe-se que na direção infinita, a partir do estado de fronteira “1”, existe um comportamento de repetição com um número fixo de N estados ao longo do tempo, onde as transições entre estes estados são representadas pelos índices $A_{i,j}$ e $B_{i,j}$. Por um lado, a dimensão infinita pressupõe que haja uma capacidade de chegada de infinitos Jobs, por outro, a dimensão finita considera, por exemplo, uma máquina em funcionamento ou em preparação. Considere que os estados da cadeia são representados pelos pares (N_0, N) .

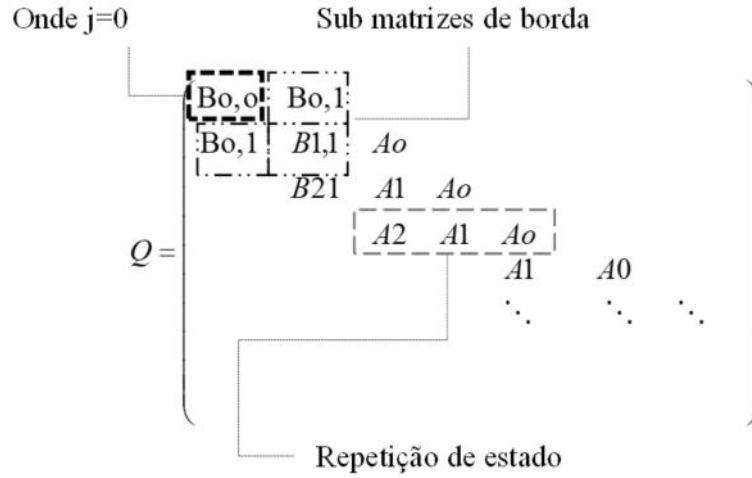
A cadeia de Markov com processo QBD é uma extensão do processo de nascimento e morte. Conforme exposto na seção 2.4.1, o processo de nascimento e morte representa as chegadas e saídas de uma estação, o que caracteriza transições entre estados adjacente. Todavia, o processo QBD, além de ter transições entre estados adjacentes, permite transições dentro do mesmo estado.

Por conta das características de repetição das transições é possível representar a estrutura da cadeia por um conjunto finito de matrizes, onde B_j , $j + d$ devem ser constantes para os níveis de repetição. Logo obtém-se o conjunto de relações da equação 3.6.

$$\begin{aligned}
 \forall j, j' \geq 1 : B_{j,j+1} &= B_{j',j'+1} = A_0 \\
 \forall j, j' \geq 1 : B_{j,j} &= B_{j',j'} = A_1 \\
 \forall j, j' \geq 3 : B_{j,j-1} &= B_{j',j'-1} = A_2
 \end{aligned} \tag{3.6}$$

Sendo assim, toda a matriz geradora de uma cadeia de Markov QBD, dita como, matriz infinitesimal, pode ser descrita pelas três matrizes A_0, A_1 e A_2 e pelas cinco matrizes sobre os limites e níveis de fronteira, conforme Figura 3.7. Esta matriz é denominada como gerador infinitesimal pelo fato de sua diagonal ser formada pela soma dos elementos com valores negativos, e também, por conta das características de suas transições, que são infinitas em uma direção conforme exposto nas Figuras 3.5 e 3.6.

Figura 3.7 – Estrutura gerador infinitesimal.



Fonte: o autor.

Todavia, por conta das limitações referentes às transições, a matriz exibe um estrutura regular que pode ser resumida em oito matrizes:

- $B_{0,0} \in \mathbb{R}^{N_0 \times N_0}$.
- $B_{0,1} \in \mathbb{R}^{N_0 \times N}$.
- $B_{1,0} \in \mathbb{R}^{N \times N_0}$.
- $B_{1,1} \in \mathbb{R}^{N \times N}$.
- $B_{2,1} \in \mathbb{R}^{N \times N}$.
- $A_0, A_1, A_2 \in \mathbb{R}^{N \times N}$.

Ost (2001) afirma que um processo de QBD de ordem (N_0, N) é um processo Markoviano no espaço do estado $\{(i, j) \in \mathbb{N}_0^2 \mid (j = 0 \wedge 0 \leq i < N_0) \vee (j > 0 \wedge 0 \leq i \leq N)\}$ com uma matriz infinitesimal conforme Figura 3.7. Assim, a matriz geradora Q pode ser expressa pela equação 3.7.

$$Q = (B, B_{2,1}, A_0, A_1, A_2) \quad (3.7)$$

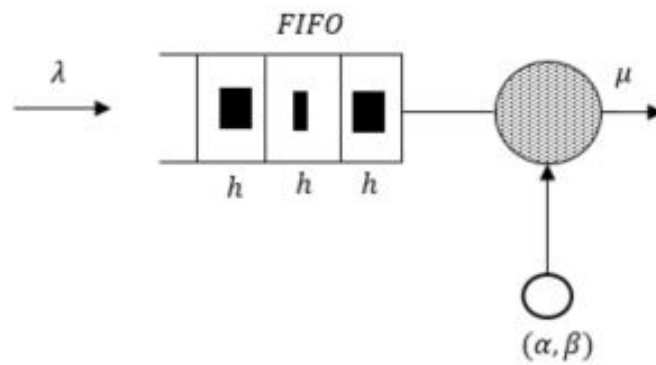
Com $B = \begin{pmatrix} B_{0,0} & B_{0,1} \\ B_{1,0} & B_{1,1} \end{pmatrix}$ e $B_{0,0} \in \mathbb{R}^{N_0 \times N_0}$, $B_{0,1} \in \mathbb{R}^{N_0 \times N}$, $B_{1,0} \in \mathbb{R}^{N \times N_0}$ e $B_{1,1}, B_{2,1}, A_0, A_1, A_2 \in \mathbb{R}^{N \times N}$.

Por conseguinte, um processo QBD representa um caso especial de duas ordens de processos estocásticos, chamados de M/G/1 e G/M/1.

3.3.1 Exemplo QBD: Servidor definido por distribuição exponencial com *buffer* limitado, considerando avarias e reparação.

O exemplo se trata de um modelo que representa uma estação de trabalho onde as peças, representadas pela variável h , podem sofrer avarias e consequentemente serem reparadas. A avaria e o reparo são representados, respectivamente, pelos parâmetros α e β , ambos distribuídos exponencialmente. Uma vez ocorrida a avaria, inicia-se o processo de reparo. Após o reparo, a mesma retorna ao processo conforme Figura 3.8.

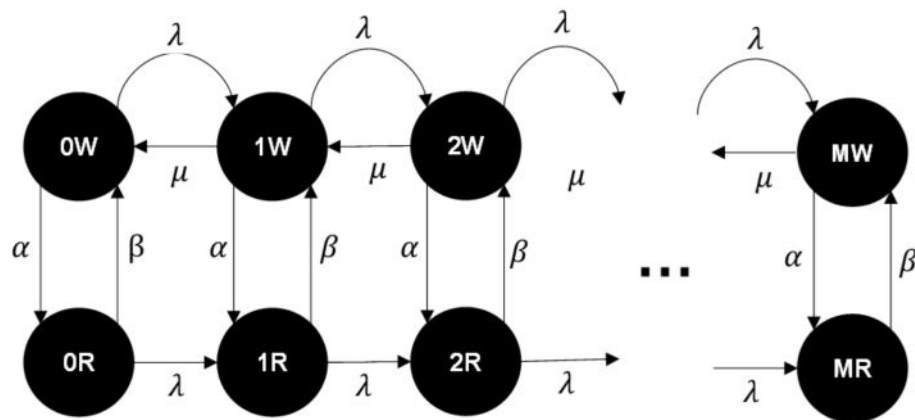
Figura 3.8 – Máquina com peças reparáveis.



Fonte: o autor.

A Figura 3.8 é representada pelo diagrama de transição de Markov, por meio da Figura 3.9. Os estados em funcionamento ou em reparo são representados respectivamente por W (*work*) e R (*repair*) .

Figura 3.9 – Representação do processo QBD.



Fonte: o autor.

Por meio do diagrama representado pela Figura 3.9 é possível desenvolver as equa-

ções de balanço que por sua vez, são regidas pela teoria de fluxo (menos o que sai mais o que entra é igual a zero), apresentadas pelas equações 3.8 - 3.19.

$$-(\alpha + \lambda) \cdot (0, W) + \beta \cdot P \cdot (0, R) + \mu \cdot P \cdot (1, W) = 0 \quad (3.8)$$

$$-(\beta + \lambda) \cdot (0, R) + \alpha \cdot P \cdot (0, W) = 0 \quad (3.9)$$

$$-(\alpha + \lambda + \mu) \cdot P \cdot (I, W) + \beta \cdot P \cdot (i, R) + \lambda \cdot P \cdot (i-1, w) + \mu \cdot P \cdot (i+1, W) = 0 \quad \text{para } 1 \leq i \leq M-1 \quad (3.10)$$

$$-(\lambda + \beta) \cdot P \cdot (i, R) + \alpha \cdot P \cdot (i, W) + \lambda \cdot P \cdot (i-1, R) = 0, \quad \text{para } 1 \leq i \leq M-1 \quad (3.11)$$

$$-(\alpha + \mu) \cdot P \cdot (M, W) + \beta \cdot P \cdot (M, R) + \lambda \cdot P \cdot (M-1, W) = 0 \quad (3.12)$$

$$-\beta \cdot P \cdot (M, R) + \alpha \cdot P \cdot (M, W) + \lambda \cdot P \cdot (M-i, R) = 0 \quad (3.13)$$

Logo, este processo pode ser observado por meio da matriz tri-diagonal Q .

$$Q = \begin{bmatrix} -(\alpha+\lambda) & \alpha & \lambda & 0 & & & & \\ \beta & -(\beta+\lambda) & 0 & \lambda & & & & \\ \mu & 0 & -(\alpha+\lambda+\mu) & \alpha & \lambda & 0 & & \\ 0 & 0 & \beta & -(\beta+\lambda) & 0 & \lambda & & \\ & & & \ddots & & & & \\ & & & \mu & 0 & -(\alpha+\lambda+\mu) & \alpha & \lambda & 0 \\ & & & 0 & 0 & \beta & -(\beta+\lambda) & 0 & \lambda \\ & & & & & \mu & 0 & -(\alpha+\mu) & \alpha \\ & & & & & 0 & 0 & \beta & -\beta \end{bmatrix} \quad (3.14)$$

A partir do gerador infinitesimal Q , as oito sub-matrizes podem ser identificadas, a saber:

$$B_{0,0} = \begin{bmatrix} -(\alpha + \lambda) & \alpha \\ \beta & -(\beta + \lambda) \end{bmatrix} \quad (3.15)$$

$$B_{0,1} = A_0 = \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} \quad (3.16)$$

$$B_{1,0} = A_2 = \begin{bmatrix} \mu & 0 \\ 0 & 0 \end{bmatrix} \quad (3.17)$$

$$B_{1,1} = A_1 = \begin{bmatrix} -(\alpha + \lambda + \mu) & \alpha \\ \beta & -(\beta + \lambda) \end{bmatrix} \quad (3.18)$$

$$B_{2,1} = \begin{bmatrix} \mu & 0 \\ 0 & 0 \end{bmatrix} \quad (3.19)$$

Conforme a estrutura já representada pela Figura 3.9, pode-se inferir que as sub matrizes $B_{0,1}$ e A_0 representam as transições entre estados adjacentes do tipo $(i, i + 1)$. Já as sub matrizes $B_{0,0}$ e $B_{1,1}$ representam as transições entre estados do mesmo nível i . E por fim, as sub matrizes $B_{1,0}$ e $B_{2,1}$ demonstram as transições entre estados adjacentes do tipo $(i + 1, i)$. Ressalta-se que, uma vez que a matriz/gerador infinitesimal possui infinitos estados, as sub matrizes simplificam o entendimento das transições e possibilita a análise numérica para estados específicos. No que se refere a análise numérica, considere o exemplo 3.2

3.3.2 Exemplo numérico: Algoritmo de HUANG-LIANG-GUO.

O Algoritmo de Huang, Liang e Guo (2006) possibilita modelar um problema que estime a quantidade de peças de reposição reparáveis em um sistema de produção, de modo a garantir o funcionamento do mesmo sob um nível confiança desejado no decorrer de um tempo t de execução. O modelo considera a variável aleatória $X(t)$, $t \geq 0$. Esta variável aleatória será representada por um processo QBD, que por sua vez representa a demanda por peças de reposição.

