



# SIMULAÇÃO POR EVENTOS DISCRETOS

## Teoria & prática

**Professor:** João Flávio de Freitas Almeida

**Site:** <http://joaoflavio.com.br/cursos/po/simulacao>



## Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Modelos simbólicos . . . . .	1
1.2	Modelos matemáticos ou analíticos . . . . .	1
1.3	Modelos de simulação . . . . .	2
1.3.1	Vantagens . . . . .	2
1.3.2	Desvantagens . . . . .	3
1.3.3	Riscos . . . . .	3
1.4	Simulação por eventos discretos . . . . .	3
1.4.1	Abordagem por eventos . . . . .	4
1.4.2	Abordagem por varredura de atividades . . . . .	4
1.4.3	Abordagem por três fases . . . . .	5
1.4.4	Abordagem por processos . . . . .	5
1.5	Condução de um estudo de simulação . . . . .	5
1.6	Aplicações da simulação . . . . .	6
<b>2</b>	<b>Geração de números e variáveis aleatórios</b>	<b>8</b>
2.1	Geradores de números aleatórios . . . . .	8
2.1.1	Geradores congruenciais de números aleatórios . . . . .	8
2.1.2	Geradores congruenciais multiplicativos . . . . .	9
2.1.3	Geradores recursivos de números aleatórios . . . . .	9
2.2	Teste de aleatoriedade . . . . .	9
2.3	Geradores de variáveis aleatórias . . . . .	10
2.3.1	Geradores de variáveis aleatórias contínuas . . . . .	10
2.3.2	Geradores de variáveis aleatórias discretas . . . . .	10
<b>3</b>	<b>Modelagem dos dados de entrada</b>	<b>11</b>
3.1	Coleta dos dados . . . . .	12
3.2	Tratamento dos dados . . . . .	13
3.3	Identificação da distribuição dos dados . . . . .	13
3.4	Quandos os dados não estão disponíveis . . . . .	14
<b>4</b>	<b>Elaboração do modelo conceitual</b>	<b>15</b>
4.1	Abstração e modelos abstratos . . . . .	15
4.2	Diagramas de Ciclos de Atividades . . . . .	15
4.3	Visão de processos . . . . .	16
4.4	Especificação de modelos de simulação . . . . .	16
<b>5</b>	<b>Elaboração do modelo computacional</b>	<b>18</b>
5.1	Linguagens: de programação vs. de simulação vs. simulador . . . . .	18
5.2	Implementação do método de três fases . . . . .	18
5.3	Abordagem por eventos . . . . .	20
5.4	Abordagem por varredura de atividades . . . . .	21
5.5	Abordagem por processos . . . . .	21
<b>6</b>	<b>Verificação e validação dos modelos</b>	<b>23</b>
6.1	Técnicas de verificação . . . . .	23
6.2	Técnicas de validação . . . . .	24



<b>7</b>	<b>Dimensionando aquecimento e replicações</b>	<b>27</b>
7.1	Medidas de desempenho . . . . .	27
7.2	Replicação e rodada . . . . .	27
7.3	Regime transitório vs. regime permanente . . . . .	27
7.4	Simulação terminal vs. Simulação não terminal . . . . .	28
<b>8</b>	<b>Análise dos resultados de uma simulação</b>	<b>33</b>
8.1	Sistemas Terminais . . . . .	34
8.2	Sistemas Não-Terminais . . . . .	34
8.3	Comparar alternativas . . . . .	37
<b>9</b>	<b>Técnicas de redução de variância</b>	<b>38</b>
<b>10</b>	<b>Planejamento e análise de experimentos de simulação</b>	<b>40</b>
<b>11</b>	<b>Otimização baseada em simulação</b>	<b>42</b>
11.1	Otimização local . . . . .	43
11.2	Otimização Global . . . . .	46
11.3	Problemas multi-objetivos . . . . .	47
<b>12</b>	<b>Elementos de probabilidade e estatística</b>	<b>48</b>
12.1	Teoria de probabilidade . . . . .	48
12.2	Variáveis aleatórias . . . . .	48
12.3	Funções de distribuição . . . . .	48
12.4	Distribuições discretas . . . . .	48
12.5	Distribuições contínuas . . . . .	48
<b>13</b>	<b>Teoria de Filas</b>	<b>49</b>
<b>14</b>	<b>Modelagem visual e <i>softwares</i> de simulação</b>	<b>52</b>
14.1	Modelagem visual em projetos de simulação . . . . .	52
14.2	<i>Softwares</i> de simulação e linguagens de programação . . . . .	52

# 1 Introdução

Referência	Capítulos	Páginas	Assunto
<a href="#">Pidd (2004)</a>	[03.1.2]	[030,035]	Condução de projetos de simulação
<a href="#">Altiok and Melamed (2010)</a>	[01.2.0]	[002,004]	Modelagem analítica x simulação
<a href="#">Law (2007)</a>	[01.2.0]	[003,006]	DES: Tipologia de modelos
<a href="#">Fishman (2001)</a>	[01.1.0]	[005,006]	DES: Definição (estados)
<a href="#">Pidd (2004)</a>	[03.1.2]	[030,035]	DES: Fatiamento do tempo
<a href="#">Pidd (2004)</a>	[06.2.0]	[085,106]	3 Fases (ABC), Eventos ou Processo
<a href="#">Kelton et al. (1998)</a>	[02.7.0]	[042,043]	Etapas de um estudo de simulação
<a href="#">Chwif and Medina (2006)</a>	[01.0.0]	[001,002]	O que não é simulação
<a href="#">Law (2007)</a>	[01.9.0]	[076,078]	Vantagens e desvantagens
<a href="#">Banks and Carson (1984)</a>	[01.2.0]	[004,005]	Vantagens e desvantagens
<a href="#">Banks and Carson (1984)</a>	[1.10.0]	[011,016]	Fluxograma das etapas

Existe um dito popular que diz que "*fazer a pergunta certa já é meia resposta*". Estudos de simulação buscam responder a perguntas sobre operações por meio de uma modelagem computacional, portanto, a intensão principal da modelagem é capturar *o que realmente é importante* no sistema para a finalidade em questão [Chwif and Medina \(2006\)](#). No entanto, o conhecimento técnico, embora seja um pré-requisito aos estudos por simulação, não é suficiente. A capacidade de resolver problemas e analisá-los é mais importante do que o simples uso de uma ferramenta computacional, ou seja, *planejar, executar e completar* um bom estudo por simulação requer mais do que boas habilidades em programação.

Em um projeto de simulação dois fluxos paralelos ocorrem simultaneamente: o projeto técnico de simulação (estruturação do problema, modelagem e a implementação) e o projeto organizacional de simulação, que envolve uma negociação inicial, a definição do projeto, o gerenciamento e controle e a renegociação, em caso de desentendimentos sobre os conceitos ou sobre os verdadeiros objetivos do projeto, o que é frequente [Pidd \(2004\)](#).

## 1.1 Modelos simbólicos

Modelos simbólicos são representados por símbolos gráficos (como *fluxogramas*) e representam um sistema de forma estática. São limitados, pela dificuldade de representação de muitos detalhes do sistema e pela falta de elementos quantitativos, porém são de grande importância, principalmente como ferramenta de comunicação [Chwif and Medina \(2006\)](#).

## 1.2 Modelos matemáticos ou analíticos

Modelos matemáticos podem fornecer soluções analíticas, como por Teoria das Filas ou Programação Linear. A diferença entre a modelagem analítica e a modelagem por simulação reside no método de execução e na natureza de suas soluções. Modelos analíticos são derivados de fórmulas matemáticas para uma representação simplificada da realidade. Seu resultado é a medida de desempenho de interesse. Se, por exemplo, em um sistema de filas M/M/1 a disciplina da fila não for FIFO, a Teoria de Filas não fornece um modelo matemático pronto para este caso.

Modelos de simulação, por outro lado, são implementados por programas de computador e produzem amostras de um histórico de rodadas. Assim, estatísticas são computadas deste histórico e usadas para formar a medida de desempenho de interesse [Altiok and Melamed](#)

(2010). Estes são geralmente usados quando os sistemas possuem muitas regras de operação tornando inviável sua representação por equações analíticas.

### 1.3 Modelos de simulação

Operações podem ser analisados por meio de experimentos reais ou representados por modelos para atender algum interesse particular. Modelos de simulação possuem natureza dinâmica (mudanças de *estado no tempo*), aleatória (usam *variáveis aleatórias*) e são formados por entidades que se interagem de forma lógica para algum fim. São particularmente usados para responder perguntas do tipo: "*o que ocorre se...*" e podem ser classificados em três dimensões Law (2007):

1. *Estáticos vs. dinâmicos*: Modelos estáticos são usados para representar sistemas em que o tempo não desempenha nenhum papel. Modelos de Monte Carlo, usados para avaliar funções matemáticas, modelos financeiros ou elaboração de cenários, são exemplos de modelos estáticos. Por outro lado, modelos dinâmicos representam sistemas que evoluem no tempo, como um sistema de uma fábrica.
2. *Determinísticos vs. estocásticos*: Modelos determinísticos não possuem componentes aleatórios. Exemplos de modelos determinísticos incluem um sistema de equações diferenciais de descrevem uma reação química, ou modelos de programação linear inteira-mista. O resultado destes modelos também não possui componentes aleatórios. A presença de elementos aleatórios gera a necessidade de elaborar modelos estocásticos. A maioria dos sistemas de filas são modelados estocasticamente.
3. *Contínuos vs. discretos*: Em modelos contínuos, os valores das variáveis se alteram de forma gradativa no tempo e são geralmente representados por equações diferenciais, como por exemplo, no crescimento de uma planta, no enchimento de um pneu de carro ou na variação do nível de um tanque de combustível. Eventos discretos, por outro lado, evoluem à medida em que os estados do sistema são alterados e facilmente identificados, como uma parada de trens em estações ou a montagem da base de uma cadeira. A simulação por eventos discretos é o nosso objeto de estudo.

Embora seja uma das ferramentas mais utilizadas no mundo da pesquisa operacional, é preciso destacar **o que não é simulação** Chwif and Medina (2006): simulação *não é* uma bola de cristal (não prevê o futuro), *não é* um modelo matemático (expressão analítica fechada), *não é* otimização (ferramenta descritiva), *não substitui* o pensamento inteligente (no processo de tomada de decisão), *não é* a técnica de último recurso (quando outras técnicas falham) e *não é* uma panaceia (só a problemas bem específicos).

Um vez que o problema definido pode ser modelado por simulação, destaca-se suas vantagens, desvantagens e riscos Banks and Carson (1984) Law (2007):

#### 1.3.1 Vantagens

1. A maioria dos sistemas reais *não* pode ser avaliado *analiticamente* com acurácia. Simulação é a única forma possível.
2. Permite estimar o desempenho de um sistema sob condições de operação *projetadas*.
3. Permite a *comparação* de projetos de sistemas operacionais.
4. Permite o *controle* das condições de experimentos (redução de variância), que seria possível apenas por experimentação.
5. Permite a *compressão do tempo* de operações longas, ou até a *expansão do tempo*.

### 1.3.2 Desvantagens

1. Simulação *estocástica* produz estimativas, e dependem de diversas rodadas. Se as características de um problemas podem ser estimadas por parâmetros exatos, a otimização é preferível.
2. *Softwares* podem ser caros e o desenvolvimento do modelo é demorado.
3. Animações realísticas podem impressionar, mas corre-se o risco do modelo não ser válido (Erro Tipo Zero).

### 1.3.3 Riscos

1. Não definir os objetivos no início do estudo de simulação.
2. Envolver todos do projeto desde o início.
3. Nível inadequado de detalhes.
4. Falha de comunicação com a gerência no decorrer do estudo.
5. Equipe de gestão não entender a simulação.
6. Tratar simulação como programação.
7. Não ter equipe com conhecimento na metodologia de simulação.
8. Não coletar bons dados.
9. Usar *software* de simulação inadequado.
10. Usar *software* de simulação mal documentado.
11. Acreditar que bom *software* de simulação requer pouca competência técnica.
12. Mal uso da animação.
13. Usar distribuições arbitrárias.
14. Analisar resultados de uma replicação apenas e tratar o resultado como "resposta".
15. Não considerar tempo de *warm-up*.
16. Usar medidas de desempenho erradas.

## 1.4 Simulação por eventos discretos

Em um sistema de eventos discretos, um ou mais fenômenos de interesse mudam seu valor, ou *estado*, em pontos discretos (ao invés de continuamente) no tempo Fishman (2001). A ocorrência destes eventos muda o estado do sistema em cada momento. Dessa forma, assumimos que não há mudanças no sistema entre um evento e outro. Mesmo em caso de haver incrementos fixos de avanço no tempo, o que não é muito comum, a evolução do sistema não ocorre de forma contínua no tempo Law (2007).

Na simulação por eventos discretos, o tempo é dividido em pequenas fatias e o estado do sistema é atualizado de acordo com as atividades que ocorrem em cada fatia do tempo. Como nem toda fatia de tempo possui ocorrência de atividade, esta simulação é mais rápida que a simulação contínua.

A técnica do *próximo-evento* possui duas vantagens sobre a técnica de *fatiamiento-do-tempo*. A primeira é que o incremento do tempo é ajustado automaticamente à períodos com alto ou baixo índice de ocorrência de atividades evitando desperdícios ou verificações desnecessárias do estado do modelo. A segunda é que a abordagem do próximo-evento deixa claro onde ocorre mais ou menos eventos. Ela é geral e engloba a técnica do fatiamiento-do-tempo. O contrário não é verdadeiro Pidd (2004). Ao invés de usar somente o paradigma

de programação estruturada baseada em eventos, a simulação por eventos discretos pode ser baseada em eventos, atividades ou em processos.

Os elementos base de uma simulação por eventos discretos são: o estado, o relógio e a lista de eventos. O *estado do evento* é representado por variáveis que representam as propriedades do sistema a ser estudado. O *relógio* mantém o controle da evolução temporal da simulação na unidade de tempo escolhida. A *lista de eventos* é chamada de lista de eventos pendentes no início da simulação. À medida em que o relógio da simulação avança, os eventos são realizados e o estado do sistema é atualizado. Os eventos pendentes são organizados em uma lista de prioridades, ordenados por duração do evento. Independentemente de como são ordenados, os eventos são removidos da lista na ordem cronológica da simulação.

A partir deste momento, a simulação computa as estatísticas do sistema, que quantifica os aspectos de interesse. Em um modelo de simulação, as estatísticas não são derivadas de distribuições de probabilidade, mas de médias de replicações de rodadas do modelo. Para avaliar a qualidade do resultado, intervalos de confiança são construídos. O fim da simulação pode ser determinado por tempo de simulação ou por uma medida estatística. Um dos três princípios governam qualquer *software* de simulação por eventos discretos:

1. Abordagem por eventos;
2. Abordagem por varredura de atividades, ou por três fases (ABC);
3. Abordagem por processos.

#### 1.4.1 Abordagem por eventos

A abordagem por eventos foi popular nos primeiros anos da simulação por eventos discretos. Foi originado no trabalho de Markowitz em 1963 na RAND Corporation Markowitz (1963). Na modelagem baseada em eventos, os elementos de execução e verificação de atividades (B e C) podem ser combinados em uma única rotina de eventos. Assim, o programa executivo (*simulation engine*) possui apenas duas fases:

1. Examine o calendário de eventos e mova o relógio para o momento do próximo evento. Mova a ocorrência de eventos para uma lista de eventos.
2. Fixe a constante do relógio, execute as rotinas dos eventos presentes na lista de eventos e esvazie a lista de eventos.

Este método de simulação é aplicável a problemas pequenos e simples. A partir dos anos 1980, a simulação por eventos perdeu popularidade para a abordagem por três fases (ABC) e por processos, que são mais simples e eficientes para problemas complexos (princípio da parcimônia). Estes podem lidar de forma modular com mudanças de estados e facilitam a alteração ou customização do modelo, o que não ocorre na abordagem por eventos Pidd (2004).

#### 1.4.2 Abordagem por varredura de atividades

O método de varredura de atividades é uma abordagem usada nas primeiras linguagens de simulação. Seu funcionamento é muito simples, porém, ineficiente. Ela, no entanto, deu origem ao método por três fases (ABC). Seu funcionamento possui a mesma estrutura da fase C do método ABC, ou seja, todos os Bs e Cs se tornam atividades e para cada uma delas é preciso avaliar uma condição, o que torna o método improdutivo. O programa executivo (*simulation engine*) possui apenas duas fases:

1. Verifique o evento do calendário para encontrar o momento do próximo evento. Mova o relógio da simulação para este momento.
2. Faça a varredura repetidamente por todas as atividades verificando se a atividade pode ocorrer ou não. Continue a varredura até que não haja atividades para serem executadas naquele momento.

### 1.4.3 Abordagem por três fases

Um refinamento da abordagem baseada em atividades é a simulação de eventos discretos por três fases (ABC). Este método é frequentemente usado por pacotes comerciais de simulação e caracterizado por utilizar de forma eficiente os recursos computacionais. A nomenclatura da etapa B vem de *book-keeping activities* ou *bound to happen*, ou seja, relacionado à execução de atividades programadas para acontecer segundo uma lista de eventos. As atividades C vem de *conditional activities*, ou seja, são avaliadas as atividades (e a liberação das entidades) que dependem de uma condição ser atendida para acontecer, portanto, não dependem do relógio da simulação Pidd (2004).

Nesta abordagem, a primeira fase (A) consiste realizar a varredura do tempo buscando a próxima atividade agendada e evoluir para a fase cronológica seguinte. Assim, é a lista de *eventos-imediatos* é criada. A segunda fase (B) consistem em executar todos os eventos que ocorrem incondicionalmente naquele momento. O programa principal remove os *eventos-imediatos* da lista e os executa sistematicamente conforme sua posição na lista. A fase (C) consiste em executar todos os eventos que ocorrem de forma condicional.

### 1.4.4 Abordagem por processos

A abordagem por processos é uma das mais frequentes utilizada no mundo. Para modelar um sistema usando abordagens baseada em eventos ou atividades, o analista considera o processo de cada classe de entidade os divide em partes fundamentais, seja em eventos independentes ou por conexões entre atividades.

A simulação baseada em processo avalia todo processo, ou sequencia de operações, em que a entidade precisa percorrer. Assim, cada classe de entidade possui seu próprio processo. Um modelo é formado por um conjunto de processos. Há pelo menos um para cada classe de entidade na simulação. Durante a simulação, entidades são criadas como membros destas classes. Elas interagem no processo até serem interrompidas por meio de atrasos incondicionais (previamente determinados) ou condicionais (aguarda uma condição para continuar) Pidd (2004). Cada processo é simulado de forma independente (*thread*) pelo computador. Neste caso, os eventos discretos fariam os outros processos ficarem inativos, ativos ou atualizarem o estado do sistema.

## 1.5 Condução de um estudo de simulação

A resolução de um problema inicia-se na formulação e evolui para a avaliação da metodologia de solução. A metodologia utilizada deve ser a que atende as necessidades ao menor custo Kelton et al. (1998). Para isso, deve-se avaliar o tipo de informação esperada e o nível de detalhamento da mesma. Estudos de simulação devem ser usados quando a avaliação de sistemas complexos não podem ser representados e analisados fielmente por “contas de cabeça”, por planilhas eletrônicas ou por programação matemática ou modelagem analítica (como Teoria de Filas). A condução de um estudo de simulação é formada por três grandes etapas Chwif and Medina (2006):

1. **Concepção ou formulação do modelo:** São analisados os *objetivos*, determinado o *escopo do modelo*, suas *hipóteses*, o *nível de detalhamento*. Nesta fase os *dados de*



*entrada* são coletados (cuidado com GIGO - *garbage in, garbage out*). O modelo deve dirigir a coleta de dados, não o contrário. O resultado desta etapa é representado por um *modelo conceitual* deve ser finalizado e compreendido por todos envolvidos;

2. **Implementação do modelo:** O modelo conceitual é convertido em *modelo computacional*. Nesta etapa são executadas tarefas de *V&V - verificação e validação*. A verificação está relacionado ao *debugging*, ou seja, se o modelo se comporta conforme projetado. Já a validação garante que o modelo se comporta como o sistema real modelado;
3. **Análise dos resultados do modelo:** Nesta etapa se tem o *modelo operacional*, que está pronto para a realização de experimentos. A experimentação e análise consiste em *executar rodadas e coletar estatísticas*. As análises podem ser candidatas, comparativas ou preditivas.
  - (a) *Análise candidata* ocorrem em sistemas com poucas informações. Avalia-se projetos de capacidade.
  - (b) *Análise comparativa* seleciona o melhor projeto de sistema de uma lista de projetos por meio de uma análise mais detalhada.
  - (c) *Análise preditiva* ocorrem em um projeto apenas e possui um grande detalhamento das atividades.

