

Modelos de Simulação de Eventos Discretos

Estudo do comportamento de sistema através de construção de simuladores.

1 Modelos de Simulação

Simulador é um algoritmo ou procedimento que representa o comportamento de um sistema em uma escala de tempo.

Através de simulação, podem ser feitos o projeto, dimensionamento, avaliação de desempenho ou reengenharia de sistemas tais como:

- Sistemas de manufatura e de processos comerciais.
- Sistemas de computação, redes de comunicação de dados ou telefônicos.
- Sistemas de transporte: obras viárias, metrô, embarque e desembarque de mercadorias em portos.
- Sistemas on-line de auto-atendimento: home-banking, e-commerce, e-business.
- Empresas de prestação de serviços: bancos, lanchonetes, postos de gasolina.
- Sistemas de defesa e equipamentos militares.

1.1 Modelos de Simulação Estáticos e Dinâmicos

Os sistemas a serem tratados neste curso são sistemas de simulação discretos, dinâmicos e estocásticos, denominados Sistemas de Simulação de Eventos Discretos. Antes de iniciarmos o estudo destes sistemas vamos identificar as categorias de modelos de simulação.

Modelo de Simulação Estático

Representa o sistema em um instante determinado, isto é, o modelo não varia ao longo do tempo. Exemplo: Cálculo de Integral de função pelo método de Monte Carlo.

Modelo de Simulação Dinâmico

Representa a evolução do sistema ao longo do tempo. Exemplo: simulação de uma linha de montagem de veículos.

1.2 Modelos de Simulação Determinísticos ou Estocásticos

Modelo de Simulação Determinístico

Se o sistema não depende de nenhuma variável probabilística (aleatória).

Modelo de Simulação Estocástico

Se o sistema depende de variáveis probabilísticas (aleatórias). Sistemas de computação, de redes de comunicação e de serviços a clientes, entre outros, estão nesta categoria. Em geral utilizam filas de chegada de tarefas em que as chegadas ocorrem de acordo com alguma distribuição de probabilidade.

1.3 Modelos de Simulação Contínuos ou Discretos

Modelo de Simulação Discreto

Se o sistema depende de variáveis que assumem valores discretos, isto é, em um domínio de valores finitos ou enumeráveis tais como o conjunto de números inteiros.

Modelo de Simulação Contínuo

Depende de variáveis que assumem valores contínuos, isto é, em um domínio de valores contínuos tais como o conjunto de números reais.

1.4 Modelos de Simulação de Tempo Real ou Simulado

Os simuladores podem operar em duas modalidades de tempo:

Tempo real

A escala de tempo é a real, isto é os eventos ocorrem e são tratados na mesma escala de tempo correspondente ao sistema real. Simuladores de jogos ou para treinamento se enquadram nesta categoria. Nestes sistemas um operador humano interage com o simulador em tempo real.

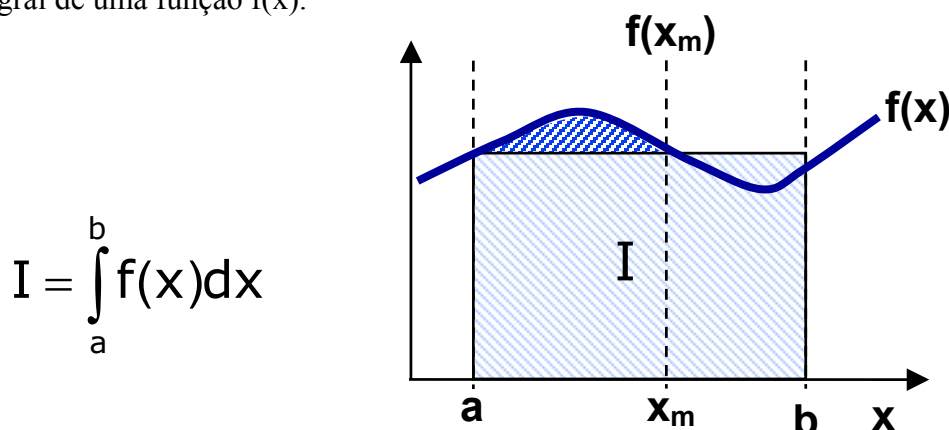
Tempo simulado

Não acompanha a escala de evolução do tempo real. Um ano do tempo de simulação pode decorrer em poucos segundos de processamento. São utilizados para análises de desempenho em que o interesse é pelas medidas de desempenho.

Exemplo 1: Modelo de Simulação pelo Método de Monte Carlo

O método de simulação de **Monte Carlo** utiliza a geração de números aleatórios para a simulação de processos ou efetuar cálculos cuja fórmula exata não é disponível. Foi originado na Segunda Guerra Mundial e aplicado ao desenvolvimento da bomba atômica.

Um exemplo de utilização é para o cálculo aproximado de integral de função cuja resolução analítica não pode ser determinada. Este é um método simulado estático, estocástico e com valores contínuos. Mostraremos como funciona em no caso da integral de uma função $f(x)$.



Existe um ponto x_m tal que a área I , definida pela integral, é igual área do retângulo de lado $(b-a)$ e altura $f(x_m)$.

O cálculo da integral de forma aproximada pelo método de Monte Carlo será feito sorteando-se aleatoriamente x_1, x_2, \dots, x_n , onde cada x_i tem distribuição uniforme entre $[a, b]$, e calculando-se a média da área dos retângulos de lado $(b-a)$ e alturas $f(x_1), f(x_2), \dots, f(x_n)$. O valor aproximado da integral é calculado pela fórmula $I \cong \frac{\sum_{i=1}^n (b-a)f(x_i)}{n}$

Aplicando-se o método de Monte Carlo à função $f(x) = \sin(x)$ para calcular

$I = \int_0^\pi \sin(x) dx$, cujo valor exato é 2, obtém-se os seguintes valores obtidos em 6 execuções da simulação, com diferentes valores de n :

n	Integral de Sen(x) por Monte Carlo	(2 – Monte Carlo)
10	1,2566	0,7434
50	1,8221	0,1770
100	1,9478	0,0522
500	2,0986	-0,0986
1000	2,0703	-0,0703
10000	1,9955	0,0045

1.5 Vantagens e Desvantagens da Simulação

Vantagens

- Sistemas do mundo real com elementos estocásticos podem não ser descritos de forma precisa através de modelos matemáticos que possam ser calculados analiticamente.
- Permite estimar o desempenho de sistemas existentes sob condições de operação projetadas, por exemplo, para verificar o seu comportamento quando aumenta a demanda de serviço.
- Permite manter maior controle sob as condições dos experimentos o que muitas vezes não é possível com o sistema real.
- Permite estudar o sistema durante um longo período de tempo simulado.

Desvantagens

- Cada execução da simulação estocástica produz apenas estimativas dos parâmetros analisados.
- O modelo de simulação em geral é caro e consome muito tempo para desenvolver.
- Os resultados da simulação, quando apresentados em grandes volumes de dados, e com efeitos de animações e gráficos, podem levar a uma confiança nos resultados acima da justificada. Se o modelo não for uma representação válida do modelo em estudo, este não terá utilidade, mesmo que os resultados causem boa impressão.