O algoritmo calcula a probabilidade de o sistema permanecer em funcionamento, sendo que tal probabilidade precisa ser no mínimo igual ao valor da confiança. Para isso, o sistema deve ter no mínimo h peças. A probabilidade de funcionamento e o valor da confiança são representados respectivamente pelas variáveis E e P . O valor de E pode ser calculado pela equação 3.20:

$$E = \gamma \cdot \expm(Q \cdot t) \cdot e \quad (3.20)$$

Onde:

- γ : vetor linha relacionado à uma distribuição do tipo fase (PH).
- Q : matriz de macro estados de dimensão $h + 1$.
- e : vetor coluna com todas as entradas 1's.
- t : tempo de trabalho do maquinário para teste.
- $\expm(Q \cdot t)$: calcula a exponencial da matriz Q no decorrer do tempo t .

Sendo assim, o algoritmo calcula E de forma iterativa até o momento em que a sua probabilidade é menor que a confiança P . Sendo $E > P$, o algoritmo termina as iterações e propõe um número mínimo de peças de reposição que possibilitem o funcionamento do sistema dentro da margem de confiança.

3.3.3 Gerador infinitesimal

Antes da execução do algoritmo é necessário calcular o gerador infinitesimal Q . Por meio do gerador, obtém-se as sub matrizes que servirão de entrada para a execução do código. O gerador foi obtido em ambiente *MATLAB* versão 2015 por meio de uma adaptação do código proposto por [Sales e Morales \(2011\)](#) em ambiente *Scilab* 5.2.2.

Como parâmetro de entrada do código do gerador Q , foram considerados:

- $\alpha = (1, 0, 0)$: Vetor de probabilidade que representa as m fases operacionais.
- $\beta = (1, 0, 0)$: Vetor de probabilidade que representa as k fases de reparação.
- $T = \begin{bmatrix} -0.0027 & 0.0027 & 0 \\ 0 & -0.008 & 0.008 \\ 0 & 0 & -0.02878 \end{bmatrix}$: Matriz que governa o tempo operacional.
- $S = \begin{bmatrix} -0.02 & 0.02 & 0 \\ 0.01 & -0.08 & 0.07 \\ 0.005 & 0 & -0.1 \end{bmatrix}$: Matriz que governa o tempo de reparação.

Os dados contidos nas matrizes S e T foram propostos no trabalho de [Pérez-Ocón e Montoro-Carzola \(2004\)](#). Os mesmos afirmam que a estrutura dessas matrizes indica a maneira pela qual as unidades do sistema operam e são reparadas. As fases operacionais α significam por exemplo, os níveis degradantes sucessivos que a unidade (h) atravessa enquanto está em operação. Uma falha só pode ocorrer na última fase, já que esta, representa o estado mais deteriorado.

A matriz infinitesimal do presente exemplo possui a estrutura representada pela matriz Q :

$$Q = \begin{bmatrix} B_{0,0} & B_{0,1} & & & & & \\ B_{1,0} & A_1 & A_0 & & & & \\ & A_2 & A_1 & A_0 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & A_2 & A_1 & A_0 & \\ & & & & A_2 & A_1 & \\ & & & & & \ddots & \ddots \end{bmatrix} \quad (3.21)$$

Sendo que pelas operações de Kronecker se pode obter as sub matrizes, tais como:
 $B_{0,0} = T$, $B_{0,1} = T^0 \alpha \otimes \beta$, $B_{1,0} = I \otimes S^0$, $A_1 = T \otimes S$, $A_0 = T^0 \alpha \otimes I$, $A_2 = I \otimes S^0 \beta$, $B_{n,n-1} = T^0 \otimes I$, $B_{n-1,n} = S^0 \alpha \otimes \beta$ e $B_{n,n} = S$.

Onde:

- I = Matriz identidade de ordem 3×3 .
- $T^0 \geq 0 = -T_e$.
- $S^0 \geq 0 = -S_e$.
- e : vetor coluna com todas as entradas 1's.

De acordo com as operações descritas, implementa-se o Código [3.2](#).

Código 3.2 – Código gerador infinitesimal.

```

1- m=3;
2- T=[-0.0027 0.0027 0; 0 -0.008 0.008; 0 0 -0.02878];
3- S=[-0.02 0.02 0; 0.01 -0.08 0.07; 0.005 0 -0.1];
4- alfa=[1 0 0];
5- betta=[1 0 0];
6- e=ones(m,1);
7- I=eye(m,m);
8- BZ=zeros(m,m);
9- B1K_Z=zeros(m,m^2);
10- BLK_Z=zeros(m^2,m^2);
11- BL1_Z=zeros(m^2,m);
12- To=T*e;
13- So=S*e;
14- Bo1=To*alfa;
15- Ao2=So*betta;
16- Bn=So*alfa;
17- B00=T;
18- B01=[Bo1(1,1)*betta Bo1(1,2)*betta Bo1(1,3)*betta; Bo1(2,1)*betta
      Bo1(2,2)*betta Bo1(2,3)*betta; Bo1(3,1)*betta Bo1(3,2)*betta Bo1
      (3,3)*betta];
19- B10=[I(1,1)*So I(1,2)*So I(1,3)*So; I(2,1)*So I(2,2)*So I(2,3)*So;
      I(3,1)*So I(3,2)*So I(3,3)*So];
20- BN_1N=[To(1,1)*I; To(2,1)*I; To(3,1)*I];
21- BNN_1=[Bn(1,1)*betta Bn(1,2)*betta Bn(1,3)*betta; Bn(2,1)*betta Bn
      (2,2)*betta Bn(2,3)*betta; Bn(3,1)*betta Bn(3,3)*betta Bn(3,3)*betta
      ];
22- BNN=S;
23- A1=[T(1,1)*I+I(1,1)*S T(1,2)*I+I(1,2)*S T(1,3)*I+I(1,3)*S; T(2,1)*I
      +I(2,1)*S T(2,2)*I+I(2,2)*S T(2,3)*I+I(2,3)*S; T(3,1)*I+I(3,1)*S
      T(3,2)*I+I(3,2)*S T(3,3)*I+I(3,3)*S];
24- A0=[Bo1(1,1)*I Bo1(1,2)*I Bo1(1,3)*I; Bo1(2,1)*I Bo1(2,2)*I Bo1
      (2,3)*I; Bo1(3,1)*I Bo1(3,2)*I Bo1(3,3)*I];
24- A2=[I(1,1)*Ao2 I(1,2)*Ao2 I(1,3)*Ao2; I(2,1)*Ao2 I(2,2)*Ao2 I(2,3)*
      Ao2; I(3,1)*Ao2 I(3,2)*Ao2 I(3,3)*Ao2];
25- Q=[B00 B01 B1K_Z B1K_Z B1K_Z BZ; B10 A1 A0 BLK_Z BLK_Z BL1_Z; BL1_Z
      A2 A1 A0 BLK_Z BL1_Z; BL1_Z BLK_Z A2 A1 A0 BL1_Z; BL1_Z BLK_Z BLK_Z
      A2 A1 BN_1N; BZ B1K_Z B1K_Z B1K_Z BNN_1 BNN]
26- end

```

Fonte: o autor.

Conforme o Código 3.2, os parâmetros de entrada do código estão descritas nas linhas de 1 a 11. Dentre eles, destacam-se as sub matrizes formadas por zeros nas linhas 8, 9, 10 e 11, onde as mesmas preenchem os espaços alocados fora das diagonais. A partir da linha 25, se definem as operações de *Kronecker* para a construção das sub matrizes B_{01} , B_{10} , $B_{n,n-1}$,

$B_{n-1,n}$, A_0 e A_2 , via produto de *Kronecker*. Após gerar as sub matrizes que compõem as diagonais e as sub-matrizes que complementam as bordas, obtém-se a gerador infinitesimal Q (Anexo A).

3.3.4 Algoritmo de HUANG-LIANG-GUO

O algoritmo de *HUANG-LIANG-GUO* é dividido em três etapas, a saber:

- 1^a etapa: $h = 0$, $Q = B_{0,0}$ e $\gamma = \alpha$, calcular E . Se $E > P$ não há necessidade de peças de reposição, logo não há necessidade de novas iterações.

- 2^a etapa: seja $h = 1$, $Q = \begin{bmatrix} B_{0,0} & B_{0,1} \\ B_{1,0} & A_1 \end{bmatrix}$ e $\gamma = (\alpha, 0)$, calcular E . Se $E > P$ é necessário uma peça de reposição e o algoritmo termina.

- 3^a etapa: $h = h + 1$, $Q = Q + \begin{bmatrix} 0 & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & A_0 \\ 0 & \cdots & A_2 & A_1 \end{bmatrix}$ e $\gamma = (\alpha, 0, \dots, 0)$, calcular E . Se $E > P$ necessita-se de h peças de reposição e o algoritmo termina. Caso contrário, executa-se novamente a 3^a etapa.

A partir do procedimento descrito pelo algoritmo de HUANG-LIANG-GUO, foi implementado o Código 3.3.

Código 3.3 – Código de Huang-Ling-Guo.

```

1- P=0.65;
2- h=0;
3- Q=T;
4- gama=alfa;
5- E=gama*expm(Q*t)*e
6-   if E>P
7-       fprintf( '\n#_numero_de_pecas_de_reposicao:_%d#\n',h)
8-   else
9-       h=h+1
10-      Q=[B00 B01; B10 A1];
11-      gama=[ alfa ,zeros(1,m^2)];
12-      E=gama*expm(Q*t)*ones(3*(3*h+1),1)
13-      if E>P
14-          fprintf( '\n#_numero_de_pecas_de_reposicao:_%d#\n',h)
15-      else
16-          h=h+1
17-          Q=[B00 B01 B1K_Z; B10 A1 A0; BL1_Z A2 A1];
18-          gama=[ alfa zeros(1,h*m^2)];
19-          E=gama*expm(Q*t)*ones(3*(3*h+1),1)
20-          if E>P
21-              fprintf( '\n#_numero_de_pecas_de_reposicao:_%d#\n
',h)
22-          else
23-              h=h+1;
24-              Q=[B00 B01 B1K_Z B1K_Z; B10 A1 A0 BLK_Z; BL1_Z A2
A1 A0; BL1_Z BLK_Z A2 A1];
25-              gama=[ alfa zeros(1,h.*m^2)];
26-              E=gama*expm(Q.*t)*ones(3*(3*h+1),1);
27-              if E>P
28-                  fprintf( '\n#_numero_de_pecas_de_reposicao
:_%d#\n',h)
29-              end
30-          end
31-      end
32-  end
33-  if E<P
34-      h=h+1;
35-      fprintf( '\n#_Terminou_a_operacao:_%d#\n',h);
36-  end

```

Fonte: o autor.

No que se refere a execução do Código 3.3, é possível observar que o mesmo efetua uma iteração a cada vez que $E < P$, criando um comportamento que simule o acréscimo de peças de reposição a medida que a probabilidade de o sistema permanecer em funcionamento seja inferior ao valor da confiança estipulada. Logo, quanto maior o número de peças

de reposição, maior a probabilidade do sistema permanecer em funcionamento. Sendo assim, para uma confiança de 65% e um tempo de serviço do maquinário de 1500, 2500 e 3000, obtém-se os seguintes valores expostos na Tabela 3.3.

Tabela 3.3 – Valor de E para que o sistema permaneça em funcionamento.

Quantidade de peças	Valor de E com $t = 2000$	Valor de E com $t = 2500$	Valor de E com $t = 3000$
0	0.75%	0.2%	0.05%
1	59.99%	57.48%	55.08%
2	68.59%	68.49%	68.38%
3	-	-	-

Fonte: o autor.