O estudo por simulação é finalizado na etapa de apresentação dos resultados e disseminação do modelo. Nesta etapa, deve-se certificar que as perguntas corretas foram respondidas *de forma clara e simples* por meio de um *sumário executivo*. A documentação complementar e as referências sobre o conteúdo de simulação devem estar disponíveis no relatório.

As etapas de um estudo de simulação podem ser resumidas pelo fluxograma apresentado na Figura 1:

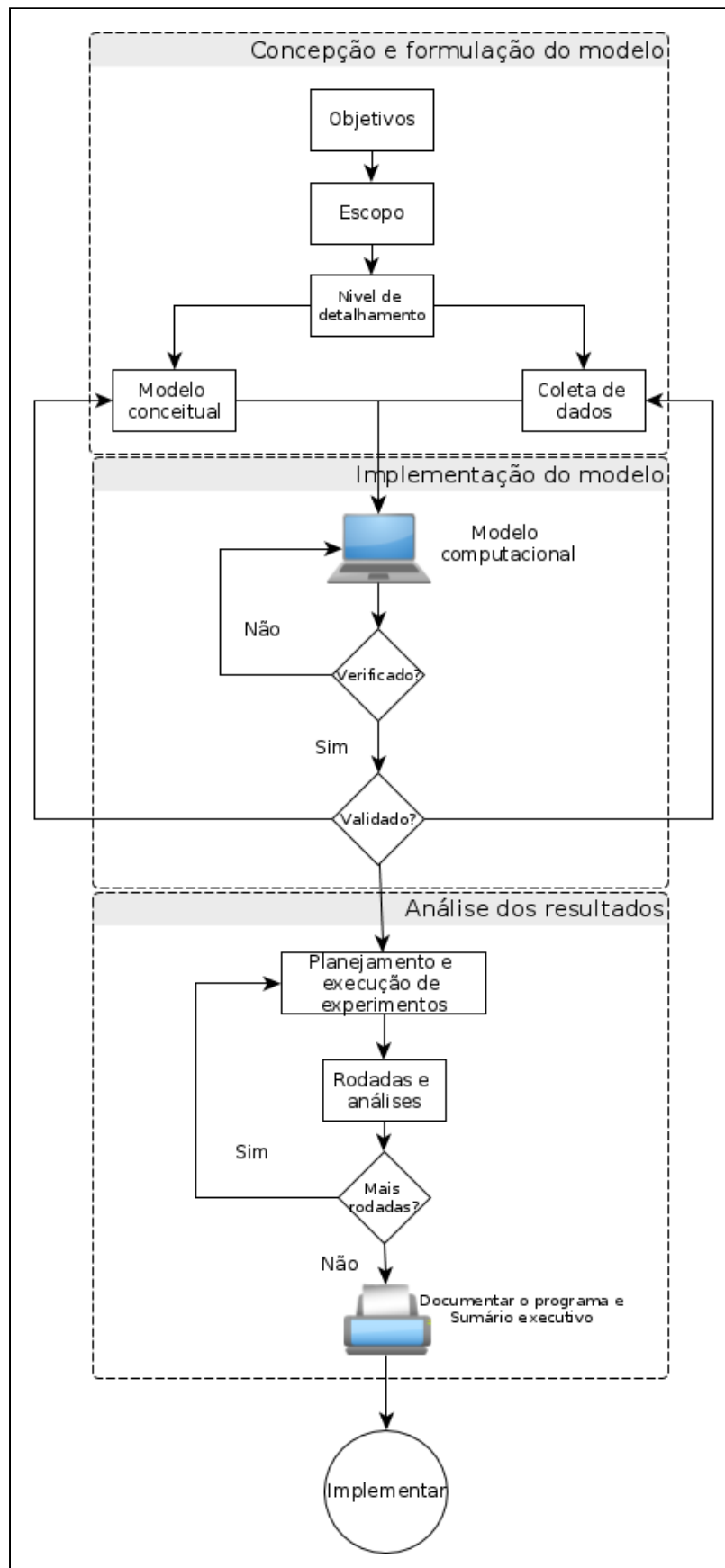
## 1.6 Aplicações da simulação

### Serviços:

1. Aeroportos e portos: Dimensionar postos de *check in*.
2. Bancos: Dimensionar caixas automáticos ou *layout*.
3. Cadeias logísticas: Estoques, transporte.
4. Call centers: Dimensionar equipes.
5. Escritórios: Fluxo de documentos.
6. Hospitais: Dimensionar ambulâncias em emergências.

### Manufatura:

1. Linhas de montagem: Balancear linhas.
2. Células de manufatura: Dimensionar equipes.
3. Programação da produção: Determinar sequências.
4. Estoques: Dimensionar níveis.
5. Kanbans: Determinar fluxo.
6. Logística interna: Determinar fluxo.



**Figura 1:** Metodologia de estudo por simulação. Fonte: Banks and Carson (1984) e Chwif and Medina (2006)

## 2 Geração de números e variáveis aleatórios

Referência	Capítulos	Páginas	Assunto
<a href="#">Chwif and Medina (2006)</a>	[Ap.0.3]	[244,245]	Método do meio-quadrado
<a href="#">Altiok and Melamed (2010)</a>	[04.0.0]	[055,061]	Variáveis e processos aleatórios
<a href="#">Kelton et al. (1998)</a>	[11.0.0]	[471,481]	Números e variáveis aleatórias
<a href="#">Law (2007)</a>	[07, 08]	[389,471]	Números e variáveis aleatórias
<a href="#">Pidd (2004)</a>	[10.2.0]	[179,189]	Números aleatórios e testes
<a href="#">Fishman (2001)</a>	[09.0.0]	[417,447]	Geradores congruenciais (técnico)
<a href="#">Chwif and Medina (2006)</a>	[Ap.0.3]	[247,249]	Geração de variáveis aleatórias
<a href="#">Banks and Carson (1984)</a>	[07.0.0]	[256,286]	Números aleatórios e testes

Números aleatórios são os blocos construtores da simulação. Simuladores precisam gerar variáveis aleatórias de diversos tipos. Isso ocorre por meio de geradores de números pseudo-aleatórios. O uso destes geradores é benéfico, pois permite que a simulação seja re-executada apresentando exatamente o mesmo comportamento.

Os geradores de números aleatórios mais simples são dados, moedas ou sacos com bolas coloridas. Outros geradores envolvem tabelas de números aleatórios, rotação de roletas divididas uniformemente em segmentos, etc. No entanto, estes geradores não são úteis para simulações computacionais de grande porte.

Em 1946, John Von Neumann apresentou o *método do meio-quadrado* para a geração de números aleatórios. A cada iteração, eleva-se o número "semente" ao quadrado e extraem-se seus dígitos do meio. Este novo número é a semente para a iteração seguinte e processo se repete [Chwif and Medina \(2006\)](#). No entanto, o método é pouco eficiente, gerando ciclos em intervalos muito curto. Além disso, toda vez que o número gerado for zero (000), é necessário escolher uma nova semente.

Abordagens modernas utilizam de geradores de números pseudo-aleatórios por meio de fórmulas [Kelton et al. \(1998\)](#) [Law \(2007\)](#) [Altiok and Melamed \(2010\)](#). O método mais popular de geração de números pseudo-aleatórios é chamado de *método da congruência* ou *resíduo* [Chwif and Medina \(2006\)](#).

Estes devem atender as seguintes necessidades [Pidd \(2004\)](#):

1. Variáveis aleatórias devem ser uniformemente distribuídas no intervalo (0,1). Podemos simular qualquer variável aleatória, discreta ou contínua, se temos a geração de variáveis aleatórias  $U(0,1)$  [Kelton et al. \(1998\)](#).
2. Os números gerados devem ser independentes entre si, em termos estatísticos, a sequência de valores não pode ter correlação serial, ou seja, os valores devem ser igualmente prováveis de ocorrerem em qualquer ponto de uma sequência.
3. O tamanho do ciclo, determinado pelo módulo de  $m$ , deve ser o maior possível.
4. Como simulações requerem uma grande quantidade de números aleatórios, a aritmética para sua geração deve ser rápida para garantir eficiência durante a execução.

### 2.1 Geradores de números aleatórios

#### 2.1.1 Geradores congruenciais de números aleatórios

Uma abordagem detalhada pode ser vista em [Fishman \(2001\)](#). Representações matemáticas de fórmulas de geração de números aleatórios são:

$$u_{i+1} = (\pi + u_i)^5 \pmod{1} \quad i = 1, \dots, n \quad (1)$$

onde  $u_0$  é o termo conhecido como *semente* e possui domínio  $0 \leq u_0 \leq 1$ . Fórmulas recursivas, como em (1) são adequadas para o uso em computadores e possuem propriedades que podem ser investigadas matematicamente. Estes geradores são úteis em momentos em que é necessário conhecer a  $u_0$  para re-executar a simulação usando os mesmos números aleatórios e avaliar métodos de redução de variância, apresentados no capítulo 9. Geralmente, a fórmula mais adotada é:

$$x_{i+1} = ax_i + b \pmod{m} \quad i = 1, \dots, n \quad (2)$$

onde  $a$ ,  $b$  e  $m$  são escalares inteiros adequadamente escolhidos e a *semente* é um inteiro  $x_0$ . A equação (2) gera uma sequência de inteiros que situam-se no intervalo  $0 \leq x_n \leq (m-1)$ . Como podem ser investigados pela teoria da congruência, estes geradores são chamados de congruenciais. Aproximações de variáveis  $U(0, 1)$  podem ser obtidas com  $u_i = x_i/m$ .

### 2.1.2 Geradores congruenciais multiplicativos

Geradores congruenciais multiplicativos são mais simples que a forma geral, pois o incremento  $b$  é zerado.

$$x_{i+1} = ax_i \pmod{m} \quad i = 1, \dots, n \quad (3)$$

Um gerador multiplicativo comumente usado Pidd (2004) pelas boas propriedades estatísticas tem os seguintes valores  $a = 16.807$  e  $m = 2^{31} - 1.147.483.647$ , ou  $a = 630.360.016$  e  $m = 2^{31} - 1$ .

### 2.1.3 Geradores recursivos de números aleatórios

Dentre as propriedades investigadas em fórmulas geradoras de números aleatórios, observa-se a distribuição dos números uniformemente no intervalo. É importante que os números dispersos não apresentem padrões como: agrupamento em blocos, tendências, etc, quando dispostos em coordenadas cartesianas. Métodos recursivos de geração de números aleatórios são apresentados nas fórmulas (4) e (5) onde  $p$  e  $m$  são números primos adequadamente escolhido.

$$x_j = x_{j-2} + x_{j-3} \pmod{p} \quad (4)$$

$$x_{j+1} = a_1x_j + a_2x_{j-1} + a_t x_{j-t-1} \pmod{m} \quad (5)$$

## 2.2 Teste de aleatoriedade

Os geradores de números aleatórios são a base do método de simulação por eventos discretos, no entanto, como os números gerados são pseudo-aleatórios, há sempre a chance de efeitos inesperados ocorrerem em alguma aplicação particular, por isso é preciso cuidado e verificações constantes nas propriedades destas fórmulas. As verificações podem ser realizadas tanto em distribuições uniformes, quanto em variáveis aleatórias geradas a partir destas distribuições. Elas podem ser feitas por *testes teóricos*, que avaliam os mecanismos de geração, ou por *testes empíricos*, que avaliam a replicabilidade do método e a uniformidade dos números aleatórios gerados. Os testes podem ser Pidd (2004):

1. Visuais, pelo uso de gráficos de dispersão. Veja que, para  $i = 1, \dots, 1000$  que  $x_{i+1} = 16.807x_i \pmod{2^{31}-1}$  é melhor distribuído que  $x_{i+1} = 19.031x_i + 9.298x_{i-1} \pmod{65.536}$ .

2. Por sequências auxiliares, utilizando outros números inteiros.
3. Por testes de frequência, para avaliar a uniformidade de sequências em um período completo.
4. Por testes em série, em que pares, tripla ou  $k$ -tuplas de valores são combinados para avaliar se são independentemente distribuídos.
5. Teste do qui-quadrado, ou de Kolmogorov-Smirnov, para testar a aderência de uma distribuição uniforme de números entre 0 e 1 sobre a sequência de números gerada [Chwif and Medina \(2006\)](#).

## 2.3 Geradores de variáveis aleatórias

### 2.3.1 Geradores de variáveis aleatórias contínuas

Um dos métodos mais populares de geração de variáveis aleatórias contínuas é o método da *Transformada Inversa* [Chwif and Medina \(2006\)](#). Dada uma função de densidade de probabilidade  $f(x)$  para uma variável aleatória  $X$ , faça:

1. Obtenha a *função acumulada* (ou *função de repartição*)  $F(x)$ :

$$F(x) = \int_{-\infty}^x f(x)dx \quad (6)$$

2. Gere um número aleatório ( $r$ ) no intervalo  $[0, 1]$ .
3. Faça  $F(x) = r$  e resolva em  $x$ . A variável  $x$  é uma variável aleatória, cuja distribuição é dada pela função densidade de probabilidade  $f(x)$ .

### 2.3.2 Geradores de variáveis aleatórias discretas

O métodos da *Transformada Inversa* também pode ser aplicado a variáveis aleatórias discretas. Neste caso, o método é um pouco diferente [Chwif and Medina \(2006\)](#):

1. Obtenha a *função acumulada* (ou *função de repartição*)  $F(x)$ :

$$F(x) = P(X \leq x) = \sum_{x_k \leq x} P(X = x_k) \quad (7)$$

2. Gere um número aleatório ( $r$ ) no intervalo  $[0, 1]$ .
3. Escolha o menor inteiro  $i$ , tal que:

$$r \leq F(x_i) \quad (8)$$

4. Faça a variável aleatória  $X$  igual a  $x_i$

### 3 Modelagem dos dados de entrada

Referência	Capítulos	Páginas	Assunto
<a href="#">Chwif and Medina (2006)</a>	[02.0.0]	[019,041]	Modelagem dos dados
<a href="#">Kelton et al. (1998)</a>	[04.4.0]	[142,158]	Determinísticos x aleatórios
<a href="#">Kelton et al. (1998)</a>	[04.4.0]	[142,158]	Input Analyzer
<a href="#">Altiok and Melamed (2010)</a>	[07.0.0]	[123,128]	Input Analyzer
<a href="#">Law (2007)</a>	[06.0.0]	[275,380]	Distribuição de input
<a href="#">Banks and Carson (1984)</a>	[09.0.0]	[332,352]	Coleta, estimativa e teste
<a href="#">Fishman (2001)</a>	[10.0.0]	[454,469]	Deduções teóricas do input

Em modelos de simulação, a ideia de modelar dados significa obter *modelos probabilísticos* que permitam *inferir* as propriedades de um dado *fenômeno aleatório*. Chamamos de *modelos de entrada*, os modelos probabilísticos que representam a *natureza aleatória* de um dado fenômeno. A *modelagem dos dados* é o processo de escolher a *melhor* representação deste fenômeno. Um estudo de modelagem dos dados pode ser resumido em três etapas [Chwif and Medina \(2006\)](#):

1. **Coleta de dados:** Corresponde ao processo de *amostragem*.
2. **Tratamento dos dados:** Técnicas para *descrever os dados* (estatística descritiva: medidas de posição e de dispersão) e identificar possíveis *falhas nos valores* (como *outliers*) para aumentar o conhecimento do fenômeno.
3. **Inferência:** Inferir o comportamento da *população* a partir da *amostra*. Como resultado, temos um *modelo probabilístico*, ou seja, uma *distribuição de probabilidade* que será incorporada ao modelo.

Os dados de entrada são usados para inicializar *parâmetros*, como número de máquinas, nível máximo de estoque, etc, e *variáveis aleatórias*, como tempo de serviço, tempo de reparo, intervalo entre chegadas, entre outros. A atividade de modelar os componentes aleatórios de entrada é chamada análise de *input* e consiste em coletar os dados, analisá-los, modelá-los e verificar os testes de aderência.

A coleta inadequada de dados leva a modelos com previsões errôneas e inúteis. É importante diferenciar o que são "dados de entrada"(valores *fornecidos* ao modelo) e o são "dados de saída"(valores *obtidos* do modelo) [Chwif and Medina \(2006\)](#).

Também deve-se fazer uma a avaliação qualitativa (dados corretos e relevantes) e quantitativa (tamanho das amostras devem ser representativas e grande o suficiente). A avaliação contra medidas empíricas do sistema (como utilização média, atrasos médios) é chamada de análise de sensibilidade [Kelton et al. \(1998\)](#) e usada para validar a credibilidade do modelo [Altiok and Melamed \(2010\)](#). A *análise de sensibilidade* é importante para comparar parâmetros de *input* determinísticos x aleatórios. Dados determinísticos não produzem resultados aleatórios, no entanto, seu uso incorreto pode causar erros grosseiros no modelo. Geralmente, dados determinísticos são: número de máquinas, servidores, funcionários, etc. Dados aleatórios são: intervalo entre chegadas, duração do atendimento, etc [Kelton et al. \(1998\)](#).

A análise de dados envolve a computação e estatísticas descritivas de posição (média, mediana, moda), dispersão (desvio-padrão, variância), distribuição (histogramas) e relações

de dependência temporal (correlações entre séries temporais) [Altiok and Melamed \(2010\)](#). A análise dos dados também pode indicar erros operacionais, pela presença de *outliers*. Estes dados devem ser retirados da amostra. Além disso, a avaliação de custo de coleta pode levar a coletas insuficientes ou estimativa de alguns dados. Para isso é necessário a realização de análises de sensibilidade para avaliar a qualidade dos dados. A expressão “*garbage in, garbage out*” é aplicável quando nenhuma análise dos dados é feita antes de inicializar a simulação [Kelton et al. \(1998\)](#).

Uso do dados históricos para realizar a simulação não é recomendado. Pelo ponto de vista teórico, o que ocorre é uma representação do passado, além disso os dados podem estar corrompidos, levando a previsões tendenciosas. Pelo ponto de vista prático, rodadas de simulação são demoradas, portanto, a modelagem dos dados por representação de distribuições é recomendado [Kelton et al. \(1998\)](#).

Na etapa de modelagem dos dados por séries temporais, as coletas podem ser classificadas nas categorias de: observações independentes, em que as variáveis aleatórias são independentes identicamente distribuídas (*i.i.d.*), e observações dependentes, em que é preciso que se determine distribuições adequadas aos dados empíricos observados. A forma de ajuste mais fácil é construir um histograma com os dados coletados e determinar a distribuição de probabilidade que mais se aproxima do mesmo [Altiok and Melamed \(2010\)](#).

Muitas vezes, os dados de entrada não estão disponíveis em registros, neste caso, o analista deve confiar na opinião de especialistas e usuários do sistema. Dentre as possíveis simplificações estão: a determinação de valores determinísticos para variáveis aleatórias; uso de distribuições simples, como a triangular, para representar operações com distribuições desconhecidas [Kelton et al. \(1998\)](#), [Altiok and Melamed \(2010\)](#).

### 3.1 Coleta dos dados

A coleta de dados é uma das atividades mais trabalhosas em um estudo por simulação. Mesmo com a estrutura do modelo válida, se os dados de entrada são coletados de forma não acurada, analisados de forma imprópria ou não representativa, o resultado poderá implicar em tomada de decisões custosas, portanto, enumera-se 7 dicas que facilitam a coleta de dados [Banks et al. \(2005\)](#):

1. Use boa parte do tempo para observar e planejar. Faça coletas intensivas de dados enquanto aprende a operação. Provavelmente os formulários elaborados serão modificados diversas vezes antes de iniciar a verdadeira coleta de dados. Garanta que os dados apropriados estejam disponíveis e em um formato utilizável.
2. Tente analisar os dados enquanto eles são coletados. Descubra quais dados formas as distribuições de entrada e quais dados são inúteis para a simulação. Não há necessidade de coletar dados supérfluos.
3. Tente combinar conjuntos de dados homogêneos. Verifique homogeneidade em períodos de tempo (exemplo: 14:00 às 15:00; 15:00 às 16:00) ou dias (exemplo: 14:00 às 15:00 todos os dias da semana) avaliando médias e desvios.
4. Esteja atento para a possibilidade de falta de registros. Muitas vezes, parte do tempo de uma operação completa não é coletada ocasionando amostras com tempos menores de operação.
5. Descubra se há relação entre duas variáveis. Construa diagramas de dispersão.
6. Considere a possibilidade de haver correlações entre eventos parecem independentes.