1.6 Causas de Insucesso no Desenvolvimento da Simulação

- Falha na obtenção de um conjunto bem definido de objetivos no início do estudo da simulação.
- Nível inadequado de detalhes:
 - Pouco detalhamento ou
 - Muito detalhamento.
- Falha de comunicação com a gerência do sistema a ser simulado durante o estudo da simulação.
- Interpretações equivocadas por parte da equipe da simulação da operação do sistema a ser simulado.
- Falha de compreensão da simulação por parte da gerência.
- Tratar a simulação de forma amadora, como um exercício de curso.
- Falha em formar uma equipe com conhecimentos de metodologias e técnicas de simulação.
- Falha na obtenção de dados representativos do comportamento do sistema.
- Software de simulação inadequado.
- Software de simulação muito complexo e com documentação inadequada.
- Crença de que software de simulação sofisticado e com recursos amigáveis, prescindem de conhecimentos técnicos da teoria de simulação.
- Uso inadequado de animação.

- Falha na consideração dos fatores aleatórios no comportamento do sistema sendo simulado.
- Uso de distribuições incorretas, isto é, que não correspondem ao comportamento real, como dados de entrada da simulação.
- Análise dos dados de uma execução da simulação utilizando fórmulas que supõem independência (usualmente os dados de saída não são IID).
- Executar uma única vez a simulação e considerar os dados obtidos como a resposta verdadeira.
- Utilizar medidas de desempenho inadequadas.

2 Metodologia de Análise de Sistemas por Simulação

A metodologia a ser utilizada se baseia na definida no capítulo 1. Nos próximos tópicos são detalhados alguns aspectos considerados importantes no processo de análise e simulação.

2.1 Formular o problema e planejar a análise

Nas reuniões de planejamento devem ser discutidos os seguintes itens que resultarão em um documento de especificação da análise e simulação a ser realizada:

- Objetivo do sistema a ser modelado;
- Objetivos da análise e simulação;
- Questões específicas a serem respondidas pelo estudo;
- Métricas de desempenho a serem utilizadas para avaliar a eficácia de uma configuração de sistema;
- Carga a ser submetida ao sistema;
- Escopo do modelo;
- Configuração do sistema a ser modelado;
- Indicação de ferramentas a serem utilizadas na simulação;
- Cronograma para realização do estudo e recursos necessários.

2.2 Coletar dados

Coletar informações sobre a arquitetura do sistema e sobre os procedimentos operacionais:

- Consultar documentações disponíveis;
- Conversar com os especialistas no assunto (SME-Subject-Matter Experts);
- Conversar com pessoas que operam o sistema;
- Identificar os parâmetros do sistema e os fatores a serem analisados;
- Planejar e realizar a coleta dos dados dos parâmetros do sistema;
- Os dados podem ser resultantes de monitoração do sistema real ou projeções sobre um sistema a ser desenvolvido.

2.3 Análise estatística dos parâmetros

Realizar análises estatísticas dos parâmetros do sistema e de carga. Esta análise deve resultar na escolha de distribuições de probabilidades que representem os parâmetros ou então, caso não possam ser identificadas estas distribuições, obter amostras de dados a serem utilizados nas simulações.

Para a análise estatísticas dos dados podem ser utilizados ferramentas e pacotes disponíveis no mercado ou que são fornecidos com as ferramentas de simulação.

2.4 Definir o modelo

Utilizando as informações obtidas nos passos anteriores, definir um documento de modelo conceitual que especifica a concepção que se tem do sistema e tudo o que foi assumido. Os detalhes do modelo dependem de:

- Objetivos do projeto
- Medidas de desempenho
- Disponibilidade dos dados
- Questões de credibilidade das informações obtidas
- Limitações de recursos de computação
- Opiniões dos especialistas SME
- Limitações em tempo e dinheiro
- Não precisa haver correspondência 1 a 1 entre os elementos do modelo e o sistema real.

2.5 Validar o modelo conceitual

Realizar um walk-through do modelo conceitual utilizando os documentos disponíveis para garantir que o que foi assumido é correto e completo.

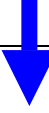



Apresentar o modelo aos gerentes, analistas e SMEs para que todos dêem a aceitação do modelo proposto.

Se existir algo que invalide o modelo voltar à etapa 2.2 para realizar a correção necessária.

2.6 Escolha da Ferramenta de Simulação

A escolha da ferramenta considera fatores tais como;

- Características da simulação a ser realizada;
- Características das ferramentas disponíveis;
- Ambiente disponível para simulação;
- Capacitação da equipe de projetistas que realizarão a implementação do simulador;
- Orçamento disponível para compra da ferramenta caso não exista disponível;
- Tempo disponível para realizar a simulação.

Ferramentas de simulação	Custo da ferramenta	Custo do desenvolvimento da simulação
Linguagens de programação: C, C++, FORTRAN, Java.	Menor 	Maior 
Pacotes de simulação: Arena, Promode, Comnet, Optnet, etc.	Maior 	Menor 

2.7 Construir o Modelo

Uma vez escolhida a ferramenta, o modelo deve ser detalhado, modelado e implementado. Neste caso podem ser utilizadas as metodologias de engenharia de software orientada a objeto ou tradicional, conforme a ferramenta escolhida.

2.8 Validar o Modelo Simulado

Realizar execuções piloto para validar o programa de simulação. Utilizar, caso exista, dados de outro sistema do qual se possua medida de desempenho e com o qual o sistema possa ser comparado.

A equipe que desenvolveu a simulação bem como os especialistas SME devem revisar o modelo e verificar se os resultados estão corretos e dentro do esperado.

Utilizar análises de sensibilidade para determinar quais fatores tem impacto mais significativo sobre as medidas de desempenho e devem ser modelados com maior cuidado.

Utilizar técnicas de probabilidade e estatística para analisar se os dados obtidos na simulação são aderentes aos dados obtidos de sistemas reais utilizados como referência. Para isto podem ser utilizados ferramentas e pacotes disponíveis no mercado ou que são fornecidos com as ferramentas de simulação.

Se o modelo não for válido voltar à etapa 2.2.

2.9 Planejar os experimentos

Para cada configuração de interesse, especificar:

- Duração ou número de passos de cada execução.
- Duração do transitório, isto é, período inicial até o sistema entrar em regime estável. Este período é excluído das medidas.
- Número de execuções independentes (com diferentes números aleatórios) para permitir a determinação de intervalos de confiança.

2.10 Executar as simulações de produção

Executar as simulações correspondentes aos experimentos planejados na etapa 2.9.

2.11 Analisar os dados de saída

Determinar o desempenho das diversas configurações do sistema. Comparar configurações alternativas do sistema utilizando as técnicas já utilizadas na etapa 2.8 **Documentar os resultados**

Documentar:

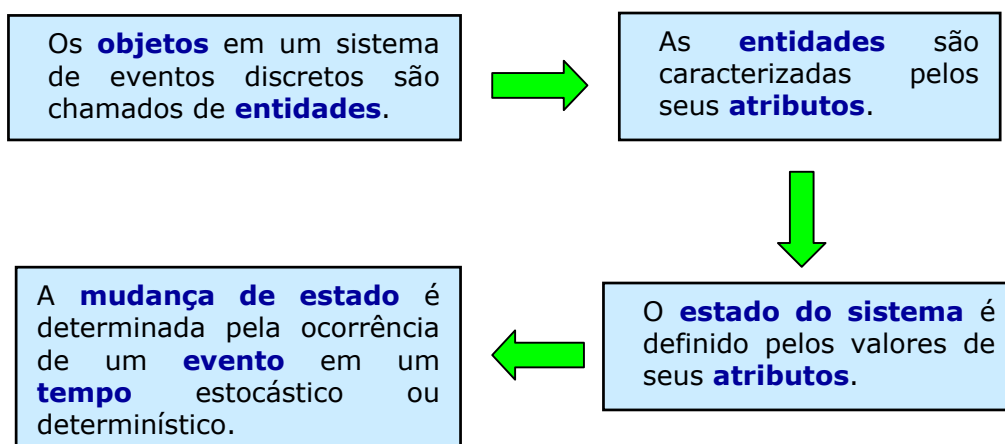
- O que foi assumido na etapa 2.1;
- O programa de simulação;
- A análise dos dados de entrada e dos resultados.
- Apresentar os resultados do estudo aos gerentes e às demais pessoas envolvidas, através de:
 - Animações do modelo;
 - Gráficos e tabelas das análises;
- Discussão da construção do modelo e do processo de validação para promover a sua credibilidade.
- Utilizar os resultados nos processos de tomada de decisão se estes forem válidos e tiverem credibilidade.

