Um algoritmo para Modelo Determinístico D/D/1/k-1

Pedro Humberto de Almeida Mendonça Gonzaga (Universidade Estadual de Montes Claros - UNIMONTES)
pedrogonza13@gmail.com
Nilson Luíz Castelúcio Brito (Universidade Estadual de Montes Claros – UNIMONTES)
casteluciobrito@gmail.com

Resumo

Este trabalho fornece uma introdução rápida e intuitiva em Teoria das Filas, partindo desde o conceito até a aplicação de técnicas computacionais para o cálculo de uma das mais importantes medidas de desempenho do modelo determinístico D/D/1/k-1, o número de usuários no sistema em um instante de tempo qualquer. Ao término do estudo deste trabalho, o leitor obterá um suporte para desenvolver de forma simples, intuitiva e gratuita ferramentas para resolver os sistemas de filas do modelo supracitado.

Palavras-Chaves: teoria das filas, R, congestionamento, filas determinísticas.

1. Introdução

A Teoria das Filas é um ramo da Probabilidade que analisa o fenômeno de formação de filas e suas características através de conceitos básicos de processos estocásticos e de matemática aplicada, proporcionando modelos matemáticos que preveem o comportamento de um determinado sistema cuja demanda cresce aleatoriamente. O estudo desses sistemas de filas tem larga utilidade: análise de desempenho em sistemas de transportes e tráfego, manutenção de máquinas, sistemas de saúde, etc.

A partir dessa análise supracitada, obtém-se as denominadas medidas de desempenho de um sistema ou processo de filas que expressam a produtividade ou operacionalidade dos mesmos. Entre essas medidas, podem-se citar: número médio de elementos na fila e no sistema, tempo médio de espera na fila e no sistema, dentre outras.

A partir do instante que se possui os resultados dessas medidas, adquire-se, também, uma infinidade de informações valiosas sobre o futuro comportamento do sistema, facilitando tomadas de decisões, tais como a manutenção ou modificação na sua infraestrutura. Por exemplo, um administrador de um supermercado pode decidir se é necessário aumentar o número de caixas para evitar congestionamentos e insatisfações dos clientes.

Gross e Harris (2008) afirmam que a medida de desempenho mais procurada em um sistema de filas é o número de clientes no sistema em um instante de tempo qualquer, $N(t)$. Este trabalho fornece uma técnica computacional que serve de grande ajuda para o cálculo do de $N(t)$ em um modelo D/D/1/k-1, bastando o usuário ter como entrada os parâmetros tempo entre chegadas, tempo de serviço e a capacidade do sistema.

O R é um software livre para computação estatística e gráficos. Ele compila e roda em uma ampla variedade de plataformas UNIX, Windows e MacOS. Porém, o leitor pode fazer o uso dos pseudocódigos nos Anexos para desenvolver suas rotinas nas linguagens que desejarem.

2 Teria das filas

A teoria de filas é uma das mais interessantes aplicações da teoria da probabilidade. Filas acontecem sempre que a procura por determinado serviço é maior que a capacidade de atendimento oferecido. Estão bastante presente no cotidiano, seja nos supermercados, nos bancos, etc. Conforme Gross e Harris (2008), um modelo ou sistema de filas pode ser descrito como sendo clientes que chegam para receber determinado serviço, esperam pelo serviço caso não haja atendimento imediato e deixam o sistema após serem servidos. O termo "cliente" é usado num sentido geral, não implicando, necessariamente, ser um cliente humano. Por exemplo, o termo cliente pode ser usado para se referir a um programa de computador esperando para ser executado, a um carro aguardando por conserto em uma oficina, etc.

Sendo assim, "a Teoria das Filas consiste na modelagem analítica de processos ou sistemas que resultam em espera e tem o objetivo de determinar e avaliar quantidades, denominadas medidas de desempenho". (FOGLIATTI; MATOS, 2007, P.01).

Essas medidas de desempenho expressam a produtividade ou a operacionalidade do sistema. Dessas medidas, pode-se destacar: número médio de elementos na fila, tempo médio de espera pelo atendimento e tempo médio ocioso dos prestadores de serviço. A partir das mesmas, é possível saber se é necessária uma manutenção ou modificação na operação do estado atual do sistema.