7. Saiba sempre a diferença entre dados de entrada e dados de saída ou desempenho, e colete apenas dados de entrada. Estes são geralmente representados por quantidades desconhecidas. Dados de saída, por outro lado, são valores de desempenho que queremos melhorá-lo. Em um sistema de filas, por exemplo, intervalo entre chegadas de clientes são dados de entrada, enquanto atrasos em filas são dados de saída.

### 3.2 Tratamento dos dados

Nesta etapa são usadas ferramentas da Estatística Descritiva. Usamos medidas de posição (média, mediana, moda, etc.) e medidas de dispersão (variância, amplitude, etc.)

Valores não usuais são conhecidos como *outliers*. Eles podem ser provenientes de um erro na coleta dos dados ou evento raro. *Outliers* distorcem as estimativas. Técnicas dos Quartis ( $A = Q_3 - Q_1$ ) são eficientes para identificar *outliers*.

A análise de correlação é utilizada para ver se as observações são *independentes identicamente distribuídas (i.i.d.)*. Um dos testes mais simples é o de *diagrama de dispersão*. Neste caso, pode-se observar o que se denomina *série temporal*, ou seja, uma correlação de uma variável com outra (tempo).

### 3.3 Identificação da distribuição dos dados

A etapa de *inferência* é iniciada pela elaboração de um *histograma*. Nesse caso, a *Regra de Sturges* pode ser aplicada Chwif and Medina (2006):

$$K = 1 + 3,3 \log_{10} n \quad (9)$$

Onde:

$K$  : Número de classes

$n$  : Número de observações na amostra

A amostra deve ser dividida em  $K$  classes. O tamanho  $h$  de cada classe deve ser:

$$h = \frac{\text{Amplitude}}{K} \quad (10)$$

A última etapa é qualificarmos a *aderência do modelo*. Nos testes estatísticos de aderência uma hipótese nula determina que a distribuição candidata é suficientemente boa para representar os dados, enquanto que a hipótese alternativa determina que não é. Os testes não devem ser usados como regras definitivas, mas meramente disponibilizar uma evidência sugestiva. Os testes mais usados são *Chi-quadrado* e *Kolmogorov-Smirnov* Banks et al. (2005). Softwares, como o *Input Analyzer* do Arena<sup>TM</sup> possuem a funcionalidade de ajustar distribuições aos dados amostrais e elaborar testes de aderência de curvas Altiok and Melamed (2010) Chwif and Medina (2006).

Distribuições de família de dados são especificadas quando estimamos seus parâmetros. Na prática, especificar distribuições teóricas com valores plausíveis para os dados que se tem em mãos é chamado de *preparação de input* Fishman (2001). Um histograma é útil para identificar a forma da distribuição de maneira a inferir sobre uma *função de densidade de probabilidade - fdp (contínua)* ou *função massa de probabilidade - fmp (discreta)*. Em seguida, é necessário selecionar a distribuição que mais se ajusta aos dados. Algumas distribuições comuns à simulação são descritas Banks et al. (2005):

**Binomial:** Modela  $n$  jogadas sucessivas independentes, com probabilidade de sucesso  $p$ .



**Binomial negativa:** Modela o número de jogadas até obter  $k$  sucessos.

**Poisson:** Modela o número de eventos independentes que ocorrem em um período fixo.

**Normal:** Modela um processo como a soma de processos menores. Admite valor negativo.

**Exponencial:** Modela tempo entre eventos independentes ou tempo de processo sem memória.

**Uniforme:** Modela incerteza completa. Todas as saídas são igualmente prováveis.

**Triangular:** Modela um processo onde apenas o mínimo, mais provável e máximo são conhecidos.

**Gamma:** Distribuição flexível usada para modelar variáveis aleatórias não negativas.

Após a seleção da distribuição, é preciso estimar os parâmetros da distribuição. Estatísticas preliminares envolvem estimar a média (11) e variância (12) da amostra:

$$\bar{X} = \frac{\sum_{i=1}^n X_i}{n} \quad (11)$$

$$S^2 = \frac{\sum_{i=1}^n X_i^2 - n\bar{X}^2}{n-1} \quad (12)$$

Os seguintes estimadores são indicados para as distribuições presentes na Tabela 1:

Distribuição	Parâmetros	Estimador
Poisson	$\alpha$	$\hat{\alpha} = \bar{X}$
Exponencial	$\lambda$	$\hat{\lambda} = \frac{1}{\bar{X}}$
Normal	$\mu, \sigma^2$	$\mu = \bar{X}, \sigma^2 = S^2$

Tabela 1: Estimadores das distribuições em simulação

### 3.4 Quando os dados não estão disponíveis

## 4 Elaboração do modelo conceitual

Referência	Capítulos	Páginas	Assunto
<a href="#">Chwif and Medina (2006)</a>	[03.0.0]	[043,067]	DCA e outros
<a href="#">Pidd (2004)</a>	[05.3.0]	[066,079]	DCA
<a href="#">Kelton et al. (1998)</a>	[00.0.0]	[000,000]	X
<a href="#">Law (2007)</a>	[00.0.0]	[000,000]	X
<a href="#">Fishman (2001)</a>	[00.0.0]	[000,000]	X

### 4.1 Abstração e modelos abstratos

Diferentemente da modelagem dos dados de entrada, a elaboração do modelo conceitual relaciona os elementos essenciais para o processo de tomada de decisão. Nesta etapa, o modelo busca representar a lógica fundamental do sistema [Kelton et al. \(1998\)](#). A etapa de criação do modelo conceitual é o aspecto mais importante de um estudo de simulação [Law \(2007\)](#).

### 4.2 Diagramas de Ciclos de Atividades

Sistemas de eventos discretos envolvem sistemas de filas de uma forma ou de outra e podem ser representados por Diagramas de Ciclos de Atividades (DCA), um dispositivo que segue o *princípio da parcimônia* (pois utiliza apenas *dois símbolos*) e adequado para a explicitar as relações lógicas destes sistemas. Por meio do DCA são explicitadas as interações entre as principais entidades do modelo [Pidd \(2004\)](#).

A terminologia relacionada aos diagramas de ciclos de atividades são divididas em dois grupos, o primeiro é relacionado aos objetos que constituem o sistema, enquanto que o segundo define as operações em que estes objetos se interagem ao longo do tempo.

**Objetos:** *Entidades e recursos.* Entidades são elementos do sistema cujo comportamento é explicitamente rastreado. Podem ser permanentes ou temporários. São agrupados em classes, conjuntos ou atributos (informações). Recursos são elementos contáveis e não são modelados ou rastreados individualmente. Entidades temporárias podem ser modeladas como recursos.

**Operações:** *Eventos, atividades, processos ou relógio da simulação.* Eventos são as mudanças de estado do sistema. Atividades são operações que envolvem entidades e mudam seus estados. Processos podem ser considerados um agrupamento de eventos e atividades.

Diagramas de ciclos de atividades são úteis para sistemas que possuem estruturas de filas, que são a maioria. O estado passivo representa as filas. Entidades movem de uma fila para outra ao passarem por algum estado ativo. Dessa forma, os DCAs são formas úteis para o desenvolvimento do modelo conceitual. Eles auxiliam na identificação de objetos dinâmicos do sistema; classificam os sistemas em entidades ou recursos e auxiliam o analista a entender as condições que governam as mudanças de estado do sistema modelado. Ao elaborar DCAs, é preciso ter atenção a alguns detalhes:

1. É obrigatória a alternância entre filas e atividades.

2. Filas são de entidades exclusivas. Não há uma fila para 2 entidades.
3. Se 2 ou mais atividades disputam 1 entidade, a prioridade deve ser explicitada.
4. As disciplinas das filas devem ser explicitadas (exceto FIFO).
5. A quantidade e posição inicial de entidades permanentes deve ser explicitada.
6. Entidades temporárias devem ser criadas e destruídas (fonte/sumidouro).
7. Desvios e atribuições devem ter regras explicitadas.
8. Duração das atividades devem ser explicitadas.
9. Uma entidade não pode ter descontinuidade no processo.

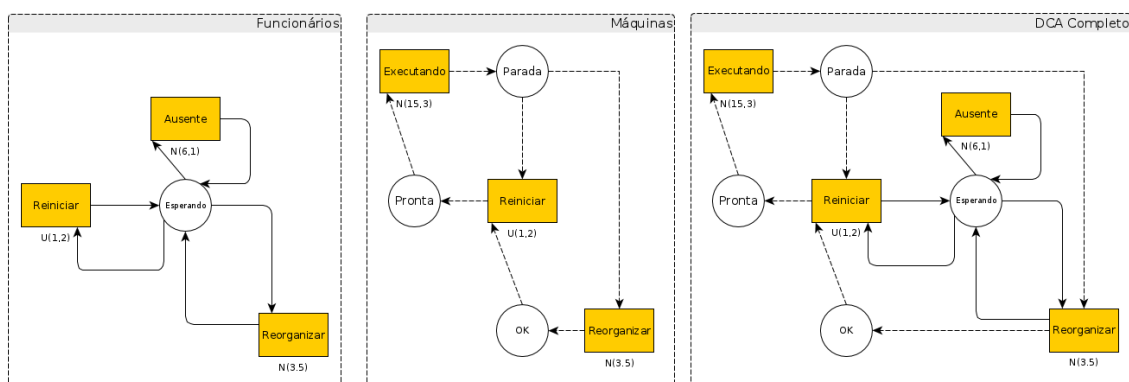


Figura 2: Elaboração do DCA da operação de uma oficina

A Figura 2 apresenta o DCA de uma operação de *job-shop* em que as entidades *operadores* e *máquinas* precisam estar juntos para realizarem as atividades de manufatura, enquanto que a Figura 3 representa o DCA da atividade de um atendente que faz o agendamento de um teatro. O agendamento pode ser feito por atendimento ao pessoal ou por atendimento telefônico. A forma gráfica de descrição da lógica do sistema constrói o esqueleto do modelo de simulação, pois viabilizam uma especificação precisa das condições de operação. Recomenda-se a criação de modelos suficientemente pequenos, viabilizando seu descarte e reconstrução em caso de erros graves. No entanto, mesmo em sistemas complexos, recomenda-se o particionamento do sistemas em blocos de lógica menores. Esta abordagem permite sua interpretação, verificação e validação dos resultados de forma muito mais ágil.

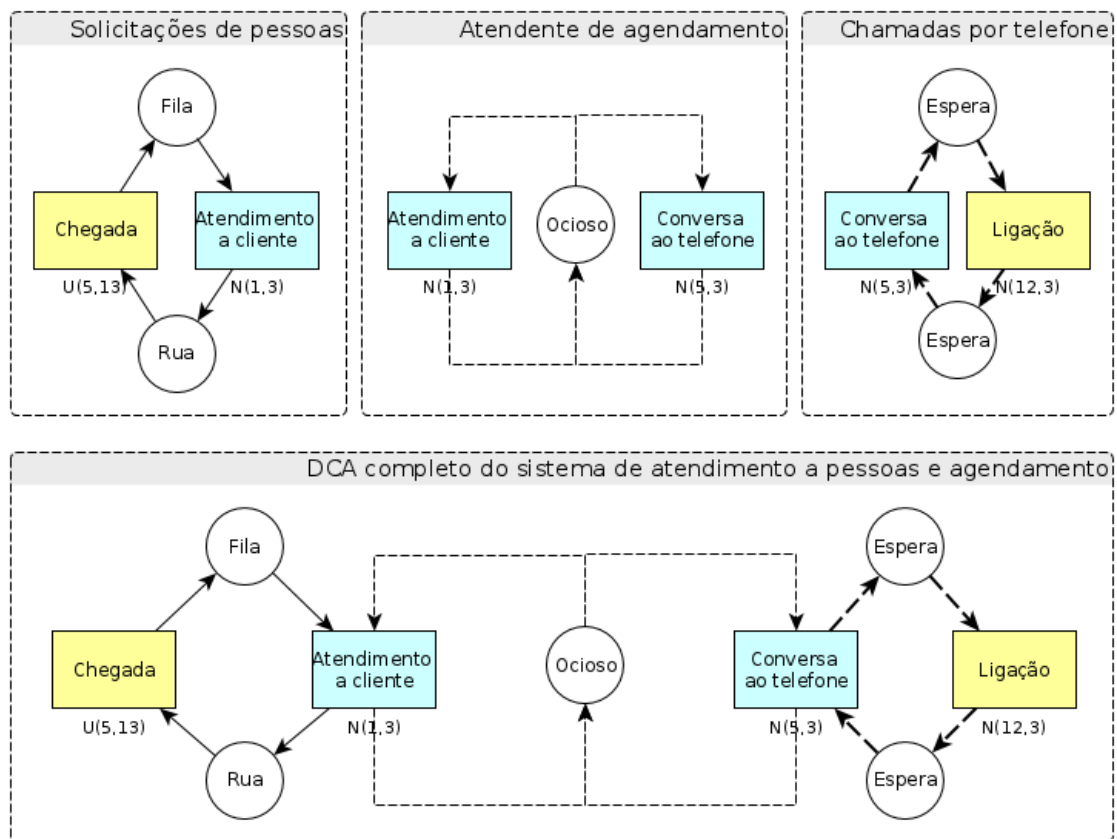
Linhas de produção, cadeias de suprimentos, sistemas de transporte ou sistemas de informações são exemplos de sistemas modelados por *softwares* de simulação [Altiok and Melamed \(2010\)](#).

### 4.3 Visão de processos

Modelos conceituais também podem ter uma visão de processos. Denominados *Process Networks*, são uma das formas mais antigas de representação de modelos de simulação. Diferentemente do DCA, os PN usam símbolos para: *chegada*, *fila*, *atraso*, *processo com recurso*, *condicional*, *saída*. A atividade "atraso + processo com recurso" é equivalente à uma "atividade" no DCA.

### 4.4 Especificação de modelos de simulação

A *complexidade* de um modelo é definido pelo binômio: [*escopo* vs. *nível de detalhamento*].



**Figura 3:** Elaboração do DCA da atividade do atendente de agendamento

## 5 Elaboração do modelo computacional

Referência	Capítulos	Páginas	Assunto
<a href="#">Chwif and Medina (2006)</a>	[04.0.0]	[069,088]	Implementação e softwares
<a href="#">Pidd (2004)</a>	[06.0.0]	[083,105]	3 Fases (ABC), eventos e processos
<a href="#">Kelton et al. (1998)</a>	[00.0.0]	[000,000]	X
<a href="#">Law (2007)</a>	[00.0.0]	[000,000]	X
<a href="#">Fishman (2001)</a>	[00.0.0]	[000,000]	X

### 5.1 Linguagens: de programação vs. de simulação vs. simulador

Até meados de 1980, a elaboração de modelos computacionais era feita por programação. Desde então, é possível a elaboração desses modelos por meio de interfaces interativas. *Visual Interface Modeling Systems* (VIMS) [Pidd \(2004\)](#) possuem tabelas para o armazenamento de dados, blocos de lógica, sistemas de ajuste de curvas de probabilidade, geradores de relatórios e estatísticas dos resultados, entre outros [Altiok and Melamed \(2010\)](#). Além disso, viabilizam a rápida elaboração de cenários e flexibilidade de alteração dos modelos [Kelton et al. \(1998\)](#). Diferentes sistemas possuem vantagens e desvantagens. O uso de cada um depende da aplicação. Isto porque as operações podem envolver lógicas e cálculos específicos não previstos em um sistema de modelagem alto-nível. De qualquer forma, é importante entender os princípios nos quais os *softwares* operam. Quando uma organização possui profissionais com *expertise* em programação de computadores, pode-se optar por desenvolver um modelo específico ao invés de investir em *softwares* especialistas.

Os sistemas possuem dois componentes principais: o método de simulação - *simulation engine* - (como abordagens por três fases (ABC), por eventos ou por processos [Pidd \(2004\)](#), descrito em 1.4) e os componentes específicos (VIMS ou código implementado), que determinam como as entidades mudam de estado.

### 5.2 Implementação do método de três fases

Adotamos o exemplo da Figura 3 em que um atendente que faz o agendamento de um teatro. O agendamento pode ser feito por atendimento ao pessoal ou por atendimento telefônico. A atividade representada na Tabela 2 possui a seguinte classificação ABC:

- B1** : Chegada-de-pessoal
- B2** : Fim-do-atendimento-pessoal
- B3** : Chegada-de-ligacao
- B4** : Fim-da-ligacao
- C1** : Inicia-atendimento-pessoal
- C2** : Inicia-atendimento-ligacao

O método de três fases também é apresentado para uma empresa que opera com 3 sondas de perfuração de petróleo. Estas trabalham em regime contínuo, sendo interrompidas apenas para manutenção corretiva. O tempo entre falhas é descrito por  $\text{Exp}(3)$  dias. A manutenção é feita por uma equipe e possui a duração de  $\text{Exp}(1)$  dia. Deseja-se simular o problema

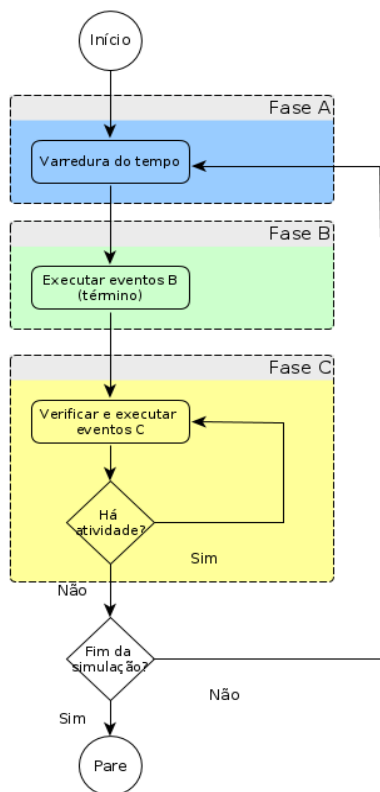


Figura 4: Método das 3 fases

Fase A	Fase B (booking)		Fase C (conditional)		$\Delta t$	Término
$t$	<i>térm. ativ.</i>	<i>libera ent.</i>	<i>ini ativ.</i>	<i>retém ent.</i>	( <i>min</i> )	
0			cheg. pessoa 1	pessoa 1	4	4
0			cheg. ligacao 1	ligacao 1	6	6
4	cheg. pessoa 1	pessoa 1	cheg. pessoa 2	pessoa 2	5	9
4			atend. pessoa 1	pes. 1 + balc.	5	9
6	cheg. ligacao 1	ligacao 1	cheg. ligacao 2	ligacao 2	3	9
9	cheg. pessoa 2	pessoa 2				Fila
9	atend. pessoa 1	pes. 1 + balc.	atend. ligacao 1	lig. 1 + balc.	6	15
9	cheg. ligacao 2	ligacao 2				Fila
...						

Tabela 2: Método de três fases aplicado ao atendimento no bar

Fase A	Fase B (booking)		Fase C (conditional)		$\Delta t$	Término
$t$	<i>térm. ativ.</i>	<i>libera ent.</i>	<i>ini ativ.</i>	<i>retém ent.</i>	( <i>dias</i> )	
0			operação	Máq 1	4,5	0+4,5 = 4,5
0			operação	Máq 2	3,8	3,8
0			operação	Máq 3	5,7	5,7
3,8	operação	Máq 2	manutenção	Máq 2 + Equ 1	1,3	5,1
4,5	operação	Máq 1				Fila
5,1	manutenção	Máq 2 + Equ 1	manutenção	Máq 1 + Equ 1	0,9	6,0
5,7	operação	Máq 3				Fila
...						

Tabela 3: Método de três fases aplicado ao problema da sonda

para avaliar o tempo que as sondas ficam paradas por falta de manutenção e analisar a ocupação média da equipe de manutenção. O resultado é apresentado na Tabela 3.

O algoritmo de simulação do método de três fases (ABC) é apresentado na Tabela 4 para a atividade onde 4 clientes chegam em um bar com 3 copos e são atendidos por uma garçonete. A atividade de beber a cerveja dura 4 minutos e de encher o copo (pela garçonete) leva 3 minutos.