3 Simulação de Sistemas de Eventos Discretos

Os sistemas a serem tratados neste curso através de simulação são sistemas

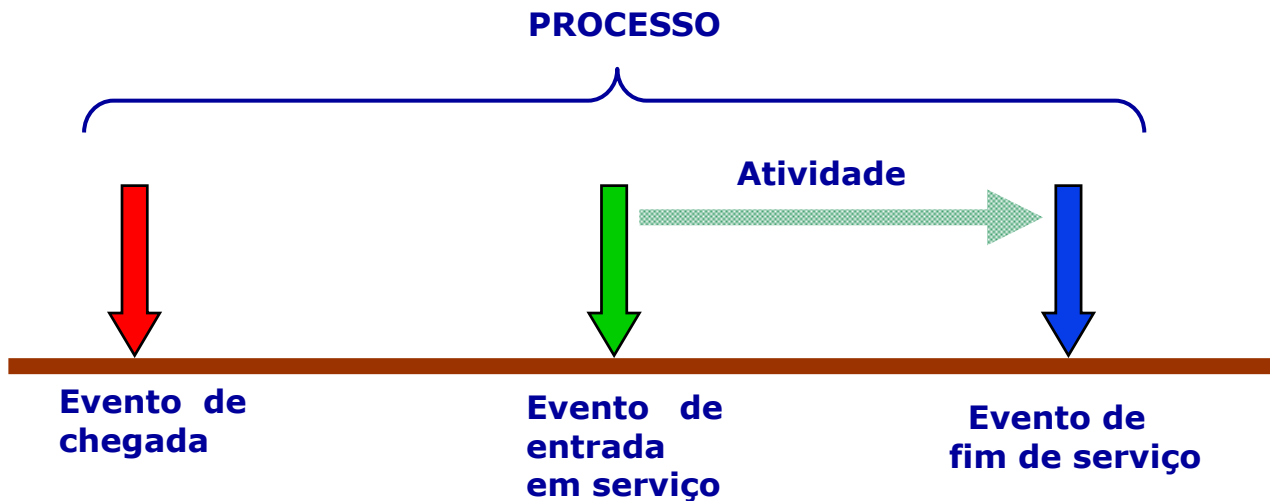
- Discretos,
- Dinâmicos e
- Estocásticos,

denominados **Sistemas de Eventos Discretos**.



O objetivo de um modelo de simulação de eventos discretos é reproduzir as atividades das **entidades** que compõe o sistema e, a partir daí, conhecer o comportamento e desempenho do sistema. Para isto precisamos definir o **estado do sistema** e as **atividades** que conduzem o sistema de um estado a outro.

Na simulação discreta, a mudança de estado é determinada pela ocorrência de um **evento** em um **tempo** determinístico ou estocástico.



Exemplo 2: Entrada de Estacionamento de Shopping Center

Em um “Shopping Center” chegam ao estacionamento 5 veículos por minuto, em horário de almoço. Existe um único bloqueio de entrada e os veículos que chegam entram em uma única fila. Cada motorista, ao chegar sua vez, gasta 10 segundos entre avançar o veículo até o controle do bloqueio, pressionar o botão, retirar o ticket de entrada, esperar a cancela ser aberta e seguir em frente com o veículo.

Perguntas:

1. Qual é o tamanho médio da fila de veículos?
2. Qual é o tempo médio que um motorista gasta desde que chega à fila até entrar no estacionamento?
3. Quantos bloqueios devem ser colocados para melhorar o atendimento?

Variáveis de simulação:

t_i instante de chegada de um veículo à fila

A_i intervalo de tempo entre chegadas

$$A_i = t_i - t_{i-1}$$

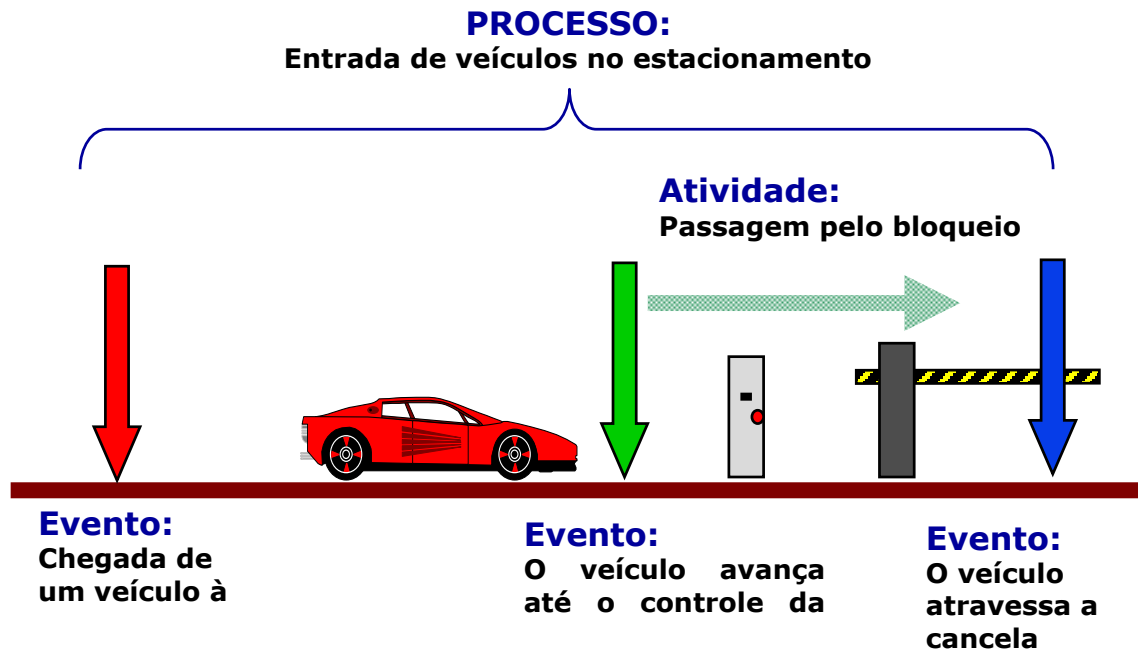
S_i tempo de serviço do usuário i , isto é, tempo para obter o ticket e passar pela cancela

D_i espera do usuário i na fila de veículos

r_i tempo de resposta do usuário i , entre a chegada do veículo à fila e a passagem pela cancela.

$$r_i = D_i + S_i$$

f_i instante de passagem do veículo i pela cancela. $f_i = t_i + r_i$



Este exemplo pode ser generalizado para outros sistemas com uma fila onde os usuários ficam a espera e um servidor que realiza atendimento de um usuário de cada vez. Quando o servidor terminar o atendimento de um usuário, verifica se a fila está livre e inicia o atendimento do próximo usuário. Uma agência de banco com um único caixa também pode ser descrita da mesma maneira.