2.1 Principais características de uma fila

Para abreviar a descrição dos sistemas de filas, utiliza-se a notação proposta por Kendall (1953), atualmente padronizada pela literatura de filas, consistindo numa série de símbolos e barras do tipo A/B/c/k/Z, que representam o processo de chegadas (A), processo de atendimento (B), número de servidores em paralelo (c), capacidade do sistema (k) e disciplina de atendimento (Z), sendo essas as principais características de uma fila.

Em muitos casos, apenas os três primeiros símbolos são usados. É comum a omissão das letras k e Z se o sistema tiver capacidade infinita e disciplina FCFS (first come, first served). Por exemplo, uma fila $M/M/1$ significa chegada exponencial, serviço exponencial, um único servidor, capacidade infinita e atendimento por ordem de chegada.

O processo de chegada é descrito pela distribuição do tempo entre chegadas, já o processo de atendimento pela distribuição do tempo de serviço. Essas distribuições podem ser Markovianas (M), Determinísticas (D), Erlang (E_K), Hiperexponencial (H_k) ou Genérica (G).

Pode parecer estranho o símbolo M ser usado para distribuições exponenciais. Isso se deve ao fato de que o símbolo E poderia ser, facilmente, confundido com o símbolo E_k, que é usado para distribuição Erlang do tipo K. Sendo assim, M é usado e representa a propriedade markoviana e sem memória da exponencial.

Na maioria das vezes, de acordo com Gross e Harris (2008), o que mais interessa na resolução de sistemas de filas é encontrar a distribuição de probabilidade da variável aleatória $N(t)$, "número total de clientes no sistema no instante de tempo t", que é composta pelos clientes que estão na fila, $N_q(t)$ mais os que estão no serviço, $N_s(t)$.

2.3 O Modelo Determinístico D/D/1/k-1

Considere o caso trivial de uma taxa constante de chegadas a um único servidor que processa a uma taxa de serviço constante. Essas chegadas espaçadas regularmente serão servidas na disciplina FCFS (First Come, First Served). Como λ é a taxa de chegada por unidade de tempo e μ a taxa de saída, então o tempo entre chegadas será $1/\lambda$ e o tempo entre saídas será $1/\mu$. Assuma que no tempo $t = 0$ não haja elementos no sistema e que a taxa de chegada λ seja maior que a taxa de saída μ; ou seja, $1/\lambda < 1/\mu$.

Gross e Harris (2008) abordam que com o sistema nessas características, a fila cresceria para sempre. Para evitar isso, torna-se necessário haver uma recusa forçada assim que o sistema atinja um determinado número de elementos; que é a capacidade do mesmo.

Para efeitos didáticos, considere a capacidade do sistema como $K = K - 1$, então a notação ficará da seguinte forma: $D/D/1/K-1/FCFS$. Isso significa que o k-ésimo elemento será recusado.

Esse é um exemplo de modelo determinístico. Em modelos de filas como esse, é possível prever o número exato de clientes no sistema, $n(t)$, em qualquer instante de tempo t. Para isso, basta desenvolver a equação do modelo e fazer os cálculos.

Voltando ao exemplo $\mathrm{D/D/1/K{-}1/FCFS}$, sob a hipótese de que assim que um serviço é completado, outro é iniciado, o número de usuários no sistema (incluindo o usuário em serviço) no tempo t é dado pela Equação (1):

$$n(t) = [\lambda t] - \left[\mu t - \frac{\mu}{\lambda}\right] \tag{1}$$

em que $[x], x \geq 0$ é o maior inteiro $\leq x$ e $n(0) = 0$.