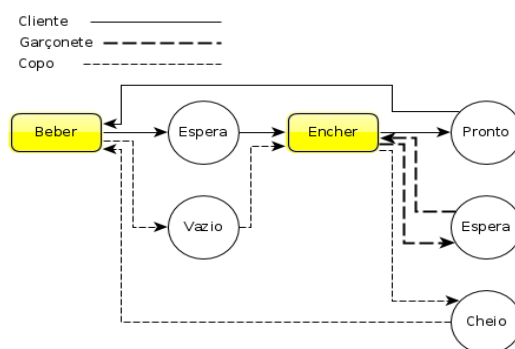


Figura 5: DCA relativo ao atendimento dos clientes no bar

Fase A	Fase B (booking)		Fase C (conditional)		$\Delta t$	Término
$t$	<i>térm. ativ.</i>	<i>libera ent.</i>	<i>ini ativ.</i>	<i>retém ent.</i>	( <i>minutos</i> )	
0			encher	copo 1 + garc	3	$0 + 3 = 3$
3	encher	copo 1 + garc	beber	copo 1 + cli 1	4	7
3			encher	copo 2 + garc	3	6
6	encher	copo 2 + garc	beber	copo 2 + cli 2	4	10
6			encher	copo 3 + garc	3	9
7	beber	copo 1 + cli 1				
9	encher	copo 3 + garc	beber	copo 3 + cli 3	4	13
9			encher	copo 1 + garc	3	12
10	beber	copo 2 + cli 2				
12	encher	copo 1 + garc	beber	copo 1 + cli 1	4	16
12			encher	copo 2 + garc	3	15
...						

Tabela 4: Método de três fases aplicado ao atendimento no bar

### 5.3 Abordagem por eventos

Consideramos o exemplo do atendente que faz o agendamento de um teatro Pidd (2004): O DCA deste sistema possui 4 eventos apenas. A redução, em comparação com o método de três fases (que possui 4 Bs e 2 Cs) é possível porque cada rotina de evento pode capturar *todas as possíveis* consequências de uma mudança de estado. Assim, por esta abordagem, os eventos equivalentes a B e C são combinados em uma única rotina de evento:

1. Chegada: Chegada de pessoa querendo comprar ingresso
2. Chamada: Chegada de uma chamada telefônica
3. Fim-atendimento: Fim do atendimento à pessoas
4. Fim-chamada: Fim do atendimento de uma chamada telefônica

O pseudo-código apresentado na Figura 6 descreve a rotina da simulação por eventos:

O *engine de simulação* baseada em eventos é gerenciado como segue:

1 # Rotina para evento Fim-atendimento ou Fim-chamada	1 # Rotina para evento Chegada ou Chamada
2	2
3 Libera PESSOA i para o mundo	3 Compute o tempo de Intervalo-entre-chegada
4 Se (fila > 0) então	4 Programe chegada-de-pessoa i+1 após Intervalo-entre-chegada
5 Segure a PESSOA i+1 na fila	5 Se (fila = 0) e (ATENDENTE está ocioso) então
6 Compute o tempo-de-atendimento	6 Pegue ATENDENTE do estado ocioso
7 Programe o fim-do-atendimento para ocorrer após o tempo-de-atendimento	7 Insira PESSOA da posição i
8 Senão Se (fila-de-ligacao > 0) então	8 Compute o tempo-de-atendimento
9 Segure a LIGACAO da fila-de-ligacao	9 Programe o fim-do-atendimento i para ocorrer após tempo-de-atendimento
10 Compute o tempo-de-conversa	10 Senão
11 Programe o fim-da-conversa dessa ligação para ocorrer após tempo-de-conversa	11 Adicione PESSOA i na fila
12 Senão	
13 Libere o ATENDENTE para estado ocioso	

Figura 6: Pseudo-códigos para eventos de chegada e fim de atendimento

1. Examine o calendário do evento para encontrar o próximo evento da lista de execução e mova o relógio para este momento. Mova todos os eventos que estão programados para este novo momento para uma lista de eventos atuais.
2. Mantenha o relógio constante. Execute cada rotina dos eventos que estão na lista de eventos atuais.

#### 5.4 Abordagem por varredura de atividades

A abordagem por varredura de atividades muitas vezes é confundida - erradamente - com o método de três fases por atividades. A diferença entre os métodos é que pelo método de três fases, as atividades da lista Bs não precisam verificação de condição para execução, o que ocorre no método de varredura de atividades. Desta forma, é como se tivessem apenas eventos Cs, que necessitam verificação de condições para execução. Embora a estratégia traga simplicidade, o método se torna ineficiente.

O *engine de simulação* da abordagem por varredura de atividades é gerenciado como segue:

1. Examine os eventos do calendário para encontrar o próximo evento. Mova o relógio da simulação para este momento.
2. Faça a varredura repetidamente por todas as atividades verificando se a atividade pode ocorrer ou não. Continue a varredura até que não haja atividades para serem executadas naquele momento.

#### 5.5 Abordagem por processos

Ao definir um processo para uma entidade, o analista precisa determinar como a entidade será adiada. Estes são conhecidos como pontos de reativação. O pseudo-código apresentado na Figura 7 descreve uma forma de modelagem da simulação por processos:

1 # Modelagem de serviço do atendente por processo
2
3 PESSOA i chega;
4 Compute o momento de chegada da PESSOA i+1;
5 Crie uma instância do processo para PESSOA i+1;
6 Adie PESSOA i+1 até o momento da chegada;
7 PESSOA i, espera-ate(primeiro da fila) e (ATENDENTE livre);
8 Ocupe ATENDENTE;
9 Pegue PESSOA i da fila;
10 Compute o tempo-de-atendimento i;
11 Adie PESSOA i até duração do tempo-de-atendimento;
12 Libere ATENDENTE para ocioso;
13 Libere PESSOA i

Figura 7: Pseudo-códigos para o processo de chegada e fim de atendimento



O processo apresentado na Figura 7 possui dois adiamento incondicionais e um adiamento condicional. Os termos **adie** apresentados nas linhas 6 e 11 representam *adiamentos incondicionais* e são equivalentes ao estado B da simulação por três fases. O *adiamento condicional* ocorre na linha 7 pelo termo **espera-ate** em que uma entidade é bloqueada até que uma condição é alcançada.

## 6 Verificação e validação dos modelos

Referência	Capítulos	Páginas	Assunto
<a href="#">Chwif and Medina (2006)</a>	[05.0.0]	[089,099]	Verificação e validação
<a href="#">Law (2007)</a>	[05.0.0]	[243,272]	Nível de detalhe. Validação
<a href="#">Pidd (2004)</a>	[12.0.0]	[232,239]	Filosofia de validação
<a href="#">Pidd (2004)</a>	[12.4.0]	[240,246]	Black (White) Box. Erros I,II,Zero
<a href="#">Banks and Carson (1984)</a>	[10.0.0]	[376,401]	Métodos de validação
<a href="#">Kelton et al. (1998)</a>	[00.0.0]	[000,000]	X
<a href="#">Fishman (2001)</a>	[00.0.0]	[000,000]	X

Uma vez que se tem o modelo funcionando, é hora de verificar e validar o modelo. Na verdade, este processo é cíclico, e deve acompanhar todas etapas do projeto.

*Verificar* está sempre relacionado ao *modelo computacional*. É a atividade de garantir que o modelo se comporta como você pretendia, ou seja, é fazer o *debugging*, rodadas de teste, verificar se as estatísticas são consistentes [Altiok and Melamed \(2010\)](#) e se não há erros no código. A pergunta que se faz é [Chwif and Medina \(2006\)](#):

"Estamos desenvolvendo *o modelo corretamente*"?

*Validar* está relacionado ao modelo *conceitual*. A pergunta é [Chwif and Medina \(2006\)](#):

"Estamos desenvolvendo *o modelo correto*"?

### 6.1 Técnicas de verificação

Em sistemas complexos diferentes atividades podem ocorrer simultaneamente, e ocasionar interações não pretendidas. A verificação de modelos de grande porte é dificultada pelo excesso de caminhos lógicos e interações do sistema [Law \(2007\)](#). Neste momento, é interessante desenvolver a parte visual da simulação para facilitar a verificação ao realçar um eventual comportamento não pretendido [Altiok and Melamed \(2010\)](#). A simulação visual é abordada na seção 14. Adicionalmente, sugere-se uma análise de consistência de desempenho do sistema por meio de uma avaliação por teoria de filas [Altiok and Melamed \(2010\)](#). Estes conceitos são vistos na seção 13. Finalizados os testes óbvios, deve-se visualizar os tipos de cenários a serem considerados na análise. A sugestão é elaborar o máximo de cenários possíveis por períodos longos de simulação. Sugere-se [Kelton et al. \(1998\)](#) que esta simulação seja feita de um dia para o outro. Em seguida, analise cuidadosamente os resultados buscando filas muito longas, recursos não utilizados, etc, para avaliar se os resultados fazem sentido. Em resumo, a *verificação* pode ser realizada em 5 etapas [Law \(2007\)](#):

1. Fazer o *debug* em módulos;
2. Solicitar a revisão do código a diferentes pessoas;
3. Alterar o *input* e avaliar de o *output* do modelo é razoavelmente consistente;
4. Avaliar estatísticas sob condições extremas;

5. Rodar o modelo em condições simplificadas (modulares) de forma que suas características possam ser facilmente computadas e comparadas por formulações analíticas por filas;
6. Usar animação para comparar as lógicas das operações com o sistema real.

## 6.2 Técnicas de validação

A etapa de *Validação* é a atividade de garantir que o modelo se comporta como o sistema real [Kelton et al. \(1998\)](#). Esta etapa não deve ser feita de forma inocente e simplista, do tipo: "se o sistema se comporta como a realidade, então ele é válido, senão, não é". Este é um processo de negociação entre o analista de simulação e seu cliente. O conceito de "sistema correto" é subjetivo e pouco conclusivo [Altiok and Melamed \(2010\)](#). Sabe-se que a simplificação é parte de qualquer modelo, ou seja, um modelo de simulação é sempre uma aproximação de um sistema real [Law \(2007\)](#). O reconhecimento das limitações é o primeiro passo para o processo de validação [Pidd \(2004\)](#).

A validação da simulação como parte de um projeto de gestão, ao invés de uma simulação de grande porte, envolve alguns aspectos práticos e teóricos [Pidd \(2004\)](#). Os modelos são usados para fornecer uma visão futura sob uma determinada operação. Assim, as considerações a serem usadas devem estar próximas da realidade. Sob este aspecto, a validação nunca deve ser feita após o término do desenvolvimento do modelo. Corre-se o risco de incorrer no Erro Tipo Zero (pergunta errada) [Law \(2007\)](#). Seu mecanismo pode ser entendido como a comparação entre 2 conjuntos de observação: um gerado a partir do modelo e outro, do observador (cliente) do sistema no mundo "real". Os conjuntos são amostras da realidade. É preciso salientar que toda amostra é limitada. Em um exemplo o observador diz: "todos os cisnes são brancos", sua amostra retrata a sua realidade. No entanto, na Nova Zelândia, por exemplo, cisnes negros são facilmente encontrados, como na Figura 8 [Pidd \(2004\)](#).

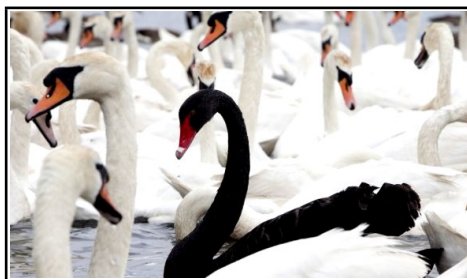


Figura 8: Cisne negro: amostragem é limitada.

Obter sucesso na validação de um modelo não é simples. Estudos de simulação são cíclicos. Dessa forma, os modelos devem ser gradualmente refinados, adotando-se, assim, o princípio da parcimônia (opção pelo mais simples). Ressalta-se que a comparação de modelos de simulação com sistemas reais só é permitido em um estado: o atual. Não há como afirmar que o modelo é válido sob quaisquer circunstâncias. É preciso fazer experimentos, comparações e garantir que a etapa de verificação foi concluída com êxito [Pidd \(2004\)](#). Comparações podem ser *black-box* e *white-box*

**Black-box:** São comparações entre os resultados (*output*) dos conjuntos (modelo e realidade) por inferências estatísticas. Observações respondem a testes de hipótese. A significância estatística é expressa por alguma probabilidade de valor. Deve-se também avaliar auto-correlações entre as observações e ocorrências de erros do Tipo I (se está certo e conclui-se que está errado) e Tipo II (se está errado e conclui-se que está certo) e o Tipo Zero (quando a pergunta formulada está errada).

**White-box:** A validação por esta estratégia ocorre geralmente no momento da confecção do modelo. São aplicados aos componentes do modelo e nas interações destes em comparação com o sistema real. A avaliação ocorre na estrutura interna do modelo. Neste caso, a estrutura interna de ambos sistemas deve ser bem compreendida. Assim, no *white-box* a validação foca na avaliação:

1. *Na distribuição de entrada:* Por avaliações de teste de aderência da distribuição, avaliação se a variável é discreta (binomial, poisson) ou contínua (normal, exponencial); se a variável é finita (triangular, uniforme) ou infinita (normal, exponencial) e o processo de geração de valores (processos Markovianos de poisson, variáveis independentes i.i.d.)
2. *Lógica estática:* Avaliação de regras (se <condição> então <ação>) que governam o sistema. Neste caso, é importante consultar o cliente e os usuários do sistema (operação real).
3. *Lógica dinâmica:* O funcionamento de componentes individuais não é suficiente. É preciso avaliar a interação entre eles. Nessa fase é importante usar a simulação visual Pidd (2004).

Na etapa de validação podemos cometer três tipos de erros (em analogia aos erros de teste de hipótese da estatística) Chwif and Medina (2006):

**Erro Tipo I :** O modelo é válido, mas é rejeitado.

**Erro Tipo II :** O modelo é inválido, mas é aceito.

**Erro Tipo Zero :** O modelo se desvia dos objetivos estabelecidos.

O **Erro Tipo Zero** é o pior de todos. Ocorre quando o analista de simulação e o cliente fazem a pergunta errada. Neste caso, se tem um modelo completamente errado. Este é o erro mais importante a ser evitado em qualquer validação. Este erro ocorre quando o modelo *pode até ser válido* tanto em lógica quanto em resultados estatísticos, *mas é inútil*, porque aborda *questões erradas* de desempenho Pidd (2004).

Para evitar Erros Tipo Zero e obter credibilidade, é importante elaborar programas modulares, de fácil descarte ou modificação. Deve-se evitar tanto elaborações extremamente detalhadas que busquem o máximo de realismo quanto a super-simplificação, pois podem tornar o modelo insuficiente para tomada de decisão. Assim, o nível de detalhamento do modelo deve ser acordado no projeto. Se o projeto exige uma operação extremamente detalhada, detalhe-a, ao custo necessário. Senão, recomenda-se começar simples e evoluir junto com toda equipe à medida da necessidade, atendendo sempre às perguntas corretas Pidd (2004). Modelos não são universalmente válidos. É preciso definir as medidas de desempenho a serem avaliadas Law (2007).

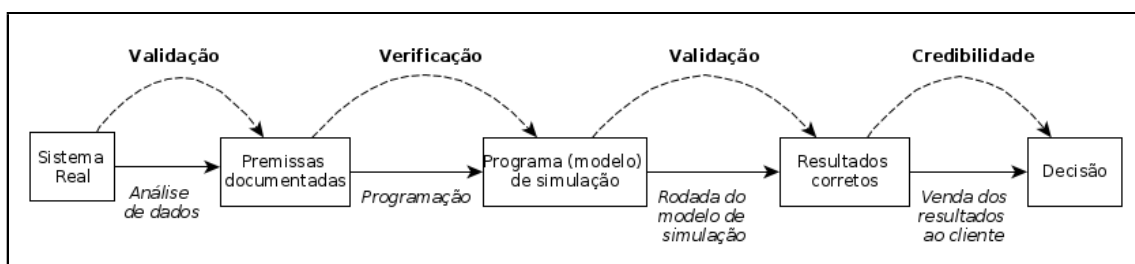


Figura 9: Ciclo de verificação, validação e credibilidade Law (2007).

A credibilidade pode ser obtida após reunir uma equipe de clientes e avaliar se estes estão convencidos de que o modelo está rodando como previsto. Se o sistema ainda não

existe, a comparação é uma atividade difícil. Muitas vezes, os registros de dados são para a contabilidade da empresa, ao invés de dados operacionais de engenharia. Se não há registros acurados do sistema real é praticamente impossível validar o modelo. Nesses casos, o foco deve ser em verificar o modelo e obter os melhores julgamentos de indivíduos com conhecimento da capacidade do sistema. Estes podem prever o comportamento do sistema com alguma precisão. Feito isto, tem-se uma poderosa ferramenta estatística, ou seja, o modelo está pronto para responder algumas questões. Em resumo, a *validação* pode ser realizada em 5 etapas Law (2007). Em todas elas o gerente do projeto de simulação deve ter um papel ativo:

1. Coletar os dados e informações de qualidade com especialistas de *cada* processo (não há uma pessoa que sabe tudo). Deve-se comparar os *outputs* com dados teóricos (ex: intervalos entre chegadas *i.i.d.*); avaliar resultados de estudos de simulação similares e considerar a experiência e intuição do analista de simulação.
2. Interaja com o cliente do modelo (gerente) em uma frequência regular: para aumentar o entendimento do problema; para envolver e manter o cliente interessado; para que o cliente entenda melhor as premissas adotadas.
3. Mantenha as premissas anotadas por escrito e faça um *check-list* de forma estruturada com o cliente: para evitar erros de comunicação, reforçando a avaliação do modelo conceitual. Esta etapa determina os indicadores a serem avaliados, portanto, deve ser feita antes de coletar os dados com os especialistas.
4. Valide os componentes do modelo usando técnicas quantitativas: determine fatores-chave (como valor de um parâmetro, distribuição de probabilidade, etc.) que mais impactem na análise de sensibilidade.
5. Valide o resultado global do modelo completo. Compare os resultados e indicadores com o sistema real para determinar se o nível de acurácia adotado é válido. Avalie a opinião de especialistas e use animação para facilitar o entendimento e a aceitação pelo cliente.

## 7 Dimensionando aquecimento e replicações

Referência	Capítulos	Páginas	Assunto
<a href="#">Chwif and Medina (2006)</a>	[06.0.0]	[101,138]	Dimensionamento de corridas
<a href="#">Law (2007)</a>	[09.0.0]	[485,540]	Análise modelo único
<a href="#">Banks and Carson (1984)</a>	[11.0.0]	[406,440]	Análise modelo único
<a href="#">Pidd (2004)</a>	[11.2.0]	[210,217]	Sistemas (não) terminais
<a href="#">Kelton et al. (1998)</a>	[00.0.0]	[000,000]	X
<a href="#">Fishman (2001)</a>	[00.0.0]	[000,000]	X

Após a etapa de V&V, o modelo torna-se *operacional*. Tem-se uma *poderosa ferramenta de experimentos estatísticos*, que permite responder questões do tipo: "*o que ocorre se...*". No entanto, entradas dos modelos de simulação são aleatórias, portanto, as saídas também são (RIRO: *random in, random out*) e não podemos concluir muita coisa com apenas uma replicação. Busca-se avaliar os resultados das medidas de desempenho por meio de sua *precisão*, que pode ser avaliada pela variância, e pela *confiança estatística*, que determina a probabilidade do intervalo de confiança conter a média da população relativa à medida de desempenho observada [Chwif and Medina \(2006\)](#). No entanto, é necessário discutir alguns conceitos pré-requisitos para análise dos resultados. São eles:

1. Medidas de desempenho
2. Replicação e rodada
3. Regime transitório vs. regime permanente
4. Simulação terminal vs. Simulação não terminal

### 7.1 Medidas de desempenho

A análise dos resultados requer uma reflexão sobre a seleção de medidas de desempenho. A seleção de medidas, como por exemplo: tempo médio de espera em fila, ou número de clientes no sistema, são medidas indiretas que *umentam o conhecimento* sobre o sistema. No entanto, medidas de desempenho devem ser orientadas ao resultado. Neste caso, para o exemplo apresentado, boas medidas seriam: *número médio de clientes que desistem porque a fila está grande*, ou *probabilidade do cliente ter que esperar mais que 3 minutos* [Chwif and Medina \(2006\)](#).

