UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM ESTATÍSTICA

CAIO COELHO MOREIRA

Estimação via simulações de Monte Carlo em uma classe de sistemas de filas G/G/C

Belo Horizonte
2023

Caio Coelho Moreira

Estimação via simulações de Monte Carlo em uma classe de sistemas de filas G/G/C

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Estatística da Universidade Federal de Minas Gerais, como requisito parcial à obtenção do título de Mestre em Estatística.
Orientador: Sokol Ndreca.

Belo Horizonte
2023

Moreira, Caio Coelho.

M838e Estimação via simulações de Monte Carlo em uma classe de sistemas de filas G/G/C [recurso eletrônico] / Caio Coelho Moreira. — 2023.
54 f. il.; 29 cm.

Orientador: Sokol Ndreca.
Dissertação (mestrado) - Universidade Federal de Minas Gerais, Instituto de Ciências Exatas, Departamento de Estatística.
Referências: f.50.

1. Estatística – Teses. 2. Monte Carlo, Método de – Teses. 3. Teoria das filas – Teses. 4. Algoritmos – Teses. I. Ndreca, Sokol. II. Universidade Federal de Minas Gerais, Instituto de Ciências Exatas, Departamento de Estatística. III. Título.

CDU 519.2(043)

Ficha catalográfica elaborada pela bibliotecária Irénquer Vismeg Lucas Cruz
CRB 6/819 - Universidade Federal de Minas Gerais - ICEx

UNIVERSIDADE FEDERAL DE MINAS GERAIS

PROGRAMA DE PÓS-GRADUAÇÃO EM ESTATÍSTICA

ATA DA DEFESA DE DISSERTAÇÃO DE MESTRADO DO ALUNO CAIO COELHO MOREIRA, MATRICULADO, SOB O Nº 2021664699, NO PROGRAMA DE PÓS-GRADUAÇÃO EM ESTATÍSTICA DO INSTITUTO DE CIÊNCIAS EXATAS DA UNIVERSIDADE FEDERAL DE MINAS GERAIS, REALIZADA NO DIA 30 DE JUNHO DE 2023.

Aos 30 dias do mês de junho de 2023, às 16h, em reunião pública virtual 269 (conforme orientações para a atividade de defesa de dissertação durante a vigência da Portaria PRPG nº 1819), reuniramse os professores abaixo relacionados, formando a Comissão Examinadora homologada pelo Colegiado do Programa de Pós-Graduação em Estatística, para julgar a defesa de dissertação do aluno Caio Coelho Moreira, nº matrícula 2021664699, intitulada: *"Estimação via simulações de Monte Carlo em uma classe de sistemas de filas G/G/C",* requisito final para obtenção do Grau de mestre em Estatística. Abrindo a sessão, o Senhor Presidente da Comissão, Prof. Sokol Ndreca, passou a palavra ao aluno para apresentação de seu trabalho. Seguiu-se a arguição pelos examinadores com a respectiva defesa do aluno. Após a defesa, os membros da banca examinadora reuniram-se reservadamente sem a presença do aluno e do público, para julgamento e expedição do resultado final. Foi atribuída a seguinte indicação:

(X) Aprovada.
( ) Reprovada com resubmissão do texto em ____ dias.
( ) Reprovada com resubmissão do texto e nova defesa em ____ dias. (
) Reprovada.

Sokol Ndreca (DEST/UFMG)
Orientador

André Luiz Fernandes Cançado (EST-UNB)

Frederico Rodrigues Borges da Cruz (EST-UFMG)

Luiz Henrique Duczmal (EST-UFMG)

O resultado final foi comunicado publicamente ao aluno pelo Senhor Presidente da Comissão. Nada mais havendo a tratar, o Presidente encerrou a reunião e lavrou a presente Ata, que será assinada por todos os membros participantes da banca examinadora. Belo Horizonte, 30 de junho de 2023.

Resumo

Muitos dos problemas reais que envolvem filas são caracterizados por processos nos quais soluções matemáticas exatas não são conhecidas ou são de difícil obtenção de forma analítica. Neste sentido, soluções que envolvem análises computacionais, simulações ou aproximações são de grande importância. Neste trabalho foi desenvolvido um algoritmo para realizar simulações de Monte Carlo em sistemas de filas gerais e, por meio dos resultados simulados foram obtidas estimativas para as medidas de interesse do sistema. A ideia central consiste em gerar aleatoriamente, e de forma independente, duas sequências que representem o tempo entre chegadas e o tempo de serviço. Uma vez que o momento da chegada de cada usuário está determinado, bem como seu tempo de serviço, tudo que irá ocorrer no sistema pode ser conhecido de forma determinística. Para avaliar os resultados obtidos por meio do algoritmo desenvolvido nesta pesquisa, estes são comparados a alguns resultados exatos ou aproximados que são apresentados ao longo do texto.

Palavras-chaves: Simulações de Monte Carlo; Teoria das filas; Algoritmos.

Abstract

Many of the real problems involving queues are characterized by processes in which exact mathematical solutions are not known or are difficult to obtain analytically. In this sense, solutions involving computational analysis, simulations or approximations are of great importance. In this research, an algorithm was developed to perform Monte Carlo simulations for general queueing systems and, through the simulated results was obtained estimates for the performance measure of interest of the system. The central idea is to randomly and independently generate two sequences that represent the time between arrivals and the service time. Once the arrival time of each user is determined, as well as his service time, everything that will happen in the system can be known in a deterministic way. To evaluate the results obtained by the algorithm developed in this research, these are compared to some exact or approximate results that are presented throughout the text.

Keywords: Monte Carlo simulations; Queuing theory; Algorithms.

Sumário

1 Introdução

Esperar em uma fila é uma situação por qual todos já passaram. Esperamos na fila do banco ou do supermercado, esperamos pelo atendimento de algum serviço de *call center*, esperamos em fila nos carros, seja por conta de um pedágio ou congestionamento, esperamos na fila dos restaurantes, seja literalmente em uma fila aguardando pelo atendimento, ou sentados esperando pelo momento em que seremos finalmente atendidos.

A verdade é que nós, enquanto clientes, não gostamos dessas esperas, e aqueles que fornecem os serviços que usufruímos, em geral, também não desejam que tenhamos de esperar. Então por que a espera ocorre? A resposta é fácil: Existe uma alta demanda e o fluxo com que essa demanda é atendida não é alto o suficiente, seja por uma questão do tempo de atendimento ou limitação de recursos, quanto mais próxima a demanda é da capacidade de atendimento, maior tende a ser a espera. Por exemplo, um restaurante pode não possuir funcionários o suficiente para atender todos seus clientes de modo que o tempo de espera seja pequeno, logo os clientes precisam esperar mais do que desejam para serem atendidos, e isso pode acontecer por ser economicamente inviável manter tantos funcionários no restaurante.

Neste sentido, conhecer a demanda por um serviço e a capacidade do atendimento é de extrema importância para responder perguntas como: "Quanto tempo um cliente irá esperar?" e "Quantas pessoas estarão esperando?". A teoria das filas tenta (e em muitos casos consegue) responder a essas e outras perguntas através de análises matemáticas detalhadas.

Entretanto muitos dos problemas reais que envolvem filas são caracterizados por processos nos quais soluções matemáticas exatas não são conhecidas ou são de difícil obtenção de forma analítica. Neste sentido, soluções que envolvem análises computacionais, simulações ou aproximações são de grande importância.

O objetivo deste trabalho é apresentar um algoritmo que pode ser aplicado para realizar a simulação de um sistema de filas, e por meio desta simulação fazer estimativas a respeito de algumas medidas de interesse. Além disso, está entre os objetivos comparar os resultados gerados por esse algoritmo a alguns resultados exatos que já são conhecidos na literatura. Por meio desta comparação, espera-se convencer o leitor a respeito da acurácia das estimativas geradas através das simulações. Por outro lado, não está entre os objetivos desta pesquisa propror que os resultados estimados pelo algoritmo sejam utilizados em detrimento dos resultados exatos, mas sim apresentá-los como uma alternativa viável e de fácil aplicação.

Ao longo do texto, o termo "simulações" é utilizado frequentemente e se refere às simulações de Monte Carlo. Este método é amplamente utilizado e bem consolidado na literatura, e consiste em realizar amostragens aleatórias de forma repetida para se obter estimativas de interesse, que seriam de difícil obtenção de forma analítica.

1.1 Notações e definições básicas

Em geral, em um sistema de filas, temos quatro medidas principais:

- Tempo de espera total no sistema.

- Tempo de espera na fila.
- Número de usuários no sistema.
- Número de usuários na fila.

O tempo médio de espera de um usuário na fila será denotado por W_q, e o tempo médio de espera no sistema denotado por W. O número médio de usuários na fila é denotado por L_q, e o número de usuários no sistema denotado por L. Neste sentido, a fila pode ser compreendida como a parte do sistema em que os usuários aguardam pelo atendimento. Já o sistema pode representar qualquer tipo de operação composta por um fluxo de chegada e saída após um certo tipo de atendimento ou serviço. Por fim, o termo "usuários" pode ser entendido no sentido mais amplo possível, sendo estes os sujeitos ao serviço ou atendimento realizado no sistema.

Uma análise quantitativa de um sistema de filas requer a caracterização de 6 componentes:

1. Padrão de chegadas dos usuários.
2. Padrão de serviço/atendimento.
3. Número de servidores/canais de atendimento.
4. Capacidade do sistema.
5. Disciplina da fila.
6. Número de estágios do atendimento.

Para caracterizar este processo usamos a notação proposta por Kendall [2], que se tornou um padrão na literatura. Seja A/B/C/Y/Z onde A denota a distribuição do tempo entre chegadas, B a distribuição do tempo de serviço, C o número de servidores, Y a capacidade do sistema e Z a disciplina da fila. Na prática, se Y = ∞ omitimos a capacidade do sistema, e caso a disciplina da fila seja o atendimento por ordem de chegada também omitimos o termo Z. Como neste texto serão considerados apenas sistemas cujo atendimento é por ordem de chegada e sem limitação de capacidade, usaremos então a notação A/B/C.

O sistema mais geral é denotado por G/G/C. G indica que a distribuição deste componente no sistema é geral. Caso fosse substituída pela letra M a distribuição referida é a exponencial, e caso fosse a letra D a referência seria para o cenário determinístico, isto é, com intervalos de tempo fixo para as chegadas ou tempo exato de atendimento.

O leitor também pode encontrar em outras referências a notação GI/G/C, que deixa explícito que a distribuição do tempo entre chegadas é geral e que as chegadas dos usuários ao sistema ocorrem de forma independente. Neste texto adotaremos a notação G/G/C e estudaremos esse sistema sob as seguintes premissas:

1. O tempo entre chegadas e o tempo de serviço podem assumir qualquer distribuição e C denota o número de servidores que atuam em paralelo.

2. O tempo entre chegadas e o tempo de serviço são independentes.

3. Só existe uma fila de capacidade infinita na qual os usuários aguardam e são atendidos por ordem de chegada.

1.2 Organização do texto

Este texto está organizado da seguinte forma: No Capítulo 2 será apresentado o algoritmo que foi utilizado para realizar todas as estimativas presentes neste trabalho. Espera-se que ao final deste capítulo o leitor compreenda a lógica por trás das simulações e como são estimadas as medidas de interesse.

No Capítulo 3 serão apresentados alguns resultados gerais válidos para sistemas G/G/C, entre eles a Lei de Little, talvez o resultado mais famoso e também mais importante a respeito deste tipo de sistema.

No Capítulo 4 serão apresentados e demostrados os resultados teóricos exatos para as medidas de interesse em um sistema M/G/1, aquele em que as chegadas ocorrem segundo uma distribuição exponencial, mas o serviço pode ser realizado segundo uma distribuição geral por apenas um servidor.

Já no Capítulo 5 a atenção será voltada para sistemas G/M/1, em que os intervalos entre as chegadas ocorrem com distribuição geral, mas o tempo de serviço do único servidor disponível segue distribuição exponencial.

Em todos estes capítulos em que são apresentados resultados teóricos, serão vistos exemplos nos quais iremos aplicar o algoritmo de simulações para obter estimativas para as medidas de interesse e compará-las aos resultados exatos. Espera-se que por meio destas demostrações e exemplos o leitor veja a complexidade do processo de obtenção de resultados exatos e a simplicidade do processo de obtenção de resultados simulados.

No Capítulo 6 será realizada uma breve comparação dos resultados do algoritmo que será apresentado neste texto com os resultados gerados pelo *queuecomputer* [3], um pacote recentemente desenvolvido em linguagem R com o mesmo objetivo do algoritmo desenvolvido nesta pesquisa. Não é objetivo desta comparação eleger o melhor método de se realizar simulações desta natureza, mas sim mostrar a consistência dos resultados gerados pelo algoritmo quando comparados aos resultados gerados por outro método.

Por fim, no Capítulo 7 serão apresentadas algumas considerações finais a respeito de tudo que será exposto ao longo do texto. Diante de tudo que foi mencionado, podemos dizer que o algoritmo que veremos a seguir tem finalidade didática, uma vez que possibilita que o leitor entenda como simulações desta natureza podem ser realizadas. Além disso, as comparações têm a finalidade de convencer o leitor sobre a acurácia dos resultados gerados pelas simulações de Monte Carlo.

2 O algoritmo de simulações

Neste capítulo será apresentado um algoritmo que pode ser utilizado para simular um sistema de filas real. A ideia central consiste em gerar aleatoriamente, e de forma independente, duas sequências que representem o tempo entre chegadas e o tempo de serviço. Uma vez que o momento da chegada de cada usuário está determinado, bem como seu tempo de serviço, tudo que irá ocorrer no sistema pode ser conhecido de forma determinística. Para entender melhor essa afirmação basta observar que, uma vez que os tempos entre as chegadas são conhecidos então o momento em que cada usuário chegará ao sistema também é, e sabendo também o tempo que levará o atendimento de cada um, é possível saber o momento exato em que o serviço será iniciado e finalizado.

Basicamente, é necessário apenas computar os momentos de chegada, início e fim dos atendimentos para cada usuário, e para isso é necessário conhecer apenas os tempos entre chegadas e tempos de serviço, que são gerados aleatoriamente e passam a ser conhecidos antes de serem utilizados como *input* no algoritmo que veremos a seguir. Ebert *et al.* [3] recentemente realizaram um trabalho também focado no uso de simulações em sistemas G/G/C e desenvolveram um pacote em R, *queuecomputer*, que realiza simulações e retorna estimativas a respeito do sistema desejado.

O algoritmo que veremos aqui foi desenvolvido de forma independente, mas quando aplicado reflete os mesmos resultados apresentados pelos autores do *queuecomputer*. Nesta pesquisa, o algortimo que veremos abaixo foi implementado em liguagem R e é ele que será utilizado para gerar os resultados simulados ao longo de todo o texto. Não está entre os objetivos deste trabalho implementar o algoritmo que será mostrado em um pacote na linguagem R, haja vista a existência do *queuecomputer* que pode ser utilizado de forma muito eficiente. Logo o principal objetivo da criação deste algoritmo é elucidar o funcionamento de um algoritmo que realiza este tipo de simulações e comparar os resultados a outros já conhecidos.

Sejam $X_1, X_2, ..., X_n$ e $Y_1, Y_2, ..., Y_n$ sequências independentes geradas aleatoriamente para representar o tempo entre chegadas e o tempo de serviço, respectivamente. Estas sequências podem ser geradas com base nas distribuições que melhor se ajustam ao sistema.

Considere que a média do tempo entre chegadas é $\frac{1}{\lambda}$ e que a média do tempo de serviço é $\frac{1}{\mu}$, sendo estes valores levados em consideração para gerar as sequências citadas acima. Para um sistema com C servidores, seja S uma matriz $C \times 3$ tal que:

- $s_{i,1} = 1$ se o servidor i está ocupado e 0 caso contrário.
- $s_{i,2}$ indica o momento de início do atendimento pelo servidor i.
- $s_{i,3}$ indica o momento projetado para o fim do atendimento do servidor i.

Durante as iterações do algoritmo os elementos da matriz S serão atualizados representando a evolução dos atendimentos, considere S^k a matriz S no passo k. Inicialmente temos $S^0 = 0_{C\times 3}$, indicando que todos servidores estão disponíveis no tempo 0.

Seja $H = (H_1, ..., H_n)^t$ um vetor tal que

$$H_k = \sum_{i=1}^{k} X_i,$$

onde $k = 1, 2, ..., n$. Ou seja, como H é formado pela soma acumulada dos tempos entre chegadas, H_k representa o momento de chegada do usuário k no sistema. Seja Y o vetor formado pelos tempos de serviço, V o vetor que indica a vacância no sistema, T o vetor que indica o tamanho da fila, E o vetor que indica a espera na fila, A o vetor que indica o momento de início do atendimento e L o vetor que indica o número de usuários no sistema. Por fim seja $F_{m\times 3}$ uma matriz tal que:

- $f_{i,1}$ indica o momento de chegada do usuário i na fila.
- $f_{i,2}$ indica o momento em que o usuário i sairá da fila.
- $f_{i,3}$ indica o tempo que o usuário i permaneceu na fila.