Porém, essa Equação, por se tratar de um sistema com capacidade finita, só é válida até ocorrer a primeira recusa. O tempo de recusa, t_i, pode ser encontrado pela Equação (2):

$$n(t_i) = k = [\lambda t_i] - \left[\mu t_i - \frac{\mu}{\lambda}\right] \tag{2}$$

Percebe-se que após t_i, $n(t)$ permanece em $k-1$, caso o tempo de serviço seja múltiplo do tempo entre chegadas, pois a nova chegada ocorrerá no mesmo instante em que o próximo serviço for completado. Assim, para $\frac{1}{\mu} = m\left(\frac{1}{\lambda}\right)$, tem-se a seguinte Equação:

$$n(t) = \begin{cases} 0, & t < 1/\lambda \\ n(t) = [\lambda t] - \left[\mu t - \frac{\mu}{\lambda}\right], & 1/\lambda \leq t < t_i \\ k-1 & t \geq t_i \end{cases} \tag{3}$$

em que t_i é encontrado em (2).

Para o caso em que o tempo de serviço não for múltiplo do tempo entre chegadas, após o primeiro bloqueio, n(t) ocasionalmente cairá para $k-2$ tornando o cálculo um pouco mais dificultoso. Observe:

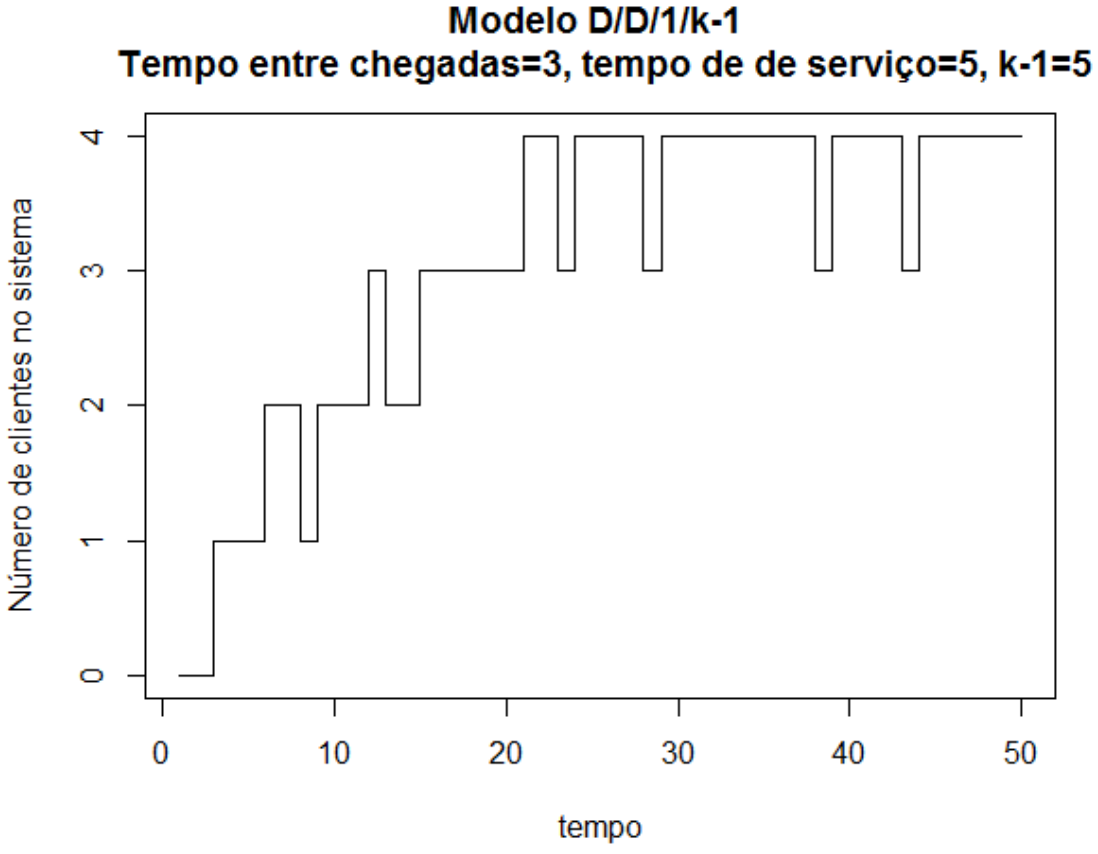
$$n(t) = \begin{cases} 0, & t < 1/\lambda \\ n(t) = [\lambda t] - \left[\mu t - \frac{\mu}{\lambda}\right], & 1/\lambda \leq t < t_i \\ k-1 \; ou \; k-2 & t \geq t_i \end{cases} \quad (4)$$

2.4 Resultados e Discussões

Como se pode notar, a complicação se encontra em saber para quais valores de $t \geq t\_i$, $n(t) = k-1$ ou $k-2$. Nos ANEXOS I, II e III encontram-se os pseudocódigos de algoritmos que automatizam os cálculos de medidas de desempenho para esse modelo. Para mais detalhes veja Gross e Harris (2008, p. 13).