### 7.2 Replicação e rodada

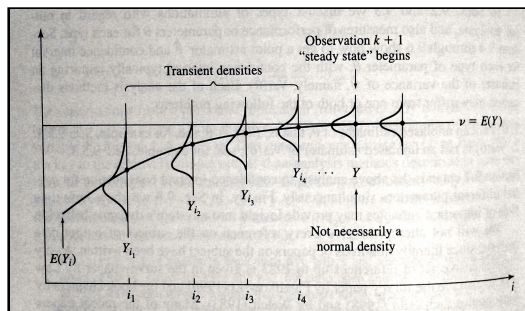
Uma **rodada** pode envolver *várias replicações*. Uma **replicação** é uma *repetição* da simulação do modelo com: a mesma configuração, a mesma duração, os mesmos parâmetros de entrada, *mas com uma semente de geração de números aleatórios diferente*.

### 7.3 Regime transitório vs. regime permanente

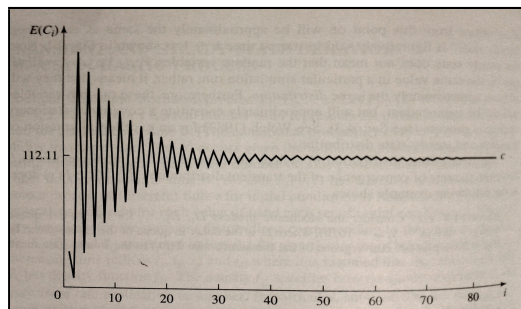
No *regime transitório* (estado transiente) a função densidade de probabilidade  $f(y)$  das variáveis de *output*  $y$ , possuem *formatos diferentes*, enquanto que, no *regime permanente* (estado-estacionário) elas se *assemelham muito*. A convergência da função  $F(y)$  pode ser *monotônica* ou *não-monotônica*. [Law \(2007\)](#).

O período de aquecimento em simulação não terminal é representado por *distribuições transientes*, que são funções de densidade de probabilidade não similares. Ao atingir o estado

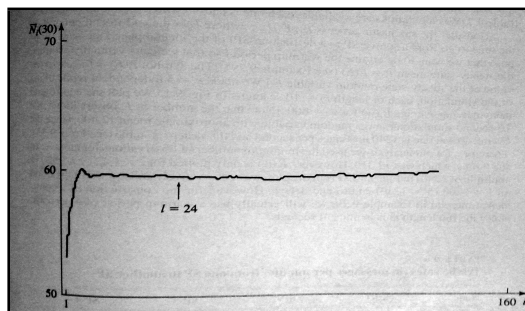
estacionário, as *distribuições de estado estacionário* se assemelham, embora não tenham que ser normalmente distribuídas. Esta relação é apresentada na Figura 10. Simulações não terminais podem evoluir para estados estacionários de forma monotônica (Figura 10) ou não monotônica (Figura 11).



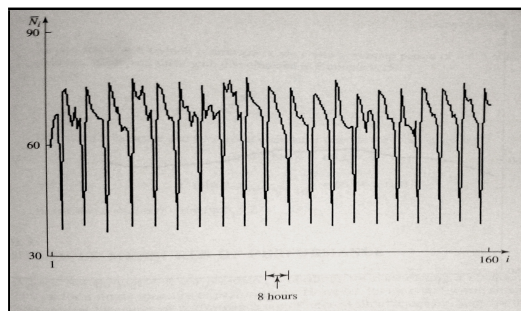
**Figura 10:** Funções de densidade transitente e estado-estacionário para um processo  $Y_i$ . Fonte: Law (2007)



**Figura 11:**  $E(C_i)$  como função não-monotônica de  $i$  em um sistema de estoques. Fonte: Law (2007)



**Figura 12:** Uso de média móvel para determinar o período de Warm-up. Fonte: Law (2007)



**Figura 13:** Produção/hora em uma pequena fábrica com paradas para o almoço: Ciclos. Fonte: Law (2007)

## 7.4 Simulação terminal vs. Simulação não terminal

A simulação, ao ser avaliada sob o ponto de vista da análise de *output*, pode ser classificada em *simulação-terminal* ou *simulação não-terminal* Law (2007) Altiook and Melamed (2010).

1. **Simulação terminal:** O fim é natural. Rodadas independentes produzem variáveis *i.i.d.*, por isso, o número de replicações é o parâmetro crítico, uma vez que é este número que determina o tamanho da amostra, afetando diretamente a acurácia das estatísticas.

Neste tipo de simulação, a *condição inicial afeta* as medidas de desempenho desejadas. O analista está interessado na dinâmica de curto prazo da operação. A inicialização do sistema pode ser feita de duas formas: iniciar a simulação a partir do momento que as primeiras entidades chegam ao sistema e considerar as estatísticas do momento específico a ser avaliado, ou fazer diversas observações do estado inicial de forma a elaborar uma distribuição de probabilidade de ocorrência da variável no início da simulação (ex: avaliar funcionamento de um banco de 11:00 às 12:00 simulando desde 08:00, momento de sua abertura e computar estatísticas somente a partir de 11:00 ou computar chegadas às 11:00 em diversos dias de forma a descrever uma distribuição de chegadas nestes momento).

2. **Simulação não terminal:** Não há um evento que determine o fim da simulação. As estatísticas de um sistemas inicial **não afetam** as medidas de desempenho desejadas

(elas podem ser diluídas se somadas infinitos eventos estacionários). Nesses casos, o analista está interessado na dinâmica e estatísticas de longo prazo.

A medida de desempenho é um *parâmetro do estado estacionário*. O início de uma simulação não-terminal é caracterizada por uma perturbação, chamada de estado transiente, período de aquecimento, ou *warm-up* (ver Figura 12).

Para determinar o tempo de aquecimento (*warm-up*) e duração da rodada, estes sistemas podem ser avaliados por 3 alternativas [Kelton et al. \(1998\)](#):

- (a) **Avaliação visual:** A duração do tempo de aquecimento pode ser determinada visualmente, por meio de gráficos ao longo da simulação.
- (b) **Replicações interrompidas:** Se o tempo de aquecimento é relativamente curto, pode-se fazer replicações interrompidas. Nesse caso, faça replicações *iid* e elimine o tempo de aquecimento e faça as análises estatísticas das simulações. Mais precisão é obtida por meio do aumento do número de replicações ou a duração da simulação (aumenta o tamanho da amostra).
- (c) **Abordagem por lotes:** Ao invés de fazer replicações *iid*, faça uma longa replicação com reinicializações e aquecimento. Quebre esta longa rodada em "*lotes*" de forma a evitar correlações entre os dados da mesma (única) replicação, para elaborar "observações independentes" e computar as estatísticas.

O momento da estabilização, chamado de *estado estacionário*. As estatísticas que antecedem o estado estacionário, portanto, são descartadas para evitar análises de *output* tendenciosas, ou seja, a computação de estatísticas deve feita somente após esta etapa.

A simulação "terminal" também pode ser abordada por parâmetros de ciclo de estado estacionário, como apresentado pela Figura 13, que são divisões da simulação em períodos de tempo de igual duração (turnos, por exemplo) de forma que as variáveis de ciclo possam ser comparáveis.

No início da simulação não-terminal, os simuladores não apresentam as distribuições de um sistema no estado-estacionário. Para isso, é necessário um procedimento (*bootstrapping*) de computar as estatísticas a partir de um período de aquecimento (*warm-up*), em que são desconsiderados os eventos anteriores ao momento de estado estacionário das distribuições. O objetivo em se dimensionar a duração do *estado transiente*, ou seja, o tempo de aquecimento (*warm-up*) e o número de replicações é obter o máximo de informações estatísticas das rodadas de simulação ao menor custo computacional possível, dessa forma estatísticas provenientes de replicações fornecem dados para estimar intervalos de confiança dos parâmetros de interesse do sistema.

É importante ressaltar que processos estocásticos reais geralmente não possuem distribuições de estado estacionário, uma vez que as características de estado dos sistemas mudam ao longo do tempo (modele regras de sequenciamento, por exemplo). Por outro lado, podemos usar modelos de simulação (que são abstrações da realidade) abordando distribuições de estado estacionário, quando se avalia uma característica de uma operação no longo prazo, sem que esta varie ao longo do tempo. Dessa forma, *a simulação para um sistema particular pode ser abordada tanto por um sistema terminal quanto não terminal. Depende do objetivo do estudo de simulação* [Law \(2007\)](#).

### Exemplo I [ [Kelton et al. \(1998\)](#) ]

Em sistemas *terminais*, como os resultados das replicações são independentes identicamente



distribuídos (*iid*) Kelton et al. (1998), pode-se elaborar um **intervalo de confiança** para o valor esperado  $\mu$  das replicações. Intervalos de confiança são medidos segundo (13):

$$\bar{X} \pm t_{n-1, 1-\alpha/2} \frac{s}{\sqrt{n}} \quad (13)$$

onde  $\bar{X}$  é a média amostral,  $s$  é o desvio padrão amostral,  $n$  é o número de replicações e  $t_{n-1, 1-\alpha/2}$  é o ponto crítico superior  $1 - \alpha/2$  da distribuição  $t$ -student com  $n - 1$  graus de liberdade. A distribuição  $t$ -student aparece naturalmente em problemas de se determinar a média de uma população (que segue a distribuição normal) a partir de uma amostra. Neste problema, não se sabe qual é a média ou o desvio padrão da população, mas ela deve ser normal, por serem *iid*.

Por exemplo, se em  $n = 5$  replicações de simulação, obtém-se a média amostral de um indicador de desempenho  $\bar{X} = 3,80$  min e desvio padrão amostral  $s = 1,64$  min, podemos ver, pela Figura 14, que para um  $t = 2,776$ , a distribuição  $t$ -student com 4 (ou seja,  $n - 1$ ) graus de liberdade representa, por um intervalo de confiança de  $1 - \alpha/2 = 95\%$  (ou seja,  $\alpha = 0,05$ ), o valor da média amostral. Dessa forma, há a probabilidade de 95% do valor da média amostral estar no intervalo  $3,80 \pm 2,04$ , conforme a equação (14): Como percebemos, a metade da largura (*half-width*) deste intervalo é muito grande, se comparado ao valor central, ou seja, a *precisão* não é boa. Isto pode ser melhorado aumentando o número de replicações de simulação Kelton et al. (1998).

$$\bar{X} \pm t_{n-1, 1-\alpha/2} \frac{s}{\sqrt{n}} = 3,80 \pm 2,776 \frac{1,64}{\sqrt{5}} = 3,80 \pm 2,04 \quad (14)$$

Para aumentar a *precisão* e reduzir o valor da metade da largura do intervalo  $h$  (*half-width*) para um valor menor, determina-se um valor aproximado para o **número mínimo de replicações**  $n$ . Para isso, adota-se a fórmula (15), onde  $n_0$  é o número inicial de replicações e  $h_0$  é a *meia-largura* que se tem para a quantidade atual de replicações:

$$n \cong n_0 \frac{h_0^2}{h^2} \quad (15)$$

Por exemplo, uma simulação usou  $n_0 = 10$  replicações de um modelo e obteve a *meia-largura* do intervalo de confiança de 95% de  $h_0 = 1.573,73$ . Para obter mais precisão e reduzir a *meia-largura* do intervalo de confiança para  $h = 500$ , precisa-se de aproximadamente 99 replicações, como visto na equação (16):

$$n \cong n_0 \frac{h_0^2}{h^2} = n \cong 10 \frac{(1.573,73)^2}{(500)^2} = 99,1 \quad (16)$$

Em sistemas de *estado-estacionário* as rodadas de simulação precisam ser longas. Nestas simulações, os estados iniciais são tendenciosos (errados), portanto, pode-se adotar as estratégias de (i) iniciar com um número de entidades e estados de máquinas, (ii) rodar a simulação por um tempo tão grande que as estatísticas do estado inicial se tornem insignificativas ou (iii) computar estatísticas após um tempo de aquecimento (*warm-up*) em que o sistema aparente estável.

## Exemplo II [ Chwif and Medina (2006) ]

Para avaliarmos as medidas, precisamos entender as variáveis de saída do problema. Neste exemplo do número médio de clientes na fila, devemos perguntar:

1. Qual a *confiança estatística* dessa variável? (intervalo de confiança - probabilidade).
2. Qual a *precisão*? (tamanho do intervalo).

<b>Unicaudal</b>	<b>75%</b>	<b>80%</b>	<b>85%</b>	<b>90%</b>	<b>95%</b>	<b>97,5%</b>	<b>99%</b>	<b>99,5%</b>	<b>99,75%</b>	<b>99,9%</b>	<b>99,95%</b>
<b>Bicaudal</b>	<b>50%</b>	<b>60%</b>	<b>70%</b>	<b>80%</b>	<b>90%</b>	<b>95%</b>	<b>98%</b>	<b>99%</b>	<b>99,5%</b>	<b>99,8%</b>	<b>99,9%</b>
<b>1</b>	1,000	1,376	1,963	3,078	6,314	12,71	31,82	63,66	127,3	318,3	636,6
<b>2</b>	0,816	1,061	1,386	1,886	2,920	4,303	6,965	9,925	14,09	22,33	31,60
<b>3</b>	0,765	0,978	1,250	1,638	2,353	3,182	4,541	5,841	7,453	10,21	12,92
<b>4</b>	0,741	0,941	1,190	1,533	2,132	2,776	3,747	4,604	5,598	7,173	8,610
<b>5</b>	0,727	0,920	1,156	1,476	2,015	2,571	3,365	4,032	4,773	5,893	6,869
<b>6</b>	0,718	0,906	1,134	1,440	1,943	2,447	3,143	3,707	4,317	5,208	5,959
<b>7</b>	0,711	0,896	1,119	1,415	1,895	2,365	2,998	3,499	4,029	4,785	5,408
<b>8</b>	0,706	0,889	1,108	1,397	1,860	2,306	2,896	3,355	3,833	4,501	5,041
<b>9</b>	0,703	0,883	1,100	1,383	1,833	2,262	2,821	3,250	3,690	4,297	4,781
<b>10</b>	0,700	0,879	1,093	1,372	1,812	2,228	2,764	3,169	3,581	4,144	4,587
<b>11</b>	0,697	0,876	1,088	1,363	1,796	2,201	2,718	3,106	3,497	4,025	4,437
<b>12</b>	0,695	0,873	1,083	1,356	1,782	2,179	2,681	3,055	3,428	3,930	4,318
<b>13</b>	0,694	0,870	1,079	1,350	1,771	2,160	2,650	3,012	3,372	3,852	4,221
<b>14</b>	0,692	0,868	1,076	1,345	1,761	2,145	2,624	2,977	3,326	3,787	4,140
<b>15</b>	0,691	0,866	1,074	1,341	1,753	2,131	2,602	2,947	3,286	3,733	4,073
<b>16</b>	0,690	0,865	1,071	1,337	1,746	2,120	2,583	2,921	3,252	3,686	4,015
<b>17</b>	0,689	0,863	1,069	1,333	1,740	2,110	2,567	2,898	3,222	3,646	3,965
<b>18</b>	0,688	0,862	1,067	1,330	1,734	2,101	2,552	2,878	3,197	3,610	3,922
<b>19</b>	0,688	0,861	1,066	1,328	1,729	2,093	2,539	2,861	3,174	3,579	3,883
<b>20</b>	0,687	0,860	1,064	1,325	1,725	2,086	2,528	2,845	3,153	3,552	3,850

**Figura 14:** Uma distribuição t-student bicaudal com 4 graus de liberdade e nível de confiança de 95% deve ter  $t$  de 2,776, ou seja,  $-\infty < t < 2,776$  é de 95%

O *Intervalo de Confiança* (IC) é representado das formas (17) e (18):

$$P(\bar{X} - h \leq \mu \leq \bar{X} + h) = 1 - \alpha \quad (17)$$

$$\bar{X} \pm -t_{n-1, 1-\alpha/2} \frac{s}{\sqrt{n}} \quad (18)$$

Onde:

$\bar{X}$	Média amostral da medida de desempenho
$h$	Precisão (metade do intervalo)
$\mu$	Média real da população
$t_{n-1, 1-\alpha/2}$	Distribuição t-student
$s$	Desvio padrão amostral
$n$	Tamanho da amostra

Neste exemplo, uma rodada de simulação com 10 replicações foi usada para realizar uma amostra-piloto. A medida de desempenho (número médio de pessoas em fila) possui média amostral  $\bar{X}$  de 1,54 pessoas e desvio-padrão amostral  $s$  de 2,034. A *precisão*  $h$  (metade

do intervalo) e o intervalo de confiança (IC) da média para um *nível de confiança* de 99%, 95%, 90% e 80% são apresentados na Tabela 5:

Replicações $n$	Confiança $100(1 - \alpha)\%$	$\alpha$	t-student $t_{n-1, 1-\alpha/2}$	Precisão (metade do intervalo) $h = t_{n-1, 1-\alpha/2} \frac{s}{\sqrt{n}}$	Int. de confiança para $\bar{X} = 1,54$
10	99%	0,01	3,25	2,09	$-0,55 \leq \mu \leq 3,63$
10	95%	0,05	2,26	1,45	$-0,09 \leq \mu \leq 3,00$
10	90%	0,10	1,83	1,18	$-0,37 \leq \mu \leq 2,72$
10	80%	0,20	1,38	0,89	$0,65 \leq \mu \leq 2,43$

**Tabela 5:** Confiança estatística diminui à medida que a precisão aumenta

Caso se deseje aumentar a *precisão* devemos aumentar o tamanho da amostra (replicação). Nesse caso, aumentamos a *amostra-piloto* para 20 replicações. A medida de desempenho (número médio de pessoas em fila) possui média amostral  $\bar{X}$  de 1,08 pessoas e desvio-padrão amostral  $s$  de 1,49. O resultado é apresentado na Tabela 6:

Replicações $n$	Confiança $100(1 - \alpha)\%$	$\alpha$	t-student $t_{n-1, 1-\alpha/2}$	Precisão (metade do intervalo) $h = t_{n-1, 1-\alpha/2} \frac{s}{\sqrt{n}}$	Int. de confiança para $\bar{X} = 1,08$
20	99%	0,01	2,86	0,95	$0,13 \leq \mu \leq 2,03$
20	95%	0,05	2,09	0,70	$0,39 \leq \mu \leq 1,78$
20	90%	0,10	1,73	0,57	$0,51 \leq \mu \leq 1,66$
20	80%	0,20	1,33	0,44	$0,64 \leq \mu \leq 1,52$

**Tabela 6:** A precisão aumenta à medida que o tamanho da amostra aumenta

Para determinar o número de amostras ( $n^*$  replicações) para obter uma *precisão*  $h^*$  sob *confiança estatística*  $100(1 - \alpha)\%$ , usamos a fórmula (19):

$$n^* = \left\lceil n \left( \frac{h}{h^*} \right)^2 \right\rceil \quad (19)$$

Neste exemplo, queremos determinar uma *precisão* de  $h^*$  para 0,5 sob um nível de *confiança estatística* de 99%. Para isso, usamos os resultados da *amostra-piloto* (ou seja,  $n = 20$  e  $h = 0,95$ ). A equação (20) determina que são necessárias 73 replicações para obter este resultado:

$$n^* = \left\lceil 20 \left( \frac{0,95}{0,5} \right)^2 \right\rceil = \lceil 72,2 \rceil = 73 \quad (20)$$

Para  $n = 73$ , temos:

$$\begin{aligned} \bar{X} &= 0,76 \\ s &= 1,40 \\ t_{73-1, 1-(0,01/2)} &= 2,65 \\ h = t_{73-1, 1-(0,01/2)} \frac{1,40}{\sqrt{73}} &= 0,43 \end{aligned}$$

A precisão obtida (de 0,43) é menor que 0,5, portanto, podemos construir o intervalo:

$$\bar{X} - h \leq \mu \leq \bar{X} + h \quad 0,76 - 0,43 \leq \mu \leq 0,76 + 0,43 \quad 0,33 \leq \mu \leq 1,19 \quad (21)$$

Assim, podemos afirmar com probabilidade de 99% que o intervalo  $[0,33; 1,19]$  contém o número *médio* de pessoas em fila.

## 8 Análise dos resultados de uma simulação

Referência	Capítulos	Páginas	Assunto
Chwif and Medina (2006)	[06.0.0]	[101,138]	Análise dos resultados
Law (2007)	[09.0.0]	[485,540]	Análise modelo único
Pidd (2004)	[11.0.0]	[208,210]	Princípios em output
Fishman (2001)	[00.0.0]	[000,000]	X

Muito esforço é feito para elaborar a programação de modelos de simulação, mas pouco esforço é feito para avaliar os resultados dos mesmos. Frequentemente, avalia-se de forma arbitrária os resultados dos modelos e adota-se como verdadeiro as sugestões do modelo para uma operação real. Além disso, a programação do modelo pode ser baseada em um documento de premissas de projeto. Para avaliação, adota-se uma única rodada de simulação. Como os dados são resultantes de distribuições de probabilidade geradas por números aleatórios, estes resultados são errôneos. Portanto, técnicas estatísticas adequadas devem ser utilizadas para avaliar os resultados destes sistemas.