O número de usuários na fila é dado por m, número de linhas na matriz F, logo a matriz F pode ter seu tamanho alterado durante a execução do algoritmo. Os vetores V, T, A e L e as matrizes F e S serão alterados ou criados pelos valores encontrados durante a execução do algoritmo, seguindo os passos abaixo para todo elemento $k \in \{1, 2, ..., n\}$, representando os usuários no sistema. Os resultados obtidos após a execução do algoritmo devem ser vistos na perspectiva de um processo com tempo discreto marcado pela chegada de um novo usuário no sistema. Portanto, as medidas que serão descritas a seguir refletem o cenário que um novo usuário encontraria no sistema ao chegar. Iniciando com $k = 1$ faça:

1. Verifique e atualize a disponibilidade de cada servidor no momento da chegada do usuário k,

$$s_{i,1}^{k} = \begin{cases} 1 & \text{se} \quad s_{i,3}^{k-1} > H_k, \\ 0 & \text{caso contrário.} \end{cases}$$

   onde $s_{i,1}^{k}$ é o elemento da i-ésima linha e primeira coluna da matriz S no passo k, que indica a vacância do servidor i, e $s_{i,3}^{k-1}$ é o elemento da terceira coluna no passo $k-1$, que indica o momento do fim do atendimento do usuário anterior. Ou seja, servidor i está ocupado se o momento do fim do atendimento anterior é maior que o momento de chegada do usuário atual, para todo $i = 1, 2, ..., C$.

2. Registre a vacância dos servidores,

$$V_k = C - \sum_{i=1}^{C} s_{i,1}^{k}.$$

   A soma da coluna 1 da matriz S^k indica o total de servidores ocupados no passo k, logo a quantidade acima registra o número de sevidores livres.

3. Se $V_k > 0$, vá para o passo 4, caso contrário, vá para o passo 5.

4. Se $V_k > 0$, significa que existe pelo menos um servidor livre e o usuário não precisará esperar na fila. Nesse caso siga os passos abaixo.

   (a) Escolha i tal que $s_{i,1}^k = 0$, este é o servidor livre que irá realizar o atendimento, e faça $s_{i,1}^k = 1$, este servidor agora está ocupado.

   (b) Faça $s_{i,2}^k = H_k$, ou seja, o momento de início do atendimento é o momento em que o usuário k chegou no sistema.

   (c) Faça $s_{i,3}^k = H_k + Y_k$, isto é, o momento em que o servidor i terminará o atendimento é dada pelo momento de chagada do usuário mais o seu tempo de serviço.

   (d) Como existe pelo menos um servidor livre, logo não existe fila, então faça $m = 0$, ou seja, a matriz $F_{m,3}$ não existe.

   (e) Faça $T_k = m = 0$, isto é, o tamanho da fila no momento k é zero.

   (f) Faça $E_k = 0$ pois não houve espera na fila.

   (g) Faça $A_k = H_k$ pois o início do atendimento é no momento da chegada.

   (h) Faça $L_k = \left(\sum_{i=1}^{C} s_{i,1}^k\right) - 1$. Este é o número de usuários no sistema sem contar o usuário que acabou de chegar, ou seja, é a ocupação do sistema no momento da chegada do usuário k.

   (i) Vá para o passo 6.

5. Se $V_k = 0$, significa que não existem servidores livres e o usuário precisará esperar na fila. Nesse caso siga os passos abaixo.

   (a) Faça $m = m + 1$, ou seja, a matriz que representa a fila ganha mais uma linha para representar o novo usuário que irá esperar pelo atendimento.

   (b) Faça $f_{m,1} = H_k$, isto é, o momento que o índivíduo chega na fila é o momento que chega no sistema.

   (c) Faça $f_{m,2} = \{s_{i,3}^{k-1} : s_{i,3}^{k-1} < s_{j,3}^{k-1}, \forall j = 1, .., C, j \neq i\}$, ou seja o momento em que o usuário k sairá da fila é o momento em que o primeiro servidor finalizar o atendimento.

   (d) Faça $f_{m,3} = f_{m,2} - f_{m,1}$, isto é, o tempo de espera na fila será dado pelo momento da saída menos o momento da chegada.

   (e) Faça $E_k = f_{m,3}$, o tempo de espera fica registrado no vetor E.

   (f) Pelo passo c temos $\{i : s_{i,3}^{k-1} < s_{j,3}^{k-1}, \forall j = 1, .., C, j \neq i\}$, em que i está associado ao servidor que ficará livre primeiro e realizará o próximo atendimento.

   (g) Faça $s_{i,1}^k = 1$, pois o servidor se manterá ocupado.

   (h) Faça $s_{i,2}^k = s_{i,3}^{k-1}$, ou seja, o momento de início do atendimento é o momento em que o servidor ficou livre do atendimento anterior.

   (i) Faça $s_{i,3}^k = s_{i,3}^{k-1} + Y_k$, isto é, o momento em que o servidor i terminará o atendimento é dada pelo momento de início do atendimento mais o tempo de serviço.

(j) Atualize o tamanho da fila. Caso $f_{i,2} < H_k$ significa que o inidivíduo i já deveria ter saído da fila antes da chegada do usuário k, portanto elimine da matriz F todas as linhas nessa situação e atualize m com o número de linhas restantes.

(k) Faça $T_k = m - 1$ para representar o número de usuários na fila no momento da chegada do usuário k.

(l) Faça $A_k = s_{i,2}^k$, para registrar o momento em que iniciará o atendimento.

(m) Faça $L_k = C + m - 1$, o número de usuários no sistema no tempo k é dado pelo tamanho da fila sem contar o usuário k mais o número de servidores, uma vez que estão todos ocupados.

(n) Vá para o passo 6.

6. Se $k = n$ encerre o algoritmo, caso contrário faça $k = k + 1$ e retorne ao passo 1.

Ao final da última iteração, as matrizes S e F podem ser descartadas, por outro lado, os vetores A, V, T, E, L e Y agora armazenam os registros do sistema para todos os usuários $k = 1, 2..., n$, lembrando que estas medidas são, respectivamente, o momento do atendimento, a vacância, tamanho da fila, espera na fila, o número de usuários no sistema e o tempo de serviço. Por meio destas medidas podemos gerar estimativas relevantes para a avaliação do sistema como será mostrado nas próximas seções. Basicamente as medidas de interesse são estimadas pela média dos valores registrados nestes vetores, isto é, a média dos registros que cada usuário simulado observou ao entrar no sistema. Ou seja, a esperança do número de usuários no sistema, pode ser estimada pela média dos valores registrados no vetor L. Analogamente, a esperança do número de usuários na fila pode ser estimada pela média dos valores resgistrados no vetor T, o tempo médio de espera na fila por meio dos valores registrados em E, e o tempo total de permanência no sistema por meio da soma dos registros em E e Y.

3 Alguns Resultados Gerais

Neste capítulo veremos resultados válidos para a classe de sistemas G/G/C. Ou seja, não existe a restrição de nenhuma distribuição para os tempos de chegada ou serviço. Conhecer esses resultados é muito útil para avaliar o algoritmo apresentado anteriormente, pois espera-se que, caso as simulações sejam confiáveis, os resultados obtidos estejam próximos aos calculados pelos métodos que já são conhecidos na literatura.

Em sistemas G/G/C não existem restrições a respeito das distribuições consideradas no sistema, mas podemos destacar algumas que frequentemente são utilizadas, como as distribuições Exponencial, Gama, Weibull, Log-Normal, ou outras distribuições com suporte nos números reais positivos. Por outro lado, é comum que se assuma independêcia dos tempos entre chegadas, e para evidenciar essa condição a notação GI/G/C também é usada com frequência, onde GI indica que as variáveis aleatórias do tempo entre chegadas seguem alguma distribuição geral e são independentes.

3.1 Lei de Little

A Lei de Little indica a relação que pode ser obtida a partir de três quantidades fundamentais: A taxa λ com que os usuários chegam no sistema, o tempo médio W que um usuário permanece no sistema, e o número médio L de usuários no sistema, sendo atendidos ou aguardando atendimento. Conhecendo duas destas três medidas, é possível obter a terceira por meio da relação apresentada no teorema que será visto abaixo. Este é um resultado muito poderoso, uma vez que não estabelece restrições para os tempos entre chegadas e tempo de serviço, e pode ser aplicado em qualquer classe de sistemas de filas.

Seja $A(t)$ o total acumulado de chegadas ao sistema até o tempo t. Seja W^k o tempo que o k-ésimo usuário permaneceu no sistema. Seja $N(t)$ o número de usuários no sistema no tempo t. Defina os seguintes limites, quando existem, como

$$\lambda = \lim_{t\to\infty} \frac{A(t)}{t},$$

$$W = \lim_{k\to\infty} \frac{1}{k} \sum_{i=1}^{k} W^{(i)},$$

$$L = \lim_{T\to\infty} \frac{1}{T} \int_0^T N(t)dt.$$

O primeiro limite λ é a taxa de chegadas no sistema. O segundo limite W é o tempo médio de permanência por usuário no sistema. E o terceiro limite é o número médio de usuários no sistema.

Teorema 1. *Se os limites λ e W existem e são finitos, então o limite L existe e*

$$L = \lambda W.$$

Não será apresentada a demostração deste teorema, para mais detalhes veja Stidham [4] e Wolff [5]. Usando a Lei de Little também podemos obter outros resultados para a classe de sistemas G/G/C. É possível estabelecer relações para as quatro medidas de interesse: L, L_q, W e W_q, como mostrado na Figura 1.

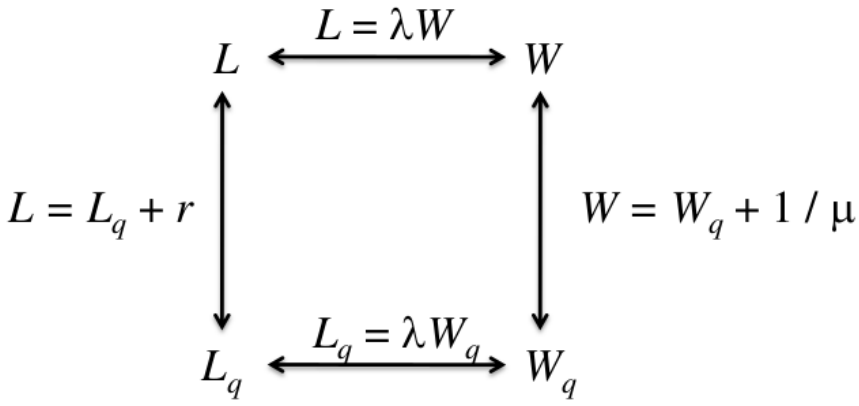

Figura 1: Relações entre as medidas de interesse. Fonte: SHORTLE et al. (2018).

Na Figura 1, $r = \frac{\lambda}{\mu}$ é a carga de trabalho do sistema, que pode ser entendida como o número médio de servidores ocupados, e μ é a taxa média de usuários atendidos por cada servidor.

Por meio destas relações é possível obter estimativas para novas medidas utilizando os resultados conhecidos para outras. Em especial, veremos como aproximar os resultados para W_q e, com isso, poderemos obter novas estimativas para as quantidades apresentadas na Figura 1.

3.2 Cotas para o tempo de espera na fila

Para muitos sistemas de filas resultados analíticos exatos não são conhecidos. Portanto, uma maneira útil de abordar esses casos é utilizando cotas inferiores e superiores, pois fornecem o pior e o melhor cenário para uma métrica de interesse. Pode-se mostrar que, para uma classe de sistemas G/G/C o tempo médio de espera na fila W_q é limitado por

- Cota superior:

$$W_q \leq \frac{\lambda(c\sigma_A^2 + \sigma_B^2/c)}{2c(1-\rho)},$$

onde σ_A^2 é a variância do tempo entre chegadas, σ_B^2 é a variância do tempo de serviço, c é o número exato de servidores, λ é a taxa de chegadas no sistema e $\rho = \frac{\lambda}{c\mu}$ é a intensidade de tráfego do sistema.

- Cota inferior:

$$W_q \geq \left(\frac{\lambda^2 \sigma_B^2 + c^2 \rho(\rho - 2)}{2\lambda c^2 (1 - \rho)} - \frac{\mu(c-1)(\sigma_B^2 + \frac{1}{\mu^2})}{2c} \right)^+ ,$$

onde $(x)^+ = \max(0, x)$.

Não é objetivo desta pesquisa apresentar os detalhes das demonstrações dos resultados acima, para tal veja [1]. Estes resultados serão utilizados mais adiante, com o objetivo de verificar se as estimativas geradas pelo algoritmo respeitam os limites mostrados acima para W_q.

3.3 Aproximações

Esta seção apresenta uma forma de realizar aproximações para avaliar o desempenho de sistemas G/G/C. Embora existam outros métodos de se realizar aproximações, que podem ser melhor explorados em [1], somente dois serão apresentados. Na seção anterior encontramos os limites para essa classe de sistemas. Limites são comprovadamente sempre válidos, mas aproximações podem fornecer resultados não acurados.

Segundo Shortle et al. [1], embora haja rigor matemático que motive a aproximação, não são apresentados resultados para avaliar sua acurácia ou precisão. Portanto, devemos ter isto em mente no momento em que forem comparados os resultados aproximados e os gerados pela simulação, seguindo os passos do algoritmo. Por outro lado, ao longo deste texto será mostrado que, diferente das aproximações, os resultados gerados pelas simulações são sim acurados, logo, podemos entender isto como uma vantagem deste método em relação ao outro.

Marchal [6] propôs aproximações para W_q em sistemas G/G/1 que se baseiam nos limites apresentados na seção aterior, basicamente, sua solução consiste em multiplicar o limite superior de W_q por um fator que se aproxima de 1 quando $\rho \to 1$. Para isso, se baseou no fato de que o limite superior se torna melhor em sistemas de alta intensidade de tráfego e o fator escolhido faz com que a aproximação seja exata para sistemas M/G/1 e D/D/1.

A fórmula proposta pode ser escrita como o produto de três fatores: variabilidade, intensidade e escala de tempo, e é dada por

$$\hat{W}_q = \left(\frac{C_A^2 + C_B^2}{2} \right) \left(\frac{\rho}{1 - \rho} \right) \left(\frac{1}{\mu} \right),$$

onde C_A é o coeficiente de variação do tempo entre chegadas e C_B do tempo de serviço. Para sistemas M/M/1 temos o resultado exato dado por

$$W_q(M/M/1) = \left(\frac{1+1}{2}\right)\left(\frac{\rho}{1-\rho}\right)\left(\frac{1}{\mu}\right) = \left(\frac{\rho}{1-\rho}\right)\left(\frac{1}{\mu}\right),$$

e baseado no fato de que para o caso G/G/1 a aproximação consiste na multiplicação do resultado para M/M/1 por um fator de variabilidade, então uma nova proposta de aproximação para os casos G/G/C é

$$\hat{W}_q = \left(\frac{C_A^2 + C_B^2}{2}\right) W_q(M/M/C),$$

em que $W_q(M/M/C)$ possui resultado exato. Esta aproximação é chamada de Allen-Cunneen (AC) [7], e pode ser mostrado que quando o sistema é formado por apenas um servidor, $C = 1$, então a aproximação se reduz à proposta por Marchal [6].

Mais recentemente, em 2022, Chaves e Gosavi [10] propuseram uma nova aproximação para sistemas G/G/C com intensidade de tráfego moderada, entre 0.5 e 0.8. Além da suposição a respeito da intensidade, também se supõe que a função de densidade da distribuição para o tempo de serviço possui a característica de ser uma função crescente do valor mínimo até a moda e em seguida decrescente até o valor máximo. Podemos citar como exemplos de distribuições com essa característica, a depender da escolha de seus parâmetros, a Gama, Weibull, Triangular, entre outras. A aproximação proposta pelos autores é dada por

$$L_q = \begin{cases} \frac{\rho^2(C_A^2+C_S^2)}{2C(1-\rho)} exp\left(\frac{(1-\rho)(1-C_A^2)}{C_A^2+4C_S^2}\right) & \text{se} \quad C_A^2 < 0.3 \text{ e } 0.15 < C_S^2 \leq 1, \\ \frac{\rho^2(1+C_S^2)(C_A^2+\rho^2 C_S^2)}{2C(1-\rho(1+\rho^2 C_S^2))} & \text{caso contrário,} \end{cases}$$

em que

$$\hat{C}_S = \begin{cases} \frac{1}{C}\frac{\sigma_B^2}{(1/\mu)^2} & \text{se} \quad C_A^2 < 0.3, \\ \frac{C\sigma_B^2}{(1/\mu)^2} & \text{caso contrário.} \end{cases}$$

Foi mostrado que a aproximação acima gera bons resultados para o tempo que os usuários permanecem no sistema, W_q, quando comparada a outras aproximações. Para mais detalhes veja Chaves e Gosavi [10].