Veja na Figura 1 um gráfico gerado no R para o algoritmo dos pseudocódigos, onde apresenta o resultado de um exemplo de caso para o modelo estudado acima, em que o tempo entre chegadas não é múltiplo do tempo de serviço.

Figura 1 – Exemplo Modelo D/D/1/k-1

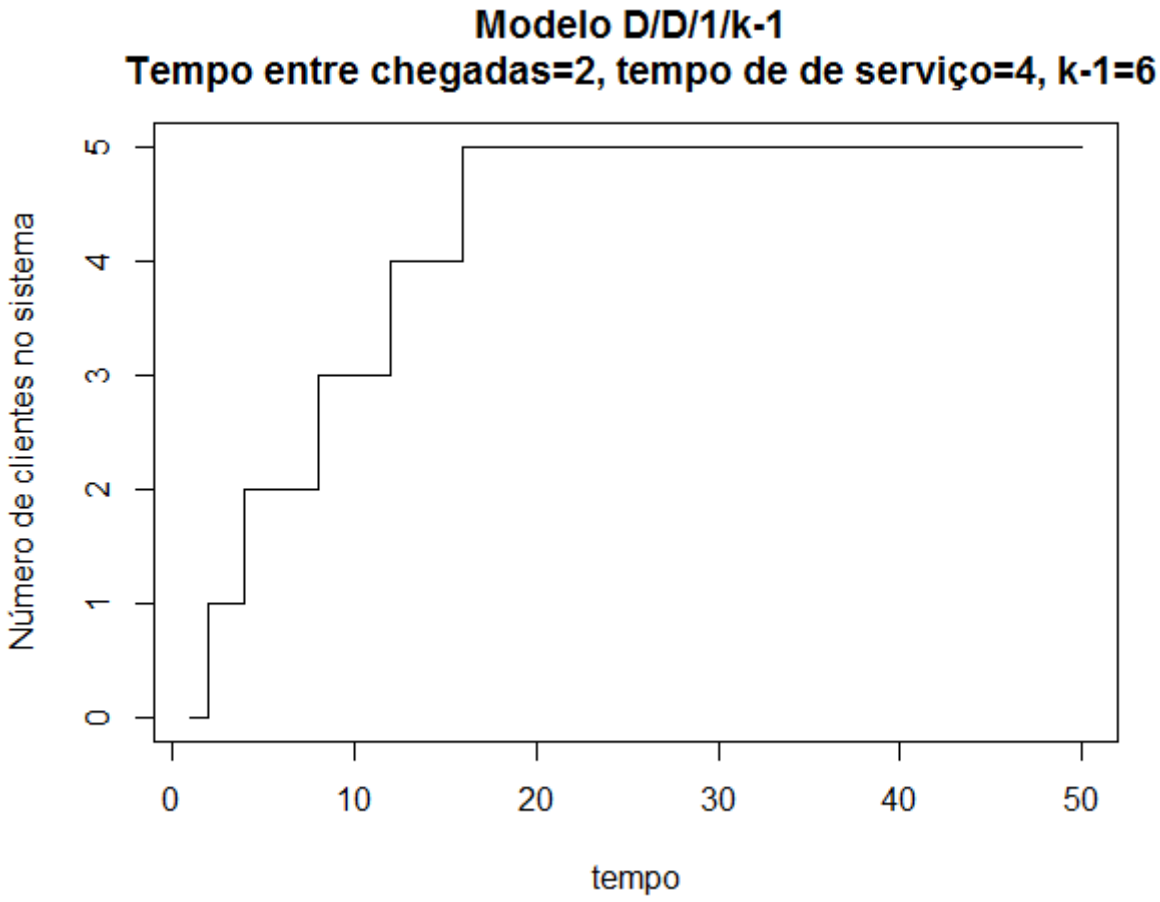


Fonte: Desenvolvida pelo próprio autor

O tempo de recusa, t_i, para esse tipo de fila é igual a 27, perceba que após esse tempo, $n(t)$ oscila entre 3 e 4 ($k-1$ e $k-2$).