De acordo com a Tabela 3.3, percebe-se que na falta total de peças de reposição o sistema possui uma probabilidade pequena de permanecer em funcionamento para os três valores de t . No entanto, a medida que se aumenta uma peça, os resultados se mostram sensíveis. Como já citado no algoritmo, no momento em que a probabilidade do sistema permanecer em funcionamento supera o valor da confiança, assim, o código finaliza suas iterações e propõe o número mínimos de peças para garantir a confiabilidade desejada do sistema.

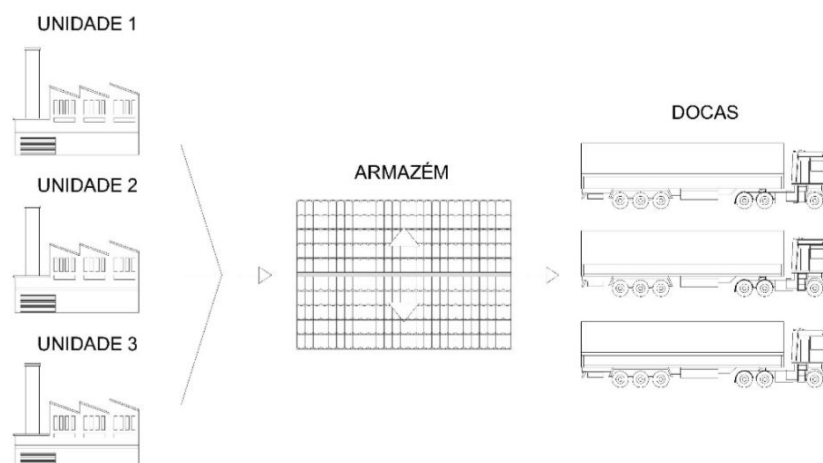
Com o intuito de flexibilizar o Código 3.3 e garantir uma maior aplicabilidade em problemas da indústria, o código em questão foi generalizado no Capítulo 4 por meio de modificações em sua estrutura. Após esta modificação, na Seção 4.1, foi feita uma análise de sensibilidade, que teve como objetivo, testar a confiabilidade e a robustez do algoritmo de HUANG-LIANG-GUO através da perturbação dos parâmetros de entrada.

Capítulo 4

Dimensionamento de equipes de mantenedores

Neste capítulo será apresentado um estudo de caso hipotético onde se busca modelar um sistema de produção com três fábricas, de modo a estipular o número mínimo de mantenedores que cada unidade produtiva deve ter, para que assim, a possibilidade de funcionamento de ambas, seja garantida. Para isto, considere o seguinte problema: Uma fábrica de produtos alimentícios é composta por três unidades de produção independentes que fabricam produtos do tipo A, B e C de forma contínua (ver Figura 4.1). A produção das unidades é responsável pelo abastecimento de um armazém, que por sua vez, tem a responsabilidade de suprir a demanda dos clientes ao longo do ano. Um dos desafios enfrentados pela gerência de produção se trata do dimensionamento do número mínimo de mantenedores que cada unidade produtiva deve ter, de modo a garantir o reparo imediato das máquinas em casos de falhas.

Figura 4.1 – Fábrica de produtos alimentícios.



Fonte: o autor.

Cada unidade produtiva é representada por um processo QBD, sendo que os estados “em funcionamento (L)” ou “em reparo (F)” são representados por um conjunto de transições aleatórias que se repetem ao longo do tempo ao passo que se aumenta ou diminui a quantidade de mantenedores. Logo, cada unidade pode ser representada por um diagrama de transição de Markov conforme Figura 3.9, sendo α e β parâmetros distribuídos exponencialmente.

Para que o armazém tenha um nível de estoque satisfatório, cada unidade deve funcionar com um percentual específico de confiabilidade, sendo de 70%, 76% e 80% para as unidades X , Y e Z respectivamente. Além disso, tem-se como característica para cada planta, os tempos operacionais (T) e de reparo (S) de ordem 3. Ambas representam as três fases operacionais e de reparo do sistema, estipuladas por $\alpha = (1, 0, 0)$ e $\beta = (1, 0, 0)$. Sendo assim, observa-se as matrizes:

$$X : T = \begin{bmatrix} -0.0030 & 0.0030 & 0 \\ 0 & -0.023 & 0.023 \\ 0 & 0 & -0.030 \end{bmatrix} \text{ e } S = \begin{bmatrix} -0.040 & 0.040 & 0 \\ 0.03 & -0.12 & 0.09 \\ 0.007 & 0 & -0.12 \end{bmatrix} \quad (4.1)$$

$$Y : T = \begin{bmatrix} -0.035 & 0.035 & 0 \\ 0 & -0.9 & 0.9 \\ 0 & 0 & -0.0035 \end{bmatrix} \text{ e } S = \begin{bmatrix} -0.95 & 0.95 & 0 \\ 0.01 & -0.19 & 0.18 \\ 0.00 & 0 & -0.03 \end{bmatrix} \quad (4.2)$$

$$Z : T = \begin{bmatrix} -0.0032 & 0.0032 & 0 \\ 0 & -0.025 & 0.025 \\ 0 & 0 & -0.0032 \end{bmatrix} \text{ e } S = \begin{bmatrix} -0.06 & 0.06 & 0 \\ 0.05 & -0.16 & 0.11 \\ 0.008 & 0 & -0.14 \end{bmatrix} \quad (4.3)$$

As matrizes T e S , por características do processo QBD, devem cumprir os seguintes requisitos: ordem m (3×3); não singularidade e; elementos das diagonais negativos e fora das diagonais todos positivos (SALES; MORALES, 2011). A utilização de matrizes não singulares se dá pelo fato de que para o desenvolvimento do gerador infinitesimal e das respectivas operações matriciais se faz necessário que as mesmas possuam determinantes não nulos, caso o contrário, pode não haver soluções para o problema.

Sendo assim, para que a probabilidade de funcionamento (E) de cada unidade seja encontrada, necessita-se primeiramente, calcular um gerador infinitesimal específico para cada unidade, já que os mesmos são capazes de representar o comportamento aleatório ao longo do tempo. Covenientemente, os geradores infinitesimais foram descritos através dos

produtos de Kronecker, conforme a equação 4.4.

$$Q(X, Y, Z) = \begin{bmatrix} B_{00} & T_{0\alpha} \otimes \beta & 0 & & \\ I \otimes S_0 & T \otimes I + I \otimes S & T_{0\alpha} \otimes I & 0 & \\ 0 & IS_0 \otimes \beta & T \otimes I + I \otimes S & 0 & \\ 0 & 0 & IS_0 \otimes \beta & 0 & S_{0\alpha} \otimes \beta \\ 0 & 0 & 0 & T_0 \otimes i & S \end{bmatrix} \quad (4.4)$$

Todavia, por meio do Código 4.1, tornou-se possível calcular os geradores para as respectivas unidades. Ressalta-se que o produto de Kronecker pode ser representado pela função *kron*, já inclusa em Matlab. Sendo assim, pode-se observar os geradores infinitesimais (Q_n) das unidades X , Y e Z nos respectivos anexos B , C e D .

Código 4.1 – Geradores infinitesimais.

```

1-      B01 = kron(To*alfa , betta) ;
2-      B10 = kron(I , So) ;
3-      A1 = kron(T, I) + kron(I , S) ;
4-      A0 = kron(To*alfa , I) ;
5-      A2 = kron(I , So*betta) ;
6-      BN_1N= kron(To, I) ;
7-      BNN_1= kron(So*alfa , betta) ;
8-      BNN=S;
9-      Qn=[B00 B01 B1K_Z B1K_Z B1K_Z BZ; B10 A1 A0 BLK_Z BLK_Z BL1_Z;
      BL1_Z A2 A1 A0 BLK_Z BL1_Z; BL1_Z BLK_Z A2 A1 A0 BL1_Z; BL1_Z BLK_Z
      BLK_Z A2 A1 BN_1N; BZ B1K_Z B1K_Z B1K_Z BNN_1 BNN]
```

Fonte: o autor.

A partir das matrizes Q_n , obtém-se um conjunto de sub matrizes que, conforme explicitado na Figura 3.6 na seção 3.3, permitem que um estado específico seja analisado. Assim, uma unidade de produção pode ser analisada em termos de sua probabilidade de funcionamento, caso tenha zero ou dois mantenedores, por exemplo. Esta situação pode ser melhor compreendida analisando a Figura 4.2.

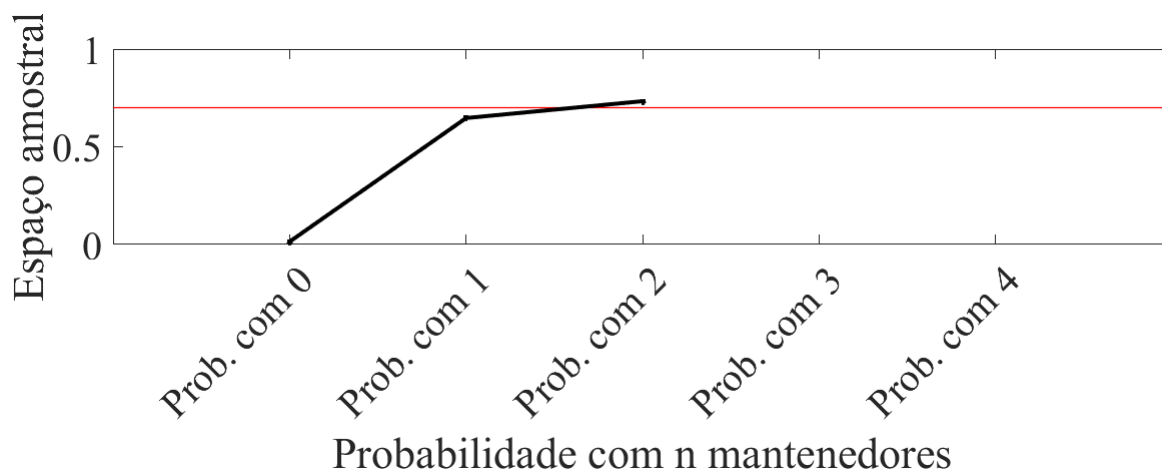
Figura 4.2 – Matriz de transição.

	0 mant.	1 mant.	2 mant.	3 mant.	4 mant.
0 mant.	B00	B01	B1K_Z	B1K_Z	B1K_Z
1 mant.	B10	A1	A0	BLK_Z	BLK_Z
2 mant.	BL1_Z	A2	A1	A0	BLK_Z
3 mant.	BL1_Z	BLK_Z	A2	A1	A0
4 mant.	BL1_Z	BLK_Z	BLK_Z	A2	A1

Fonte: o autor.

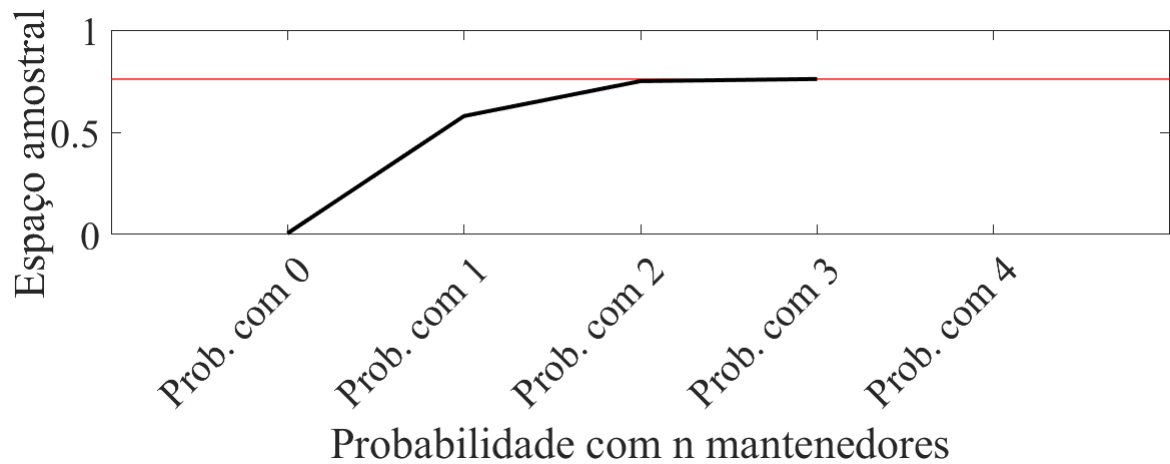
A Figura 4.2 demonstra o conjunto de sub matrizes correlacionadas com os respectivos números de mantenedores (mant), formando assim, estados específicos no que se refere às probabilidades de transição. Logo, o algoritmo de HUANG-LIANG-GUO referenciado na seção 3.3.2 é aplicado por meio do código apresentado no Apêndice E para a resolução do problema do dimensionamento dos mantenedores. Com isto, obtém-se as Figuras 4.3, 4.4 e 4.5, que representam, respectivamente, o número mínimo de mantenedores que as unidades X, Y e Z devem ter, dada a confiabilidade preestabelecida pela gerência de produção.