Utilizando as aproximações aqui apresentadas para uma das medidas de interesse podemos utilizar as relações apresentadas na Figura 1 para encontrar aproximações para as demais.

3.4 Simulações

A fim de avaliar se os resultados gerados pelo algoritmo são confiáveis, primeiramente vamos compará-los aos obtidos através da Lei de Little da seguinte forma: como temos todas as

estimativas por meio das simulações, utilizaremos duas delas, calcularemos a terceira por meio das relações conhecidas e compararemos este resultado àquele de fato estimado nas simulações.

3.4.1 Lei de Little

Considere $X_1, ..., X_{2000}$ variáveis aleatórias *iid* com $X_i \sim Exp(2)$ para todo $i = 1, 2, ..., 2000$ representando os tempos entre chegadas dos usuários no sistema, e $Y_1, ..., Y_{2000}$ variáveis aleatórias *iid* com $Y_i \sim Gama(3, 2)$ representando os tempos de serviço destes usuários, com

$$f(y_i) = \frac{\beta^\alpha}{\Gamma(\alpha)} y_i^{\alpha-1} e^{-\beta y_i},$$

onde $\alpha = 3$, $\beta = 2$ e $y_i > 0$. Por fim, considere que este sistema conta com 4 servidores. Serão realizadas 10 réplicas, sendo que cada réplica consiste em avaliar o comportamento do sistema com a chegada destes 2000 usuários.

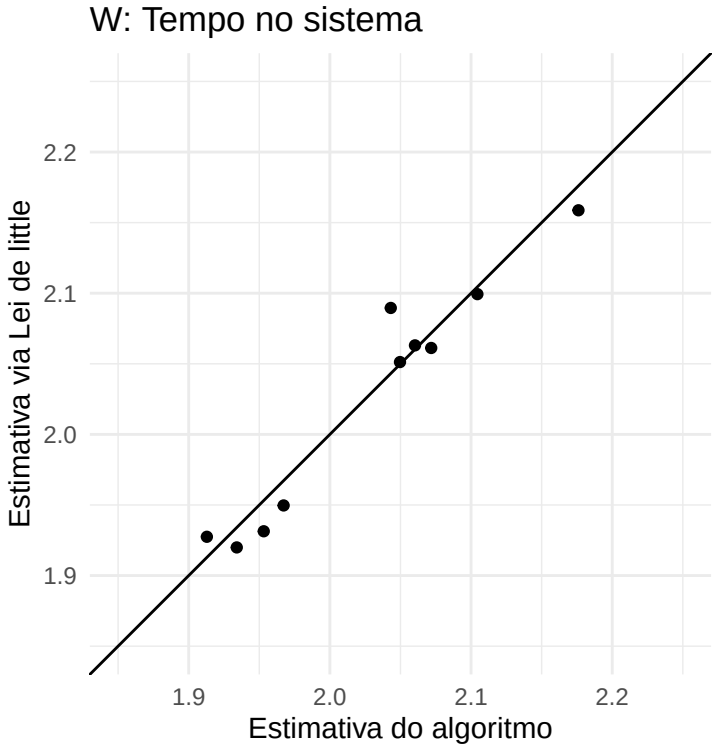

Figura 2: Tempo de espera no sistema estimado por meio das relações estabelecidas pela Lei de Little.

Conforme estabelecido pela Lei de Little, $W = \frac{L}{\lambda}$, o resultado que vemos na Figura 2[1] foi obtido considerando a estimativa do algoritmo para o número de usuários no sistema e a

[1]Este texto considera o ponto como separador decimal devido ao uso do *software* R que é desenvolvido em língua inglesa.

dividindo por $\hat{\lambda}$, que é a estimativa para λ. Este resultado é apresentado no eixo y do gráfico, onde pode ser comparado à estimativa direta de W que foi realizada pelo algoritmo considerando os tempos médios de espera da simulação e é aprentada no eixo x. Na diagonal temos a função identidade, ou seja, se os pontos recaem sobre esta reta significa que os resultados das duas abordagens são exatamente os mesmos. Já na Figura 3, em menor destaque, temos todas as demais comparações que foram realizadas em procedimento análogo.

Como pode ser analisado nas figuras, os resultados das duas abordagens são muito próximos, ou seja, a estimativa de uma quantidade, obtida diretamente por meio da simulação, leva a resultados muito próximos daqueles obtidos pela Lei de Little, em que duas quantidades são utilizadas para se encontrar a terceira. Com isso, podemos concluir que neste exemplo o comportamento do sistema simulado obedece às relações estabelecidas pela Lei de Little.

Considere agora que $X_i \sim Gama(2,3)$ para todo $i = 1, 2, ..., 5000$, e que desta vez o tempo de serviço segue distribuição $Y_i \sim Exp(4)$. Considere o sistema com apenas 1 servidor e agora 100 replicações da simulação. Como pode ser comprovado pelos gráficos da Figura 4, os resultados obtidos pelo algoritmo não mais condizem com os calculados pelas relações estabelecidas na Lei de Little.

Isto ocorre pois, conforme estabelecido no algoritmo, o número de usuários no sistema L_k é computado a cada instante k que representa a chegada de um novo usuário no sistema, e ao final da simulação a estimativa do número de usuários no sistema é dada por $L = \sum_{k=1}^{n} \frac{L_k}{n}$. Por outro lado, a Lei de Little define que

$$L = \lim_{T \to \infty} \frac{1}{T} \int_0^T N(t)dt.$$

Ou seja, o algoritmo avalia o número de usuários em tempo discreto considerando apenas os momentos de novas chegadas, enquanto a Lei de Little considera o tempo contínuo durante todo intervalo de tempo T. Logo, é natural que os resultados não sejam próximos quando o tempo entre chegadas não é exponencial.

Quando o tempo entre chegadas segue uma distribuição exponencial, a quantidade de chegadas é regida pela distribuição Poison, e neste caso vale a propriedade PASTA (*Poisson Arrivals See Time Average).* Basicamente, esta propriedade diz que, ao chegar em um sistema um usuário observa, em média, a mesma situação que um observador externo poderia ver em qualquer instante arbitrário do tempo. Para mais detalhes veja [1]. Consequentemente, estimar o número de usuários no sistema considerando apenas os momentos de chegada equivale a estimar esta quantidade para qualquer instante de tempo apenas quando as chegadas são descritas por um processo de Poisson, que não é o caso deste exemplo.

No Capítulo 5 veremos os resultados exatos para sistemas G/M/1 e este mesmo exemplo será retomado. Será mostrado que o número médio de usuários no sistema no momento da chegada de um novo usuário é 0.333, muito próximo aos mostrados na Figura 4 referentes à estimativa do algoritmo. Logo, fica claro que temos duas interpretações diferentes, uma diz respeito ao sistema em qualquer instante e outra considera apenas o momento em que um novo usuário chega ao sistema, e quando o tempo entre chegadas tem distribuição exponencial vimos que estas duas situações levam ao mesmo resultado.

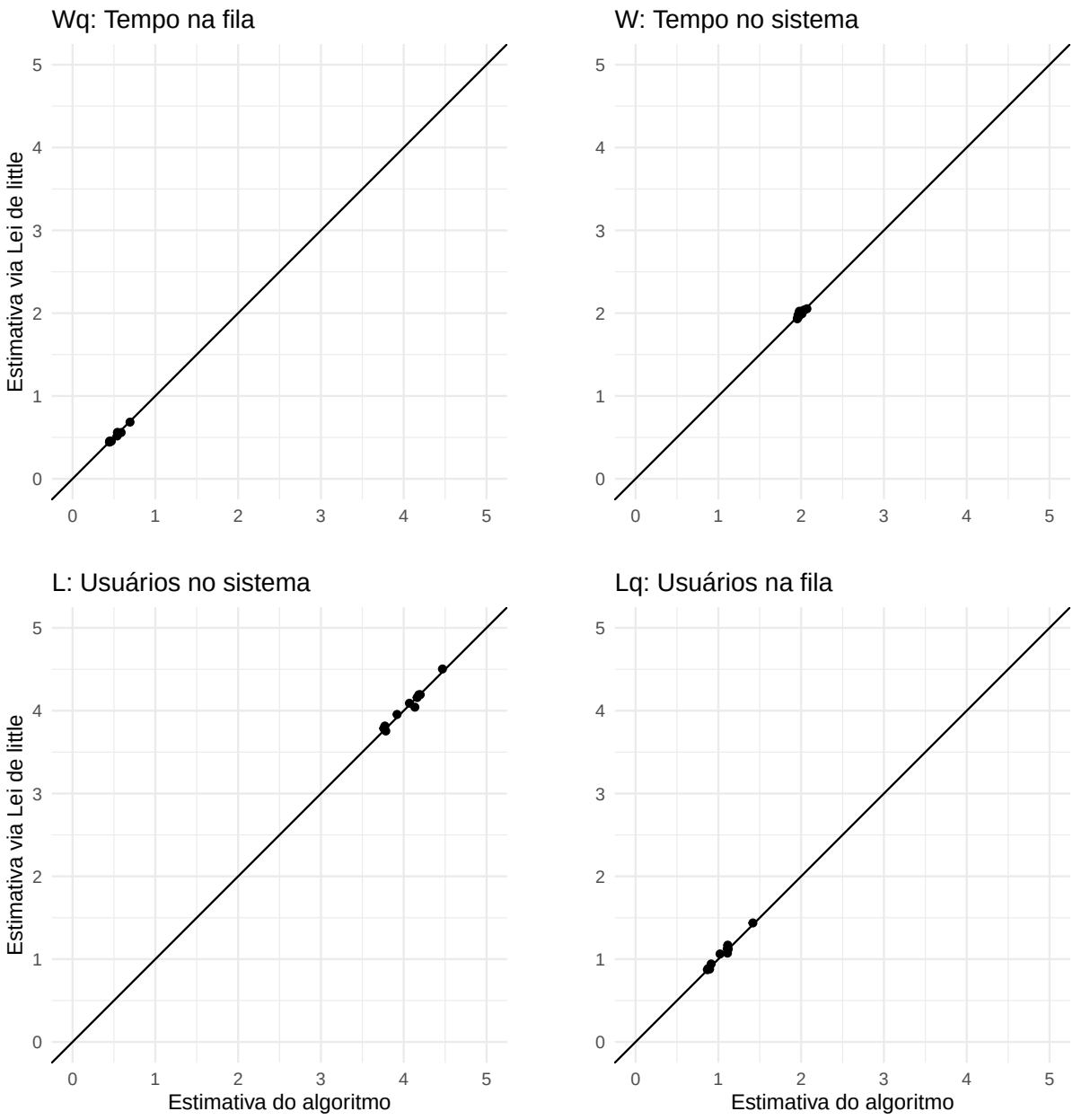

Figura 3: Medidas de interesse do sistema estimadas por meio das relações estabelecidas pela Lei de Little.

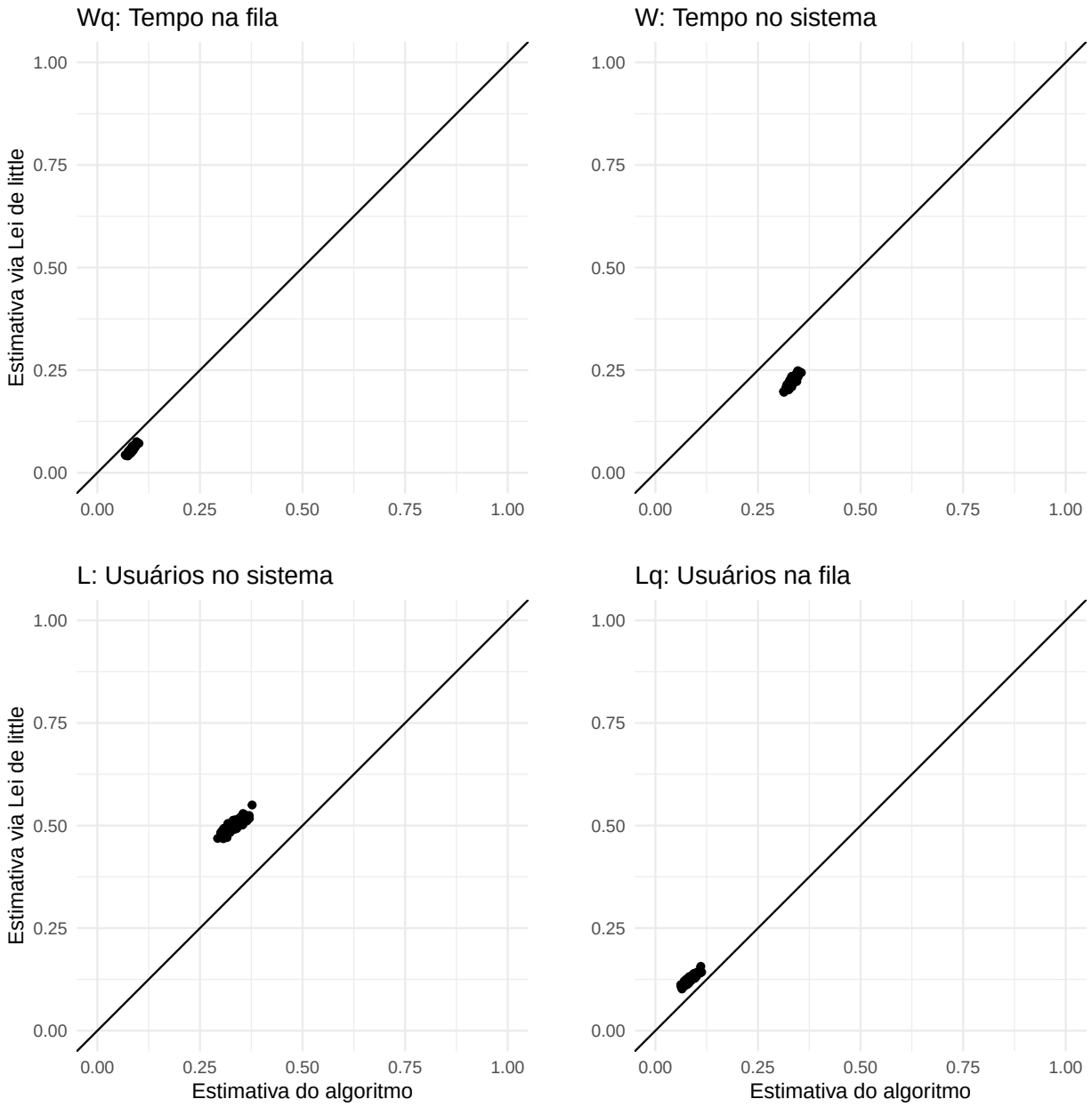

Figura 4: Medidas de interesse estimadas por meio das relações estabelecidas pela Lei de Little.

O algoritmo apresentado no Capítulo 2, considera apenas os momentos em que um novo usuário chega ao sistema. Contudo, de posse dos valores estimados para W e λ podemos utilizar a Lei de Little para estimar L considerando o cenário de tempo contínuo.

No Capítulo 6 veremos que as estimativas geradas pelo *queuecomputer* [3] refletem o cenário de tempo contínuo e portanto, quando a distribuição do tempo entre chegadas não é exponencial, os resultados divergem daqueles gerados pelo algoritmo desenvolvido nessa pesquisa com relação ao número de usuários no sistema e na fila. Por outro lado, devido à propriedade PASTA, os resultados são próximos quando a distribuição do tempo entre chegadas é exponencial.

3.4.2 Limites e aproximações

Como vimos no exemplo anterior, os resultados da Lei de Little só irão concordar com os obtidos na simulação caso os tempos entre chegadas possuam distribuição exponencial. Então, para visualizar os resultados da execução do algoritmo em um sistema G/G/C, considere que o sistema conta com 50 servidores e que desejamos obter uma alta intensidade de tráfego. Seja $\rho = \frac{\lambda}{c\mu} = 0.95$ e suponha que os tempos entre chegadas sejam $X \sim Gama(6, 2)$.

Deste modo podemos adotar a seguinte distribuição para o tempo de serviço de modo a obter a intensidade desejada $Y \sim Gama(6, \beta)$, onde $\beta = \frac{2}{50\rho}$ e $\rho = 0.95$. Ou seja, temos $E[X] = \frac{6}{2}$ e $E[Y] = \frac{6}{\beta} = 150\rho = 142.5$. Consequentemente $\lambda = \frac{1}{E[X]} = \frac{1}{3}$, $\mu = \frac{1}{142.5}$ e $\rho = \frac{\lambda}{50\mu} = 0.95$. Deste modo, podemos imaginar um sistema de 50 servidores que recebe um novo usuário a cada 3 segundos, sendo o tempo médio de atendimento igual a 142.5 segundos, ou 2.38 minutos.