Observe na Figura 2 o resultado o resultado de um exemplo de caso para o modelo estudado acima, em que o tempo entre chegadas é múltiplo do tempo de serviço.

Figura 2



Fonte: Desenvolvida pelo próprio autor

Já para esse caso, o tempo de recusa é igual a 20. Observe que após esse tempo, o número de clientes se mantém em 5 eternamente ($k-1$).

O algoritmo proposto neste artigo foi desenvolvido em linguagem R, sendo que alguns outros resultados triviais também foram automatizados, tais como tempo de recusa, o primeiro

cliente a ser rejeitado pelo sistema, a primeira chegada no sistema após a primeira recusa e a duração do ciclo para o caso do tempo de serviço ser não múltiplo do tempo entre chegadas. O ciclo é o tempo que o número de clientes no sistema começa a se repetir novamente e é encontrado simplesmente pelo cálculo do mmc entre o tempo de serviço e o tempo entre chegadas.

Para o exemplo da Figura 1 foram encontrados os seguintes resultados:

```
Tempo de Recusa: 27
O usuario 9 foi o primeiro a ser rejeitado pelo sistema
A primeira saida no sistema apos a primeira recusa ocorreu no tempo 28
A primeira chegada no sistema apos a primeira recusa ocorreu no tempo 30
Duracao do ciclo: 15 unidades de tempo.
```

Já para o exemplo da Figura 2:

```
Tempo de Recusa: 20
O usuario 10 foi o primeiro a ser rejeitado pelo sistema
A primeira saida no sistema apos a primeira recusa ocorreu no tempo 22
A primeira chegada no sistema apos a primeira recusa ocorreu no tempo 22
```

O leitor conseguirá automatizar facilmente o tempo de recusa simplesmente com uma boa analisada na Equação 2. Para a primeira chegada ao sistema após a primeira recusa, basta perceber que, como ti sempre vai ocorrer no momento de uma chegada, pode-se afirmar que essa chegada sempre vai ocorrer no tempo tc+ti (tempo entre chegadas mais o tempo de recusa). Já para calcular a primeira saída do sistema após a primeira recusa, pode ser seguido o seguinte procedimento contido no ANEXO IV.

Conclusão Considerações Finais

Foram realizados vários testes com diversos exemplos diferentes para tempo entre chegadas e tempos de serviços. Os resultados foram comparados com os cálculos manuais e os de exemplos prontos em livros didáticos. Todos os resultados foram positivos demonstrando a eficácia do algoritmo.

O R também possui um pacote que automatiza o cálculo das medidas de desempenho para modelos markovianos e alguns outros tipos de redes de fila; o pacote queueing. Para instalá-lo basta digitar o comando *install.packages("queueing")* no console e, após instalado,

carregue-o com o comando *library(queueing)* e, por fim, utilize o comando help(queueing) para seguir as instruções de uso.

REFERÊNCIAS

Fogliatti, Maria Cristina; Neli Maria Costa Mattos. **Teoria de filas**. Rio de Janeiro: Interciência, 2007.

Gross, D.; Harris, C. M. **Fundamentals of queueing theory**. John Wiley & Sons, 4 edição, 2008.

ANEXOS

ANEXO I

Algorithm 1 ALGORITMO PARA MODELO DETERMINÍSTICO - Parte 1: Tempo de serviço (t_s) múltiplo do Tempo entre chegadas (t_c)