Figura 4.3 – Número mínimo de mantenedores - Unidade X.



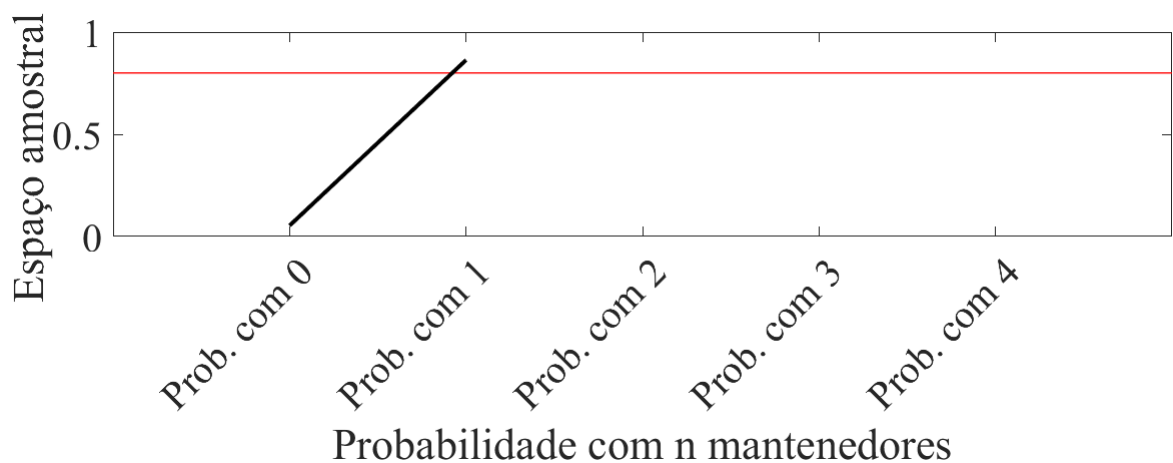
Fonte: Elaborado pelo autor.

Figura 4.4 – Número mínimo de mantenedores - Unidade Y.



Fonte: Elaborado pelo autor.

Figura 4.5 – Número mínimo de mantenedores - Unidade Z.



Fonte: Elaborado pelo autor.

Portanto, observa-se por meio das Figuras 4.3, 4.4 e 4.5, que a unidade X deve operar com 2 mantenedores para obter uma probabilidade de permanecer em funcionamento maior ou igual a 70%. Já a unidade Y, com 3 mantenedores, possui praticamente o mesmo valor da confiabilidade requerida, que é de 76%. Por fim, a unidade Z, com apenas um mantenedor, propicia uma probabilidade de permanência de 86,32%, cerca de 6% acima do estipulado.

Uma das questões cruciais em problemas relacionados a cadeias de Markov, está no fato do entendimento dos parâmetros que compõe determinados modelos, haja vista a volatilidade que tais parâmetros podem assumir quando aplicados a sistemas dinâmicos, como é o caso dos biológicos, físicos e sociais. Perturbar um sistema permite entender como os pa-

râmetros determinam as propriedades da cadeia (SENETA, 1993; MEYER, 1991; CASWELL, 2013). Portanto, o presente estudo avaliou os resultados obtidos por meio de uma análise de sensibilidade, que objetiva descrever o efeito sobre a probabilidade de funcionamento de uma unidade de produção, quando o gerador infinitesimal Q_n é perturbado.

4.1 Análise de Sensibilidade do Algoritmo HUANG-LIANG-GUO

A análise de sensibilidade foi dividida em dois momentos. No primeiro, avaliou-se o comportamento de E_0 , E_1 , E_2 , E_3 e E_4 (E_n – Probabilidade de funcionamento com n mantenedores) quando se varia a matriz de tempo de reparo (S) e o tempo de funcionamento de produção (t). No segundo, analisou-se apenas ao tempo de produção (t), com o intuito de ver o comportamento de E_n ao longo do tempo.

Para análise inicial dos parâmetros de S e t , manteve-se como padrão a matriz de tempo operacional (T), sendo uma escolha feita por conveniência, tendo em vista que se tem como premissa que a matriz de tempo de reparo tem maior possibilidade de variar, haja vista que a detecção de uma falha, assim como o seu reparo, varia de acordo com a complexidade da falha.

O parâmetro t , variou de forma aleatória em um intervalo de 10.000 a 100.000 unidades de tempo, enquanto que a matriz S , variou por meio de um comportamento pseudo-aleatório, já que a mesma deve ser formada levando em consideração as restrições de não singularidade, diagonais negativas, todos elementos fora das diagonais positivos e soma das linhas iguais a zero. Tanto a matriz S , quanto o parâmetro t , foram iterados 10.000 vezes. Para a geração das matrizes pseudoaleatórias S , considere o Código 4.2.

Código 4.2 – Matriz pseudoaleatória S.

```

1- function MS=Matrizes_SENS(S)
2-  A=-1+(1+1)*rand(1);
3-  B=-1+(1+1)*rand(1);
4-  C=0;
5-  M=[A B C];
6-  while ~(-0.001<sum(M) && sum(M)<0.001 && A<0)
7-    A=-1+(1+1)*rand(1);
8-    B=-1+(1+1)*rand(1); C=0;
9-    M=[A B C];
10-  end
11-  M=[A B C];
12-  Mt=M;
13-  A1=-1+(1+1)*rand(1);
14-  B1=-1+(1+1)*rand(1);
15-  C1=-1+(1+1)*rand(1);
16-  M1=[A1 B1 C1];
17-  while ~(-0.001<sum(M1) && sum(M1)<0.001 && B1<0 && A1>0 && C1>0)
18-    A1=-1+(1+1)*rand(1);
19-    B1=-1+(1+1)*rand(1);
20-    C1=-1+(1+1)*rand(1);
21-    M1=[A1 B1 C1];
22-  end
23-  M1=[A1 B1 C1];
24-  Mt1=M1;
25-  e=0;
26-  f=0;
27-  g=-1+(1+1)*rand(1);
28-  qq=[e f g];
29-  while ~(g<0)
30-    e=0;
31-    f=0;
32-    g=-1+(1+1)*rand(1);
33-    qq=[e f g];
34-  end
35-  qq=[e f g];
36-  Q=[M; M1;qq];
37-  MS=Q
38- end

```

Fonte: o autor.

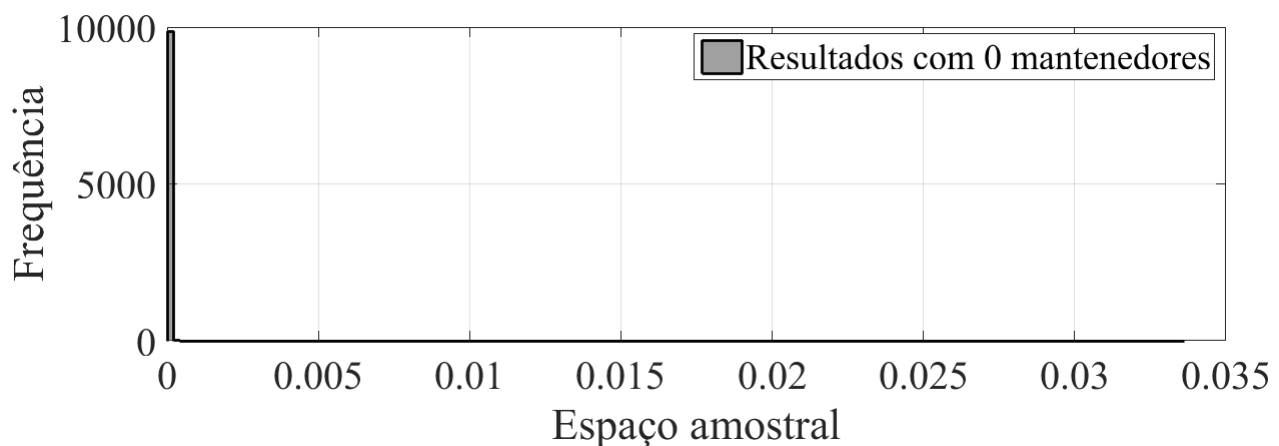
Para o desenvolvimento do Código 4.2, criou-se primeiramente uma função, que por sua vez, foi executada em uma *main function* (função global), onde serviu como entrada para o código que representa o algoritmo, situado no Apêndice E. Para a formação da matriz, criou-se cada variável com uma característica independente, que posteriormente, fo-

ram alocadas em um vetor de 3 posições. O vetor só foi consolidado, quando as condições de restrições foram atendidas, o que por consequência, culminou na formação da matriz pseudoaleatória S de ordem 3×3 .

Executando o código, pode-se observar os histogramas representados pelas Figuras 4.6, 4.7 (a), (b), (c) e (d), que correlacionam as frequências de ocorrência de um resultado E_n , com as respectivas faixas de probabilidade no eixo x , sendo um espaço amostral alocado em um intervalo de zero a um. Logo se pode inferir a possibilidade de ocorrência de um resultado dada a quantidade de mantenedores estipulada.

Na Figura 4.6 pode-se notar que a possibilidade de uma unidade permanecer em funcionamento sem contar sequer com um mantenedor é zero, haja vista que dos 10.000 pontos gerados, 100% estão na faixa de zero, o que permite dizer que a chance de permanecer em funcionamento ao longo do tempo é praticamente nula.

Figura 4.6 – Probabilidade de funcionamento E_0 .



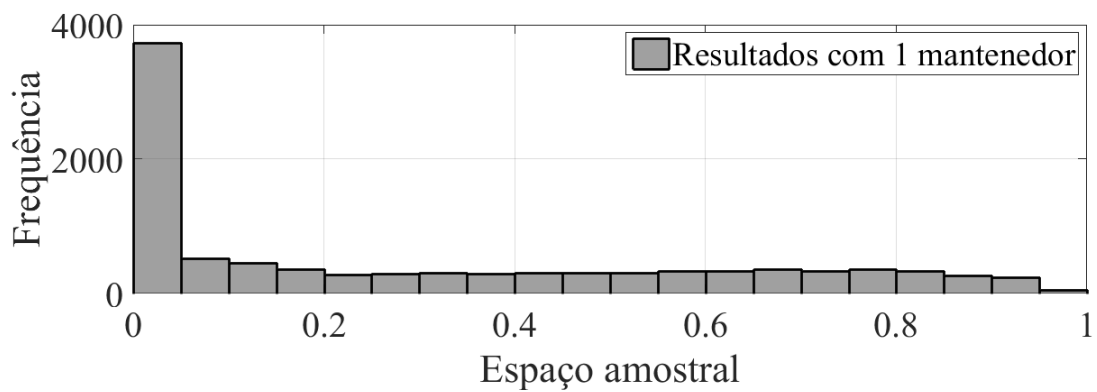
Fonte: o autor.

No entanto, percebe-se que a medida que uma unidade de produção conta com no mínimo um mantenedor, os resultados assumem outra característica de distribuição, gerando situações de permanência de estado (de funcionamento). Na Figura 4.7(a), com um mantenedor, os valores E_1 estão em grande parte nas faixas de 0 a 0.1 e de 0.8 a 1, ocupando os extremos do espaço amostral, o que faz com que a permanência do sistema se torne não confiável, tendo em vista que variando os dados de entrada, os resultados podem ser bons ou ruins em proporções similares.