O algoritmo foi aplicado utilizando amostras de tamanho 2000 geradas aleatoriamente de X e Y. Os resultados foram computados e este procedimento foi repetido 100 vezes. Com base nos limites e aproximações apresentados anteriormente, vemos a comparação dos resultados. Os histogramas mostrados na Figura 5 indicam os resultados gerados pelo algoritmo, a linha pontilhada marca os resultados aproximados e a linha contínua marca as cotas para as estimativas das medidas de interesse.

Para encontrar a aproximação e as cotas de W_q foram utilizados os resultados apresentados nas Seções 3.2 e 3.3, e os demais resultados para as outras medidas de interesse, apresentados na Figura 5, foram derivados da Lei de Little, utilizando as estimativas obtidas para W_q. Sabe-se que $W = W_q + \frac{1}{\mu}$, onde W_q tem seus limites conhecidos, mas $\frac{1}{\mu}$ é o tempo médio de serviço e para este não foram apresentados limites, e cujo os resultados de suas estimativas podem variar durante a execução do algoritmo. Logo, é possível que após a simulação sejam observados valores fora dos intervalos para L_q, W e L. Contudo, pela Lei Forte dos Grandes Números, uma vez que as estimativas tendem a convergir para os valores reais, à medida que a amostra cresce, é natural que os valores sempre estejam dentro dos intervalos quando as simulações são feitas com amostras maiores. Para ilustrar este fato, o mesmo cenário é simulado novamente, mas agora considerando a chegada de 50000 usuários no sistema para garantir que o sistema tenha atingido a estabilidade.

Como pode ser visto na Figura 6, considerando uma amostra muito maior na simulação

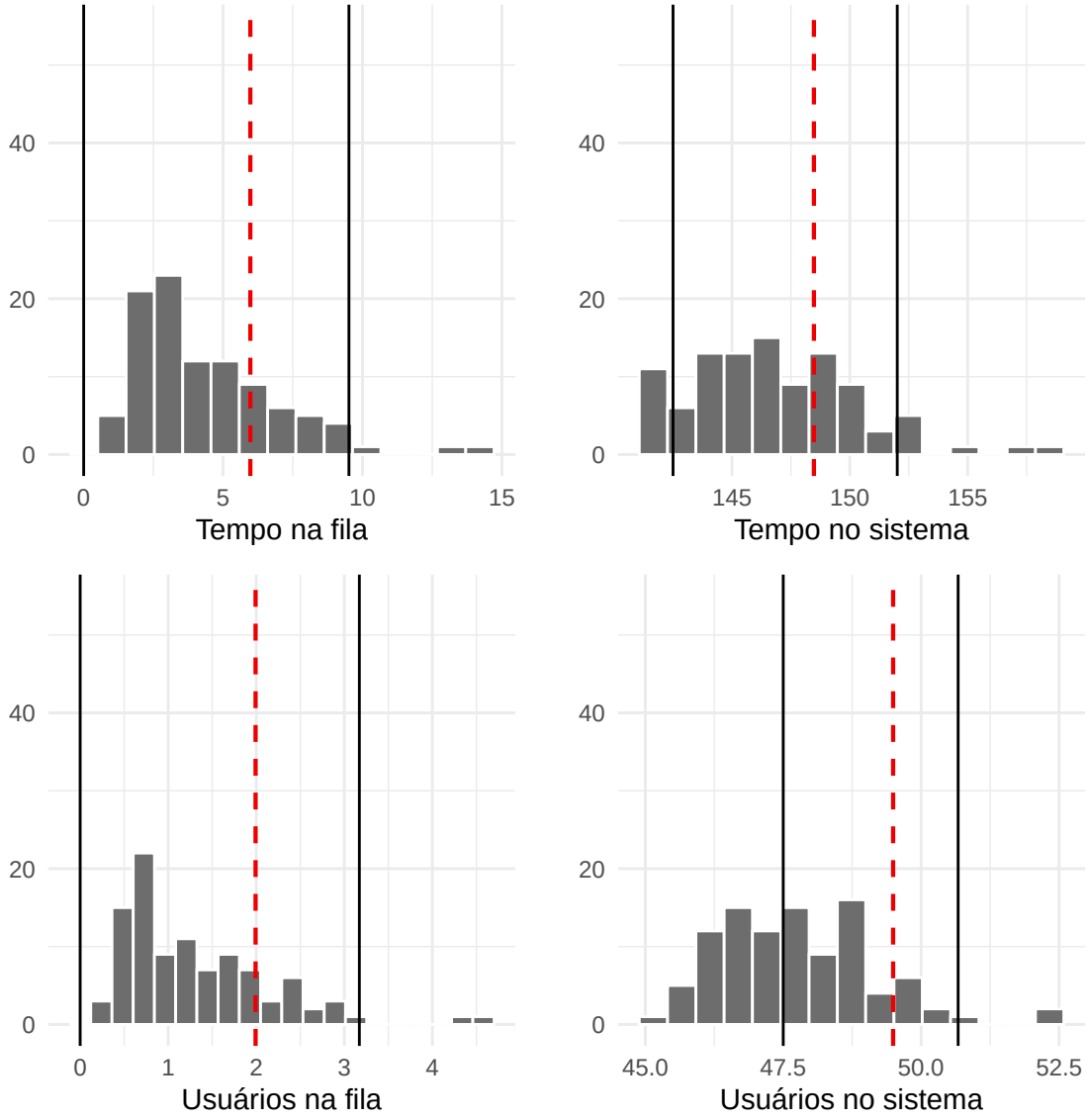

Figura 5: Limites para as medidas de interesse.

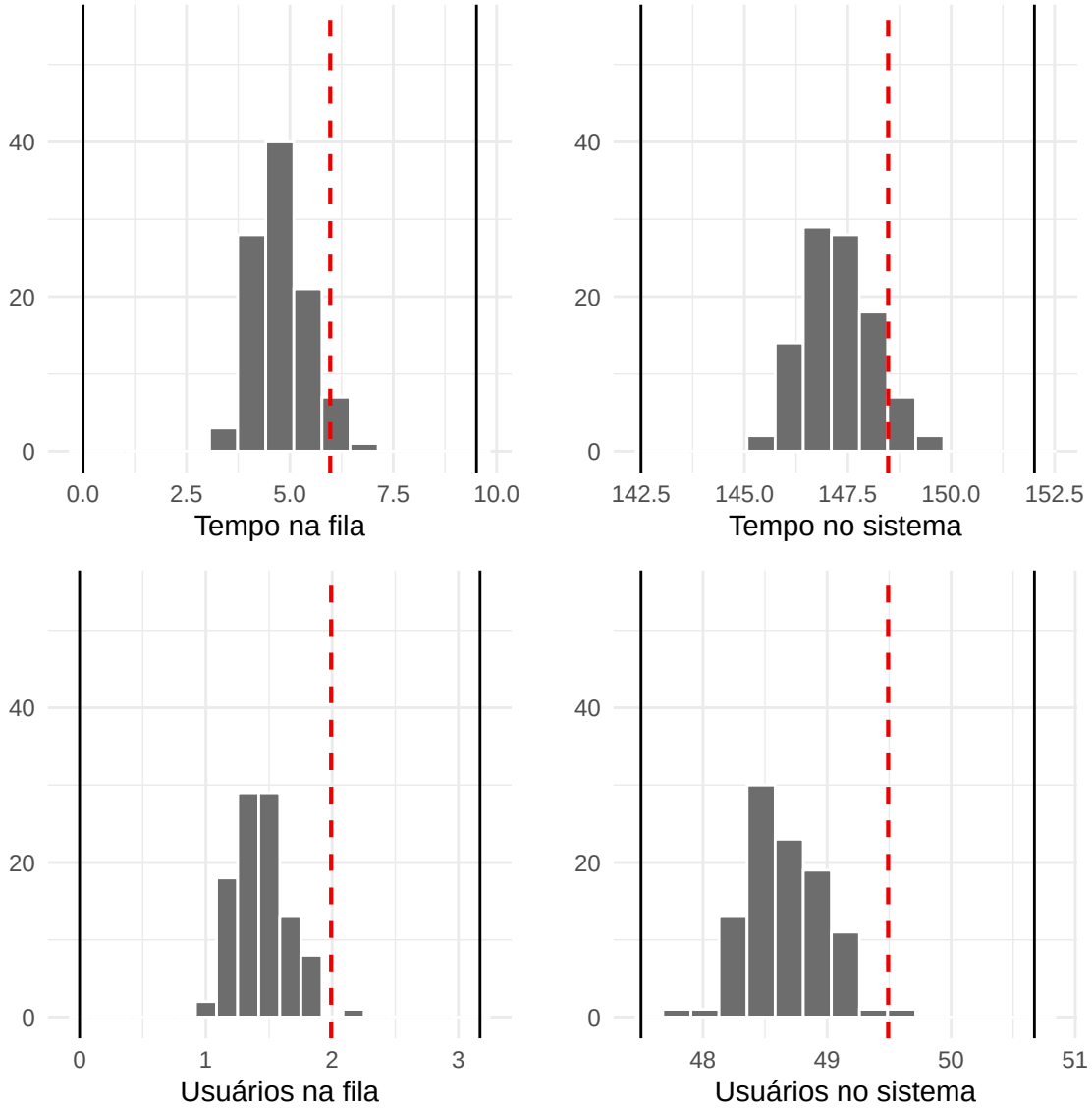

Figura 6: Limites para as medidas de interesse.

conseguimos obter a convergência dos resultados garantindo que os limites sejam válidos. Mas note que a linha pontilhada que marca os resultados aproximados está distante da média dos valores estimados nas simulações. Isto reflete o que antes foi mencionado a respeito da acurácia das aproximações e reforça a vantagem das simulações em detrimento deste método, levando em conta, obviamente, que as simulações sejam realizadas com um tamanho de amostra grande o suficiente para garantir a convergência das estimativas para os valores exatos.

A fim de comparar os resultados do algoritmo com as aproximações propostas por Chaves e Gosavi [10], considere agora um sistema com intensidade de tráfego moderada, $\rho = 0.75$, tempos entre chegadas $X \sim Gama(5, 2)$ e tempos de serviço $Y \sim Gama(5, 1.33)$, sendo este sistema formado por 2 servidores.

O algoritmo foi aplicado considerando amostras de tamanho 5000 geradas aleatoriamente para os tempos entre chegadas e tempos de serviço. O procedimento para gerar as estimativas foi replicado 100 vezes e os resultado são apresentados na Figura 7, em que as linhas pontilhadas indicam os resultados das aproximações e os histogramas são formados pelas estimativas geradas pelo algoritmo. A partir da aproximação proposta em [10] as demais foram obtidas por meio das relações estabelecidas pela Lei de Little.

Podemos notar por meio da Figura 7 que os resultados foram bem acurados para o tempo de permanência na fila e no sistema. Contudo, o mesmo não pode ser dito com relação ao número de usuários. Isso acontece pois, conforme mencionado anteriormente, sistemas em que as chegadas não ocorrem segundo um processo Poison não se valem da propriedade PASTA, ou seja, como o algoritmo computa o número de usuários no sistema e na fila apenas no momento da chegada de um novo usuário, essas estimativas não são válidas para qualquer instante de tempo.

Diante desta característica podemos considerar apenas as estimativas para W e W_q geradas pelo algoritmo e estimar as demais por meio das relações $L = \lambda W$ e $L_q = \lambda W_q$. Ou seja, por meio da Lei de Little agora temos as estimativas para o número de usuários no sistema e na fila válidas para qualquer instante de tempo, e não mais apenas considerando o momento da chegada de novos usuários. A Figura 8 considera este cenário e apresenta os resultados obtidos para o mesmo sistema, pode-se notar que os resultados obtidos desta forma são muito mais acurados para L e L_q.

Este exemplo ilustra uma grande vantagem do algoritmo, podemos obter as estimativas para o número de usuários no sistema e na fila sob duas perspectivas, considerando um instante de tempo qualquer ou apenas o momento da chegada de novos usuários.

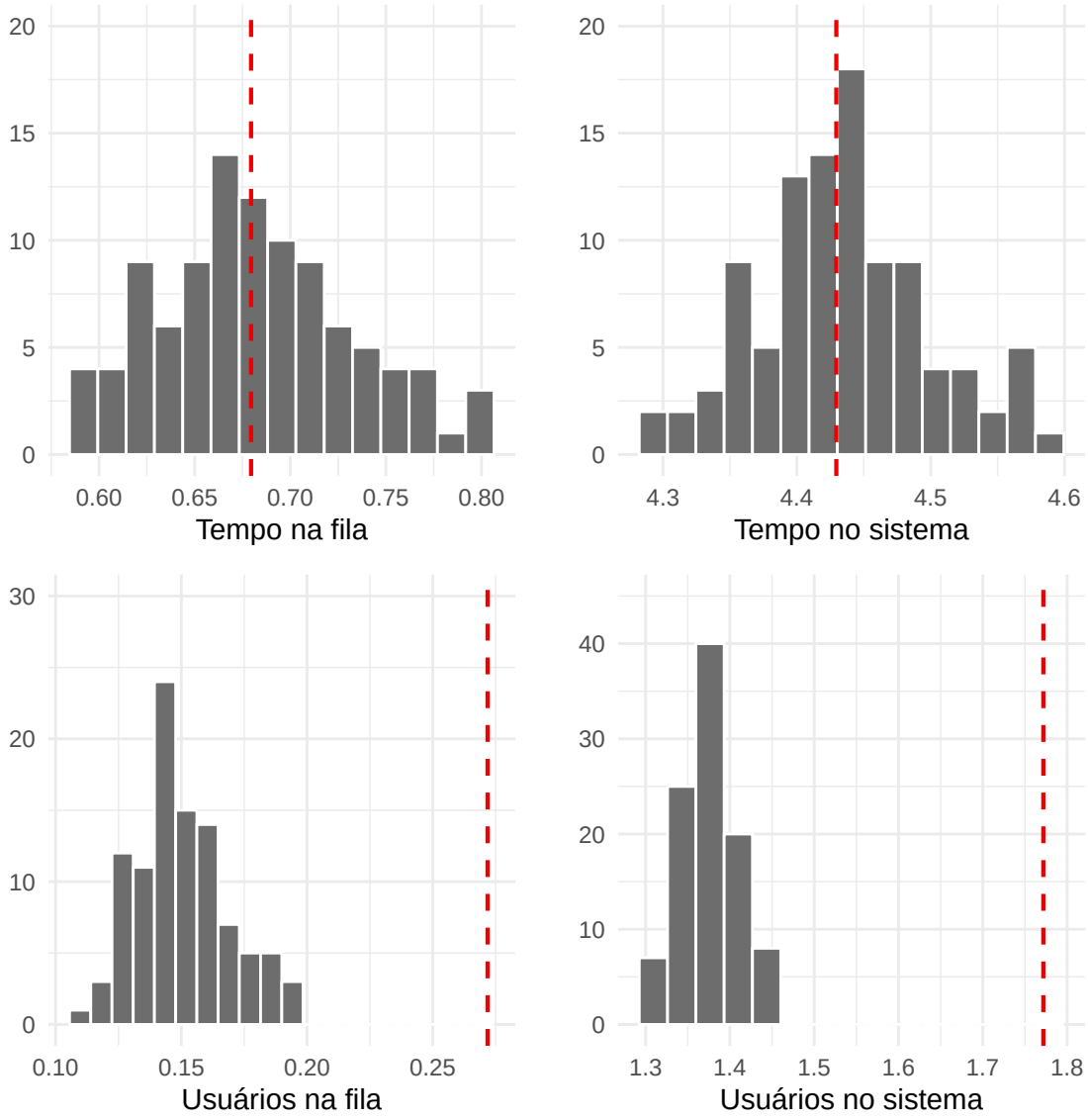

Figura 7: Aproximações para as medidas de interesse.