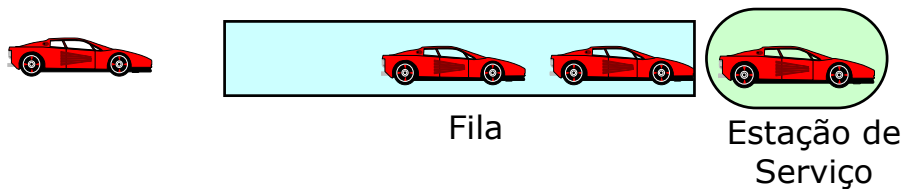
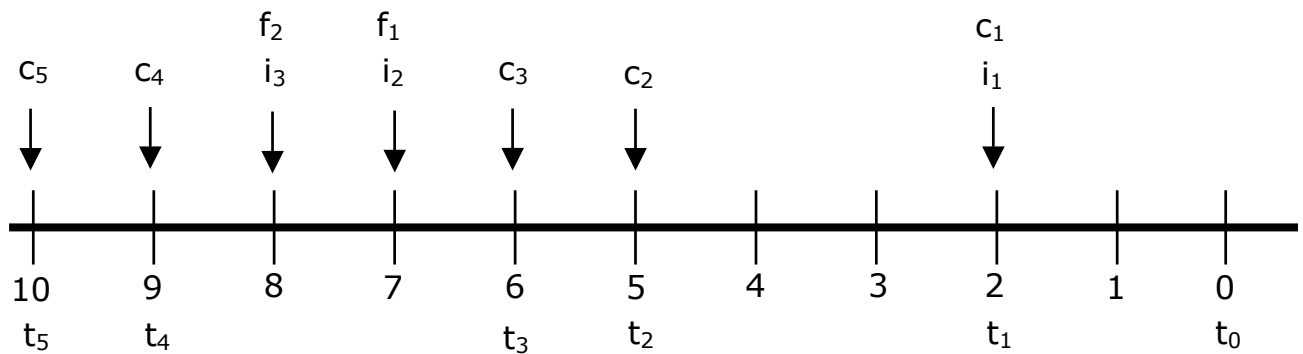


Gráfico dos eventos no tempo



Tempo de simulação	Evento	Prox.Evento chegada	Prox.Evento Inic. Atend.	Prox.Evento Fim Atend.	Número na fila	Estado do servidor
0		c_1 em $t_1=2$			0	livre
1					0	livre
2	c_1	c_2 em $t_2=5$	i_1 em $t=2$		1	livre
	i_1			f_1 em $t=7$	0	ocupado
3					0	ocupado
4					0	ocupado
5	c_2	c_3 em $t_3=6$			1	ocupado
6	c_3	c_4 em $t_4=9$			2	ocupado
7	f_1		i_2 em $t=7$		2	livre
	i_2			f_2 em $t=8$	1	ocupado
8	f_2		i_3 em $t=8$		1	livre
	i_3			f_3 em $t=11$	0	ocupado
9	c_4	c_5 em $t_5=10$			1	ocupado
10	c_5	c_6 em $t_6=12$			2	ocupado
11	f_3		i_4 em $t=11$		2	livre
	i_4			f_4 em $t=13$	1	ocupado

4 Simuladores de Eventos Discretos

Os principais modelos de Simulação de Eventos Discretos são os seguintes:

- Modelos orientados a evento:** definido pelas mudanças de estado que podem ocorrer em cada evento.
- Modelos orientados a processo:** definido pelo processo através do qual as entidades do sistema são conduzidas. Neste caso, o processo consiste da sequência de eventos.

4.2 Simulador Orientado a Eventos

Na simulação orientada a eventos existe um procedimento associado com cada tipo de evento no sistema. O simulador ciclicamente escala eventos, atualiza o relógio para o próximo evento a ocorrer e executa o procedimento associado ao evento.

Os elementos de um simulador orientado a eventos são:

Eventos

Os eventos podem ser agendados para um determinado instante no tempo existindo uma lista dos próximos eventos a ocorrerem, ordenados por tempo de ocorrência. A ocorrência de um evento afeta o estado da simulação. Também são atualizados os contadores de estatísticas que permitem a geração dos relatórios da simulação.

Estado do sistema

Coleção de variáveis de estado necessárias para descrever o sistema em um determinado momento no tempo. O estado do sistema pode ser visto como resultante do estado de seus componentes.

Mecanismo de temporização

O avanço de tempo, nos simuladores de **tempo simulado**, pode ser definido por

- Incrementos fixos de tempo: o relógio é incrementado em intervalos fixos de tempo e após o incremento é verificado se existe um evento agendado para este momento.
- Próximo evento: o tempo é incrementado pela ocorrência de um evento. É mais eficiente que o anterior em termos de atualização de tempos.

Lógica de Simulação

A lógica de simulação consiste em verificar o próximo evento agendado e atualizar o relógio de simulação e as demais variáveis de estado do sistema. O término da execução pode ser controlado através de um limite no tempo de simulação ou de um limite do número de eventos ocorridos. Ao final da simulação serão emitidos relatórios com os resultados da simulação.

Métodos estatísticos

Conjunto de funções e procedimentos entre os quais se incluem variáveis que armazenam informações estatísticas para determinar o desempenho do sistema, geradores de números aleatórios para diferentes distribuições de probabilidade e analisadores de dados de entrada e saída.

Lógica de Simulação

A simulação realizada manualmente no exemplo 2 é o que se chama simulação orientada a evento.

Na simulação orientada a evento, em cada instante a simulação é definida por x que representa o estado da simulação, t que é o relógio de simulação e $L = \{(e_1, t_1), (e_2, t_2), \dots\}$ que é uma lista de eventos e_i escalados para ocorrerem nos instante t_i . A lista L é mantida ordenada por ordem crescente de tempo.

A Simulação Orientada a Eventos consiste das seguintes etapas:

Passo 0: No início da simulação, define-se o estado inicial $x = x_0$, o relógio de simulação $T=0$ e a lista L é inicializada com todos os eventos viáveis no estado x_0 .

Passo 1: Retirar uma entrada de (e_i, t_i) de L com menor tempo t_i .

Passo 2: Atualizar o tempo de simulação para o valor $T = t_i$.

Passo 3: Atualizar o estado da simulação de acordo com uma função de transição de estado f . Sendo x o estado atual, o novo estado $x' = f(x, e_i)$.

Passo 4: Retirar da lista L todos os eventos que não são viáveis no estado x' .

Passo 5: Acrescentar à lista L em ordem crescente de tempo, todos os eventos viáveis no estado x' que ainda não estejam na lista. Ao acrescentar um evento (e_k, t_k) à lista. O instante, t_k de ocorrência do evento e_k deve ser gerado a partir de um gerador de números aleatórios de acordo com a distribuição de ocorrência do evento e_k .

Passo 6: Voltar ao passo 1 da simulação.

O programa de simulação também inclui funções para contagens estatísticas e de emissão do relatório ao final da simulação.

4.3 Simulador Orientado a Processos

A operação deste simulador consiste da interação de um número de processos executando em paralelo. O gerenciamento de eventos é implícito aos processos. O sistema de simulação prove mecanismos para manipular processos: colocar um processo em espera, escalar um processo e terminar um processo entre outras operações. Por esta razão, as linguagens utilizadas para este tipo de simulação costumam trabalhar com o conceito de processo ou thread.

Os principais componentes de um simulador orientado a processo são:

Entidades

São objetos que requisitam serviços. Exemplos: usuários em uma fila de banco, carros na entrada de um estacionamento, partes em um sistema de manufatura, mensagens em um sistema de computação.

Atributos

Informações que caracterizam uma entidade em particular. Exemplo: modelo de item a ser produzido.

Processos

Ações realizadas sobre a entidade ao longo da simulação.

Recursos

Objetos que provêm serviços às entidades. Exemplo: máquinas em um sistema de manufatura, caixa de um banco, UCP de um computador, linha de comunicação de dados.