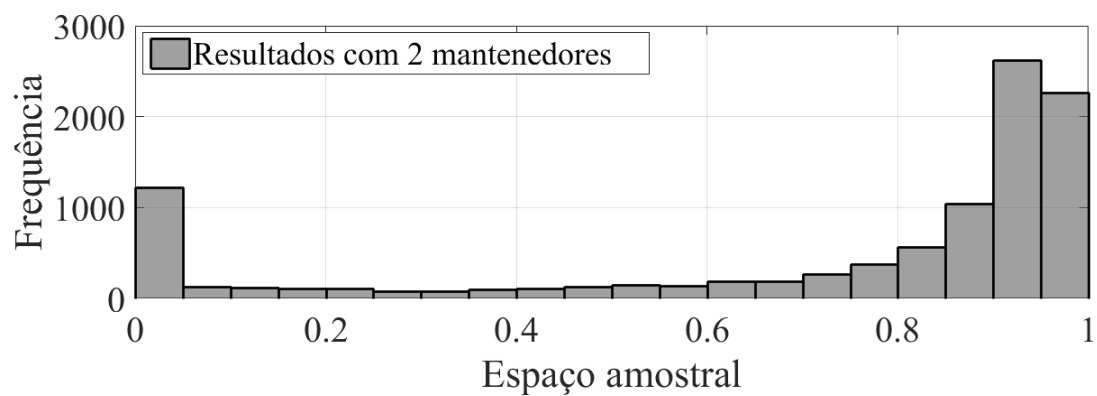
Contudo, ao passo que se acrescentam 2, 3 e 4 mantenedores, os valores E_n migram gradativamente para as faixas próximas a 1, ou seja, aproximadamente 100% de permanência de estado de funcionamento, diminuindo os valores relativos às ocorrências ligadas às faixas próximas de zero.

Figura 4.7 – Possibilidade de permanecer em funcionamento com mantenedores.

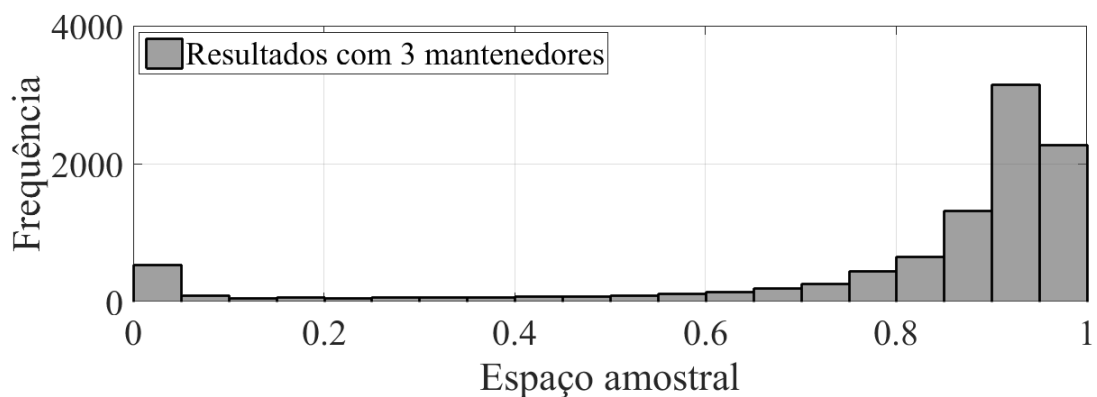
(a) Probabilidade de funcionamento E1.



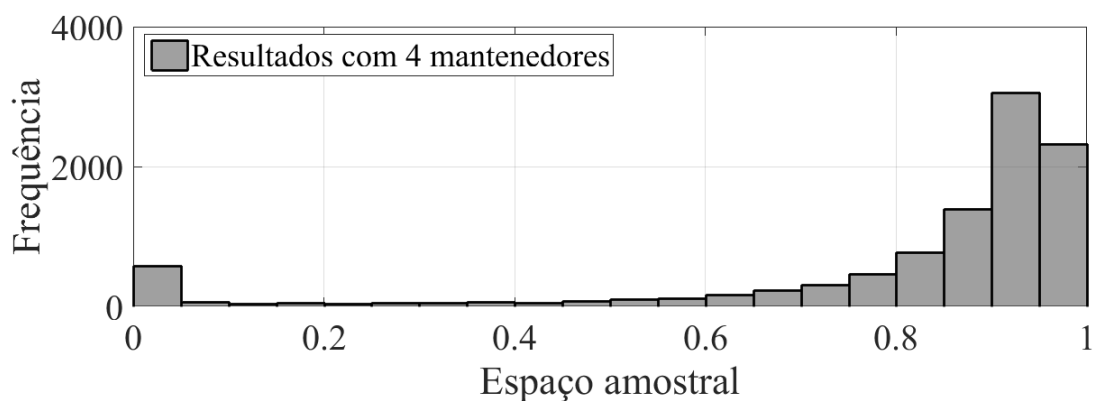
(b) Probabilidade de funcionamento E2.



(c) Probabilidade de funcionamento E3.



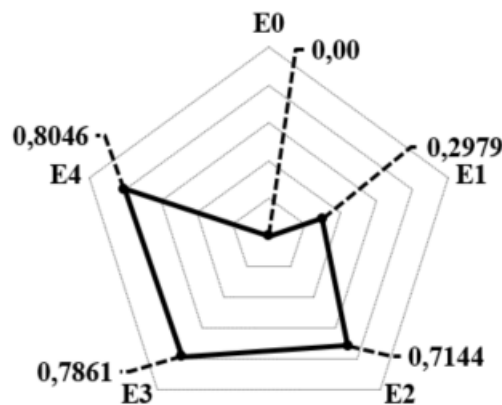
(d) Probabilidade de funcionamento E4.



Fonte: Elaborado pelo autor.

O aumento da probabilidade de uma unidade permanecer em funcionamento quando aumentado o número de mantenedores, também pode ser evidenciado por meio das médias dos valores de E_n , conforme evidenciado na Figura 4.8. Nas condições em que se pode perceber que de três para quatro mantenedores, a média é similar, o que pode ser explicado por Ost (2001), onde o mesmo afirma que o processo QBD a partir do estado 1 apresenta um processo de repetição de estrutura, conforme já exposto na Figura 3.5, e que o gerador infinitesimal é representado por oito sub matrizes em toda sua extensão. Sendo assim, as sub matrizes que governam as transições a partir do estado três, tendem a se repetir.

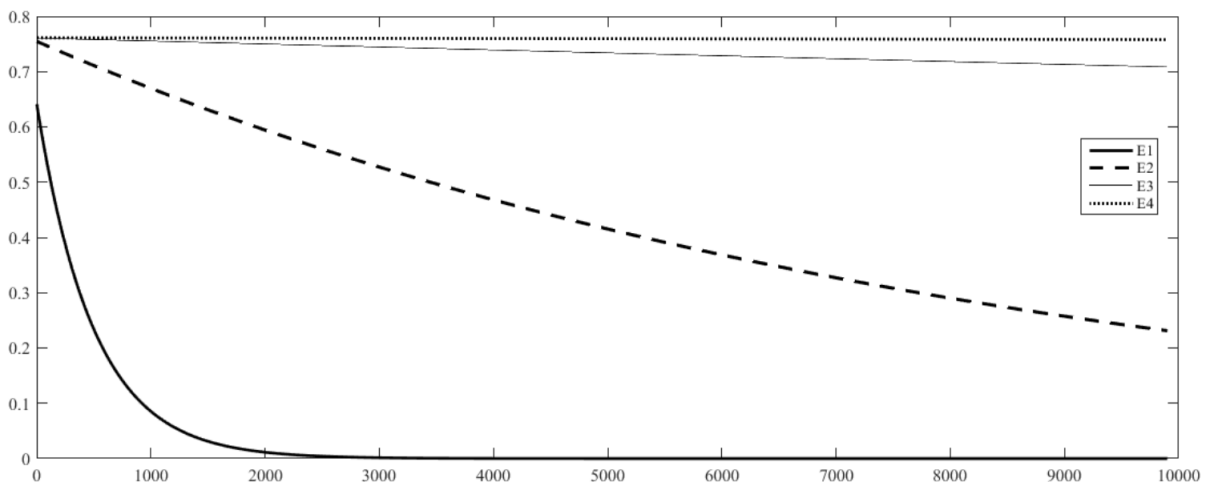
Figura 4.8 – Médias das probabilidades E_n .



Fonte: o autor.

Em relação à variação de apenas t , manteve-se como default as matrizes S e T . Além disso, em vez de ter um comportamento aleatório, como anteriormente, t variou de maneira crescente, o que permite avaliar uma unidade de produção ao longo do tempo e visualizar até quando o sistema é confiável. Para isso, considere a Figura 4.9.

Figura 4.9 – Variando t de forma crescente.



Fonte: o autor.

A Figura 4.9 demonstra uma relação inversa de proporcionalidade, haja vista que, à medida que se aumenta o tempo de trabalho, diminui-se a probabilidade de uma unidade permanecer em funcionamento. Não obstante, nota-se que uma unidade de produção como apenas um mantenedor (E1) tem um comportamento extremo em relação à uma unidade com 4 mantenedores, ao passo que a confiabilidade de E1 diminui drasticamente ao longo do tempo. Já para E4, existe uma probabilidade alta de permanência, mesmo quando “estressado” o valor de t .

Capítulo 5

CONCLUSÕES E TRABALHOS FUTUROS

Diante da dinâmica e da eminente complexidade dos sistemas de produção, atender a demanda não se trata de uma tarefa trivial. Diante disto, o presente trabalho, propôs inicialmente, uma abordagem básica referente à aplicação de teoria das filas em problemas de produção, e a partir disto, abordou modelos mais flexíveis e robustos como são os modelos markovianos, que nesta situação, foram discutidos em um âmbito de confiabilidade e dimensionamento de equipes.

O problema do dimensionamento de equipes de trabalho, caso solucionado de forma empírica, pode ocasionar situações negativas, como o excedente de mão de obra ou a falta da mesma, que por vezes, compromete as estratégias de atendimento da demanda de uma organização. Sendo assim, diversas técnicas podem ser empregadas no planejamento da força de trabalho, desde as abordagens clássicas, como as elaboradas por Erlang e Jackson, ou até mesmo, métodos relacionados aos modelos probabilísticos, como os markovianos.

O uso de modelos markovianos, por meio do método de QBD, possibilita a modelagem de sistemas que possuem características distintas dentro de um mesmo estado. Assim, o processo QBD proporciona ao ambiente estudado a possibilidade de representar situações de estado de funcionamento ou estado de reparo, que alinhados ao algoritmo de HUANG-LIANG-GUO possibilita determinar quantos mantenedores uma unidade de produção deve ter de modo a garantir o funcionamento ao longo do tempo.

Portanto, diante do primeiro objetivo, foram modeladas as três unidades de produção X, Y e Z e realizado o dimensionamento mínimo de mantenedores que cada unidade deve ter, de modo a garantir a confiança preestabelecia. Após avaliação, a unidade X com dois mantenedores proporciona 70% de probabilidade de permanecer em funcionamento, a unidade Y com três proporciona 76%, e por fim, a unidade Y, com apenas um mantenedor possibilita 80% de chance de permanecer em funcionamento.

O Segundo objetivo do trabalho se tratou da análise de sensibilidade dos parâmetros de entrada, onde notou-se que ao perturbar o gerador infinitesimal por meio da variação aleatória e pseudoaleatória das variáveis, os valores de probabilidade de funcionamento tendem a se aproximar de 100%, ao passo que se acrescenta um mantenedor. No entanto, nota-se que quando se acrescenta o quarto mantenedor, existe pouca variação no sistema, fato que fica evidenciado quando comparado os valores médios de probabilidade.