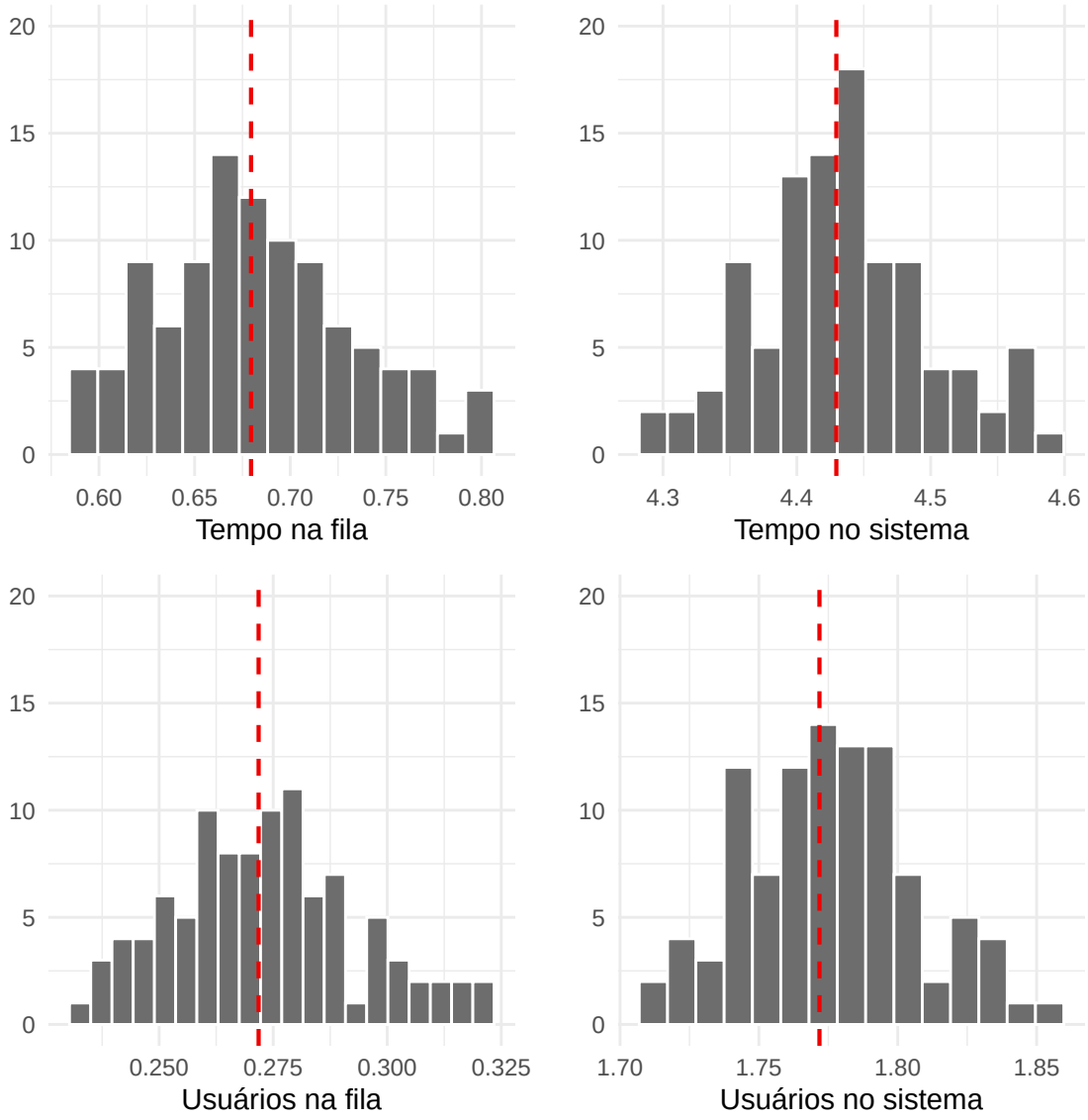

Figura 8: Aproximações para as medidas de interesse.

4 Sistema M/G/1

Neste capítulo será considerado um sistema com servidor único e chegadas seguindo um processo de Poisson, ou seja, os tempos entre as chegadas seguem distribuição Exponencial. Seja S a variável aleatória que representa o tempo de serviço com distribuição geral. Seja $\mu = \frac{1}{E[S]}$ a taxa de serviço do sistema.

Uma coleção de fórmulas para as medidas de desempenho (ou interesse) podem ser obtidas para esse sistema: W_q, W, L_q e L. Fórmulas para essas medidas são tipicamente referidas à Pollaczek-Khintchine (PK), cuja abordagem consiste em encontrar uma fórmula para uma das medidas e obter as demais utilizando as relações estabelecidas pela Lei de Little. Pensando nisso, veremos como encontrar o valor exato para W_q e as demais medidas serão encontradas como consequência das relações entre elas.

No sistema M/G/1, dada a chegada de um novo usuário, seu tempo de espera é determinado pelo número de usuários que já estão no sistema. Cada um destes usuários contribui em média $E[S]$ no tempo em que este novo usuário irá esperar. Como existem em média L_q usuários no sistema, a espera deste novo usuário causada pelos usuários da fila então será em média $L_q E[S]$.

Além disso, existe ainda o usuário que está sendo atendido pelo único servidor disponível. Este contribui no tempo de espera com uma quantidade menor que $E[S]$ uma vez que parte de seu atendimento já foi realizado, logo sua contribuição é chamada de tempo residual de serviço. Em resumo, o tempo médio de espera de um usuário novo é

$$W_q = L_q E[S] + \mathbb{P}\{\text{servidor ocupado}\} E[\text{tempo residual} \mid \text{servidor ocupado}].$$

Como $L_q = \lambda W_q$,

$$W_q = \lambda W_q E[S] + \mathbb{P}\{\text{servidor ocupado}\} E[\text{tempo residual} \mid \text{servidor ocupado}]$$

$$\implies W_q(1 - \lambda E[S]) = \mathbb{P}\{\text{servidor ocupado}\} E[\text{tempo residual} \mid \text{servidor ocupado}],$$

e como $\lambda E[S] = \lambda \frac{1}{\mu} = \rho$, temos

$$W_q = \frac{\mathbb{P}\{\text{servidor ocupado}\} E[\text{tempo residual} \mid \text{servidor ocupado}]}{1 - \rho}.$$

$\mathbb{P}\{\text{servidor ocupado}\}$ é dada pela fração de tempo em que o servidor está ocupado, ou seja, esta é a intensidade do sistema, ρ. Logo, o problema se reduz à encontrar a esperança do tempo residual dado que o servidor está ocupado no momento da chegada de um novo usuário. Será mostrado que

$$E[\text{tempo residual} \mid \text{servidor ocupado}] = \frac{E[S^2]}{2E[S]} = \frac{1 + C_B^2}{2} E[S],$$

onde C_B^2 é o coeficiente de variação do tempo de serviço ao quadrado. Para encontrarmos a quantidade acima podemos entender o problema como um processo de renovação, isto é, sempre que um serviço termina e outro se inicia temos uma renovação no sistema, e estamos interessados no tempo médio até a próxima renovação a partir de um instante t, que marca a chegada de um novo usuário. Para entender em mais detalhes a demonstração que veremos deste resultado veja Gallager [8].

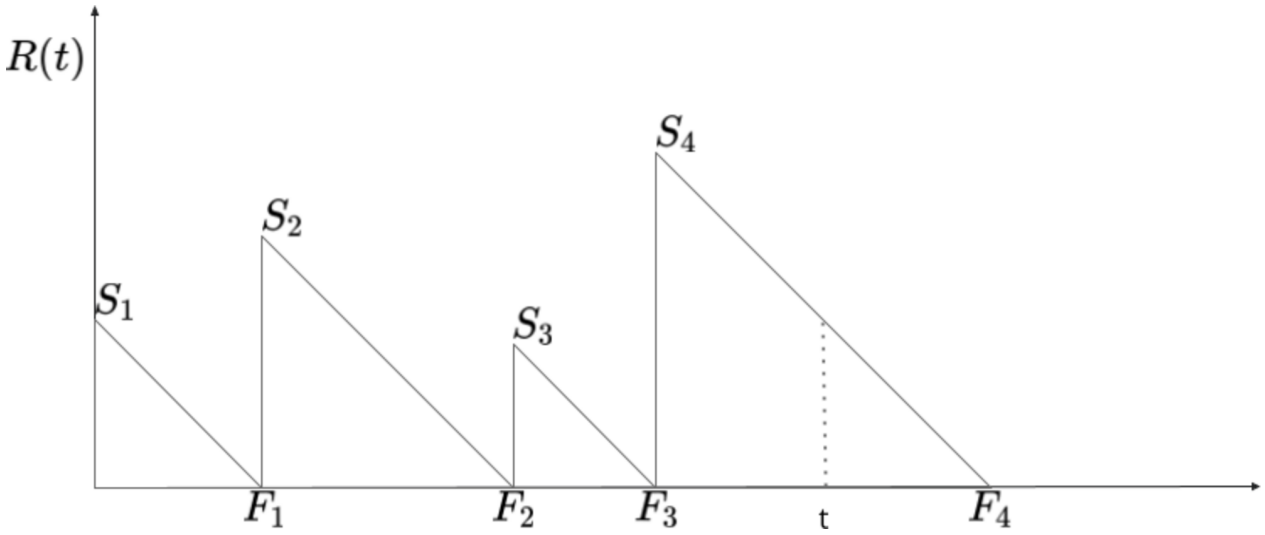

Figura 9: Tempo residual no tempo t.

A Figura 9 ilustra o cenário em que estamos interessados. F_i denota o instante em que um serviço é finalizado e S_i é o tempo de serviço do usuário i, logo, é fácil notar que cada triângulo formado na figura possui tamanho da base igual ao da altura. Seja t o momento da chegada de um novo usuário, podemos definir a função $R(t)$ para denotar o tempo residual até a próxima renovação como a altura do triângulo mostrado na figura.

O tempo médio residual $R(t)$ durante o intervalo $(0, t]$ é dado por

$$\frac{1}{t}\int_0^t R(\tau)d\tau.$$

Estamos interessados no limite desta média quando $t \to \infty$, ou seja, quando o sistema já está em equilíbrio. Note que a integral acima é basicamente a soma das áreas de triângulos isóceles, considerando apenas parte do último triângulo, ou seja

$$\int_0^t R(\tau)d\tau = \frac{1}{2}\sum_{i=1}^{N(t)} S_i^2 + \int_{F_{N(t)}}^t R(\tau)d\tau,$$

onde $N(t)$ é o número de serviços concluídos até o intante t. Diante da expressão acima, note que

$$\frac{1}{2t}\sum_{i=1}^{N(t)} S_i^2 \leq \frac{1}{t}\int_0^t R(\tau)d\tau \leq \frac{1}{2t}\sum_{i=1}^{N(t)+1} S_i^2.$$

Analisando os limites da desigualdade acima note que quando $t \to \infty$

$$\lim_{t\to\infty} \frac{1}{2t} \sum_{i=1}^{N(t)} S_i^2 = \lim_{t\to\infty} \sum_{i=1}^{N(t)} \frac{S_i^2}{N(t)} \frac{N(t)}{2t}.$$

Analisando separadamente os termos do resultado acima vemos que $N(t) \to \infty$ quando $t \to \infty$, e pela Lei Forte dos Grandes Números

$$\lim_{t\to\infty} \sum_{i=1}^{N(t)} \frac{S_i^2}{N(t)} = \lim_{k\to\infty} \sum_{i=1}^{k} \frac{S_i^2}{k} = E[S^2].$$

Já para o segundo termo, pela lei forte para um processo de renovação [8], temos

$$\lim_{t\to\infty} \frac{N(t)}{2t} = \frac{1}{2E[S]}.$$

O resultado acima é bem intuitivo, uma vez que $\frac{1}{E[S]}$ é a média de usuários servidos por instante de tempo. Combinando as duas quantidades encontradas conclui-se que

$$\lim_{t\to\infty} \frac{1}{2t} \sum_{i=1}^{N(t)} S_i^2 = \frac{E[S^2]}{2E[S]}.$$

Portanto ambos os limites existem e, como o resultado que buscamos é limitado superiormente e inferiormente pela mesma quantidade vale que

$$\lim_{t\to\infty} \frac{1}{t} \int_0^t R(\tau)d\tau = \frac{E[S^2]}{2E[S]} = \frac{1 + C_B^2}{2} E[S].$$

Finalmente, agora que todas as expressões necessárias foram encontradas, combinando todos esses resultados temos

$$W_q = \frac{1 + C_B^2}{2} \frac{\rho}{1 - \rho} E[S].$$

É fácil notar que caso o tempo de serviço também seguisse distribuição exponencial teríamos $C_B^2 = 1$ e portanto os resultados coincidiriam com os já bem conhecidos para sistemas M/M/1. Finalmente, todas as demais medidas de interesse agora podem ser facilmente encontradas utilizando as relações estabelecidas pela Lei de Little. Temos:

L	W	L_q	W_q
$\frac{1+C_B^2}{2}\frac{\rho^2}{1-\rho} + \rho$	$\frac{1+C_B^2}{2}\frac{\rho}{\mu-\lambda} + \frac{1}{\mu}$	$\frac{1+C_B^2}{2}\frac{\rho^2}{1-\rho}$	$\frac{1+C_B^2}{2}\frac{\rho}{\mu-\lambda}$

4.1 Simulações

A fim de avaliar o algoritmo que realiza as simulações considerando sistemas M/G/1 a seguir veremos exemplos assumindo diferentes distribuições para os tempos de serviço. O procedimento consistirá em realizar as simulações utilizando as distribuições escolhidas, coletar as estimativas geradas pelo algoritmo e compará-las aos resultados exatos calculados por meio das fórmulas apresentadas.

4.1.1 Distribuição Gama para o tempo de serviço

Considere um sistema com tempo entre chegadas exponencialmente distribuído com média 2 e tempos de serviço com distribuição Gama, $S \sim Gama(3, 2)$. Logo temos $E[S] = 3/2$ e $Var[S] = 3/4$, consequentemente $C_B^2 = 1/3$. Conhecendo todas as fórmulas apresentadas anteriormente e as distribuições agora descritas a respeito do sistema, podemos afirmar que as medidas de interesse exatas são conforme mostra a tabela a seguir

Tabela 2: Medidas de iteresse

Lq	Wq	W	L
1.5	3	4.5	2.25

Para exemplificar o bom funcionamento do algoritmo e a qualidade de suas estimativas, considere a simulação da chegada de 50000 usuários no sistema descrito acima. Além disso considere 100 réplicas desta simulação, os seja, como resultado temos uma amostra de tamanho 100 para as estimativas de cada uma das quantidades de interesse, onde cada estimativa é calculada considerando a simulação com os 50000 usuários. Os gráficos da Figura 10 mostram histogramas com a distribuição das estimativas de W_q, W, L_q e L, e as linhas pontilhadas marcam os valores exatos apresentados na tabela acima.

Pelos gráficos mostrados na Figura 10 vemos que as estimativas oscilam em torno de seus respectivos valores reais. Como cada estimador consiste na média dos valores observados nas simulações, conforme foi definido no Capítulo 2, vale o Teorema Central do Limite para afirmar que a distribuição dos estimadores é assintoticamente Normal. Ou seja, para cada medida de interesse temos uma amostra de tamanho 100 de uma variável aleatória normalmente distribuída. Podemos assumir a variância como sendo desconhecida e aplicar um teste de hipóteses para a média com o objetivo de testar se a média dos estimadores é igual à média real. Para tal será aplicado o teste t de Student por meio do *software* R.

Para o tempo médio de espera na fila temos as seguintes hipóteses:

$H_0 : W_q = 3$
$H_1 : W_q \neq 3$

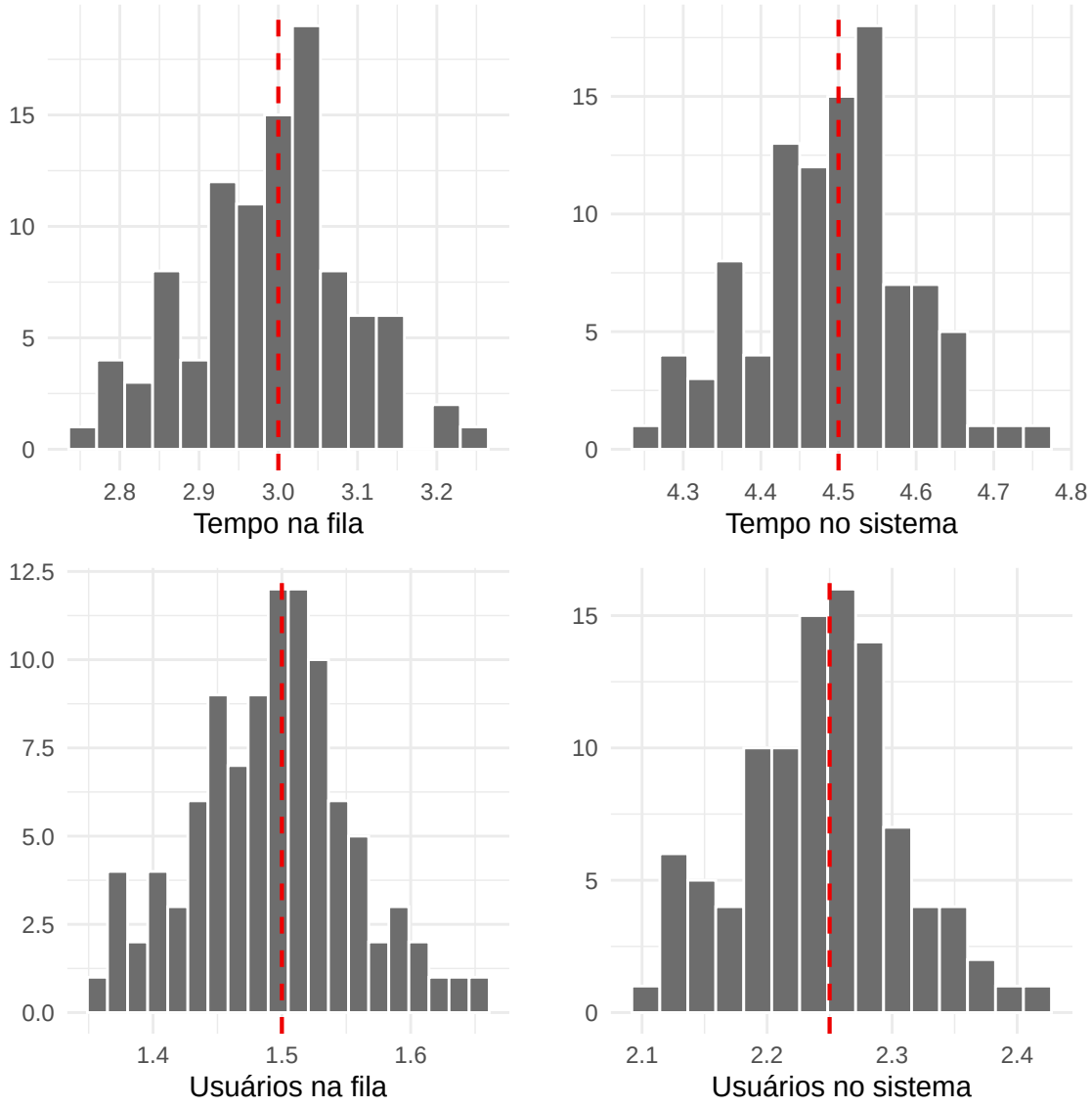

Figura 10: Estimativas para as medidas de interesse geradas pelo algoritmo.