Filas

Local onde as entidades esperam por um recurso. A política de atendimento mais comum é a FCFS (First Come First Served).

Tipo de sistema	Entidades	Atributos	Recursos	Filas
Manufatura	Partes	Código de peças, datas de entrega	Máquinas, trabalhadores	Filas, estoques
Comunicações	Mensagens	Destino, comprimento da mensagem	Nós, enlaces	Buffers
Aeroporto	Aviões	Número do voo Capacidade	Pistas, terminais	Filas de passageiros, de aviões.
Supermercados	Compradores	Tamanho da compra	Caixas	Filas

5 Ferramentas de Simulação

5.1 Linguagens e bibliotecas de funções de simulação

Os projetistas em geral já conhecem uma linguagem de uso geral. A implementação do modelo mais eficiente. O custo do software é menor mas o de projeto em geral é maior. Permite maior flexibilidade. Exemplos:

- C
- C++
- FORTRAN
- Java
- Biblioteca Simlib para C
- Biblioteca JavaSim para Java

A biblioteca de simulação Simlib foi desenvolvida em C por W. David Kelton e Averill M. Law, apresentada no livro: Law, A. M., Kelton, W. D., "Simulation Modeling and Analysis", 3rd ed., McGraw-Hill Companies Inc, 2000, ISBN 0-07-059292-6, 760p. Foi adaptada da SUPERSIMLIB, escrita por Gregory Glockner. Possui funções para manipular listas de eventos.

5.2 Pacotes de Simulação

Possuem embutidas características necessárias à simulação que diminuem o tempo de desenvolvimento do modelo de simulação. Possuem recursos adicionais para visualização e animação e para tratamento de dados. Em geral exigem aprendizado da ferramenta. Exemplos:

Pacotes de Uso Geral

- Arena
- Extend
- AweSim
- Symix
- GPSS/H
- Mixeo Saint
- MODSIM III, CACI e Marti
- SES/workbench
- SIMUL8
- SLX

Pacotes de Uso Geral Orientados a Objeto

- SIMPLE++
- MODSIM III

Pacotes de uso em Manufatura

- AutoMod
- Extend+ Manufacturing
- ProModel
- Quest
- Witness

Pacotes de uso em Redes de Comunicação

- NS2 (ferramentas escritas em C++ e com fonte aberto)
- COMNET III (descontinuado)
- Opnet IT Guru
- OPNET Modeler

Pacotes de uso em Reengenharia de processos e serviços

- Arena Business Edition
- Extend + BPR
- ProcessModel
- ServiceModel (ProModel)
- SIMPROCESS

Pacotes de uso em Saúde

- MedModel (ProModel)

Pacotes de uso em Call Center

- Arena Call Center Edition

Pacotes de uso em Animação

- Proof Animation

6 Arena

Foi lançado pela empresa Systems Modelling em 1993 e atualmente pertence à Rockwell. É sucessor do SIMAN, desenvolvido em 1982. Possui uma interface gráfica GUI que permite a modelagem do sistema através de módulos. Possui interface para Microsoft VBA permitindo integração com programas que suportam Active X. A versão Arena 3 Academic, disponível para uso livre de pagamento, e que será utilizada nos exemplos, possui limitações no número de entidades que podem ser criadas.

6.1 Ferramentas do Arena

Arena: A ferramenta de modelagem e simulação.

Input Analyser: Realiza a análise estatística dos dados de entrada do sistema permitindo determinar a distribuição que mais se ajusta aos dados para entrada no simulador.

Output Analyser: Realiza a análise estatística dos resultados da simulação.

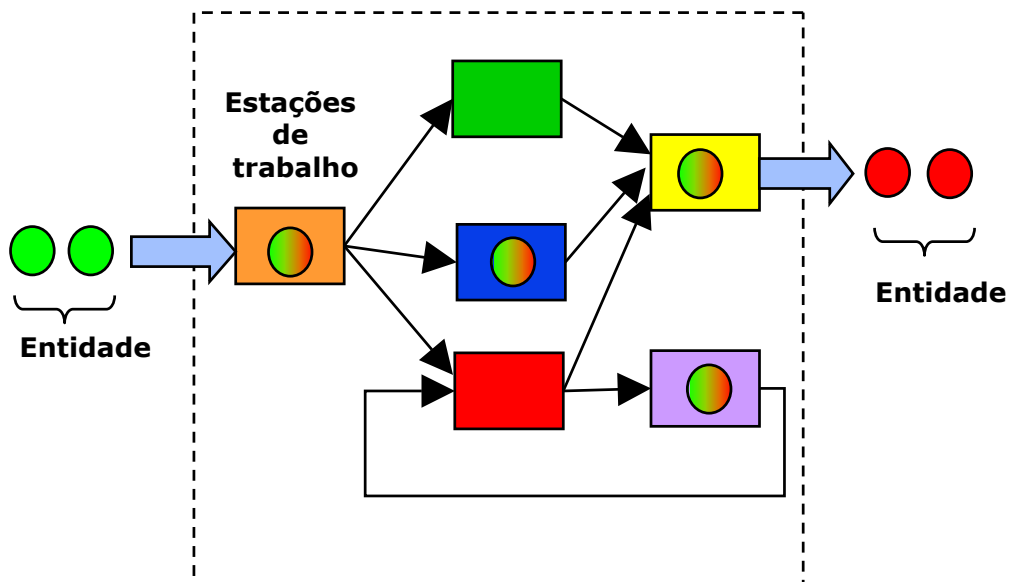
Arena Viewer: Visualizador da simulação.

6.2 Elementos da Modelagem em Arena

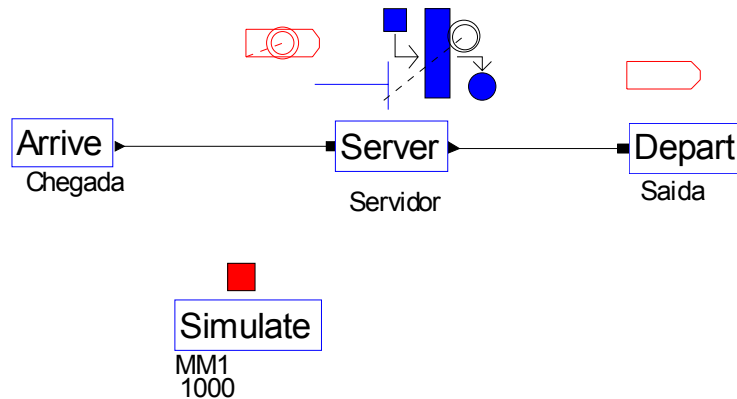
Entidades: são as pessoas, transações ou tarefas que se movem ao longo do sistema.

Estações de trabalho: onde será realizado algum serviço.

Fluxo: caminhos que a entidade irá percorrer ao longo de estações.