Todavia, quando se estressa o sistema por meio do crescimento da variável aleatória, a confiabilidade dos resultados tende a diminuir, ao passo que com um mantenedor, a probabilidade de funcionamento cai consideravelmente. Em contrapartida, com quatro mantenedores a permanência de estado de funcionamento tende a ser longínqua, tendo em vista que de 1.000 a 10.000 unidades de tempo, a variação foi próxima de zero.

Como oportunidade de trabalhos futuros, recomenda-se que o mesmo sistema seja modelado, no entanto, levando em consideração aspectos de dependência entre as unidades de produção, tratadas nesta ocasião, como independentes.

Além disso, pode-se também: implementar o modelo em uma situação real; adicionar elementos de subjetividade à lógica nebulosa e; acrescentar elementos de testes de sobrevivência ao modelo desenvolvido.

REFERÊNCIAS

A.GÓMEZ-CORRAL; GARCÍA, M. Maximum queue lengths during a fixed time interval in the m/m/c retrial queue. *Applied Mathematics and Computation*, USA, v. 235, p. 124–136, 2014. Citado na página 40.

AL-SEEDY, R. O. *et al.* Transient solution of the m/m/c queue with balking and reneging. *Computers & Mathematics with Applications*, USA, v. 57, p. 1280–1285, 2009. Citado na página 40.

ARBOS, L. C.; SANTOS, J. F.; SANCHEZ, C. V. The operations-time chart: A graphical tool to evaluate the performance of production systems—from batch-and-queue to lean manufacturing. *Computers & Industrial Engineering*, Elsevier, Espanha, v. 61, n. 3, p. 663–675, 2011. Citado na página 47.

ARENALES, M. *et al.* *Pesquisa Operacional*. 1. ed. Brasil: Campus Elsevier, 2007. 544 p. ISBN 8535214542. Citado 4 vezes nas páginas 27, 28, 32 e 33.

BAEK, J. W. *et al.* Exact time-dependent solutions for the m/d/1 queue. *Operations Research Letters*, v. 44, p. 692–695, 2016. Citado na página 39.

BAUMANN, H.; SANDMANN, W. Multi-server tandem queue with markovian arrival process, phase-type service times, and finite buffers. *European Journal of Operational Research*, Elsevier, v. 256, n. 1, p. 187–195, 2017. Citado na página 26.

BITRAN, G. R.; DASU, S. A review of open queueing network models of manufacturing systems. *Queueing Systems*, USA, v. 12, p. 95–133, 1992. Citado na página 43.

BUYUKKARAMIKLI, N. C.; BERTRAND, J. W. M.; VANOIJEN, H. P. G. Periodic capacity management under a lead-time performance constraint. *OR Spectrum*, Netherlands, v. 35, n. 1, p. 221–249, 2011. Citado na página 44.

CASWELL, H. Sensitivity analysis of discrete markov chains via matrix calculus. *Linear Algebra and its applications*, Elsevier, n. 438, p. 1727–1745, 2013. Citado na página 74.

CHWIF, L.; MEDINA, A. *Modelagem e simulação de eventos discretos: Teoria & aplicações*. 4. ed. Brasil: Elsevier, 2015. 294 p. ISBN 9788535279320. Citado 8 vezes nas páginas 28, 29, 31, 33, 35, 37, 38 e 40.

DOWNING, D.; CLARK, J. *Estatística Aplicada*. 3. ed. Brasil: Saraiva, 2011. 368 p. ISBN 8502104160. Citado 2 vezes nas páginas 29 e 53.

FARDIS, F.; ZANDI, A.; GHEZAVATI, V. Stochastic extension of cellular manufacturing systems: a queuing-based analysis. *Journal of Industrial Engineering International*, Springer, v. 9, n. 1, p. 20, 2013. Citado 2 vezes nas páginas 43 e 47.

FERNANDES, F. C. F.; GODINHO FILHO, M. *Planejamento e Controle da Produção: dos fundamentos ao essencial*. 1. ed. Brasil: Editora Atlas, 2010. 296 p. ISBN 9788522458714. Citado 2 vezes nas páginas 40 e 41.

FREITAS FILHO, P. J. *Introdução a Modelagem de Simulação de Sistemas: com aplicações em arena*. 1. ed. Brasil: Visual Books, 2001. 322 p. Citado na página 30.

GALLAGER, R. G. *Stochastic Processes: Theory for applications*. USA: Cambridge University Press, 2013. 553 p. ISBN 978-1-107-03975-9. Citado 3 vezes nas páginas 25, 47 e 49.

GANDHI, A. *et al.* Exact analysis of the m/m/k/setup class of markov chains via recursive renewal reward. *Queueing Systems*, Springer, EUA, v. 77, p. 177–209, 2014. Citado na página 47.

GHEZAVATI, V. R.; SAIDI-MEHRABAD, M. An efficient hybrid self-learning method for stochastic cellular manufacturing problem: A queuing-based analysis. *Expert Systems with Applications*, Iran, v. 38, p. 1326–1335, 2011. Citado 2 vezes nas páginas 43 e 47.

HERAGU, S. S.; SRINIVASAN, M. Analysis of manufacturing systems via single-class, semi-open queuing networks. *International Journal of Production Research*, Taylor & Francis, v. 49, n. 2, p. 295–319, 2011. Citado na página 44.

HUANG, Z.; LIANG, L.; GUO, B. A general repairable spare part demand model based on quasi birth and death process. *ACTA AUTOMATICA SINICA*, Editorial Office of Acta Automatica Sinica, v. 32, n. 2, p. 200–206, 2006. Citado na página 61.

KAMPEN, N. G. *Stochastic Processes in Physics and Chemistry*. 3. ed. [S.l.]: Elsevier, 2006. Citado 2 vezes nas páginas 25 e 47.

KIM, W. S.; LIM, D. E. Analysis of overtaking in m/m/c queues. *Computers & Industrial Engineering*, v. 101, p. 177–183, 2016. Citado na página 39.

KLEINROCK, L. *Queueing Systems*. 1. ed. USA: Wiley-Interscience, 1975. 417 p. ISBN 0471491101. Citado 6 vezes nas páginas 25, 34, 36, 39, 49 e 50.

LEE, D.; LI, S. Transient analysis of multi-server queues with markov-modulated poisson arrivals and overload control. *Performance Evaluation*, USA, v. 16, p. 49–66, 1992. Citado na página 39.

LEEUWAARDEN, J. S. H. V.; WINANDS, E. M. M. Quasi-birth-and-death processes with an explicit rate matrix. *Stochastic Models*, Taylor & Francis Online, v. 22, n. 1, p. 77–98, 2006. Citado na página 26.

MARSUDI, M. Application of queuing theory in analyzing the use of production capacity. *International Journal of Integrated Engineering*, Tunisia, v. 2, p. 19–31, 2010. ISSN 2229-838X. Citado 2 vezes nas páginas 25 e 47.

MARSUDI, M.; SHAFEEK, H. The application of queuing theory in multi-stage production line. In: *Proceedings of the 2014 International Conference on Industrial Engineering and Operations Management Bali, Indonesia*. Bali, Indonesia: [s.n.], 2014. p. 668–675. Citado na página 43.

- MEYER, C. D. Sensitivity of the stationary distribution of a markov chain. *Society for Industrial and Applied Mathematics*, SIAM, v. 15, n. 3, p. 715–728, 1991. Citado na página 74.
- MOREIRA, D. A. *Pesquisa Operacional: Curso introdutório*. 2. ed. Brasil: Cengage, 2010. 376 p. ISBN 8522110514. Citado 2 vezes nas páginas 27 e 29.
- OSOGAMI, T. *Analysis of Multi-server Systems via Dimensionality Reduction of Markov Chains*. 252 f. Tese (Doutorado) — School of Computer Science, Carnegie Mellon University, Pittsburgh, 2005. Citado 3 vezes nas páginas 25, 44 e 55.
- OSOGAMI, T.; HARCHOL-BALTER, M.; SCHELLER-WOLF, A. Analysis of cycle stealing with switching times and thresholds. *Performance Evaluation*, Elsevier, v. 61, n. 4, p. 347–369, 2005. Citado 2 vezes nas páginas 25 e 45.
- OST, A. *Performance of Communication Systems: A model-based approach with matrix-geometric methods*. [S.l.]: Springer-Verlag Berlin Heidelberg, 2001. 290 p. ISBN 978-3-662-04421-6. Citado 4 vezes nas páginas 26, 56, 58 e 78.
- PÉREZ-OCÓN, R.; MONTORO-CARZOLA, D. A multiple system governed by a quasi-birth-and-death process. *Reliability Engineering and System Safety*, Elsevier, v. 84, p. 187–196, 2004. Citado na página 62.
- PERLMAN, Y.; ELALOUE, A.; BODINGER, E. Dynamic repair priority for a transfer line with a finite buffer. *Journal of Manufacturing Systems*, Israel, v. 33, n. 1, p. 16–26, 2013. Citado na página 44.
- POLIZELLI, D. L.; RUIZ JÚNIOR, J. Gestão de conhecimento e taylorismo: Contribuições para a evolução da administração. *Revista interação*, Brasil, V, n. 2, p. 26–53, 2011. Citado na página 25.
- PRADO, D. *Teoria das Filas e Simulação*. 4. ed. Brasil: Falconi, 2009. 127 p. ISBN 9788598254401. Citado 6 vezes nas páginas 27, 28, 29, 30, 31 e 32.
- PRASAD, K.; B. USHA. A comparison between m/m/1 and m/d/1 queuing models to vehicular traffic at kanyakumari district. *IOSR Journal of Mathematics*, v. 11, n. 2, p. 13–15, 2015. Citado na página 38.
- RAMAN, D.; NAGALINGAM, S. V.; GURD, B. W. A genetic algorithm and queuing theory based methodology for facilities layout problem. *International Journal of Production Research*, Taylor & Francis, v. 47, n. 20, p. 5611–5635, 2009. Citado na página 44.
- RESNICK, S. I. *Adventures in Stochastic Processes*. 1. ed. [S.l.]: Birkhäuser Basel, 1992. 626 p. ISBN 978-0-8176-3591-6. Citado 2 vezes nas páginas 49 e 50.
- ROTHKOPF, M. H.; RECH, P. Perspectives on queues - combining queues is not always beneficial. *Operations Research*, USA, v. 35, p. 906–909, 1987. Citado na página 40.
- ROY, D. Semi-open queuing networks: a review of stochastic models, solution methods and new research areas. *International Journal of Production Research*, India, v. 54, p. 1735–1752, 2015. Citado na página 43.
- SALES, M. V. da S.; MORALES, G. Aplicações do processo de quase-nascimento-e-morte em sistemas de produção. In: XLIII SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL, 2011, Ubatuba. *Anais...* São Paulo, 2011. p. 156–167. Citado 5 vezes nas páginas 26, 48, 56, 62 e 70.

SALGADO, E. G. Análise da aplicação do mapeamento do fluxo de valor na identificação de desperdícios do processo de desenvolvimento de produtos. *Gest. Prod.*, v. 16, p. 344–356, 2009. Citado 2 vezes nas páginas 42 e 43.

SARKAR, A.; MUKHOPADHYAY, A. R.; GHOSH, S. K. Productivity improvement by reduction of idle time through application of queuing theory. *OPSEARCH*, India, v. 52, p. 195–211, 2015. Citado na página 43.

SENETA, E. Sensitivity of finite markov chains under perturbation. *Statistics & Probability Letters*, Elsevier, v. 17, n. 2, p. 163–168, 1993. Citado na página 74.

SHANTHIKUMAR, J. G.; BUZACOTT, J. A. Open queueing network models of dynamic job shops. *International Journal Of Production Research*, Taylor & Francis, v. 19, n. 3, p. 255–266, 1981. Citado na página 45.

SILVA, C. R. N. d.; MORABITO, R. Aplicação de modelos de redes de filas abertas no planejamento do sistema job-shop de uma planta metal-mecânica. *Gest. Prod.*, scielo, v. 14, p. 393 – 410, 2007. ISSN 0104-530X. Citado 2 vezes nas páginas 25 e 46.