Tabela 3: Teste de hipóteses para o tempo de espera na fila

Test statistic	df	P value	Alternative hypothesis	mean of Wq
-0.972	99	0.334	two.sided	2.99

Conforme pode ser analisado no teste acima[2], sob o nível de significância $\alpha = 0.05$, não há evidências para se rejeitar a hipótese nula, ou seja, não podemos afirmar que a média dos estimadores é diferente da média real. Em outras palavras, podemos considerar que os resultados gerados pelo algoritmo, em média, são equivalentes aos resultados exatos. O mesmo procedimento pode ser replicado para testar hipóteses a respeito das demais medidas, W, L e L_q, obtendo-se a mesma conclusão, fato que é fácil de ser notado analisando os gráficos da Figura 10.

4.1.2 Distribuição Log-normal para o tempo de serviço

Considere agora um novo sistema M/G/1 com tempo estre chegadas exponencialmente distribuídos com média 8 e tempos de serviço seguindo uma distribuição Log-normal, $S \sim Ln(1, 1)$ com

$$f(s) = \frac{1}{\sqrt{2\pi}\sigma s} e^{-(log(s)-\mu)^2/2\sigma^2}.$$

Assim como no exemplo anterior, cada simulação consistirá na chegada de 50000 usuários no sistema, e ao todo serão realizadas 100 simulações.

Com base na distribuição escolhida para o tempo de serviço temos

$$E[S] = exp\left\{\mu + \frac{1}{2\sigma^2}\right\} = exp\{1, 5\} = 4.48,$$

$$Var(S) = (e^{\sigma^2} - 1)(e^{2\mu+\sigma^2}) = (e - 1)e^3 = 34.52.$$

Consequentemente $C_B^2 = 1.72$ e assim podemos calcular exatamente todas as medidas de interesse conforme é apresentado na tabela abaixo.

Tabela 4: Medidas de interesse

Lq	Wq	W	L
0.969	7.754	12.234	1.529

Os histrogramas mostrados na Figura 11 apresentam as distribuições das medidas de interesse estimadas nas simulações. Nota-se que os valores oscilam em torno do valor real que é marcado pela linha pontilhada.

[2]Resultados apresentados diretamente da forma que foram gerados pelo R, sem tradução para Português.

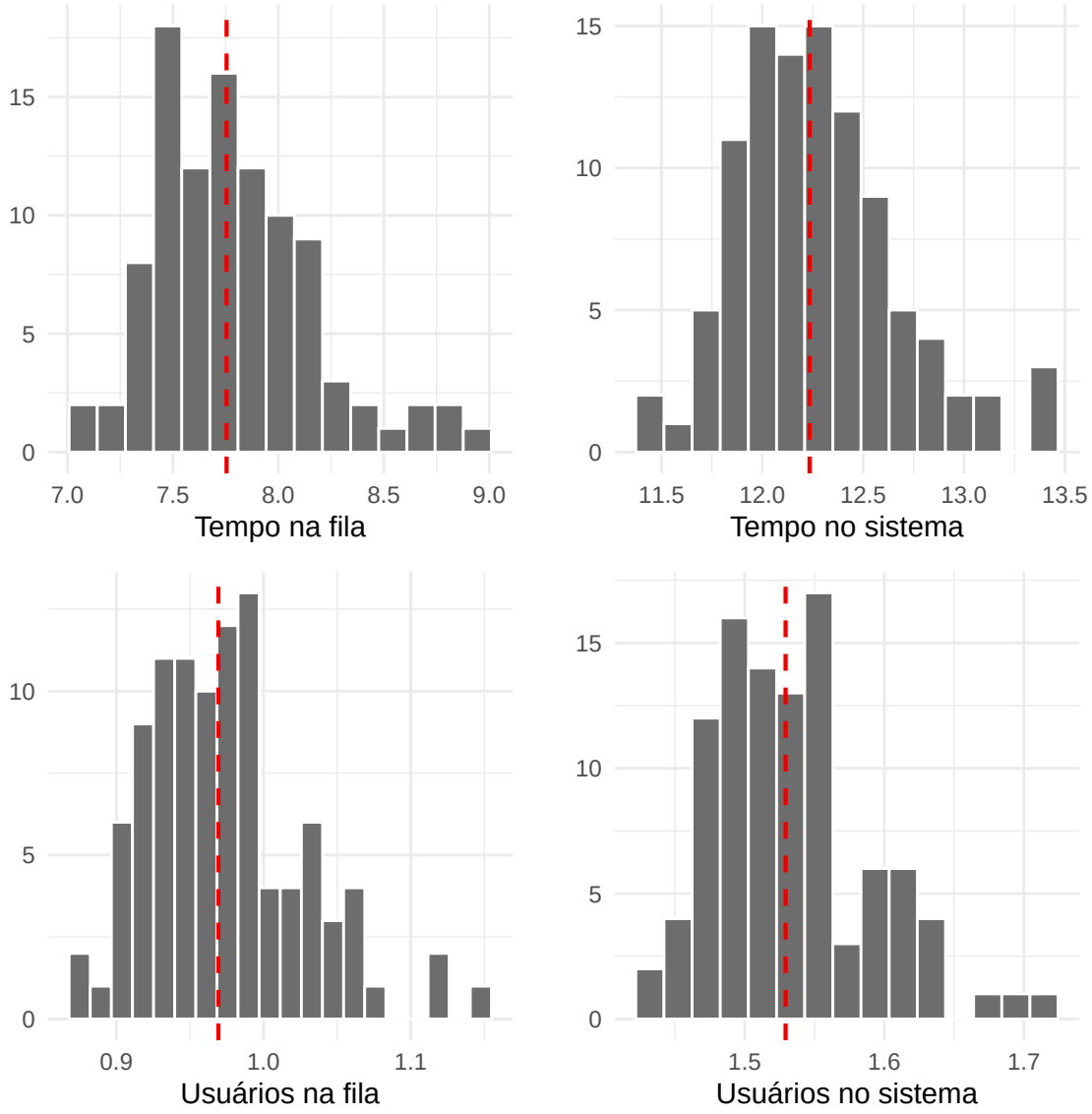

Figura 11: Estimativas para as medidas de interesse geradas pelo algoritmo.

Novamente, poderia ser aplicado um teste de hipóteses para dizer se, em média, os valores gerados pelas simulações são iguais aos reais. Mas ao invés de utilizar o Teorema Central do Limite para realizar estes testes, agora iremos nos valer da Lei Forte dos Grandes Números, que afirma que com probabilidade 1 a média aritmética converge para a média real quando o tamanho da amostra cresce. Como todas as estimativas para as medidas de interesse são calculadas como as médias dos valores observados durante as simulações, no momento da chegada de um novo usuário, é esperado então que a Lei Forte dos Grandes Números seja válida. Para ilustrar este fato considere o mesmo cenário agora simulado com a chegada de 500000 usuários no sistema. Os resultados são mostrados na tabela abaixo, em que se pode observar a proximidade dos resultados estimados aos resultados exatos.

Tabela 5: Comparação entre os resultados exato e simulado

Resultado	L	Lq	W	Wq
Estimado	1.530	0.971	12.221	7.752
Exato	1.529	0.969	12.234	7.754

Logo, podemos concluir que ter realizado uma simulação com um grande número de usuários foi suficiente para o sistema entrar em equilíbrio e as estimativas se aproximarem bastante dos resultados reais.

5 Sistema G/M/1

No sistema G/M/1 os usuários chegam ao sistema em intervalos de tempo independentes e identicamente distribuídos de acordo com alguma função de distribuição geral. O tempo médio entre chegadas é $\frac{1}{\lambda}$ e o serviço é exponencialmente distribuído com média $\frac{1}{\mu}$. Para obter estabilidade, assim como nos demais sistemas, é necessário que $\rho = \frac{\lambda}{\mu} < 1$.

Denote por L_k^a o número de usuários no sistema no momento imediatamente anterior a k-ésima chegada. Defina D_{k+1} como o número de usuários servidos entre o intervalo da k-ésima e da $(k+1)$-ésima chegada. Temos

$$L_{k+1}^a = L_k^a + 1 - D_{k+1},$$

se $L_k^a \geq D_{k+1}$. Ou seja, quando o próximo usuário chega, o número de usuários no sistema será dado pela quantidade que havia quando o usuário anterior chegou, menos a quantidade servida neste intervalo, mais um para representar o novo usuário que chega ao sistema. A partir desta equação vemos que a sequência $\{L_k^a\}_{k=0}^{\infty}$ forma uma cadeia de Markov com probabilidades de transição

$$p_{i,j} = \mathbb{P}(L_{k+1}^a = j | L_k^a = i).$$

Note que $p_{i,j} = 0$ para todo $j > i+1$ e para $j \leq i+1$, $p_{i,j}$ é a probabilidade de exatamente $i+1-j$ usuários serem servidos durante o intervalo da k-ésima e $(k+1)$-ésima chegada. Portanto temos a seguinte matriz de probabilidades de transição P

$$\begin{pmatrix} p_{0,0} & \beta_0 & 0 & \dots & & \\ p_{1,0} & \beta_1 & \beta_0 & 0 & \dots & \\ p_{2,0} & \beta_2 & \beta_1 & \beta_0 & 0 & \dots \\ \vdots & & & & & \ddots \end{pmatrix},$$

onde β_i denota a probabilidade de i usuários serem servidos durante o intervalo das chegadas. Para calcular β_i denotamos a duração do intervalo por t e levamos em conta que o número de usuários servidos pode ser descrito por uma variável aleatória Poisson com média μt, uma vez que o tempo de serviço é exponencialmente distribuído. Assim,

$$\beta_i = \int_{t=0}^{\infty} \frac{(\mu t)^i}{i!} e^{-\mu t} f_a(t) dt,$$

onde f_a é a função densidade da variável aleatória que representa o tempo entre chegadas. Uma vez que a soma de cada uma das linhas da matriz deve ser igual a um temos

$$p_{i,0} = 1 - \sum_{j=0}^{i} \beta_j.$$

Desejamos encontrar a distribuição limite $\{\pi\}_{n=0}^{\infty}$ que satisfaça $\pi = \pi P$, ou seja

$$\pi_0 = \pi_0 p_{0,0} + \pi_1 p_{1,0} + \pi_2 p_{2,0} + ... = \sum_{i=0}^{\infty} \pi_i p_{i,0},$$

$$\pi_n = \pi_{n-1}\beta_0 + \pi_n \beta_1 + \pi_{n+1}\beta_2 + ... = \sum_{i=0}^{\infty} \pi_{n-1+i}\beta_i.$$

Seja σ um operador tal que $\sigma^n = \pi_n$. Temos

$$\pi_n = \pi_{n-1}\beta_0 + \pi_n \beta_1 + \pi_{n+1}\beta_2 + ...$$

$$\implies \pi_n - (\pi_{n-1}\beta_0 + \pi_n \beta_1 + \pi_{n+1}\beta_2 + \dots) = 0$$

$$\implies \sigma^n - (\sigma^{n-1}\beta_0 + \sigma^n \beta_1 + \sigma^{n+1}\beta_2 + \dots) = 0.$$

Por meio do operador definido agora temos a seguinte equação polinomial

$$\sigma^{n-1}(\sigma - \beta_0 - \sigma\beta_1 - \sigma^2\beta_2 ...) = 0$$

$$\implies (\sigma - \beta_0 - \sigma\beta_1 - \sigma^2\beta_2 ...) = 0$$

$$\implies \sigma = \sum_{i=0}^{\infty} \sigma^i \beta_i.$$

Substituindo pelo valor de β_i temos

$$\sigma = \sum_{i=0}^{\infty} \sigma^i \int_{t=0}^{\infty} \frac{(\mu t)^i}{i!} e^{-\mu t} f_a(t) dt$$

$$= \int_{t=0}^{\infty} \sum_{i=0}^{\infty} \frac{(\sigma\mu t)^i}{i!} e^{-\mu t} f_a(t) dt,$$

onde $\sum_{i=0}^{\infty} \frac{(\sigma\mu t)^i}{i!} = e^{\sigma\mu t}$. Logo temos

$$\sigma = \int_{t=0}^{\infty} e^{\sigma\mu t} e^{-\mu t} f_a(t) dt$$

$$= \int_{t=0}^{\infty} e^{-t(\mu - \sigma\mu)} f_a(t) dt.$$

A integral acima pode ser reconhecida como a transformada de Laplace da distribuição do tempo entre chegadas f_a. Logo temos $\sigma = \tilde{A}(\mu - \mu\sigma)$, em que $\tilde{A}$ é usado para denotar a tranformada.

Note que $\sigma = 1$ é uma solução para a equação acima, uma vez que $\tilde{A}(0) = \int_{t=0}^{\infty} f_a(t)dt = 1$. Mas esta raiz não é útil, veremos agora que existe uma única raiz no intervalo $0 < \sigma < 1$ quando $\rho < 1$, temos

$$\sigma = \tilde{A}(\mu - \mu\sigma) = \sum_{i=0}^{\infty} \sigma^i \beta_i = H(\sigma).$$

Considere agora duas funções de σ separadamente no plano cartesiano, $y = \sigma$ e $y = H(\sigma)$. Observe que quando estas duas curvas se interceptam temos a solução da equação acima. Já vimos que há uma intercessão em $\sigma = 1$ e agora buscaremos por outra solução. Primeiro observe que $H(0) = \sum_{i=0}^{\infty} 0^i \beta_i = \beta_0 < 1$ e $H(1) = \sum_{i=0}^{\infty} 1^i \beta_i = 1$, ou seja, para $\sigma \in (0, 1)$ temos $H(\sigma) \in (0, 1)$. Além disso podemos provar que $H(\sigma)$ é não decrescente e convexa por meio de suas derivadas de primeira e segunda ordem. Finalmente, então existem apenas duas possibilidades para o comportamento dos gráficos de $y = \sigma$ e $y = H(\sigma)$.

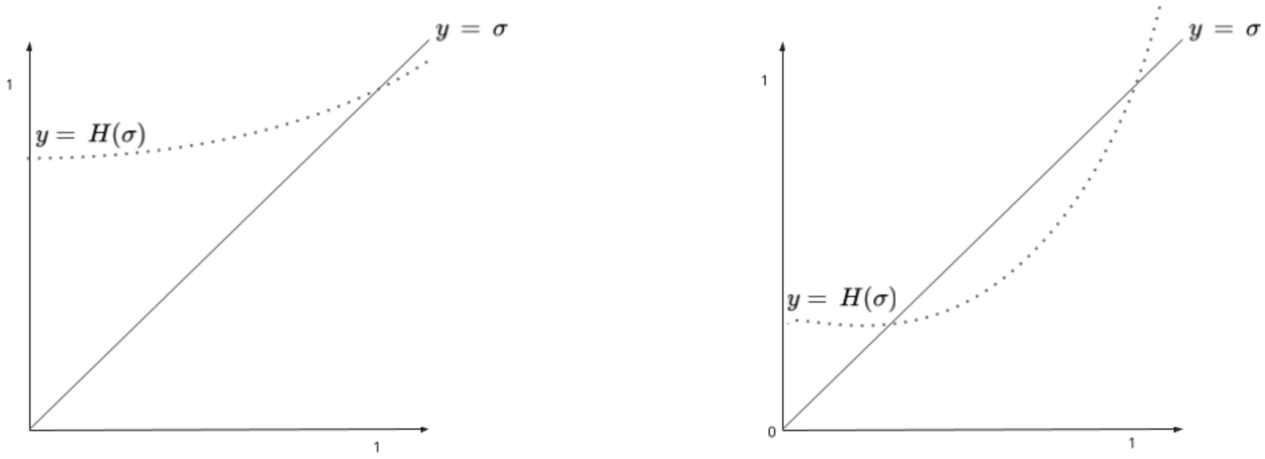

Figura 12: Possibilidades para a função H(.).