O exemplo de um sistema com uma fila e um servidor pode ser representado no ARENA como:

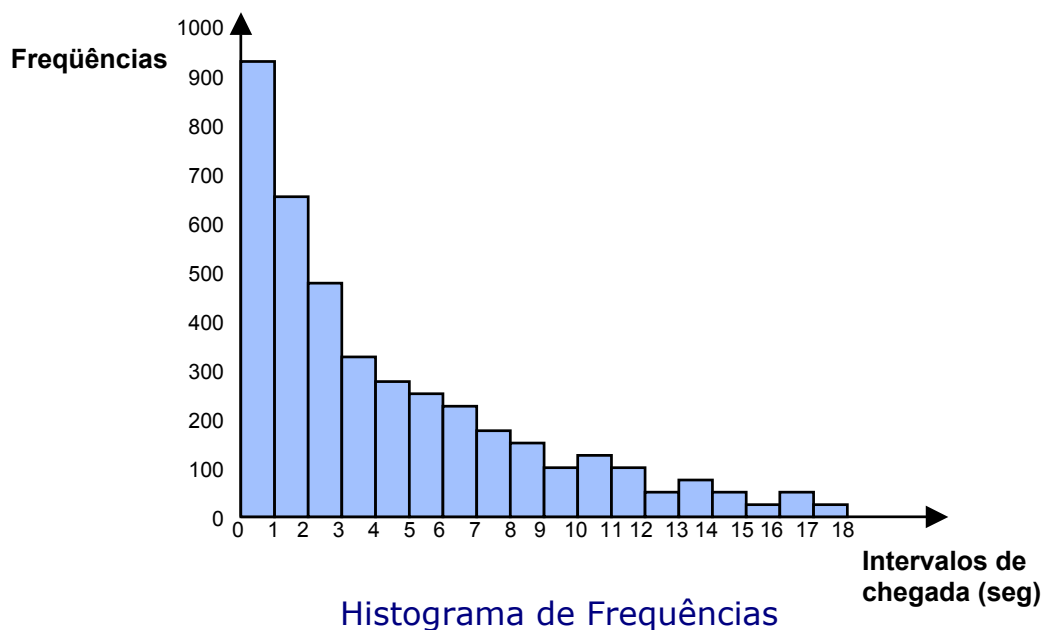


7 Escolha das Distribuições de Probabilidade dos Parâmetros de Entrada

As aplicações de simulação, em geral, dependem de parâmetros de entrada não determinísticos, isto é, que são variáveis aleatórias.

Exemplo: em comunicação de dados, a distribuição dos intervalos de chegada de mensagens, tipos de mensagens e tamanho de mensagens variam aleatoriamente de acordo com distribuições de probabilidade.

Quando for possível realizar observação do sistema real pode-se obter dados das variáveis aleatórias de entrada que serão representados através de histograma de frequências.



Existindo os dados observados do sistema real o seu uso na simulação pode ser feito de uma das formas:

1. Utilizar os próprios dados observados na simulação. Neste caso constrói-se arquivos com os dados que alimentam o simulador.
2. Ajustar uma função de distribuição empírica aos dados observados.
3. Através de técnica de inferência estatística determinar uma distribuição teórica.

A partir de dados observados podemos determinar distribuições de probabilidades que sejam aderentes aos dados observados.

Observando o histograma de frequência podemos considerar possíveis distribuições de probabilidades que sejam aderentes ao mesmo. Para isto podem ser utilizados os testes de aderência tais como Chi-quadrado e Kolmogorov-Smirnov.

A realização destas análises pode ser feita utilizando-se pacotes de estatística que já possuem os testes usuais implementados.

8 Geração de Números Aleatórios (Randômicos)

Uma vez que tenham sido determinadas as distribuições de probabilidade dos parâmetros de entrada, o simulador pode ser implementado utilizando estas distribuições. Para isto, precisamos de geradores de números aleatórios com as distribuições dos parâmetros de entrada.

8.1 Geradores de números aleatórios entre [0,1]

Em geral, as bibliotecas de linguagens de programação, possuem um gerador de números aleatórios uniformemente distribuídos entre 0 e 1.

Se a linguagem escolhida não possui um gerador de números aleatórios então se pode implementar um. A técnica mais conhecida para isto é a que se denomina de **Congruência Linear**, proposta por D. H. Lehmer em 1951 e é baseada na seguinte relação recursiva:

$$X_{k+1} = (a X_k + c) \bmod m \quad \text{para } k = 0, 1, 2, \dots$$

Onde a é um multiplicador, c é um incremento e m é o módulo. X_0 é o valor inicial denominado “semente”. Todos os números são inteiros não negativos. Deve ser observado que os números gerados serão números inteiros no intervalo $[0, m-1]$.

Para a geração de números aleatórios $\{U_1, U_2, \dots\}$ no intervalo $[0,1]$ basta fazer $U_k = X_k/m$.

Na verdade, todos os geradores computacionais devem ser considerados pseudo-aleatórios, pois partindo da mesma semente, obtém-se a mesma seqüência de números. No caso da técnica proposta para geração de números aleatórios entre 0 e 1, podemos obter apenas os valores $0, 1/m, 2/m, \dots, (m-1)/m$. Notar que, com esta técnica, a

probabilidade de ocorrer um valor entre $0,1/m$ e $0,9/m$ é 0 o que contraria a definição da distribuição uniforme entre $[0,1]$.

Para reduzir o efeito destes problemas, deve-se escolher a , c e m de forma adequada. O valor de m deve ser o maior possível. Em geral utiliza-se m que seja potência de 2 para facilitar as operações. Assim, se $m=2^n$ para algum inteiro n , o cálculo $p \bmod m$ consiste em se seleccionar os n bits mais à direita do número p . Também se pode escolher $c=0$ que a pseudo-aleatoriedade do gerador será pouco afetada.

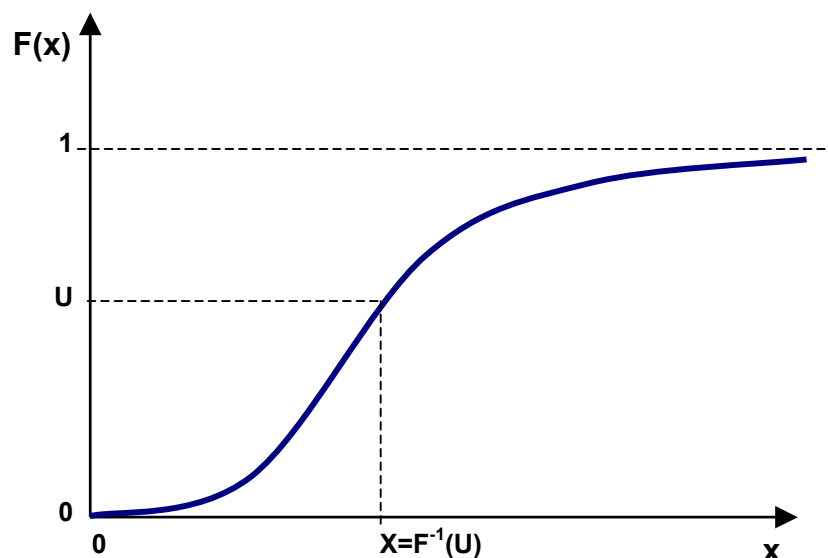
A questão agora é “Como obter números aleatórios com uma determinada distribuição de probabilidade considerando que existe disponível a função de geração de números aleatórios com distribuição uniforme”.

Métodos disponíveis:

- **Método de Transformada Inversa:** Fórmula de geração a partir da inversa da função distribuição de probabilidade.
- **Método de Aceitação/Rejeição:** Gera uma amostra de números aleatórios no intervalo desejado e aceita o subconjunto da amostra que atende à função de distribuição de probabilidades.
- **Método da Convolução:** Obtém a distribuição através de soma de outras distribuições.
- **Método de Composição:** Obtém a distribuição através de soma ponderada de outras distribuições.
- **Método baseado em propriedades especiais:** Exemplo: Através de fórmula que transforma uma distribuição em outra.

8.2 Método de Transformada Inversa

Seja $F(X)$ a função de distribuição de probabilidade da qual se quer obter a amostra X . Consideremos que a variável aleatório $U = F(X)$ possui distribuição uniforme no intervalo $[0, 1]$ (lembrar que $F(x)$ indica uma probabilidade).