Processos estocásticos podem ser entendidos como uma coleção de variáveis aleatórias similares em um espaço amostral comum. Assim, valores possíveis das variáveis representam o *estado-do-espaco*, que podem ser de tempo discreto  $(X_1, X_2, \dots)$  ou tempo contínuo  $(X_t, t \geq 0)$ . Para fazer inferências sobre processos estocásticos, precisamos assumir uma *covariância-estacionária* (caso contrário não seria possível fazer análises estatísticas), ou seja,  $\mu$  e  $\sigma^2$  são estacionários ao longo do tempo e a covariância e correlação entre os processos devem existir, ou seja, serem diferentes de zero Law (2007):

$$\begin{aligned}
 \mu_i &= \mu & i &= 1, 2, \dots \\
 \sigma_i^2 &= \sigma^2 & i &= 1, 2, \dots \\
 C_{i,i+j} &= \text{cov}(X_i, X_{i+j}) & \text{iid em } i, j &= 1, 2, \dots \\
 \rho_j &= \frac{C_{i,i+j}}{\sqrt{\sigma_i^2 \sigma_{i+j}^2}} = \frac{C_j}{\sigma^2} & j &= 1, 2, \dots
 \end{aligned}
 \tag{22}$$

A simulação, na verdade, é um *experimento de amostragem estatística baseada em computadores*, portanto, é necessário utilizar técnicas estatísticas para entender seus resultados. Além disso, simulações são processo não-estacionários e auto-correlacionados, portanto, técnicas de estatística clássica, baseadas em eventos *iid*, não são diretamente aplicáveis.

Rodadas de simulação são geradas por diferentes distribuições de probabilidade de *input*, por meio de amostragem de dados diferentes. Seja uma simulação com  $n$  ocorrência de eventos e  $m$  rodadas, ou replicações, uma variáveis de desempenho  $Y_{m,n}$  é representada pela matriz (23):

$$Y_{m,n} = \begin{pmatrix} y_{1,1} & y_{1,2} & \cdots & y_{1,n} \\ y_{2,1} & y_{2,2} & \cdots & y_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ y_{m,1} & y_{m,2} & \cdots & y_{m,n} \end{pmatrix}
 \tag{23}$$

Assim, temos que  $i = 1, \dots, n$  não são *iid*, mas entre as replicações  $j = 1, \dots, m$  temos

independência entre as rodadas. Portanto, em uma análise de *output* podemos usar as observações  $y_{ij}$  para fazer inferências sobre as variáveis aleatórias  $Y_1, \dots, Y_n$ , pois  $\bar{y}_{i(m)} = \frac{\sum_j y_{ij}}{m}$  é um estimador não-tendencioso de  $E(Y_i)$ . Dentre os métodos estatísticos usados para comparar os resultados de uma simulação com observações do sistema real, enumeramos três Law (2007):

1. **Abordagem por inspeção:** A comparação de 2 conjuntos de amostras, por meio de testes estatísticos clássicos como chi-quadrado e Kolmogorov-Smirnov, é insuficiente porque os sistemas reais são *não estacionários* e há *auto-correlação* entre os processos. Sugere-se usar os dados históricos da operação real no sistema de simulação e comparar os desvios de *output* do modelo com o real. Este deve ser bem menor que o processo por amostragem.
2. **Abordagem por intervalo de confiança:** Esta abordagem é baseada em dados independentes. Esta forma é mais confiável e factível de se comparar os resultados e obter os *outputs i.i.d.* de ambos sistemas (modelo e operação real). Determina-se os valores médios do modelo de simulação  $\mu_x = E(X_j)$  e do sistema real  $\mu_y = E(Y_j)$  e constrói-se um intervalo de confiança  $\xi = \mu_x - \mu_y$ . Esta estratégia é melhor do que realizar testes de hipótese ( $H_o : \mu_x = \mu_y$ ), que seriam facilmente rejeitadas (falsa), porque modelos são aproximações da realidade, sempre.
3. **Abordagem por séries temporais:** Nesta abordagem, conjuntos unitários de *output* podem ser comparados por um número finito de realizações. Por exemplo: 200 *outputs* de de ambos sistemas (modelo e operação real) são comparados um a um em uma escala temporal e sua *auto-correlação* dos processos são avaliadas.

## 8.1 Sistemas Terminais

A avaliação de *sistemas terminais* é feita em 7 etapas:

1. Estabelecer *medidas de desempenho* adequadas.
2. Escolher a *precisão e confiança estatística* que se pretende trabalhar.
3. Definir o *tempo* de simulação.
4. Construir uma *amostra-piloto*. Estimar o *intervalo de confiança*.
5. Determinar o número de *replicações* ( $n^*$ ) necessárias.
6. *Rodar* o modelo novamente com  $n^*$  replicações.
7. Calcular o *novo* intervalo de confiança.

## 8.2 Sistemas Não-Terminais

Em *sistemas não-terminais* a resposta para a pergunta: "*por quanto tempo cada replicação deve ser executada?*" é até que os dados atinjam o *estado permanente*. As seguintes alternativas podem ser adotadas:

- Começar a simulação no estado permanente.
- Rodar a simulação por muito tempo.
- Eliminar os dados do período de aquecimento, ou *warm-up*. (Usa-se média-móvel para suavização)

Em seguida, a avaliação de *sistemas não-terminais* é feita em 4 etapas:

1. Estabelecer *medidas de desempenho* adequadas.
2. Escolher a *precisão e confiança estatística* que se pretende trabalhar.
3. Identificar o período de aquecimento, ou *warm-up*.
4. Determinar a *duração* da simulação.
  - (a) Tempo *curto*: Elaborar um conjunto de replicações (como no sistema terminal) e estabelecer o intervalo de confiança.
  - (b) Tempo *longo*: Determinar *lotes* de uma única replicação e estabelecer o intervalo de confiança.

**Exemplo:** Centro de Distribuição (Pág. 117 de [Chwif and Medina \(2006\)](#))

1. Medida de desempenho: tempo médio de entrega de produtos.
2. Precisão:  $h^* \leq 5$  minutos. Confiança estatística: 95% ( $\alpha = 0,05$ ).
3. Período de aquecimento, ou *warm-up* (visual): 700h.
4. Determinar a *duração* da simulação.
  - (a) Tempo *curto*: Elaborar um conjunto de replicações (como no sistema terminal) e estabelecer o intervalo de confiança.
  - (b) Tempo *longo*: Determinar *lotes* de uma única replicação e estabelecer o intervalo de confiança.

**(a) Amostra-piloto:** 20 replicações. *warm-up*: 700h.

Para  $n = 20$ , temos:

$$\begin{aligned}
 \bar{X} &= 372,3 \text{ minutos} \\
 s &= 14,4 \text{ minutos} \\
 t_{20-1,1-(0,05/2)} &= 2,09 \\
 h = t_{20-1,1-(0,05/2)} \frac{14,4}{\sqrt{20}} &= 6,73 \text{ (Não Ok)} \\
 \text{Intervalo de confiança} &= P[365,5; 379,0] = 0,95
 \end{aligned}$$

$$n^* = \left\lceil 20 \left( \frac{6,73}{5,0} \right)^2 \right\rceil = \lceil 36,1 \rceil = 37 \quad (24)$$

Após 37 replicações...

$$\begin{aligned}
 \bar{X} &= 375,0 \text{ minutos} \\
 s &= 15,1 \text{ minutos} \\
 t_{37-1,1-(0,05/2)} &= 2,05 \\
 h = t_{37-1,1-(0,05/2)} \frac{15,1}{\sqrt{37}} &= 5,1 \text{ (Não Ok)} \\
 \text{Intervalo de confiança} &= P[370,0; 380,1] = 0,95
 \end{aligned}$$

$$n^* = \left\lceil 37 \left( \frac{5,1}{5,0} \right)^2 \right\rceil = \lceil 44,8 \rceil = 45 \quad (25)$$

Novamente, após 45 replicações...

$$\begin{aligned}\bar{X} &= 374,7 \text{ minutos} \\ s &= 15,2 \text{ minutos} \\ t_{45-1,1-(0,05/2)} &= 2,03 \\ h = t_{45-1,1-(0,05/2)} \frac{15,2}{\sqrt{45}} &= 4,6 \text{ (Ok)} \\ \text{Intervalo de confiança} &= P[370, 1; 379, 3] = 0,95\end{aligned}$$

(b) **Tamanho de lotes:**  $k$  observações. (correlação  $\approx$  entre os lotes).

**Correlograma:** Função de auto-correlação  $\rho_k$ ,  $p/ k = 1, 2, 3, \dots$  (Lotes  $\geq 10k$ )

**Tamanho de lotes:** 170 observações.

**Número de lotes:** 50 lotes (Recomenda-se  $\geq 30$  lotes).

**Taxa de observação:**  $\frac{1obs.}{4h} = \frac{0,25obs}{h}$ , então, o tempo de geração de lotes (TG):

**TG:**  $\frac{obs}{h}(50lotes) = 34.000$  horas. Então o tempo de simulação (TS):

**TS:**  $warm-up + TG + 0,01(TG)$ , onde  $\alpha = 0,01$  é um coeficiente de segurança.

**TS:**  $700 + 34.000 + (0,01)34.000 = 35.040$  horas (Tempo de "1 replicação").

Cada um dos 50 lotes (com 170 observações cada) é tratado como uma nova replicação independente. Nesse caso,

$$\begin{aligned}\bar{X} &= 367,2 \text{ minutos} \\ s &= 43,0 \text{ minutos} \\ t_{50-1,1-(0,05/2)} &= 2,05 \\ h = t_{50-1,1-(0,05/2)} \frac{43,0}{\sqrt{50}} &= 12,1 \text{ (Não Ok)} \\ \text{Intervalo de confiança} &= P[355, 1; 379, 3] = 0,95\end{aligned}$$

$$n^* = \left\lceil 50 \left( \frac{12,1}{5,0} \right)^2 \right\rceil = \lceil 311,0 \rceil = 311 \quad (26)$$

Após 311 replicações...

Novo tempo de simulação (TS): **TG:**  $\frac{170obs}{0,25obs/h}(311)$  lotes = 212.160 horas.

**TS:**  $warm-up + TG + 0,01(TG)$ , onde  $\alpha = 0,01$  é um coeficiente de segurança.

**TS:**  $700 + 212.160 + (0,01)212.160 = 214.982$  horas (= 313 lotes) ("1 replicação").

$$\begin{aligned}\bar{X} &= 376,2 \text{ minutos} \\ s &= 46,8 \text{ minutos} \\ t_{311-1,1-(0,05/2)} &= 1,96 \\ h = t_{311-1,1-(0,05/2)} \frac{46,8}{\sqrt{311}} &= 5,2 \text{ (Não Ok)} \\ \text{Intervalo de confiança} &= P[371, 0; 381, 4] = 0,95\end{aligned}$$

Nova alternativa: Lotes = 20  $k = 340$ , ou seja (20)(17).

Nesse caso, com  $TG = 212.160 = \frac{340_{obs}}{\frac{1ote}{0,25_{obs}}}(x \text{ lotes})$ , temos  $x = 160$  lotes.

$$\begin{aligned}\bar{X} &= 376,2 \text{ minutos (igual o anterior)} \\ s &= 30,0 \text{ minutos (menor)} \\ t_{160-1,1-(0,05/2)} &= 1,98 \\ h = t_{160-1,1-(0,05/2)} \frac{30,0}{\sqrt{160}} &= 4,7 \text{ (Ok)} \\ \text{Intervalo de confiança} &= P[371, 5; 380, 9] = 0,95\end{aligned}$$

### 8.3 Comparar alternativas

Quando há o *mesmo número de replicações* realiza-se o **teste  $t$  com amostras emparelhadas**. Nesse caso, seja  $x_{1i}$  e  $x_{2i}$  as médias de  $i = 1, \dots, m$  replicação da rodada de simulação 1 e 2, respectivamente. A avaliação é feita em 4 etapas:

1. Estabelecer a diferença:  $d_i = x_{1i} - x_{2i}, i = 1, \dots, m$ .
2. Calcule a *média* e o *desvio-padrão* de  $d$ :

$$\bar{d} = \frac{\sum_{i=1}^m d_i}{m} \quad s_d = \sqrt{\frac{\sum_{i=1}^m (d_i - \bar{d})^2}{m - 1}} \quad (27)$$

3. Construir o *intervalo de confiança*  $IC[\theta_1, \theta_2]$  para  $\bar{d}$ , onde  $\theta_1 = \mu - h$  e  $\theta_2 = \mu + h$ . Por exemplo, para uma confiança estatística de 95%, a precisão  $h$  (metade do intervalo) fica:

$$h = t_{m-1,1-(0,05/2)} \frac{s_d}{\sqrt{m}} \quad (28)$$

4. (a) Se  $\theta_1 \leq 0$  e  $\theta_2 \geq 0$ , então não se pode concluir nada.  
 (b) Se  $\theta_1 \geq 0$  e  $\theta_2 \geq 0$ , então a alternativa 01 possui a média maior.  
 (c) Se  $\theta_1 \leq 0$  e  $\theta_2 \leq 0$ , então a alternativa 02 possui a média maior.

Para o caso de amostras ou *replicações de tamanho diferente*, calcula-se a média e desvio-padrão para cada alternativa, separadamente.



## 9 Técnicas de redução de variância

Referência	Capítulos	Páginas	Assunto
<a href="#">Banks and Carson (1984)</a>	[12.1.3]	[456,462]	Common Random Numbers (CRN)
<a href="#">Chwif and Medina (2006)</a>	[06.0.0]	[137,138]	Variáveis anti-téticas
<a href="#">Kelton et al. (1998)</a>	[11.4.1]	[482,489]	Arena (CRN)
<a href="#">Law (2007)</a>	[11.0.0]	[577,609]	CRN e Variáveis anti-téticas
<a href="#">Pidd (2004)</a>	[11.4.0]	[218,226]	CRN e Variáveis anti-téticas
<a href="#">Fishman (2001)</a>	[04.2.2]	[145,149]	CRN e outros

Em simulações os dados de entrada são variáveis aleatórias provenientes de uma distribuição de probabilidade. Por isso, os dados de saída também são aleatórios (RIRO: *random in, random out*), ou seja, há *variância* associada aos dados de saída de modelos estocásticos. Quanto mais variância, menos preciso é o resultado. Uma forma de reduzi-la é simular por mais tempo e mais replicações [Kelton et al. \(1998\)](#).

À medida em que o número de simulações aumenta, o sistema converge para o estado estacionário, assim, pela lei dos grandes números, aumenta a estimativa da variável convergir para seu valor verdadeiro. O erro da estimativa em um número finito de simulações pode ser obtido pelo teorema central do limite. Apesar da simplicidade, ao utilizar números pseudo-aleatórios, a simulação demanda muito tempo computacional para obter a convergência da variável. Assim, recorre-se a técnicas de redução de variância afim de aumentar a eficiência do método. Estas auxiliam a determinar o número mínimo de replicações de forma a reduzir o intervalo de confiança dos resultados de saída [Kelton et al. \(1998\)](#).

Técnicas de *redução de variância* é dependente da correlação entre médias amostrais (replicações). Baixa correlação diminui o potencial de redução de variância [Fishman \(2001\)](#). Assim, estas técnicas visam aumentar a taxa de convergência da estimativa de uma variável por meio da diminuição da variância de amostras geradas por números pseudo-aleatórios. Dentre as técnicas estão: o uso de probabilidade condicional, números anti-téticos, variáveis de controle, amostragem estratificada, hipercubo latino e amostragem por importância.

Em *softwares* de simulação, como o Arena<sup>TM</sup> [Automation \(2007\)](#), a redução de variância pode ser obtida sincronizando a geração de números aleatórios para modelos semelhantes que avaliam diferentes estratégias de operações. A estratégia, chamada *amostragem correlacionada* ou *common random numbers* (CRN) [Banks and Carson \(1984\)](#) [Kelton et al. \(1998\)](#) [Law \(2007\)](#) é requisitada quanto estudos de simulação envolvem mais de uma *alternativa* de um modelo. Dessa forma, diferentes alternativas podem ser carregadas pela mesma semente externa de número aleatório. Os números aleatórios das replicações podem ser:

1. Independentes
2. Comuns e sem sincronização (gerados sob demanda para cada replicação)
3. Comuns e sincronizados (mesmos números para cada replicação)

Os **números comuns e sincronizados** geralmente produzem melhores resultados em termos de redução de variância entre as replicações [Banks and Carson \(1984\)](#). A estratégia tira proveito da correlação das amostras geradas e, assim como nos *variáveis antitéticas*, no entanto, é necessário que haja seleção cuidadosa dos parâmetros e elementos do modelo para que haja uma sincronização eficiente na geração dos números aleatórios.

Considere  $X$  e  $Y$  as estimativas dos parâmetros de saída obtidas após, 10 replicações, por exemplo. Nesse caso, busca-se estimar:

$$E[X] - E[Y] = E[X - Y] \quad (29)$$

Se as variáveis são independentes, a variância entre amostras é dada por

$$V[X - Y] = V[X] + V[Y] \quad (30)$$

No entanto, usando CRN sincronizado, o que está sendo feito é induzindo a correlação positiva. Como a correlação possui o mesmo sinal da covariância, o estimador da variância fica:

$$V[X - Y] = V[X] + V[Y] - 2Cov(X, Y) \quad (31)$$

**Números anti-téticos** tentam reduzir a variância ao introduzir pares de números aleatórios negativamente correlacionados pertencentes à mesma distribuição de probabilidade.

Por exemplo, sejam duas sequências de números aleatórios distribuídos uniformemente entre 0 e 1, ou seja,  $U[0, 1]$ . A sequência-01:  $r_1, r_2, \dots, r_m$ . Os *números anti-téticos* são igualmente aleatórios e formam a sequência-02:  $1 - r_1, 1 - r_2, \dots, 1 - r_m$ , no entanto, as variáveis não são independentes, de fato, elas são negativamente correlacionadas. Se em uma replicação, por exemplo, o "tempo de atendimento" é alto, na outra, ele será obrigatoriamente baixo [Chwif and Medina \(2006\)](#).

Considere  $X$  e  $Y$  as estimativas dos parâmetros de saída obtidas quando usamos as sequência-01 e sequência-02, respectivamente. Nesse caso, a média do parâmetro pode ser obtida pela média de  $X$  e  $Y$  representada em [32](#):

$$E[X + Y] = E[X] + E[Y] \quad (32)$$

No entanto, como as variáveis não são independentes, a variância do parâmetro é estimado por [33](#):

$$V[X + Y] = V[X] + V[Y] + 2Cov(X, Y) \quad (33)$$

Onde  $Cov(X, Y)$  é a covariância entre  $X$  e  $Y$ . Se a covariância entre  $X$  e  $Y$  é negativa, podemos concluir que a variância entre  $X+Y$  será menor. No entanto, em uma simulação, não é necessário calcular  $Cov(X, Y)$ , podemos calcular a variância de  $X+Y$  diretamente.

**Variáveis de controle** usam os valores dos erros nas estimativas de quantidades conhecidas para reduzir os erros nas estimativas de quantidades desconhecidas. O valor sobre as quantidades conhecidas elege a variável de controle, cuja solução analítica é conhecida.

A **amostragem estratificada** consiste em dividir a distribuição em  $n$  intervalos de mesma probabilidade para, posteriormente, realizar simulações dentro de cada intervalo de modo que  $\frac{1}{n}$  da amostra esteja dentro de cada um dos estratos. O procedimento garante que a distribuição de probabilidade da amostra fique mais próxima da distribuição desejada. Uma extensão desta estratégia para múltiplas dimensões é o hipercubo latino. Por outro lado, a *amostragem por importância* tenta reduzir a variância alterando a medida de probabilidade dando mais pesos a resultados considerados mais importantes ao fim que se propõe.