Na primeira possibilidade mostrada na Figura 12, $H(\sigma)$ não toca a bissetriz no intervalo $(0, 1)$ e toca apenas em $\sigma = 1$, já na segunda opção a curva toca a bissetriz apenas uma vez no intervalo e depois novamente em $\sigma = 1$. A segunda opção ocorre apenas se a inclinação em $\sigma = 1$ for maior que a inclinação da bissetriz, ou seja, se

$$H'(1) > 1$$

$$\implies \sum_{i=0}^{\infty} i(1)^{(i-1)} \beta_i = \sum_{i=0}^{\infty} i\beta_i.$$

Note que o resultado acima é exatamente a esperança de β_i, o número médio de usuários servidos durante o intervalo de tempo entre as chegadas, que é dado pelo número médio de usuários servidos μ, vezes a amplitude média do intervalo entre as chegadas $\frac{1}{\lambda}$. Ou seja se

$$H'(1) = \frac{\mu}{\lambda} > 1,$$

então existe apenas uma raiz no intervalo $(0, 1)$. Em outras palavras, é necessário que $\rho = \frac{\lambda}{\mu} < 1$.

Considere então que a única raiz no intervalo $(0, 1)$ é σ. Lembrando que $\pi_n = \sigma^n$ satisfaz a condição de equilíbrio, precisamos apenas normalizar a solução

$$\sum_{i=0}^{\infty} \pi_i = \sum_{i=0}^{\infty} \sigma^i = \frac{1}{1-\sigma}.$$

Finalmente, para que $\sum_{i=0}^{\infty} \pi_i = 1$ basta fazer

$$\pi_n = (1-\sigma)\sigma^n.$$

Isso conclui que o número de usuários no sistema antes da chegada de um novo usuário, quando o sistema está em equilíbrio, segue uma distribuição geométrica com parâmetro dado pela raiz σ que precisa ser encontrada e depende da distribuição do tempo entre chegadas.

Portanto, conhecendo a distribuição do número de usuários no sistema podemos encontrar L, que é simplesmente a média de uma distribuição geométrica com parâmetro em função de σ, ou seja

$$L = \sum_{n=0}^{\infty} n\pi_n = \sum_{n=0}^{\infty} n(1-\sigma)\sigma^n = \frac{\sigma}{1-\sigma}.$$

Com relação ao número de usuários na fila, basta notar que se existem $n > 0$ usuários no sistema então existem $n-1$ na fila, uma vez que o sistema conta com um servidor. Temos então

$$L_q = \sum_{n=1}^{\infty} (n-1)\pi_n = \sum_{n=1}^{\infty} n\pi_n - \sum_{n=1}^{\infty} \pi_n = L - (1-\pi_0)$$

$$\implies L_q = L - \sigma = \frac{\sigma^2}{1-\sigma}.$$

Com relação ao tempo médio de espera na fila, seja T_q a variável aleatória que representa esta quantidade e seja $W_q(t)$ sua função de distribuição. Note que se o sistema está vazio quando um novo usuário chega, então o mesmo é servido imediatamente com tempo de espera zero. Nesse caso

$$W_q(0) = \mathbb{P}(T_q \leq 0) = \pi_0 = 1 - \sigma.$$

Se existem n usuários no sistema, todos eles serão atendidos, cada um com tempo de atendimento exponencialmente distribuído. Logo a soma destes tempo segue distribuição Erlang, e portanto

$$W_q(t) = \mathbb{P}(T_q \leq t)$$

$$= (1-\sigma) + \sum_{n=1}^{\infty} \mathbb{P}(\text{n atendimentos em tempo menor que t} \mid \text{n usuários no sistema})\pi_n$$

$$= (1-\sigma) + \sum_{n=1}^{\infty} (1-\sigma)\sigma^n \int_0^t \frac{\mu(\mu t)^{n-1}}{(n-1)!} e^{-\mu t} dt$$

$$= (1-\sigma) + \int_0^t \mu(1-\sigma)e^{-\mu t} \frac{\sigma}{e^{-\sigma\mu t}} dt \sum_{n=1}^{\infty} \frac{(\mu t\sigma)^{n-1}}{(n-1)!} e^{-\sigma\mu t}$$

$$= (1-\sigma) + \sigma \int_0^t \mu(1-\sigma)e^{-\mu(1-\sigma)t} dt.$$

Pelo resultado acima vemos que $W_q(t)$ representa uma mistura de uma variável aleatória discreta e uma contínua. Resolvendo a integral temos

$$W_q(t) = (1-\sigma) + \sigma(1 - e^{-\mu(1-\sigma)t}) = 1 - \sigma e^{-\mu(1-\sigma)t}.$$

Com base na distribuição acima para o tempo de espera na fila, podemos encontrar o valor médio

$$W_q = \frac{1}{\mu} \frac{\sigma}{1-\sigma}.$$

Note que a expressão acima é composta pelo tempo médio de serviço multiplicado pelo número médio de usuários no sistema, ou seja, cada usuário que já estava no sistema contribui em média com o mesmo tempo na espera do usuário que acabou de chegar, inclusive o usuário que já estava sendo atendido. Isso ocorre devido à propriedade da falta de memória da distribuição exponencial que governa os tempos de serviço desse sistema.

Por fim, para encontrar o tempo médio de permanência no sistema basta usar a relação $W = W_q + \frac{1}{\mu}$. Abaixo temos o resumo de todos os resultados encontrados.

L^A	W	L_q^A	W_q
$\frac{\sigma}{1-\sigma}$	$\frac{1}{\mu(1-\sigma)}$	$\frac{\sigma^2}{1-\sigma}$	$\frac{\sigma}{\mu(1-\sigma)}$

Devido à forma com que o sistema foi descrito, usamos sobrescrito A para indicar que estas medidas de eficácia são referentes ao momento da chegada de um novo usuário no sistema, e não são válidas para qualquer instante de tempo, como era nos casos em que o tempo entre chegadas é exponencialmente distribuído.

Observe que, apesar das fórmulas terem sido encontradas, ainda é um desafio encontrar σ. Para ilustrar, considere que o tempo entre chegadas é exponencialmente distribuído com média $1/\lambda$. O primeiro passo é encontrar a transformada de Laplace desta distribuição,

$$\tilde{A}(s) = \int_0^\infty e^{-st} f(t)dt = \int_0^\infty \lambda e^{-t(\lambda+s)}dt = \frac{\lambda}{\lambda+s}.$$

Avaliando $\tilde{A}$ em $(\mu - \mu\sigma)$ para solucionar $\sigma = \tilde{A}(\mu - \mu\sigma)$ temos

$$\sigma = \frac{\lambda}{\lambda + \mu - \mu\sigma} \implies \sigma\lambda + \sigma\mu - \mu\sigma^2 - \lambda = 0,$$

que pode ser reescrita como

$$(\sigma - 1)(\lambda - \mu\sigma) = 0,$$

cujas soluções são obtidas em $\sigma = 1$ ou quando $\lambda - \mu\sigma = 0$, ou seja, quando

$$\sigma = \frac{\lambda}{\mu} = \rho.$$

Como esperado, o resultado acima encontrado para σ quando inserido nas fórmulas apresentadas leva aos resultados conhecidos para sistemas M/M/1. Este procedimento de encontrar a transformada de Laplace da distribuição dos tempos entre chegadas para se obter o valor de σ deve ser sempre realizado para diferentes distribuições, o que torna mais complicada a obtenção das medidas de interesse e motiva ainda mais o uso de simulações como alternativa para se obter estes resultados.

5.1 Simulações

A fim de avaliar o algoritmo que realiza as simulações considerando sistemas G/M/1 a seguir veremos exemplos assumindo diferentes distribuições para os tempos entre chegadas. O procedimento consistirá em realizar as simulações utilizando as distribuições escolhidas, coletar as estimativas geradas pelo algoritmo e compará-las aos resultados exatos calculados por meio das fórmulas apresentadas.

5.1.1 Distribuição Gama para o tempo entre chegadas

No Capítulo 3 foi apresentado um exemplo que considerava o tempo entre chegadas com distribuição $Gama(2,3)$ e o tempo de serviço com distribuição $Exp(4)$. Considerando que este sistema possui apenas um servidor, agora iremos calcular os resultados exatos para as medidas de interesse e compará-los aos resultados das simulações realizadas naquele exemplo.

Lembrando que vimos que estes resultados não eram condizentes com a Lei de Little devido ao fato das estimativas serem realizadas no momento da chegada de um novo usuário, e a lei considera o sistema em um momento qualquer, mas agora os resultados derivados também são referentes ao momento de chegada. Portanto, espera-se que os resultados simulados sejam condizentes com os valores exatos que serão calculados a seguir.

Se o tempo entre chegadas tem distribuição $Gama(\alpha,\beta)$ com densidade

$$f(t) = \frac{\beta^\alpha}{\Gamma(\alpha)} t^{\alpha-1} e^{-\beta t},$$

então a transformada de Laplace é dada por

$$\tilde{A}(s) = \int_0^\infty e^{-st} f(t) dt = \int_0^\infty e^{-st} \frac{\beta^\alpha}{\Gamma(\alpha)} t^{\alpha-1} e^{-\beta t} dt = \left(\frac{\beta}{\beta+s}\right)^\alpha.$$

Avaliando $\tilde{A}$ em $(\mu - \mu\sigma)$ para solucionar $\sigma = \tilde{A}(\mu - \mu\sigma)$ temos

$$\sigma = \left(\frac{\beta}{\beta + \mu - \mu\sigma}\right)^\alpha,$$

cuja solução pode ser encontrada com poucas iterações utilizando o método numérico de substituições sucessivas. Para isso faça

$$\sigma_{k+1} = \left(\frac{\beta}{\beta + \mu - \mu\sigma_k}\right)^\alpha,$$

onde σ_0 é um chute inicial.

Em nosso exemplo $\alpha = 2$, $\beta = 3$ e $\mu = 4$, e considerando um chute inicial $\sigma_0 = 0.1$ temos

$$\sigma_1 = \frac{3}{3+4-4\sigma_0} = 0.2066116,$$

$$\sigma_2 = \frac{3}{3+4-4\sigma_1} = 0.2361414,$$

$$\vdots$$

$$\sigma_n = 0.25$$

para n grande o suficiente. Finalmente, podemos inserir o valor de σ nas fórmulas encontradas para calcular os seguintes valores exatos para as medidas de interesse

Tabela 7: Medidas de interesse

Lq	Wq	W	L
0.083	0.083	0.333	0.333

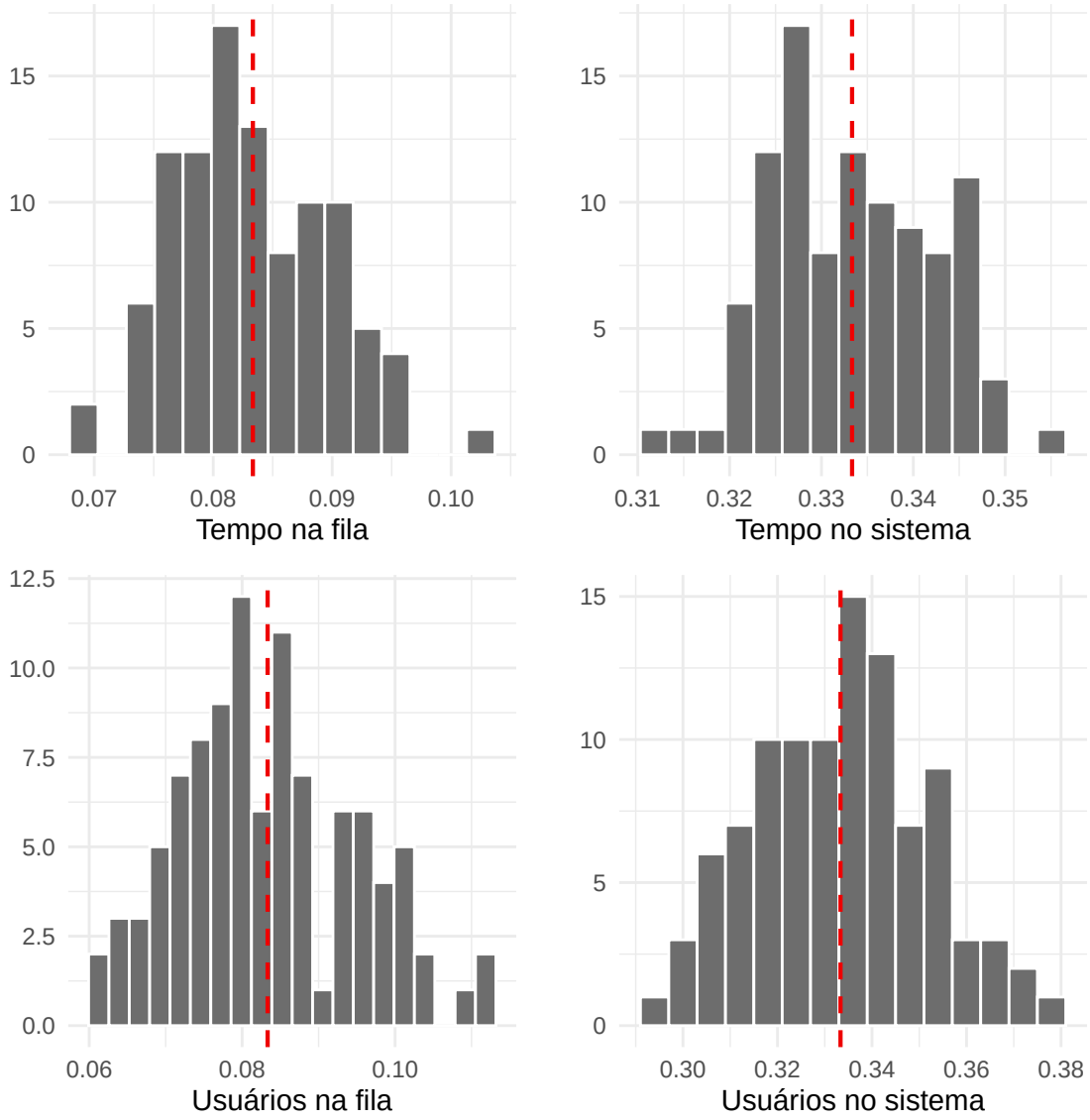

Figura 13: Estimativas para as medidas de interesse geradas pelo algoritmo.

Conforme pode ser analisado nos gráficos da Figura 13, os resultados simulados oscilam em torno dos valores reais. O teste de hipóteses abaixo visa avaliar se a média do estimador para o total de usuários no sistema, no momento da chegada de um novo usuário, é igual a 1/3, que é o valor exato calculado anteriormente.

$H_0 : W_q = 1/3$
$H_1 : W_q \neq 1/3$

Tabela 8: Teste de hipóteses para o número de usuários no sistema

Test statistic	df	P value	Alternative hypothesis	mean of Wq
0.033	99	0.974	two.sided	0.333

Conforme aponta o teste, não existem evidências para afirmar que a média do estimador, utilizando as simulações do algoritmo, é diferente do valor real. Em outras palavras, os resultados das simulações são, em média, iguais aos resultados exatos quando se considera o sistema no momento da chegada de um novo usuário.