Dado U no intervalo $[0,1]$, $U=F(X)$ para algum X . Desde que a inversa $F^{-1}(U)$ sempre existe, podemos escrever:

$$P[X \leq x] = P[F^{-1}(U) \leq x] = P[U \leq F(x)]$$

Como $U \sim U[0,1]$, isto é $P[U \leq u] = u$ para todo u no intervalo $[0,1]$, então para qualquer $F(x)$ no intervalo $[0,1]$, tem-se

$$P[U \leq F(x)] = F(x)$$

E então,

$$P[X \leq x] = F(x)$$

Concluindo, para gerar a amostra aleatória X geramos um número aleatório u e resolvemos a equação $u=F(x)$, isto é, obtemos o valor de x através de $x = F^{-1}(u)$.

Exemplo 3: Geração de números aleatórios exponenciais

A função de distribuição de probabilidade de uma variável aleatória exponencial é

$$F(x) = 1 - e^{-\lambda x}$$

onde $1/\lambda$ é o valor médio da variável x .

Fazendo $F(x) = u$, temos

$$x = -(1/\lambda) \ln(1-u)$$

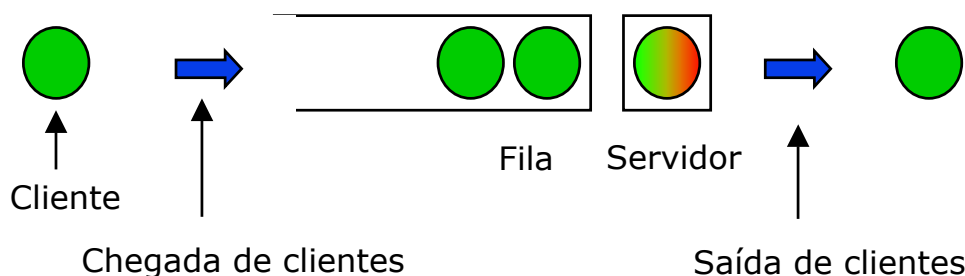
Assim se u tem distribuição uniforme no intervalo $[0,1]$, então x , calculado pela equação acima, tem distribuição exponencial com média $1/\lambda$.

9 Construção de Simuladores de Eventos Discretos

Em lugar de se utilizar pacotes, a simulação poderá ser feita através da construção de um programa simulador específico.

Exemplo 4: Simulador de Sistema de Fila Única com um Servidor

O exemplo mostra uma estrutura muito comum que é um sistema de fila com um servidor.



O sistema de fila se caracteriza por um **processo de chegada** de clientes e um **processo de serviço**. Estes processos podem ser definidos por distribuições de probabilidade.

Exemplos:

- Caixas de atendimento em um banco
- Caixas de supermercado
- Redes de comunicação de dados
- Entrada e saída de estacionamentos
- Em aeroportos: filas de embarques de passageiros, filas para decolagens e aterrissagens de aviões.
- Sistemas de manufatura: filas de serviços para serem executados em estações de fabricação ou montagem.

Neste sistema consideramos que os intervalos de chegada de clientes A_1, A_2, \dots, A_n , são variáveis aleatórias independentes e identicamente distribuídos (IID).

Os tempos de serviço S_1, S_2, \dots, S_n , também são variáveis aleatórias independentes e identicamente distribuídos (IID) e também independentes dos tempos de chegada

A disciplina de atendimento aos clientes da fila é FCFS (First Come First Served) também chamada FIFO (First In First Out). Outras disciplinas de atendimento são possíveis (LCFS, Round-robin,...).

Considerando a estrutura proposta do programa de simulação orientada a evento, devem ser implementados os procedimentos correspondentes aos eventos que causam mudanças no estado do sistema:

- Procedimento de Evento de Chegada
- Procedimento de Evento de Saída

Procedimento de Evento de Chegada

Definir o tempo do próximo evento de chegada;

se o servidor está Ocupado

então { Somar um ao número de clientes na fila

Se a fila está cheia

então retorna ("ERRO");

 Armazenar instante de chegada deste Cliente

 }

senão { /* realiza o atendimento do cliente */

 definir a espera do cliente em fila como 0

 acumular estatísticas

 somar 1 ao contador de clientes atendidos

 mudar o estado do servidor para ocupado

 definir o tempo de fim de atendimento do cliente

 }

retornar

Procedimento de Evento de Saída

```
se a fila esta vazia
    então mudar o estado do servidor para livre
    senão {/* selecionar um cliente da fila para execução */
        subtrair 1 do contador de clientes na fila
        calcular o tempo de espera em fila do cliente
        calcular o tempo de serviço do cliente
        acumular estatísticas
        mudar o estado do servidor para ocupado
        somar 1 ao contador de clientes atendidos
        avançar a fila uma posição para a frente
    }

retornar
```

A simulação terminará após a saída do sistema do n-ésimo cliente.

No Anexo D encontra-se do código em C deste simulador.

9.1 Obtenção de medidas de Desempenho

Durante a execução da simulação serão atualizadas variáveis de contagem que permitem o cálculo de medidas de desempenho.

Quatro medidas de desempenho para avaliação de sistemas de fila que podem ser utilizadas são:

- Tempo médio dos clientes na fila
- Tempo médio dos clientes no sistema
- Número médio de clientes na fila
- Utilização do servidor

Tempo Médio dos Clientes na Fila

Em uma execução do simulador, ou do sistema que o simulador representa, são observados os valores D_1, D_2, \dots, D_n dos tempos de filas dos n clientes.

O tempo médio dos clientes da fila obtido a partir de uma sequência de n observações pode ser diferente do obtido em outra rodada de observações. Assim, sendo $d(n)$ o valor esperado do tempo médio em fila em n observações, $d(n)$ é uma variável aleatória que

pode ser estimada pela fórmula $\hat{d}(n) = \frac{\sum_{i=1}^n D_i}{n}$

Tempo Médio dos Clientes no Sistema

Em uma execução do simulador, podem ser observados os valores S_1, S_2, \dots, S_n dos tempos de serviço dos n clientes. O tempo r_i que o cliente i ficou no sistema é calculado como:

$$r_i = D_i + S_i$$

Outra forma de calcular o valor r_i é através de observação dos tempos t_i de entrada do cliente i no sistema e f_i que é o tempo de sua saída do sistema e neste caso,

$$r_i = f_i - t_i$$

Assim, sendo $r(n)$ o valor esperado do tempo médio no sistema em n observações, $r(n)$ é uma variável aleatória que pode ser estimada pela fórmula:

$$\hat{r}(n) = \frac{\sum_{i=1}^n r_i}{n} = \frac{\sum_{i=1}^n (D_i + S_i)}{n} \quad \text{ou então} \quad \hat{r}(n) = \frac{\sum_{i=1}^n r_i}{n} = \frac{\sum_{i=1}^n (f_i - t_i)}{n}$$

Número Médio dos Clientes na Fila

$q(n)$ é uma variável aleatória que indica o número médio de clientes na fila para uma amostra de n clientes. Para se obter uma estimativa de $q(n)$ deve-se considerar a proporção do tempo que a fila ficou com i clientes, para todos valores de i .