SLACK, N.; CHAMBERS, S.; JOHNSTON, R. *Administração da Produção*. 3. ed. [S.l.]: Editora Atlas, 2010. 703 p. ISBN 9788522453535. Citado 2 vezes nas páginas 41 e 42.

SMITH, J. M. M/g/c/k performance models in manufacturing and service systems. *Asia-Pacific Journal of Operational Research*, Singapore, v. 25, p. 531–561, 2008. Citado na página 43.

TAHA, H. A. *Pesquisa Operacional*. 8. ed. Brasil: Pearson, 2008. 353 p. ISBN 8576051508. Citado 4 vezes nas páginas 27, 28, 29 e 32.

TIER, R. *Markov Models: Understanding markov models and unsupervised machine learning in python with real-world applications*. [S.l.]: CreateSpace Independent Publishing Platform, 2017. 66 p. ISBN 9781544657288. Citado 2 vezes nas páginas 25 e 50.

VOTTO, R. G.; FERNANDES, F. C. F. Produção enxuta e teoria das restrições: proposta de um método para implantação conjunta na indústria de bens de capital sob encomenda. *Gest. Prod.*, v. 21, p. 45–63, 2013. Citado 2 vezes nas páginas 42 e 43.

WU, K.; MCGINNIS, L. Performance evaluation for general queueing networks in manufacturing systems: Characterizing the trade-off between queue time and utilization. *European Journal of Operational Research*, Elsevier, Singapura, v. 221, n. 2, p. 328–339, 2012. Citado na página 47.

YANG, D.-Y.; CHIANG, Y.-C.; TSOU, C.-S. Cost analysis of a finite capacity queue with server breakdowns and threshold-based recovery policy. *Journal of Manufacturing Systems*, Elsevier, Taiwan, v. 32, n. 1, p. 174–179, 2013. Citado na página 47.

ZHANG, H.-Y.; CHEN, Q.-X.; MAO, N. System performance analysis of flexible flow shop with match processing constraint. *International Journal of Production Research*, Taylor & Francis Online, v. 54, n. 20, p. 6052–6070, 2016. Citado na página 25.

APÊNDICE A

Resultado Gerador Infinitesimal

Figura A.2 – Gerador infinitesimal Q - Página 2

01/09/17 16:33		MATLAB Command Window						2 of 6
0	0	0	0	0	0	0	0	0 ✓
0	0	0	0	0	0	0	0	0 ✓
0	0	0	0	0	0	0	0	0 ✓
0	0	0	0	0	0	0	0	0 ✓
0	0	0	0	0	0	0	0	0 ✓
0	0	0	0	0	0	0	0	0 ✓
0	0	0	0	0	0	0	0	0 ✓
0	0	0	0	0	0	0	0	0 ✓
0	0	0	0	0	0	0	0	0 ✓
0	0	0	0	0	0	0	0	0 ✓
0	0	0	0	0	0	0	0	0 ✓
0	0	0	0	0	0	0	0	0 ✓
0	0	0	0	0	0	0	0	0 ✓
0	0	0	0	0	0	0	0	0 ✓
0	0	0	0	0	0	0	0	0 ✓
0	0	0	0	0	0	0	0	0 ✓
0	0	0	0	0	0	0	0	0 ✓
0	0	0	0	0	0	0	0	0 ✓
0	0	0	0	0	0	0	0	0 ✓
0	0	0	0	0	0	0	0	0 ✓
Columns 13 through 24								
0	0	0	0	0	0	0	0	0 ✓
0	0	0	0	0	0	0	0	0 ✓
0	0	0	0	0	0	0	0	0 ✓
0	0	0	0	0	0	0	0	0 ✓
0	0	0	0	0	0	0	0	0 ✓
0	0	0	0	0	0	0	0	0 ✓
0	0	0	0	0	0	0	0	0 ✓
0	0	0	0	0	0	0	0	0 ✓
0	0	0	0	0	0	0	0	0 ✓
0	0	0	0	0	0	0	0	0 ✓
0	0	0	0	0	0	0	0	0 ✓

Fonte: o autor

Figura A.5 – Gerador infinitesimal Q - Página 5

01/09/17 16:33		MATLAB Command Window						5 of 6	
0	0	0	0	0	0	0	0	0	↙
0	0.0027	0	0	0	0	0	0	0	↙
0	0	0.0027	0	0	0	0	0	0	↙
0	0	0	0.0027	0	0	0	0	0	↙
0	0	0	0	0.0080	0	0	0	0	↙
0	-0.0280	0.0200	0	0.0080	0	0	0	0	↙
0	0.0100	-0.0880	0.0700	0	0.0080	0	0	0	↙
0	0.0050	0	-0.1080	0	0	0.0080	0	0	↙
0	0	0	0	-0.0488	0.0200	0	-0.0288	0	↙
0	0	0	0	0.0100	-0.1088	0.0700	0	-0.0288	↙
0	0	0	0	0.0050	0	-0.1288	0	0	↙
-0.0288	0	0	0	0	0	0	-0.0227	0.0200	↙
0	0.0027	0	0	0	0	0	0.0100	-0.0827	↙
0.0700	0	0.0027	0	0	0	0	0.0050	0	↙
-0.1027	0	0	0.0027	0	0	0	0	0	↙
0	-0.0280	0.0200	0	0	0	0	0	0	↙
0	0	0	0	0	0	0	0	0	↙
0	0.0100	-0.0880	0.0700	0	0	0	0	0	↙
0	-0.0950	0	0	0	0	0	0	0	↙
0	0.0050	0	-0.1080	0	0	0	0	0	↙
0	0	0	0	0	0	0	0	0	↙
0	0	0	0	0	0	0	0	0	↙
0	0	0	0	-0.0950	0	0	0	0	↙
0	0	0	0	0	0	0	0	0	↙
0	0	0	0	0	0	0	0	0	↙
0	0	0	0	0	0	0	-0.0950	0	↙
0	0	0	0	0	0	0	0	0	↙
Columns 37 through 42									
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

Fonte: o autor

Figura A.6 – Gerador infinitesimal Q - Página 6

```

01/09/17 16:33      MATLAB Command Window      6 of 6

```

```

0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0.0080      0      0      0      0      0
0      0.0080      0      0      0      0
0      0      0.0080      0      0      0
-0.0488      0.0200      0      -0.0288      0      0
0.0100      -0.1088      0.0700      0      -0.0288      0
0.0050      0      -0.1288      0      0      -0.0288
0      0      0      -0.0200      0.0200      0
0      0      0      0.0100      -0.0800      0.0700
0      0      0      0.0050      0      -0.1000

```

>>

Fonte: o autor

APÊNDICE B

Resultado Gerador X

Figura B.1 – Gerador infinitesimal da unidade X - Página 1

MATLAB Command Window

Page 1

$$QX =$$

Columns 1 through 12

[illegible]

Fonte: o autor

Figura B.2 – Gerador infinitesimal da unidade X - Página 2

[illegible]

Fonte: o autor

Figura B.3 – Gerador infinitesimal da unidade X - Página 3

[illegible]

Fonte: o autor

Figura B.4 – Gerador infinitesimal da unidade X - Página 4

[illegible]

Fonte: o autor

Figura B.5 – Gerador infinitesimal da unidade X - Página 5

MATLAB Command Window

Page 5

```

0      0      0      0      0      0      0      0      0 ✓
0      0.0030      0      0      0      0      0      0      0 ✓
0      0      0.0030      0      0      0      0      0      0 ✓
0      0      0      0.0030      0      0      0      0      0 ✓
0      0      0      0      0.0230      0      0      0      0 ✓
0      -0.0630      0.0400      0      0.0230      0      0      0      0 ✓
0      0      0      0      0      0.0230      0      0      0 ✓
0      0.0300      -0.1430      0.0900      0      0.0230      0      0      0 ✓
0      0      0      0      0      0      0.0230      0      0 ✓
0      0.0070      0      -0.1430      0      0      0.0230      0      0 ✓
0      0      0      0      -0.0700      0.0400      0      -0.0300      0 ✓
0      0      0      0      0.0300      -0.1500      0.0900      0      -0.0300 ✓
0      0      0      0      0.0070      0      -0.1500      0      0 ✓
-0.0300      0      0      0      0      0      -0.1500      0      0 ✓
0      0      0      0      0      0      0      -0.0430      0.0400 ✓
0      0.0030      0      0      0      0      0      0.0300      -0.1230 ✓
0.0900      0      0.0030      0      0      0      0      0.0070      0 ✓
0      0      0      0      0      0      0      0.0070      0 ✓
-0.1230      0      0      0.0030      0      0      0      0      0 ✓
0      0      0      0      0      0      0      0      0 ✓
0      -0.0630      0.0400      0      0      0      0      0      0 ✓
0      0      0      0      0      0      0      0      0 ✓
0      0.0300      -0.1430      0.0900      0      0      0      0      0 ✓
-0.1130      0      0      0      0      0      0      0      0 ✓
0      0.0070      0      -0.1430      0      0      0      0      0 ✓
0      0      0      0      0      0      0      0      0 ✓
0      0      0      0      0      0      0      0      0 ✓
0      0      0      0      0      0      0      0      0 ✓
0      0      0      0      -0.1130      0      0      0      0 ✓
0      0      0      0      0      0      0      0      0 ✓
0      0      0      0      0      0      0      0      0 ✓
0      0      0      0      0      0      0      0      0 ✓
0      0      0      0      0      0      0      0      0 ✓
0      0      0      0      0      0      0      -0.1130      0 ✓
0      0      0      0      0      0      0      0      0 ✓

```

Columns 37 through 42

```

0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0

```

Fonte: o autor

Figura B.6 – Gerador infinitesimal da unidade X - Página 6

MATLAB Command Window

Page 6

```

0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0.0230      0      0      0      0      0
0      0.0230      0      0      0      0
0      0      0.0230      0      0      0
-0.0700      0.0400      0      -0.0300      0      0
0.0300      -0.1500      0.0900      0      -0.0300      0
0.0070      0      -0.1500      0      0      -0.0300
0      0      0      -0.0400      0.0400      0
0      0      0      0.0300      -0.1200      0.0900
0      0      0      0.0070      0      -0.1200

```

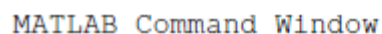
>>

Fonte: o autor

APÊNDICE C

Resultado Gerador Y

Figura C.1 – Gerador infinitesimal da unidade Y - Página 1



Page 1

 $QY =$

Columns 1 through 12

[illegible]

Fonte: o autor

Figura C.2 – Gerador infinitesimal da unidade Y - Página 2

[illegible]

Fonte: o autor

Figura C.3 – Gerador infinitesimal da unidade Y- Página 3

[illegible]

Fonte: o autor

[illegible]

Fonte: o autor

Figura C.5 – Gerador infinitesimal da unidade Y - Página 5

MATLAB Command Window

Page 5

```

0      0      0      0      0      0      0      0      0      0 ✓
0      0.0350      0      0      0      0      0      0      0 ✓
0      0      0.0350      0      0      0      0      0      0 ✓
0      0      0      0.0350      0      0      0      0      0 ✓
0      0      0      0      0.9000      0      0      0      0 ✓
0      -1.8500      0.9500      0      0.9000      0      0      0      0 ✓
0      0.0100      -1.0900      0.1800      0      0.9000      0      0      0 ✓
0      0      0      -0.9300      0      0      0.9000      0      0 ✓
0      0      0      0      -0.9535      0.9500      0      -0.0035      0 ✓
0      0      0      0      0.0100      -0.1935      0.1800      0      -0.0035 ✓
0      0      0      0      0      0      -0.0335      0      0 ✓
-0.0035      0      0      0      0      0      0      -0.9850      0.9500 ✓
0      0.0350      0      0      0      0      0      0.0100      -0.2250 ✓
0.1800      0      0.0350      0      0      0      0      0      0 ✓
-0.0650      0      0      0.0350      0      0      0      0      0 ✓
0      0      0      0      0      0      0      0      0 ✓
0      -1.8500      0.9500      0      0      0      0      0      0 ✓
0      0.0100      -1.0900      0.1800      0      0      0      0      0 ✓
-0.0300      0      0      0      0      0      0      0      0 ✓
0      0      0      -0.9300      0      0      0      0      0 ✓
0      0      0      0      0      0      0      0      0 ✓
0      0      0      0      0      0      0      0      0 ✓
0      0      0      0      -0.0300      0      0      0      0 ✓
0      0      0      0      0      0      0      0      0 ✓
0      0      0      0      0      0      0      0      0 ✓
0      0      0      0      0      0      0      -0.0300      0 ✓
0      0      0      0      0      0      0      0      0