$n \leftarrow$ Vetor nulo ▷ n é o vetor que guardará os números de usuários nos tempos t.
$k1 \leftarrow k - 1$
if $t < t_c$ **then**
    $n[t] \leftarrow 0$
else if $t_c \leq t$ **and** $t < ti$ **then** ▷ ti é o tempo de recusa.
    $nc_t \leftarrow floor(t/t_c)$ ▷ Calcula, em inteiro, o número de chegadas para o t fornecido.
    $ns_t \leftarrow floor((t - t_c)/(t_s))$ ▷ Calcula, em inteiro, o número de saídas para o t fornecido.
    $n_t \leftarrow nc_t - ns_t$ ▷ Calcula o número de usuários no sistema.
    $n[t] \leftarrow n_t$
else
    $n[t] \leftarrow k1$
end if

ANEXO II

Algorithm 2 ALGORITMO PARA MODELO DETERMINÍSTICO - Parte 2: Tempo de serviço (t_s) não múltiplo do Tempo entre chegadas (t_c)

Require: $t < ti$ ▷ Para tempos menores que o tempo de recusa.

1: **if** $t < t_c$ **then**
2: $n_t \leftarrow 0$
3: $n[t] \leftarrow n_t$
4: **else if** $t \geq t_c$ **and** $t < ti$ **then** ▷ O procedimento é análogo ao caso anterior.
5: $nc_t \leftarrow floor(t/t_c)$ ▷ Calcula, em inteiro, o número de chegadas para o ti encontrado.
6: $ns_t \leftarrow floor((t - t_c)/(t_s))$ ▷ Calcula, em inteiro, o número de saídas para o ti encontrado.
7: $n_t \leftarrow nc_t - ns_t$ ▷ Calcula o número de usuários no sistema.
8: $n[t] \leftarrow n_t$
9: **end if**

ANEXO III

Algorithm 3 ALGORITMO PARA MODELO DETERMINÍSTICO - Parte 3: Tempo de serviço (t_s) não múltiplo do Tempo entre chegadas (t_c)

Require: $t \geq ti$ ▷ Para tempos maiores ou iguais ao tempo de recusa.

$n_t \leftarrow k1$ ▷ Atribui a n_t o valor de $k1$ que foi definido como $k-1$.

while $ti \leq t$ **do** ▷ ti aumenta uma unidade a cada loop. No momento em que o valor de ti ultrapassar o valor de t o loop para.

    $cont\_tc\_ti \leftarrow 0$

    $cont\_ts\_ti \leftarrow 0$

    if $ti < primeiro\_tc\_ti$ **then** ▷ Se o valor atual de ti for menor que o primeiro tempo de chegada após a primeira recusa, manter o valor do contador como zero.

        $cont\_tc\_ti \leftarrow 0$

    else

        $primeiro\_tc\_ti \leftarrow primeiro\_tc\_ti + t_c$ ▷ Caso contrário, acrescenta-se um tempo entre chegadas ao valor atual de $primeiro\_tc\_ti$.

        $cont\_tc\_ti \leftarrow cont\_tc\_ti + 1$

    end if

    if $ti < primeiro\_ts\_ti$ **then**

        $cont\_ts\_ti \leftarrow 0$

    else

        $primeiro\_ts\_ti \leftarrow primeiro\_ts\_ti + t_s$

        $cont\_ts\_ti \leftarrow cont\_ts\_ti + 1$

        $diferencacont\_tc\_ts \leftarrow cont\_tc\_ti - cont\_ts\_ti$

        $n_t \leftarrow n_t + diferencacont\_tc\_ts$

    end if

    if $n_t > k1$ **then** ▷ Verifica condição de bloqueio

        $n_t \leftarrow k1$ ▷ Caso positivo, n_t volta a ter valor $k-1$.

    end if

    $ti \leftarrow ti + 1$ ▷ Incrementa uma unidade a ti

end while

$n[t] = n_t$

ANEXO IV

Algorithm 4 Algoritmo para cálculo do primeiro tempo de saída após primeira recusa

$primeirots \leftarrow (tc + ts)$ ▷ primeira saída do sistema. Sempre vai ocorrer assim.

$primeirotsti \leftarrow (primeirots)$ ▷ $primeirotsti$ é igual a Primeira saída após a primeira recusa

while $primeirotsti \leq ti$ **do** ▷ Enquanto $primeirotsti$ for menor que ti incremente o valor do tempo de saída (ts), o loop para quando isso não ocorrer.

    $primeirotsti \leftarrow (primeirotsti + ts)$

end while