A estimativa de $q(n)$ é $\hat{q}(n) = \sum_{i=0}^{\infty} i \hat{p}_i$

O cálculo da estimativa da proporção \hat{p}_i é feito como $\hat{p}_i = \frac{T_i}{T(n)}$ onde T_i é o tempo que a fila ficou com i clientes e $T(n) = T_0 + T_1 + T_2 + \dots + T_n$. Assim, a estimativa do

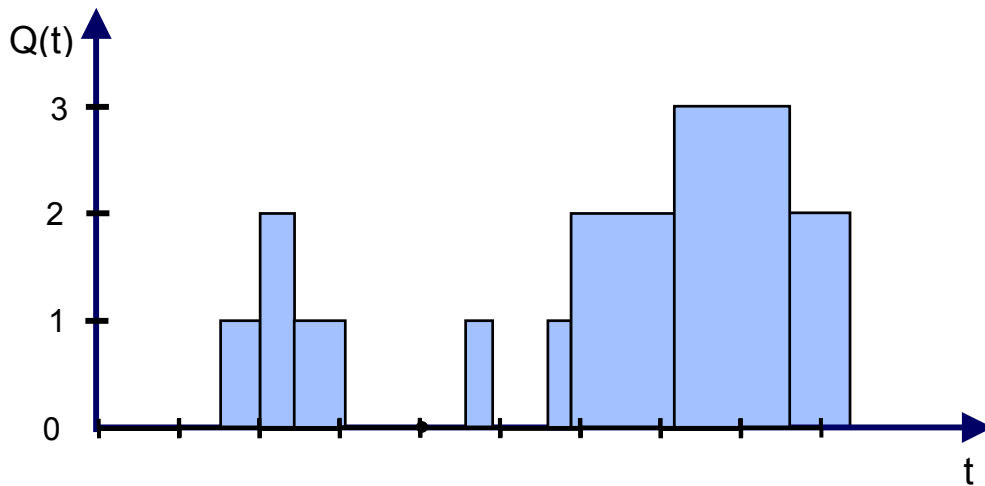
número médio de clientes na fila é $\hat{q}(n) = \frac{\sum_{i=0}^{\infty} iT_i}{T(n)}$

Pode ser observado que os valores $i T_i$, correspondem às áreas de cada retângulo da

figura a seguir, desta forma $\sum_{i=0}^{\infty} iT_i = \int_0^{T(n)} Q(t) dt$

e o novo valor do estimador de $q(n)$ é $\hat{q}(n) = \frac{\int_0^{T(n)} Q(t) dt}{T(n)}$

Esta fórmula pode ser calculada ao longo da simulação como a soma das áreas dos retângulos da figura.



Utilização do Servidor

A utilização do servidor, indicada como $u(n)$, é definida como a porcentagem do tempo que o servidor ficou ocupado ao longo da observação de n clientes.

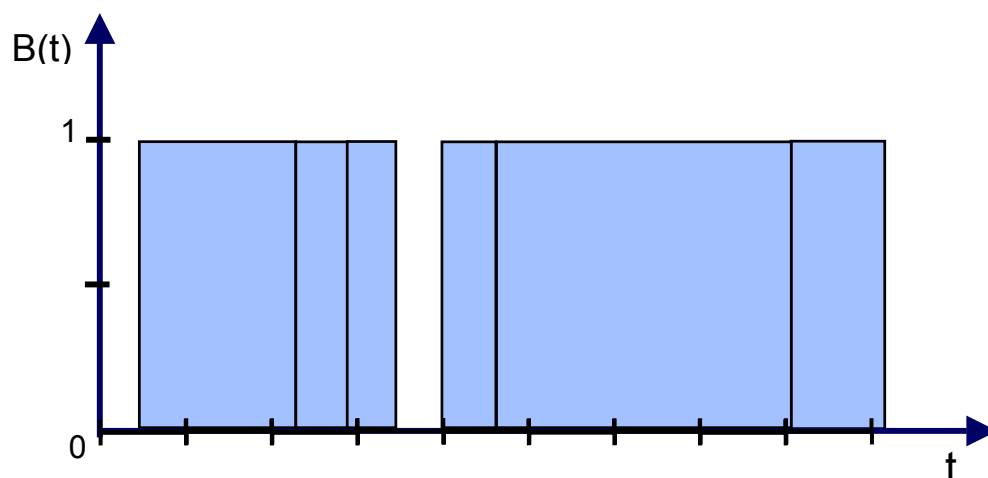
Define-se $B(t)$ como,

$$B(t) = \begin{cases} 1 & \text{se o servidor está ocupado no tempo } t \\ 0 & \text{se o servidor está livre no tempo } t \end{cases}$$

De forma análoga à que foi feita no cálculo de $q(n)$, pode-se estimar $u(n)$ como

$$\hat{u}(n) = \frac{\int_0^{T(n)} B(t) dt}{T(n)}$$

A integral da fórmula é a soma das áreas dos retângulos da figura a seguir.



10 Bibliografia

- [1] Jain, R., "The Art of Computer Systems Performance Analysis", John Wiley & Sons Inc, ISBN: 0-471-50336-3, 1991, 685 p.
- [2] Law, A. M., Kelton, W. D., "Simulation Modeling and Analysis", 3rd ed., McGraw-Hill Companies Inc, 2000, ISBN 0-07-059292-6, 760p.
- [3] Cassandras, C. G., "Discrete Event Systems: Modeling and Performance Analysis", Aksen Associates Incorporated Publishers, 1993, ISBN: 0-256-11212-6, 790p.
- [4] Soares, L.F.G., "Modelagem e Simulação Discreta de Sistemas", Editora Campus, 1992, ISBN 85-7001-703-0, 250p.
- [5] Kelton, W. D., Sadowski, R. P., Sadowski, D. A., "Simulation with Arena", McGraw-Hill Companies Inc, 1998. [Prad 99]
- [6] Prado, D., "Usando o ARENA em Simulação", Editora de Desenvolvimento gerencial, Belo Horizonte, 1999, ISBN 85-86948-19-5, 284p.
- [7] Magalhães, M. N., Lima, A. C. P., "Noções de Probabilidade e Estatística", 3 ed., IME-USP, São Paulo, 2001, 375p.

11 Exercícios

1. Simule, utilizando o pacote de simulação Arena, um posto Lava Rápido de automóveis. Para isto utilize as seguintes etapas:
 - a. Observe o funcionamento do posto durante um determinado período e identifique as etapas realizadas, o número de funcionários em cada etapa, o tempo de duração de cada etapa e o intervalo de chegada de automóveis.
 - b. A partir dos dados obtidos identifique distribuições que sejam aderentes a estes dados.
 - c. Modele e implemente o simulador, utilizando as distribuições e parâmetros medidos nas etapas anteriores.
 - d. Planeje os experimentos, execute o simulador e obtenha estatísticas de operação: tempos médios, número de automóveis na fila, etc.
 - e. Realize análises dos resultados tais como: como o aumento na taxa de chegada de carros influencia o tempo total. Faça gráficos. Como deveria ser dimensionado o posto se o intervalo médio entre chegadas caísse pela metade.
 - f. Como você modificaria o simulador para obter dados financeiros de manutenção mensal do posto?
2. Foram feitas as seguintes observações de intervalos de chegada.
5,7; 7,8; 8,0; 7,7; 5,0; 6,1; 8,0; 2,6; 3,2; 3,4; 7,7; 7,6; 7,2; 3,2; 7,6; 3,3; 6,8; 6,8; 4,4; 5,2; 4,5; 5,5; 4,1; 6,9; 7,9; 5,1; 3,7; 5,3; 3,3; 4,1; 5,7; 5,6; 4,4; 7,1; 4,1; 7,2; 4,6; 3,4; 6,6; 5,2.
 - a. Faça o histograma dos dados.
 - b. Determine para os intervalos de chegada, as funções:
 - Função densidade de probabilidade $f(x)$
 - Função distribuição de probabilidades $F(x)$
 - c. Determine um gerador de números aleatórios para gerar intervalos de acordo com a distribuição observada, utilizando o método da transformada inversa.