```

Columns 37 through 42

```

0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0

```

Fonte: o autor

Figura C.6 – Gerador infinitesimal da unidade Y - Página 6

MATLAB Command Window

Page 6

```

0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0.9000      0      0      0      0      0
0      0.9000      0      0      0      0
0      0      0.9000      0      0      0
-0.9535      0.9500      0      -0.0035      0      0
0.0100      -0.1935      0.1800      0      -0.0035      0
0      0      -0.0335      0      0      -0.0035
0      0      0      -0.9500      0.9500      0
0      0      0      0.0100      -0.1900      0.1800
0      0      0      0      0      -0.0300
>>

```

Fonte: o autor

APÊNDICE D

Resultado Gerador Z

Figura D.1 – Gerador infinitesimal da unidade Z - Página 1

MATLAB Command Window

Page 1

QZ =

Columns 1 through 12

[illegible]

Fonte: o autor

Figura D.2 – Gerador infinitesimal da unidade Z - Página 2

[illegible]

Fonte: o autor

Figura D.3 – Gerador infinitesimal da unidade Z- Página 3

[illegible]

Fonte: o autor

Figura D.4 – Gerador infinitesimal da unidade Z - Página 4

[illegible]

Fonte: o autor

Figura D.5 – Gerador infinitesimal da unidade Z - Página 5

MATLAB Command Window

Page 5

```

0      0      0      0
0      0.0032      0      0      0      0      0      0      0 ✓
0      0      0      0      0      0      0      0      0 ✓
0      0      0.0032      0      0      0      0      0      0 ✓
0      0      0      0.0032      0      0      0      0      0 ✓
0      0      0      0      0      0.0250      0      0      0 ✓
0      -0.0850      0.0600      0      0.0250      0      0      0      0 ✓
0      0      0      0      0      0.0250      0      0      0 ✓
0      0.0500      -0.1850      0.1100      0      0.0250      0      0      0 ✓
0      0      0      0      0      0      0.0250      0      0 ✓
0      0.0080      0      -0.1650      0      0      0.0250      0      0 ✓
0      0      0      0      -0.0632      0.0600      0      -0.0032      0 ✓
0      0      0      0      0.0500      -0.1632      0.1100      0      -0.0032 ✓
0      0      0      0      0.0080      0      -0.1432      0      0 ✓
-0.0032      0      0      0      0      0      0      0      0 ✓
0      0      0      0      0      0      0      -0.0632      0.0600 ✓
0      0.0032      0      0      0      0      0      0.0500      -0.1632 ✓
0      0      0      0.0032      0      0      0      0.0080      0 ✓
0.1100      0      0      0      0      0      0      0      0 ✓
-0.1432      0      0      0.0032      0      0      0      0      0 ✓
0      0      0      0      0      0      0      0      0 ✓
0      -0.0850      0.0600      0      0      0      0      0      0 ✓
0      0      0      0      0      0      0      0      0 ✓
0      0.0500      -0.1850      0.1100      0      0      0      0      0 ✓
-0.1320      0      0      0      0      0      0      0      0 ✓
0      0.0080      0      -0.1650      0      0      0      0      0 ✓
0      0      0      0      0      0      0      0      0 ✓
0      0      0      0      0      0      0      0      0 ✓
0      0      0      0      -0.1320      0      0      0      0 ✓
0      0      0      0      0      0      0      0      0 ✓
0      0      0      0      0      0      0      0      0 ✓
0      0      0      0      0      0      0      0      0 ✓
0      0      0      0      0      0      0      -0.1320      0 ✓
0      0      0      0      0      0      0      0      0 ✓

```

Columns 37 through 42

```

0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0

```

Fonte: o autor

Figura D.6 – Gerador infinitesimal da unidade Z - Página 6

MATLAB Command Window

Page 6

```

0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0      0      0      0      0      0
0.0250      0      0      0      0      0
0      0.0250      0      0      0      0
0      0      0.0250      0      0      0
-0.0632      0.0600      0      -0.0032      0      0
0.0500      -0.1632      0.1100      0      -0.0032      0
0.0080      0      -0.1432      0      0      -0.0032
0      0      0      -0.0600      0.0600      0
0      0      0      0.0500      -0.1600      0.1100
0      0      0      0.0080      0      -0.1400

>>

```

Fonte: o autor

APÊNDICE E

**Aplicação do algoritmo de
HUANG-LIANG-GUO**

Código E.1 – Algoritmo de HUANG-LIANG-GUO - Parte 1.

```

clc
clear all
close all
1. m=3;
2. T = input ( ' digite_a_matriz_de_tempo_operacional_T: ')
3. S = input ( ' digite_a_matriz_que_governa_o_tempo_de_reparo_da_avaria_
    S: ')
4. t = input ( ' digite_o_valor_da_quantidade_de_interacoes_ou_tempo_de_
    servico )
5. _alfa=[1_0_0];
6. _betta=[1_0_0];
7. _e=ones(m,1);
8. _I=eye(m,m);
9. _BZ=zeros(m,m);
10. _B1K_Z=zeros(m,m^2);
11. _BLK_Z=zeros(m^2,m^2);
12. _BL1_Z=zeros(m^2,m);
13. _To=T*_e;
14. _So=S*_e;
15. _Bo1=To*_alfa;
16. _Ao2=So*_betta;
17. _Bn=So*_alfa;
18. _B00=T;
19. _B01=_kron(To*_alfa, _betta);
20. _B10=_kron(_I, _So);
21. _A1=_kron(T, _I)+_kron(_I, _S);
22. _A0=_kron(To*_alfa, _I);
23. _A2=_kron(_I, _So*_betta);
24. _BN_1N=_kron(To, _I);
25. _BNN_1=_kron(So*_alfa, _betta);
26. _BNN=S;
27. _QZ=[B00 _B01 _B1K_Z _B1K_Z _B1K_Z _BZ; _B10 _A1 _A0 _BLK_Z _BLK_Z _BL1_Z; _
    BL1_Z _A2 _A1 _A0 _BLK_Z _BL1_Z; _BL1_Z _BLK_Z _A2 _A1 _A0 _BL1_Z; _BL1_Z _BLK_Z _
    BLK_Z _A2 _A1 _BN_1N; _BZ _B1K_Z _B1K_Z _B1K_Z _BNN_1 _BNN]
28. __confiabilidade=__input(' Digite a confiabilidade p: ')_%\n
29. __mantenedores=0;
30. __QZ=T;
31. __gama=alfa;
32. __E0=gama*expm(QZ*t)*ones(m,1);
33. __ProbE_0=[E0];
34. __ax=__gca;
35. __ax.XTick=__[1_2_3_4_5];
36. __ax.XTickLabels=__{'Prob. com 0', 'Prob._com_1', 'Prob._com_2', 'Prob._
    com_3', 'Prob._com_4'};
37. __ax.XTickLabelRotation = 45;
38. bar(ProbE_0, 'k');
39. ylabel('P(E)')
40. axis([ 0 6 0 1])
41. hold on

```

Fonte: o autor.

Código E.2 – Algoritmo de HUANG-LIANG-GUO - Parte 2.

```

42. b0=confiabilidade; b1=0;
43. x= linspace(0,6, 50);
44. y= b0-b1*x;
45. plot(x,y,'linewidth',1, 'color', 'red')
46. if E0>confiabilidade
47.     return
48. end
49. mantenedores=mantenedores+1;
50. QZ=[B00 B01; B10 A1];
51. gama=[ alfa zeros(1,m^2)];
52. E1=gama*expm(QZ*t)*ones(3*(3*mantenedores+1),1);
53. ProbE_0=[E0, E1];
54. ax = gca;
55. ax.XTick = [1 2 3 4 5];
56. ax.XTickLabels = {'Prob._com_0', 'Prob._com_1', 'Prob._com_2', 'Prob._com_3', 'Prob._com_4'};
57. ax.XTickLabelRotation = 45;
58. bar(ProbE_0, 'k');
59. ylabel('P(E)')
60. axis([ 0 6 0 1])
61. hold on
62. b0=confiabilidade; b1=0;
63. x= linspace(0,6, 50);
64. y= b0-b1*x;
65. plot(x,y,'linewidth',1, 'color', 'red')
66. if E1>confiabilidade
67.     return
68. end
69. mantenedores=mantenedores+1;
70. QZ=[B00 B01 B1K_Z; B10 A1 A0; BL1_Z A2 A1];
71. gama=[ alfa zeros(1,mantenedores*m^2)];
72. E2=gama*expm(QZ*t)*ones(3*(3*mantenedores+1),1);
73. ProbE_0=[E0, E1, E2];
74. ax = gca;
75. ax.XTick = [1 2 3 4 5];
76. ax.XTickLabels = {'Prob._com_0', 'Prob._com_1', 'Prob._com_2', 'Prob._com_3', 'Prob._com_4'};
77. ax.XTickLabelRotation = 45;
78. bar(ProbE_0, 'k');
79. ylabel('P(E)')
80. axis([ 0 6 0 1])
81. hold on
82. b0=confiabilidade; b1=0;
83. x= linspace(0,6, 50);
84. y= b0-b1*x;
85. plot(x,y,'linewidth',1, 'color', 'red')

```

Fonte: o autor.

Código E.3 – Algoritmo de HUANG-LIANG-GUO - Parte 3.

```

86. if E2>confiabilidade
87. return
88. end
89. mantenedores=mantenedores+1;
90. QZ=[B00 B01 B1K_Z B1K_Z; B10 A1 A0 BLK_Z; BL1_Z A2 A1 A0; BL1_Z
    BLK_Z A2 A1];
91. gama=[alfa zeros(1,mantenedores*m^2)];
92. E3=gama*expm(QZ*t)*ones(3*(3*mantenedores+1),1);
93. ProbE_0=[E0, E1, E2, E3];
94. ax = gca;
95. ax.XTick = [1 2 3 4 5];
96. ax.XTickLabels = {'Prob._com_0', 'Prob._com_1', 'Prob._com_2', 'Prob
    ._com_3', 'Prob._com_4'};
97. ax.XTickLabelRotation = 45;
98. bar(ProbE_0, 'k');
99. ylabel('P(E)')
100. axis([ 0 6 0 1])
101. hold on
102. b0=confiabilidade; b1=0;
103. x= linspace(0,6, 50);
104. y= b0-b1*x;
105. plot(x,y,'linewidth',1, 'color', 'red')
106. if E3>confiabilidade
107.     return
108. end
109. mantenedores=mantenedores+1;
110. QZ=[B00 B01 B1K_Z B1K_Z B1K_Z; B10 A1 A0 BLK_Z BLK_Z; BL1_Z A2 A1
    A0 BLK_Z; BL1_Z BLK_Z A2 A1 A0; BL1_Z BLK_Z BLK_Z A2 A1];
    gama=[alfa zeros(1,mantenedores*m^2)];
111. E4=gama*expm(QZ*t)*ones(3*(3*mantenedores+1),1);
112. ProbE_0=[E0, E1, E2, E3, E4];
113. ax = gca;
114. ax.XTick = [1 2 3 4 5];
115. ax.XTickLabels = {'Prob._com_0', 'Prob._com_1', 'Prob._com_2', 'Prob
    ._com_3', 'Prob._com_4'};
116. ax.XTickLabelRotation = 45;
117. bar(ProbE_0, 'k');
118. ylabel('P(E)')
119. axis([ 0 6 0 1])
120. hold on
121. b0=confiabilidade; b1=0;
122. x= linspace(0,6, 50);
123. y= b0-b1*x;
124. plot(x,y,'linewidth',1, 'color', 'red')
125. hold on

```

Fonte: o autor.