## 10 Planejamento e análise de experimentos de simulação

Referência	Capítulos	Páginas	Assunto
<a href="#">Chwif and Medina (2006)</a>	[07.0.0]	[139,154]	Experimentos e otimização
<a href="#">Law (2007)</a>	[12.0.0]	[619,643]	Design Experimental. Metamodelos
<a href="#">Pidd (2004)</a>	[11.0.0]	[208,210]	Princípios em output
<a href="#">Banks and Carson (1984)</a>	[12.0.0]	[450,486]	Análise comparativa fatorial
<a href="#">Kelton et al. (1998)</a>	[00.0.0]	[000,000]	X
<a href="#">Fishman (2001)</a>	[00.0.0]	[000,000]	X

Devemos ter 3 princípios em mente ao analisarmos os resultados de uma simulação por eventos discretos [Pidd \(2004\)](#):

1. A simulação é um processo complexo de experimentos de amostragens combinados, portanto, pode ser difícil entender os efeitos dos resultados dessa combinação.
2. Como a simulação ocorre por um tempo do modelo, muitas variáveis de saída tomam a forma de séries temporais, portanto, as observações podem incluir correlações seriais, ou seja, as observações podem não ser estatisticamente independentes, e análises estatísticas clássicas podem não ser adequadas.
3. Apesar das similaridades, há uma diferença entre simulação e *design* de experimentos (DOE): os resultados de uma simulação envolvem o tempo e as operações internas de um sistema, enquanto que em *design* de experimentos se faz uma avaliação estática dos resultados e não se pode tirar vantagens pelos conhecimentos internos dos processos, que contribuem com técnicas de redução de variância.

Em *design* de experimentos, se temos apenas um fator a ser avaliado, simplesmente rodamos a simulação em diversos níveis e determinamos o intervalo de confiança esperado para obter estabilidade na variável de desempenho (*output*) a ser observada. Se temos pelo menos 2 fatores independentes, a quantidade mínima de rodadas é de  $2^n$  replicações. Se temos mais que dois fatores, devemos determinar como os 2 fatores se relacionam com os fatores restantes do modelo. Uma forma de realizar este experimento é fixar os  $k - 1$  fatores complementares (que não serão avaliados) e rodar o modelo para avaliar o desempenho do sistema em relação ao fator de interesse. Em seguida, fazer o mesmo procedimento para cada  $k - 1$  fatores restantes, ou seja, um fator a cada momento, até se obter uma visão geral do comportamento do sistema. E estratégia é ineficiente, pois a avaliação é independente.

Uma estratégia mais econômica de avaliar os fatores é combiná-los em duplas por um processo chamado *design* fatorial de  $2^k$  em que requer a escolha de 2 níveis de combinação a ser simulado. Os pontos são previamente escolhidos de forma arbitrária, seguindo os sentimentos de especialistas, gerando possíveis valores quantitativos. Por utilizar apenas 2 fatores, espera-se uma aproximação de respostas lineares ou uma função de resultado monotônica. Os resultados podem ser apresentados em forma tabular facilitando a exposição das relações que geram resultados melhores ou piores que o cenário original [Law \(2007\)](#).

Um exemplo de *design* fatorial de  $2^k$  é um modelo de estoque (s,S), em que podemos determinar as políticas de estoque de segurança  $s$  ou o pedido ao cliente  $d = S - s$ , assim, temos dois fatores de entrada são alterados e avaliados simultaneamente por meio de uma variável de resposta financeira. A avaliação é feita em termos dos valores médios, variável

e intervalo de confiança. O resultado pode ser traçado por meio de equações lineares explicitando a combinação ideal dos dois fatores. A estratégia permite determinar uma política resultante melhor que a original Law (2007).

No entanto, se avaliarmos mais que dois fatores de experimento, a quantidade de experimentos cresce exponencialmente. Por exemplo, se avaliamos 11 fatores e 5 replicações precisamos simular  $2^{11} = 2.048$  experimentos e 10.240 experimentos no total. Se cada experimento durasse 1 minuto, levaríamos aproximadamente 1 semana para rodar a simulação completa. Por isso, uma abordagem é avaliar  $2^{k-p}$  alternativas, chamada *design* fatorial fracionais, onde  $p$  são valores de entrada impossíveis de serem combinados com outros fatores na realidade Law (2007).

Uma estratégia mais avançada é determinar quais dos fatores de entrada são os mais importantes para as variáveis de desempenho de interesse. Tal estratégia é implementada via *meta-modelos*, onde equações de regressão são utilizadas para prever os valores de respostas de combinações de interesse. Abordar todo espaço de combinações possíveis seria proibitivo em termos computacionais. Os *meta-modelos* visam otimizar uma superfície de resposta que maximize ou minimize um valor esperado de saída. Dessa forma, ao avaliar o modelo de estoques, um *meta-modelos* desenharia uma superfície de soluções baseada nas boas combinações dos fatores de entrada Law (2007).

A mais avançada estratégia de simulação está relacionada à seleção de valores de entrada que maximize ou minimize os resultados de desempenho dos valores de saída: **Otimização para simulação** Law (2007). Nesse caso, busca-se determinar, *todos* os possíveis valores de entrada que otimizariam uma função de valor esperado de saída. Os parâmetros de entrada podem ser variáveis controláveis:

1. *Quantitativas discretas*: número de máquinas, estações de trabalho.
2. *Quantitativas contínuas*: *lead-time* de suprimentos ou *throughput* de produtos.
3. *Qualitativas de escolha*: disciplina de filas.

De forma geral, o foco deste processo está relacionado à seleção de variáveis quantitativas controláveis. Diferentemente das outras estratégias  $2^n$  ou  $2^{n-p}$ , não se busca otimizar o resultado final com base em valores de entrada candidatos, previamente selecionados.

## 11 Otimização baseada em simulação

Referência	Capítulos	Páginas	Assunto
<a href="#">Chwif and Medina (2006)</a>	[07.0.0]	[139,154]	Experimentos e otimização
<a href="#">Law (2007)</a>	[12.5.0]	[655,658]	Simulação via otimização
<a href="#">Kelton et al. (1998)</a>	[00.0.0]	[000,000]	X
<a href="#">Fishman (2001)</a>	[00.0.0]	[000,000]	X

Por muitas décadas, a simulação tem sido usada como uma poderosa ferramenta de estatística *descritiva* para modelagem e análise de uma ampla variedade de sistemas complexos reais. Com os desenvolvimentos recentes, a simulação pode ser usada como uma ferramenta *prescritiva* para sistemas de apoio à decisão.

A **otimização para simulação** busca selecionar, dentre todas possíveis configurações, aqueles parâmetros de entrada que maximizam ou minimizam uma função de otimização. As variáveis de entrada devem ter seus limites (*lower-bound* e *upper-bound*) determinados ou podem fazer parte de uma relação linear, determinando uma restrição no modelo. O objetivo é maximizar um valor esperado de saída.

Resolver estes problemas demandam buscas muito extensas consumindo muito recurso computacional. Por isso, diversas técnicas de otimização são usadas para estes métodos, embora não se tenha certeza do ótimo global, pois são métodos de busca para uma quantidade limitada de condições de entrada. Adotam-se: meta-heurísticas, algoritmos genéticos, etc. A interação do sistemas se dá entre um pacote de otimização e um pacote de simulação. O pacote de simulação especifica as condições e a configuração do sistema, enquanto que o pacote de otimização otimiza os parâmetros de entrada dado um conjunto de restrições pré-determinados. Até que a regra de parada seja atingida, os valores são atualizados e simulados novamente.

O objetivo é minimizar ou maximizar uma medida de desempenho de interesse sob perturbações de natureza estocástica. Se a função de desempenho é conhecida explicitamente, então técnicas analíticas, como programação matemática, podem ser usadas.

Dentre as características desejadas nestes pacotes de otimização para simulação estão: a velocidade de otimização, qualidade em termos de intervalos de confiança dos intervalos de saída, as restrições poderiam ser lineares e não-lineares, determinação dos possíveis limites das variáveis de saída, regras de parada em função de parâmetros de simulação, consideração das sementes dos modelos de simulação de forma a viabilizar a promoção de técnicas de redução de variância dos valores aleatórios gerados e a viabilidade de programação paralela dos computadores em rede [Law \(2007\)](#). Geralmente, a avaliação está interessada no comportamento de *estado-estacionário*, como nos 2 exemplos abaixo [Fu \(1994\)](#):

1. Em sistemas de filas, como a M/M/1, as medidas de desempenho podem ser: encontrar um tempo médio de serviço do servidor que minimiza a soma do tempo médio esperado dado um número de clientes e um custo de atendimento.
2. Em sistemas de controle de estoques, as medidas de desempenho podem ser: encontrar os valores de estoque de segurança ( $s$ ) e tamanho do pedido de ressuprimento ( $q=S-s$ ) que minimiza o custo de armazenamento, pedido e atraso (*backlogging*).

Reitera-se que antes de usar a simulação, modelos e métodos analíticos devem de usados para ganhar *insight*. Como toda técnica estatística, assume-se *independência* e *normalidade*,

o que presume o uso de técnicas de redução de variância. Técnicas de otimização para simulação são classificadas quanto ao espaço de busca (local x global), espaço de parâmetros (discreto x contínuo) e funções objetivas (única x múltipla) [Tekin and Sabuncuoglu \(2004\)](#).

### 11.1 Otimização local

#### [Discreto] Ranqueamento e Seleção

No *Ranqueamento e Seleção*, o processo é sequencial e realizado em dois estágios. O conceito de seleção correta de parâmetro de desempenho é usado, portanto, o objetivo é fazer uma decisão sobre algum critério. A variância não precisa ser igual, nem conhecida. Podem ser realizados por zona de indiferença ou seleção de sub-conjuntos [Fu \(1994\)](#).

O método de Ranqueamento e Seleção (R&S) pode ter duas abordagens: *Zona de Indiferença* e *Seleção de Subconjuntos*. Uma desvantagem é que o procedimento requer independência entre as variáveis. A hipótese de normalidade pode ser assumida quando técnicas de agrupamento em lotes são realizadas [Tekin and Sabuncuoglu \(2004\)](#).

**Zona de Indiferença:** garante que a medida de desempenho do parâmetro selecionado difere da solução ótima por um valor máximo  $\sigma$  com a probabilidade de pelo menos  $P^*$ . Estudos nesta área avaliam abordagens por dois-estágios ou completamente sequencial.

**Seleção de Subconjuntos:** busca selecionar subconjuntos que contém os melhores indicadores com a probabilidade de pelo menos  $P^*$ . Esta abordagem é útil quando a quantidade de escolhas é muito grande.

#### [Discreto] Comparação Múltipla

Em procedimentos de *Comparação Múltipla* a ideia é rodar uma quantidade de replicações para elaborar alguma inferência sobre a medida de desempenho de interesse por meio de intervalos de confiança. Se o intervalo de confiança não é suficientemente justo, mais replicações devem ser rodadas. Podem ser feitas comparações por pares (MCA) ou comparações com o melhor (MCB) [Fu \(1994\)](#).

A ideia é rodar uma quantidade de replicações e elaborar conclusões sobre uma medida de desempenho construindo intervalos de confiança. De forma geral, para os todos os pares de variáveis são computadas as diferenças entre as medidas de desempenho com níveis de intervalo de confiança  $(1 - \alpha)100\%$  para cada intervalo. O sistema selecionado é aquele que possui a diferença negativa com todos os outros pares. Comparações múltiplas podem ocorrer de três formas. Para estes, técnicas de redução de variância e amostragem por dois-estágios tem sido recentemente desenvolvido [Tekin and Sabuncuoglu \(2004\)](#):

1. Comparação de todos os pares
2. Comparações múltiplas com o melhor
3. Comparação múltipla com controle

#### [Discreto] Otimização Ordinal

*Otimização Ordinal* (OO) se concentra em encontrar um subconjunto de variáveis de *design* "bom o bastante" ao amostrar grande conjunto de soluções e avaliar pequenos conjuntos de variáveis de *design* [Tekin and Sabuncuoglu \(2004\)](#).

**[Discreto] Busca Aleatória**

Algoritmos de *Busca Aleatória* podem trabalhar em parâmetros com espaço infinito. Entradas de limites superiores e inferiores de cada fator de controle definem toda região de busca. A técnica seleciona pontos de forma aleatória. O procedimento finaliza quando uma quantidade de rodadas é completada. A convergência do método é fraca porque informações prévias não são usadas nas iterações [Tekin and Sabuncuoglu \(2004\)](#).

**[Discreto] Busca Simplex/Complex**

*Busca Simplex/Complex* constrói um simplex  $p$ -dimensional e escolhendo  $p + 1$  pontos extremos para uma função de  $p$  parâmetros e simula a resposta para cada ponto extremos (vértice). A cada iteração um novo ponto é adicionado ao simplex por meio de projeção e o pior ponto é retirado do conjunto. O procedimento continua até que o tamanho do conjunto se torne suficientemente pequeno [Tekin and Sabuncuoglu \(2004\)](#).

**[Discreto] Método do Fator Único**

É um método de busca direta que pode trabalhar em espaço infinito de parâmetros. Envolve movimentos coordenados enquanto os outros fatores são mantidos constantes [Tekin and Sabuncuoglu \(2004\)](#).

**[Discreto] Busca Padrão**

Também É um métodos de busca direta que pode trabalhar em espaço infinito de parâmetros. Por outro lado, a *Busca Padrão*, ao iniciar de um ponto inicial, verifica mudanças incrementais e melhora de forma variável o valor de resposta. O método é repetido para todas as variáveis até que uma nova configuração é obtida. São geralmente utilizados em conjunto com outros métodos [Tekin and Sabuncuoglu \(2004\)](#).

**[Contínuo] Metodologia de Superfície de Resposta**

*Metodologia de Superfície de Resposta* (MSR) é um método amplamente aceito e utilizado por ser uma metodologia geral de otimização para simulação. Sua maior vantagem é que o método usa ferramentas estatísticas bem conhecidas e dominadas, além de ser relativamente eficiente. Diversos exemplos de implementação em casos reais utilizam deste método, que se resume em uma classe de procedimentos que [Tekin and Sabuncuoglu \(2004\)](#):

1. Ajustam uma série de modelos de regressão para que o modelo de simulação avalie as variáveis em diversos pontos e
2. Otimizam a função de regressão resultante.

O algoritmo básico da Metodologia de Superfície de Resposta consiste de duas fases. Na **primeira fase** um modelo de primeira-ordem é ajustado para a superfície de resposta. Em seguida a direção de maior ganhos é estimada pelo modelo. Esta etapa é repetida até que a inclinação computada comece a se aproximar de zero, indicando que variáveis de *design* de primeira-ordem não se ajustam mais à sub-região. Na **segunda-fase**, uma superfície quadrática é ajustada usando variáveis experimentais de segunda-ordem. O ótimo é determinado a partir deste ajuste [Tekin and Sabuncuoglu \(2004\)](#).

**[Contínuo] Estimativa por Diferenciação Finita**

O método *baseado em gradiente* de *Estimativa por Diferenciação Finita* é baseado em determinar derivadas parciais de uma função de uma medida de desempenho com o objetivo de estimar o gradiente em cada ponto. Pelo menos  $p + 1$  pontos de avaliação de simulação são necessários. Para tornar a estimativa mais confiável, múltiplas observações de cada derivada podem ser necessárias [Tekin and Sabuncuoglu \(2004\)](#).

**[Contínuo] Análise de Perturbação**

O método *baseado em gradiente* de *Análise de Perturbação* é aplicado a problemas que satisfazem condições específicas. Pelo método, todos os gradientes de uma função objetivo são estimados em uma *única* rodada de simulação. Análises de Perturbação podem ser classificadas como [Tekin and Sabuncuoglu \(2004\)](#):

**Análise de Perturbação Finita (APF)** : É um procedimento heurístico feito para parâmetros discretos que aproximam as diferenças em uma medida de desempenho quando um parâmetro de desempenho *discreto* é perturbado por uma unidade.

**Análise de Perturbação Infinitesimal (API)** : É usada para obter derivadas de parâmetros *contínuos* e estimar todas as derivadas parciais em única rodada mantendo o controle das estatística de eventos específicos durante uma rodada de simulação.

**[Contínuo] Análise do Domínio da Frequência**

O método *baseado em gradiente* de *Análise do Domínio da Frequência* consiste em oscilar o valor de um parâmetro de acordo com uma função senoidal durante a simulação. A magnitude da variação da medida de desempenho indica sua sensibilidade relativa à medida de desempenho do parâmetro. O método possui algumas desvantagens: requer indexação cuidadosa da simulação em conjunto com a variação senoidal das variáveis de acordo com o índice temporal [Tekin and Sabuncuoglu \(2004\)](#).

**[Contínuo] Estimativas de Taxa de Possibilidade**

O método *baseado em gradiente* de *Estimativas de Taxa de Possibilidade* (ETP) diferencia a medida de probabilidade subjacente do sistema. O método assume que a medida de desempenho é um vetor aleatório com função de distribuição cumulativa  $F(X, .)$  e de densidade  $f(X, .)$  e a dependência de  $X$  ocorre por meio de um vetor aleatório  $Y$ . Diferenciando uma função de esperança de  $Y$  em relação a  $X$  permite estimar as derivadas das medidas de desempenho juntamente com a medida de desempenho em si [Tekin and Sabuncuoglu \(2004\)](#).

**[Contínuo] Aproximação Estocástica**

*Aproximação Estocástica* assume que o problema original dado por uma fórmula de minimizar uma medida de desempenho:  $\min H(X)$  pode ser resolvida determinando que o gradiente da função é igual a zero, ou seja,  $\nabla H(x) = 0$ . O método usa a fórmula recursiva (34):

$$X_{n+1} = \prod_{\phi} (X_n - a_n \nabla H_n) \quad (34)$$



Onde  $a_n$  é uma série de passos de valores reais.  $X_n$  é estimado no início da iteração  $n$ .  $\nabla H_n$  é uma estimativa do valor médio gradiente  $\nabla H(X_n)$ . À medida que  $n$  se aproxima ao infinito,  $X_n$  se aproxima de um valor cuja função teórica de regressão da variável estocástica é minimizada. Uma dificuldade do método é a grande quantidade de iterações que o método necessita para chegar ao ótimo [Tekin and Sabuncuoglu \(2004\)](#).

## 11.2 Otimização Global

### Algoritmos Evolucionários

*Algoritmos Evolucionários* (AE) são métodos de busca heurísticos que implementam ideias de processos evolutivos. Ao contrário de métodos tradicionais de solução única, AE trabalham em populações de soluções de forma que soluções pobres se tornam extintas, enquanto que boas soluções evoluem para alcançar o ótimo [Tekin and Sabuncuoglu \(2004\)](#).

O interesse crescente no uso de AE em otimização para simulação ocorre porque o método não necessita de condições restritivas de conhecimento prévio sobre o formato da superfície de resposta. De forma geral, Algoritmos Evolucionários de otimização para simulação são descritos em três passos:

1. Gere uma população de soluções;
2. Avalie estas soluções por meio de um modelo de simulação;
3. Faça seleção, aplique operadores genéticos para produzir nova prole, descendência, (ou soluções) e as insira na população;
4. Repita até que algum critério de parada seja alcançado.

### Busca Tabu

*Busca Tabu* (BT) é um procedimento de busca sob restrições, onde cada passo consiste em resolver problemas de otimização secundários. Em cada passo, o procedimento de busca omite um subconjunto de soluções do espaço de busca. Este subconjunto é alterado à medida em que o algoritmo evolui. Este é definido por soluções previamente consideradas, que são chamadas *condições tabu reinantes* [Tekin and Sabuncuoglu \(2004\)](#).

### Anelamento Simulado

Algoritmos de *Anelamento Simulado* (AS) começam com uma solução inicial, geralmente escolhida aleatoriamente. Uma solução vizinha, então, é gerada por um mecanismo adequado e a mudança na função objetivo é calculada. Se a redução ocorre (para um problema de minimização), a solução vizinha gerada substitui a solução atual. Se não há redução, então o algoritmo de AS aceita a solução com alguma probabilidade, de forma a evitar a fixação do ponto em um ótimo local [Tekin and Sabuncuoglu \(2004\)](#).

### Algoritmos Bayesianos/Amostragem

A metodologia *Bayesiana/Amostragem* (B/A) é uma estratégia de busca, onde a cada iteração a tentativa seguinte é escolhida para ser o ponto que maximiza a probabilidade de não exceder o valor prévio por meio de uma constante positiva  $\psi_n$ , dada pela relação (35):

$$P[H(X_{n+1}) \leq H(X_n) + \psi_n] \quad (35)$$

Para um problema de minimização, o método obtém escolhas nas áreas onde o desempenho médio avaliado pela simulação é baixo. Inicialmente  $\psi_n$  é baixo, mas aumenta à medida em que a otimização se torna mais local para aumentar a velocidade de convergência. Para encontrar o mínimo global, o espaço é explorado apenas uma vez na área, até as soluções locais ficarem saturadas [Tekin and Sabuncuoglu \(2004\)](#).

### Método de Superfície Gradiente

O *Método de Superfície Gradiente* (MSG) de otimização de sistemas de eventos discretos e dinâmicos se diferencia dos outros métodos de otimização global, porque ele usa métodos de busca tradicionais para explorar globalmente a superfície de resposta. O método combina as vantagens da Metodologia de Superfície de Resposta (MSR) e técnicas eficientes de estimativa por derivadas como Análise de Perturbação (AP) e Estimativas de Taxa de Possibilidade (ETP) em conjunto com algoritmos de Aproximação Estocástica (AE)

No Método de Superfície Gradiente (MSG), a estimativa do gradiente é obtida pela AP e ETP e o gradiente da superfície da medida de desempenho é ajustada por estas estimativas de gradientes usando métodos de mínimos quadrados, como em MSG. Os pontos zeros ajustados na superfície são avaliados como estimativas do ótimo global [Tekin and Sabuncuoglu \(2004\)](#).

O MSG é um método de busca global porque a cada iteração, ele usa informações de todos os pontos ao invés de apenas um gradiente local. As características mais atrativas do método é que ele obtém estimativas de gradientes em uma única rodada e rapidamente encontra o ótimo por causa de suas orientações globais [Tekin and Sabuncuoglu \(2004\)](#).

### 11.3 Problemas multi-objetivos

Problemas reais podem ter diversos objetivos que deveriam ser otimizados simultaneamente. Diversas dificuldades surgem ao otimizar sistemas multi-objetivos devido à natureza estocástica da resposta. Por causa destas dificuldades, poucos trabalhos tem sido realizados em otimização para simulação multi-objetiva. As abordagens são combinadas e oferecem poucos *insights* genéticos ou conclusivos [Tekin and Sabuncuoglu \(2004\)](#).

## 12 Elementos de probabilidade e estatística

Referência	Capítulos	Páginas	Assunto
<a href="#">Chwif and Medina (2006)</a>	[Ap.7.0]	[273,275]	Distribuições
<a href="#">Banks and Carson (1984)</a>	[04.0.0]	[122,165]	Revisão
<a href="#">Law (2007)</a>	[04.0.0]	[214,239]	Revisão
<a href="#">Kelton et al. (1998)</a>	[00.0.0]	[000,000]	X
<a href="#">Fishman (2001)</a>	[00.0.0]	[000,000]	X

### 12.1 Teoria de probabilidade

apêndice C de [Kelton et al. \(1998\)](#)  
capítulo 03 do livro [Altiok and Melamed \(2010\)](#)

### 12.2 Variáveis aleatórias

apêndice C de [Kelton et al. \(1998\)](#)  
capítulo 03 do livro [Altiok and Melamed \(2010\)](#)

### 12.3 Funções de distribuição

capítulo 03 do livro [Altiok and Melamed \(2010\)](#)  
apêndice D de [Kelton et al. \(1998\)](#)

### 12.4 Distribuições discretas

capítulo 03 do livro [Altiok and Melamed \(2010\)](#)  
apêndice D de [Kelton et al. \(1998\)](#)

### 12.5 Distribuições contínuas

capítulo 03 do livro [Altiok and Melamed \(2010\)](#)  
apêndice D de [Kelton et al. \(1998\)](#)

## 13 Teoria de Filas

Referência	Capítulos	Páginas	Assunto
<a href="#">Chwif and Medina (2006)</a>	[Ap.2.0]	[231,241]	Introdução
<a href="#">Banks and Carson (1984)</a>	[02.0.0]	[018,032]	Sistemas de filas
<a href="#">Kelton et al. (1998)</a>	[00.0.0]	[000,000]	X
<a href="#">Law (2007)</a>	[00.0.0]	[000,000]	X
<a href="#">Fishman (2001)</a>	[00.0.0]	[000,000]	X

Um sistema de filas é matematicamente caracterizado pelo processo de chegadas, de serviço, número de servidores, disciplina do serviço e a capacidade da fila. Assumimos que o intervalo entre chegadas e a duração da atividade de serviço são variáveis aleatórias independentes identicamente distribuídas (*i.i.d.*) e definidas em termos de distribuições de probabilidade [Altiok and Melamed \(2010\)](#), [Taha et al. \(2008\)](#). As disciplinas do serviço podem ser:

1. FIFO: *First in, First out*;
2. LIFO: *Last in, First out*;
3. SIRO: *Service in random order*;
4. PS: *Priority service*;

A teoria de filas é usada para determinar uma aproximação rápida de um sistema de filas a ser estudado por simulação. A fila M/M/1 é a fórmula mais simples e popular da teoria de filas. O primeiro “M” denota um processo *Markoviano*, ou seja os intervalos entre chegadas são distribuições exponenciais de probabilidade independentes identicamente distribuídas (*i.i.d.*). O segundo “M” determina a distribuição do tempo de serviço, que também é exponencial, enquanto que o “1” indica que o sistema possui servidor único [Kelton et al. \(1998\)](#). Os indicadores de desempenho de interesse em um sistema de filas são:

1. Número médio de clientes na fila (*buffer* apenas);
2. Número médio de clientes no sistema (*buffer*);
3. Tempo médio de clientes na fila (*buffer* apenas);
4. Tempo médio de clientes no sistema (*buffer* e servidores);
5. Utilização média do servidor (fração do tempo ocupado);
6. Taxa de saída do sistema;

Modelamos uma operação considerando  $i$  como um cliente pertencente ao sistema. O intervalo entre chegadas entre o cliente  $i - 1$  e o cliente  $i$  é uma variável *iid*, representada por  $A_i$ . A taxa de chegada,  $\lambda = 1/E[A_i]$ , representa o número médio de chegadas por unidade de tempo. O tempo de serviço para cada cliente  $i$  também é uma variável *iid*, representada por  $X_i$ . Definimos a taxa de serviço,  $\mu = 1/E[X_i]$ , como o número médio de processamentos por unidade de tempo.

Uma forma de avaliar estas medidas é computando analiticamente as distribuições de probabilidade de estado-estacionário do número de clientes no sistema. Assim, se o sistema

possui capacidade  $K$  e  $P_n$  determina a probabilidade de ter  $n$  clientes no sistema, então, o número de clientes no sistema,  $L$ , é representado pela fórmula (36):

$$L = \sum_{n=1}^K nP_n \quad (36)$$

Onde  $K$  pode ser finito ou infinito. Em um sistema de servidor único, o número médio de clientes em atendimento é  $1 - P_0$  (probabilidade do sistema não estar vazio), portanto, o número médio de clientes na fila,  $L_q$ , é representado pela fórmula (37):

$$L_q = L - (1 - P_0) \quad (37)$$

Em geral,  $(1 - P_0) = \rho$  é a utilização do servidor em qualquer sistemas de filas de servidor único com capacidade finita ou infinita. Para sistemas com capacidade infinita, a utilização pode ser expressa como a taxa de entrada sobre a taxa de serviço  $\rho = \lambda/\mu$ . Quando  $\rho \geq 1$  dizemos que o sistema não está em equilíbrio. A fila tende a crescer infinitamente.

A taxa de saída pode ser expressa em termos da utilização. O servidor atende à taxa de  $\mu$  clientes por unidade de tempo, gerando uma média de saída representada por (38):

$$\bar{o} = \mu(1 - P_0) \quad (38)$$

No entanto, para filas com capacidade infinita, a equação (38) se reduz a  $\lambda$ , dada a conservação de fluxo no longo prazo em sistemas de filas. A *fórmula de Little*, representada em (39), é uma relação de estado estacionário genérica de grande importância prática.

$$L = \lambda W \quad (39)$$

Onde  $L$  é o número médio de elementos no sistema e  $W$  é o tempo médio gasto no sistema. Assim, a fórmula vale tanto para todo sistema  $L = \lambda W$ , quanto para as filas ou atendimento apenas, ou seja,  $L_q = \lambda W_q$  e  $L_s = \lambda W_s$ . Além disso, a fórmula é válida para qualquer capacidade de fila, pois representa a lei de conservação de fluxos. Em um sistema de filas com servidor único e capacidade limitada, a relação de conservação  $K$  é expressa em (40):

$$\lambda(1 - \pi_K) = \mu(1 - P_0) \quad (40)$$

Onde  $\pi_K$  é a probabilidade de um cliente encontrar a fila cheia em sua chegada, e  $P_0$ , é a probabilidade do sistema se encontrar vazio. A equação determina que, no estado-estacionário, a taxa efetiva de chegada no sistema equivale à taxa efetiva de saída do mesmo.

A modelagem analítica por teoria de filas apresenta algumas deficiências quando comparado aos modelos de simulação [Kelton et al. \(1998\)](#). Enumeramos algumas:

1. Os intervalos entre chegadas e as taxas de serviço podem não ser exatos;
2. As distribuições podem não ser exponencial, demandando fórmulas sofisticadas;
3. Fórmulas de teoria de filas são para o longo prazo;
4. As fórmulas não apresentam a variabilidade dos resultados.

Os parâmetros são sintetizados na Tabela 7, enquanto que as fórmulas especializadas para os sistemas M/M/1 e M/M/S são apresentadas na Tabela 8:

$A_i$	: O intervalo entre chegadas entre clientes $i - 1$ e $i$ . Variável <i>iid</i> .
$\lambda = 1/E[A_i]$	: Taxa de chegada: Média de chegadas por unid. de tempo.
$X_i$	: Tempo de atendimento para o cliente $i$ . Variável <i>iid</i>
$\mu = 1/E[X_i]$	: Taxa de serviço: Média de processamentos por unid. de tempo.
$L = \sum_{n=1}^K nP_n$	: Número médio de clientes no sistema.
$L_q = L - (1 - P_0)$	: Número médio de clientes na fila.
$\rho = 1 - P_0$	: Utilização do servidor único com capacidade finita ou infinita.
$\rho = \lambda/\mu$	: Utilização do servidor único com capacidade infinita.
$\bar{o} = \mu(1 - P_0)$	: Taxa de saída de clientes do sistema.
$L = \lambda W$	: Fórmula de <i>Little</i> : Lei da conservação dos fluxos.

Tabela 7: Síntese dos parâmetros de teoria das filas

Little	M/M/1	M/M/S
$L = \lambda W$	$\rho = \frac{\lambda}{\mu}$	$\rho = \frac{\lambda}{s\mu}$
$L_q = \lambda W_q$	$P_0 = 1 - \rho$	$P_0 = \left[ \sum_{n=0}^{s-1} \frac{1}{n!} \left(\frac{\lambda}{\mu}\right)^n + \frac{1}{s!} \left(\frac{\lambda}{\mu}\right)^s \left(\frac{s\mu}{s\mu - \lambda}\right) \right]^{-1}$
$L_s = \lambda W_s$	$P_j = (1 - \rho)\rho^j, j = 1, 2, \dots$	$P_j = \frac{(s\rho)^j}{j!} P_0, j = 1, 2, \dots, s - 1$
	$P_{[\geq k]} = \rho^k$	$P_{[j \geq s]} = \frac{(s\rho)^s}{s!(1-\rho)} P_0$
	$L_q = \frac{\rho^2}{1-\rho} = \frac{\lambda^2}{\mu(\mu-\lambda)}$	$L_q = \frac{P_{[j \geq s]}\rho}{1-\rho}$
	$L_s = \rho$	$L_s = \frac{\lambda}{\mu}$
	$L = \frac{\rho}{1-\rho} = \frac{\lambda}{\mu-\lambda}$	$L = \frac{\lambda}{\mu} + \frac{P_{[j \geq s]}\rho}{1-\rho}$

Tabela 8: Fórmulas das fórmulas de teoria das filas

## 14 Modelagem visual e *softwares* de simulação

Referência	Capítulos	Páginas	Assunto
<a href="#">Chwif and Medina (2006)</a>	[04.0.0]	[069,088]	Implementação e softwares
<a href="#">Banks and Carson (1984)</a>	[03.2.0]	[071,100]	Linguagens de simulação
<a href="#">Law (2007)</a>	[3.0.0]	[187,213]	Classificação e propriedades
<a href="#">Pidd (2004)</a>	[08.0.0]	[129,146]	Conceitos e exemplos
<a href="#">Kelton et al. (1998)</a>	[00.0.0]	[000,000]	X
<a href="#">Fishman (2001)</a>	[00.0.0]	[000,000]	X

### 14.1 Modelagem visual em projetos de simulação

O analista precisa compreender dois mundos. O primeiro relacionado ao conhecimento técnico e científico demanda um conhecimento em formulações matemáticas. O segundo é relacionado aos projetos de operações reais, gerenciado por uma equipe em que o cliente precisa avaliar as funcionalidades desenvolvidas e usá-las para tomar decisões no dia-a-dia. Por isso a interface gráfica integrada aos modelos matemáticos facilita a comunicação entre estes dois mundos. A modelagem visual apresenta três benefícios:

1. Elaborar um sistema gráfico facilita o entendimento do programa e sua comparação à operação real modelada.
2. Ao desenvolver o modelo de simulação, erros básicos podem ser facilmente identificáveis, como, por exemplo, o crescimento infinito de uma fila, no entanto, corre-se o risco de analistas viciarem seus desenvolvimentos baseando-se apenas em comportamentos visuais. Com isso, perde-se a capacidade técnica e analítica de elaboração e avaliação do modelo.
3. O programador também é beneficiado, pois sistemas visuais viabilizam o *debug* de erros lógicos de programação.

Em projetos de ciências gerenciais, a elaboração de modelos computacionais requer do analista habilidades de representação de operações, conhecimento tecnológico de modelagem, programação, estatística, além do envolvimento deste com a equipe de projeto e clientes. Sob este aspecto, a modelagem visual, que tem sido facilitada pelos poderosos computadores pessoais, viabilizando uma aproximação maior do analista aos seus clientes. Estes permitem a prototipagem rápida, adotando o princípio da parcimônia, ao apresentar, de forma simples, resultados robustos por poderosas ferramentas de análise. Assim, é preciso usar estas funcionalidades para tirar proveitos dos sistemas interativos visuais.

### 14.2 *Softwares* de simulação e linguagens de programação

Alguns princípios gerais norteiam os *softwares* de simulação. Analistas elaboram modelos computacionais em linguagens de programação de sua preferência, como: **C**, **C++**, **Java**, **Python**, etc, e em filosofias de programação específicas, como: orientação a objeto, programação funcional ou estruturada. Assim, não existe "a melhor compra" desses sistemas, porque *softwares* de simulação são desenvolvidos em diferentes linguagens e filosofias de programação. *Journals* e conferências abordam, por meio de *reviews*, as características, vantagens e desvantagens relacionadas aos *softwares* de simulação mais recentes. *Softwares* de simulação podem seguir três caminhos distintos. O programa pode:

1. Ser inteiramente desenvolvido por uma linguagem de baixo nível de propósito geral, como C, C++, Java, Python, FORTAN, etc.
2. Ser desenvolvido por linguagem de alto nível especialistas de simulação, como: GPSS, Simula, SIMAN, Simscript.
3. Ser elaborado por sistemas de interativos visuais alto nível, onde o usuário carrega caixinhas e as vincula por meio de setas. Exemplos de sistemas interativos são o: Arena<sup>TM</sup>, Flexsim<sup>TM</sup>, Promodel<sup>TM</sup>, MicroSaint<sup>TM</sup>, Simul8<sup>TM</sup> e Witness<sup>TM</sup>.

Um *software* de simulação é formado por uma *engine* de simulação, que adota uma abordagem de cálculo do executor principal, como por processos, por três fases, por varredura de atividades ou por eventos; por componentes específicos relacionados à sua aplicação e por uma biblioteca de funções. Todos estes componentes são integrados de forma harmoniosa e apresentados por meio de uma interface gráfica intuitiva.

O analista ou organização, ao adotar uma linguagem de programação de funcionalidade genérica, precisa ter em mente os prós e contras de sua utilização. Muitas vezes, as organizações não têm dinheiro para investir em *softwares* de simulação e preferem usar o corpo técnico, com experiência em programação, para elaborar um programa de simulação. No entanto, esta estratégia além de demorada, pode ser custosa. O desenvolvimento de sistemas em linguagens genéricas muitas vezes requer o desenvolvimento de bibliotecas de funções, demandando conhecimento profundo em programação. Organizações possuem analistas com habilidades em linguagens de programação diferentes. Analistas desenvolvedores podem abandonar o projeto no meio de sua execução em função de uma nova proposta de trabalho ou mudança de estilo de vida, por exemplo. Além disso, uma extensa documentação deve ser feita, para que outros desenvolvedores, ou clientes, possa vir a utilizá-la. Às vezes, somente documentação não é suficiente. É necessário que o usuário ou cliente tenha por um treinamento intenso no modelo para que este não seja descartado no futuro.

O uso de linguagens de programação de simulação possuem funcionalidades de linguagens genéricas condensadas em poucas linhas. Todas as linguagens de simulação possuem características específicas semelhantes:

1. O sistema executivo de simulação, que pode ser implementado em processos, eventos ou atividades, fica escondido, ou seja, não demanda o conhecimento de sua funcionalidade pelo analista.
2. Os sistemas possuem linguagens com sintaxe bem definida, portanto, livre de erros de programação.
3. Os sistemas de manipulação de dados e rastreamento de variáveis específicos para a linguagem.
4. Facilidades na elaboração de experimentos.

Em um nível superior estão os *softwares* de simulação. Sua seleção requer o conhecimento de alguns princípios: *Softwares* possuem funcionalidade aplicada a eventos específicos. Alguns são aplicados à manufatura, outros à pessoas, etc. Os pacotes vêm com interface gráfica e objetos específicos às aplicações em que são destinados. Outra funcionalidade importante é a facilidade para o usuário final. Se este for um cliente, os *softwares* de simulação permitem o desenvolvimento de animações, o que facilita a dinâmica do sistema. Se o usuário final for um analista, que busca saber os resultados e estatísticas coletadas dos indicadores de desempenho, o *software* permite o desenvolvimento da lógica, apenas. Um terceiro ponto está relacionado à política de suporte. *Softwares* livres podem se tornar pesadelos quando se busca uma solução para um problema desconhecido ou *bug*. Assim, *softwares* comerciais possuem um suporte ao usuário.





**Figura 15:** Analogia da pintura de um mosaico com o desenvolvimento de um modelo de simulação

Sugere-se para aplicações de menor porte, com funcionalidades conhecidas, a prototipagem e desenvolvimento rápido de sistemas por meio de *softwares* de simulação. No caso de sistemas complexos, que resultam da interação de diversos pequenos blocos de funções específicas, indica-se o desenvolvimento de sistemas por meio de linguagens de simulação, ou de programação de uso genérico. Uma analogia é feita na pintura de um muro enorme, como na Figura 15, feito por pequenos mosaicos. Sua pintura é feita por um rolo de tinta grande ou pequenos pincéis específicos para cada componente?

## Referências

- Altiook, T. and Melamed, B. (2010). *Simulation modeling and analysis with Arena*. Academic press.
- Automation, R. (2007). *Arena-User's Guide*.
- Banks, J. and Carson, J. (1984). *Discrete-Event System Simulation*. Industrial and Systems Engineering. Prentice hall Englewood Cliffs, NJ, USA.
- Banks, J., Carson, J., Nelson, B., and Nicol, M. (2005). *Discrete-Event System Simulation*. Prentice hall Englewood Cliffs, NJ, USA, 4 edition.
- Chwif, L. and Medina, A. C. (2006). *Modelagem e simulação de eventos discretos*. Afonso C. Medina.
- Fishman, G. S. (2001). *Discrete-event simulation: modeling, programming, and analysis*. Springer Science & Business Media.
- Fu, M. C. (1994). Optimization via simulation: A review. *Annals of Operations Research*, 53(1):199–247.
- Kelton, W. D., Sadowski, R. P., and Sadowski, D. A. (1998). *Simulation with ARENA*, volume 47. WCB/McGraw-Hill New York.
- Law, A. M. (2007). *Simulation modeling and analysis*, volume 4. McGraw-Hill New York.
- Markowitz, H. M. (1963). SIMSCRIPT: A Simulation Programming Language. Technical report, Rand Corporation.
- Pidd, M. (2004). *Computer Simulation in Management Science*. John Wiley & Sons, Inc., 5 edition.
- Taha, H. A., Marques, A. S., and Scarpel, R. A. (2008). *Pesquisa operacional*. Pearson Education do Brasil.
- Tekin, E. and Sabuncuoglu, I. (2004). Simulation optimization: A comprehensive review on theory and applications. *IIE Transactions*, 36(11):1067–1081.