5.1.2 Distribuição Weibull para o tempo entre chegadas

Considere agora um sistema em que os tempos entre chegadas seguem um distribuição Weibull com parâmetros α e β, temos

$$f(t) = \alpha\beta t^{\alpha-1} e^{-\beta t^{\alpha}},$$

$t \geq 0$, $0 < \alpha \leq 1$. Assim como no exemplo anterior, para obter as medidas de interesse é preciso primeiro econtrar a transformada de Laplace

$$\tilde{A}(s) = \int_0^{\infty} e^{-st} f(t) dt,$$

e solucionar a equação $\sigma = \tilde{A}(\mu - \mu\sigma)$. Veremos abaixo a solução para este problema, mas o leitor interessado em uma solução mais detalhada pode consultar [9]. Neste caso a transformada de Laplace é dada por

$$\tilde{A}(s) = \int_0^{\infty} e^{-st} \alpha\beta t^{\alpha-1} e^{-\beta t^{\alpha}} dt,$$

onde $e^{-\beta t^{\alpha}} = \sum_{k=0}^{\infty} \frac{(-\beta t^{\alpha})^k}{k!} = \sum_{k=0}^{\infty} \frac{(-\beta)^k t^{\alpha k}}{k!}$. Logo pode-se reescrever

$$\tilde{A}(s) = \alpha\beta \sum_{k=0}^{\infty} \frac{(-\beta)^k}{k!} \int_0^{\infty} e^{-st} t^{\alpha(k+1)-1} dt = \alpha\beta \sum_{k=0}^{\infty} \frac{(-\beta)^k}{k!} \frac{\Gamma(\alpha(k+1))}{s^{\alpha(k+1)}}.$$

Fazendo $n = k + 1$, finalmente obtemos

$$\tilde{A}(s) = \sum_{n=1}^{\infty} \frac{\alpha(-1)^{n-1}}{(n-1)!}\beta^n \frac{\Gamma(\alpha n)}{s^{\alpha n}}.$$

De posse do resultado acima, novamente podemos solucionar $\sigma = \tilde{A}(\mu - \mu\sigma)$ por meio do método numérico de substituições sucessivas, isto é

$$\sigma_{k+1} = \sum_{n=1}^{\infty} \frac{\alpha(-1)^{n-1}}{(n-1)!}\beta^n \frac{\Gamma(\alpha n)}{(\mu - \mu\sigma_k)^{\alpha n}},$$

$k = 0, 1, 2...$, fazendo $\sigma_0 = 0.5$, $\alpha = 0.5$, $\beta = 2$ e $\mu = 4$, veremos que $\sigma_k \to 0.767$ para k grande o suficiente.

Portanto, dadas essas configurações temos um sistema com média de tempo entre chegadas igual a 0.5, que é simplesmente a esperança da variável aleatória Weibull com os parâmetros que escolhemos. Além disso, sabemos que esse sistema conta com apenas um servidor e o tempo médio de serviço é igual 0.25, ou seja, o servidor tem capacidade de atender em média $\mu = 4$ usuários por unidade de tempo. Agora que conhecemos a raiz $\sigma = 0.767$ podemos encontrar as seguintes medidas de interesse:

Tabela 9: Medidas de interesse

Lq	Wq	W	L
2.525	0.823	1.073	3.292

Para avaliar o algoritmo, na Figura 14 são apresentados os resultados de 100 réplicas de simulações, cada uma considerando a chegada de 50000 usuários no sistema. Lembrando que o tamanho da fila e o número de usuários no sistema são referentes ao momento de uma nova chegada ao sistema. Vemos que as estimativas geradas pelo algoritmo se distribuem em torno do valor real, como era de se esperar.

O teste de hipóteses a seguir confirma que não existem evidências para se afirmar que a média das estimativas geradas pelo algoritmo é diferente da média real de usuários no sistema. Em outras palvras, podemos aceitar que as estimativas geradas pelo algoritmo são, em média, iguais aos valores médios reais. O mesmo procedimento poderia ser aplicado para as demais medidas de interesse e seria obtida a mesma conclusão.

$H_0 : L = 3.292$
$H_1 : L \neq 3.292$

Tabela 10: Teste de hipóteses para o número de usuários no sistema

Test statistic	df	P value	Alternative hypothesis	mean of L
0.925	99	0.357	two.sided	3.299

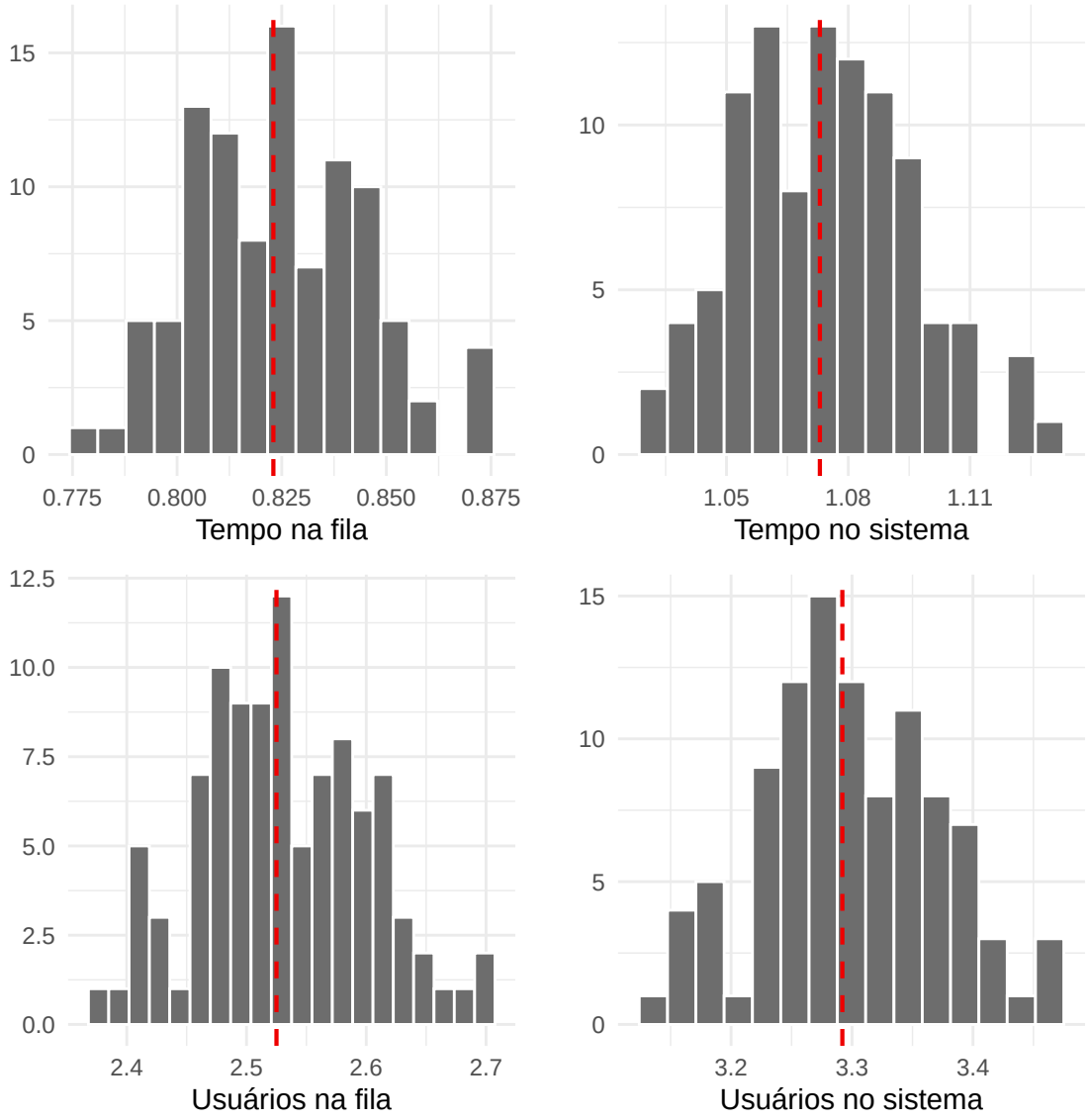

Figura 14: Estimativas para as medidas de interesse geradas pelo algoritmo.

6 Comparação com o queuecomputer

Foi mencionada anteriormente a existência de um pacote em R, o *queuecomputer,* que realiza simulações da mesma natureza do algoritmo aqui apresentado. Agora será realizada uma breve comparação entre as medidas geradas pelo algoritmo e as medidas geradas pelo pacote.

Esta comparação não tem como objetivo eleger o melhor método, uma vez que o algoritmo aqui descrito possui apenas uma função didática, com o objetivo de possibilitar que o leitor entenda como podem ser realizadas simulações desta natureza. Logo, objetivo desta comparação é apenas validar que os resultados vistos ao longo do texto são iguais aos que poderiam ser obtidos por meio do pacote.

Considere um sistema com 6 servidores, tempo entre chegadas $X \sim Exp(3)$ e tempo de serviço $Y \sim Ln(0,1)$. Foi realizada uma simulação considerando a chegada de 5000 usuários neste sistema, com base na mesma amostra gerada para os tempos entre chegada e de serviço, os resultados foram computados pelo algoritmo e pelo *queuecomputer*. As estimativas obtidas pelos dois métodos podem ser comparadas na tabela abaixo.

Tabela 11: Comparação entre os resultados do pacote e do algoritmo

Medida	Estimativa do algoritmo	Estimativa do queuecomputer
Média de usuários no sistema	7.013	7.199
Média de usuários na fila	2.093	2.259
Tempo médio no sistema	2.424	2.424
Tempo médio na fila	0.759	0.759

Note que as estimativas referentes ao tempo de espera no sistema e ao tempo de espera na fila são exatamente as mesmas. Contudo, para o número de usuários no sistema e na fila as estimativas, apesar de serem próximas, são diferentes.

Considere agora um novo sistema com 2 servidores, tempo entre chegadas $X \sim \text{Weibull}(2,3)$ e tempo de serviço $Y \sim \text{Gama}(12,4)$. Novamente a simulação será realizada considerando a chegada de 5000 usuários neste sistema.

Tabela 12: Comparação entre os resultados do pacote e do algoritmo

Medida	Estimativa do algoritmo	Estimativa do queuecomputer
Média de usuários no sistema	0.780	1.164
Média de usuários na fila	0.013	0.044
Tempo médio no sistema	3.112	3.112
Tempo médio na fila	0.118	0.118

Outra vez podemos afirmar que as estimativas para W e W_q são exatamente as mesmas, mas desta vez as estimativas para L e L_q são muito diferentes. Isto ocorre pois, como vimos, o

algoritmo apresentado neste trabalho restringe as medidas ao momento da chegada de um novo usuário, e as medidas apresentadas pelo *queuecomputer* valem para qualquer momento.

Isso pode ser comprovado por meio das relações estabelecidas pela Lei de Little: $L = \lambda W$ e $L_q = \lambda W_q$. A simulação do algoritmo no segundo caso resultou em $\hat{\lambda} = 0.37404$, e considerando as estimativas para W e W_q, temos $\hat{L} = 1.164045$ e $L_q = 0.04408963$, resultados exatamente iguais aos estimados pelo pacote. Ou seja, conhecendo todas as estimativas geradas pelo algoritmo e as relações existentes entre essas medidas podemos mensurar o comportamento em dois cenários: um que considera o sistema em qualquer instante, e outro que considera o sistema apenas quando um usuário chega.

Portando pode-se chegar à conclusão de que os resultados apresentados na Tabela 11 são próximos pois, apesar de serem obtidos de forma diferente, vale a propriedade PASTA pois se trata de um sistema com tempo entre chegadas exponencialmete distribuído. Ou seja, enquanto o algoritmo descrito neste trabalho computa o número de usuário na fila e no sistema a cada momento de chegada de um novo usuário, o *queuecomputer* aparentemente computa apenas os tempos de espera no sistema e na fila e obtém as demais medidas através das relações conhecidas por meio da Lei de Little. Por outro lado no exemplo em que os tempos entre chegadas não são exponencialmente distribuídos, a propriedade não vale. Isso significa que ao chegar em um sistema um usuário interno observa, em média, um cenário diferente se comparado a um observador externo que acompanha o sistema a todo instante. Por esse motivo, os resultados apresentados na Tabela 12 são diferentes com relação ao número de usuários no sistema e na fila, uma vez que se referem a formas distintas de se observar um sistema em que a propriedade PASTA não é válida.

Apesar do algoritmo possibilitar que sejam encontradas as mesmas estimativas do pacote, vale destacar que a eficiência do *queuecomputer* é muito superior. Ebert *et al.* [3] mostraram que o pacote que desenvolveram em liguagem R é computacionalmente mais eficiente se comparado à outros concorrentes que realizam simulações de natureza similar.

De fato, a superioridade do *queuecomputer* é notória, o pacote apresentou tempos de processamento muito inferiores se comparados aos levados pelo algoritmo aqui apresentado. Logo, recomenda-se fortemente que este pacote seja utilizado em análises de sistemas de filas utilizando a linguagem R, e que o algoritmo aqui descrito seja considerado apenas de forma didática.

Além disso, espera-se que, diante dos resultados antes apresentados e das dificuldades inerentes ao processo de descoberta dos resultados exatos, o leitor tenha se convencido da importância do uso das simulações, não como uma forma de substituir os resultados teóricos mas como uma alternativa mais simples e que pode trazer excelentes resultados.

7 Considerações Finais

Como foi explorado ao longo do texto, os resultados gerados pelo algoritmo são muito satisfatórios e possibilitam estimativas para as medidas de interesse bem próximas aos valores reais. Encontrar soluções exatas é um grande desafio, uma vez que cada configuração para o sistema necessita de uma abordagem diferente para se encontrar as soluções. E ainda, como vimos em alguns exemplos deste trabalho, muitas soluções necessitam da implementação de métodos numéricos que levam a resultados diferentes à depender dos parâmetros das distribuições que regem o sistema. As simulações, por outro lado, requerem apenas uma amostra gerada aleatoriamente para os tempos entre chegadas e tempos de serviço. Feito isso, o procedimento do algoritmo é o mesmo independente das distribuições utilizadas.

Por esse motivo, esta pesquisa teve como objetivo incentivar o uso de simulações em sistemas de filas e ilustrar em detalhes o funcionamento de um algoritmo que pode ser utilizado para realizar essas simulações. O tempo de processamento poderia ser citado como uma desvantagem em relação aos resultados exatos, uma vez que é necessário um grande número de usuários na simulação para se obter resultados próximos aos verdadeiros. Contudo, já existem soluções computacionais extremamente eficientes, como é o caso do pacote *queuecomputer* [3], que realiza simulações da mesma natureza do algoritmo aqui descrito em tempo extremamente rápido.

Além desta vantagem, as simulações podem também ser utilizadas como forma de validação de resultados matemáticos encontrados para diferentes sistemas de filas, a fim de avaliar se tais resultados fazem sentido e estão corretos. Logo, simulações também podem contribuir com pesquisas de cunho mais teórico, auxiliando na checagem dos resultados matemáticos obtidos pelos pesquisadores.

Também podemos destacar como vantagem das simulações a ausência das suposições a respeito das distribuições. Podemos, por exemplo, simular sistemas com alguma dependência determinada entre os tempos de chegadas ou serviço e estimar as medidas de interesse, situação que seria de enorme dificuldade quando se trata de resultados exatos.

Finalmente, conclui-se este texto reforçando mais uma vez a importância das simulações de Monte Carlo, não só para o estudo de sistemas de filas como foi feito neste trabalho, mas também para diversos outros casos que podem se valer desta ferramenta poderosíssima. Sua grande importância se dá por sua simplicidade, basta entender como se pode simular um evento de interesse e tudo que se deseja saber sobre este evento pode ser estimado diretamente.

Referências

[1] SHORTLE, J. F.; THOMPSON, J. M.; GROSS, D.; HARRIS, C. M. Fundamentals of queueing theory, vol. 399. John Wiley and Sons, 2018.

[2] KENDALL, D. G. Stochastic processes occurring in the theory of queues and their analysis by the method of the imbedded Markov chain. The Annals of Mathematical Statistics, p. 338-354, 1953.

[3] EBERT, A.; WU, P.; MENGERSEN, K.; RUGGERI, F. Computationally Efficient Simulation of Queues: The R Package queuecomputer. Journal of Statistical Software, v.95, n.5, p. 1-29, 2020.

[4] STIDHAM, S.; PRABHU, N. U. Optimal control of queueing systems. Mathematical Methods in Queueing Theory: Proceedings of a Conference at Western Michigan University, 1973. Springer-Verlag Berlin Heidelberg, p. 263-294, 1974.

[5] WOLFF, R. W. Little's law and related results. Wiley encyclopedia of operations research and management science, New York, v. 4, 2011.

[6] MARCHAL, W. G. Some simpler bounds on the mean queuing time. Operations Research, v. 26, n. 6, p. 1083-1088, 1978.

[7] ALLEN, A. O. 1990. Probability, Statistics, and Queueing Theory with Computer Science Applications, 2 ed. Academic Press, New York, 1990.

[8] GALLAGER, R. G. Discrete stochastic processes. Springer, 1995.

[9] STRELKOVSKAYA, I. V.; GRYGORYEVA, T. I.; SOLOVSKAYA, I. N. Self-similar traffic in G/M/1 queue defined by the Weibull distribution. Radioelectronics and Communications Systems, v. 61, n. 3, p. 128-134, 2018.

[10] CHAVES, C.; GOSAVI, A. On general multi-server queues with non-poisson arrivals and medium traffic: A new approximation and a COVID-19 ventilator case study. Operations Research, p. 5205–5229, 2022.

Apêndice

Função desenvolvida em linguagem R que implementa o algoritmo apresentado no Capítulo 2 e foi utilizada em todas as simulações mostradas neste trabalho.

```
simula_fila = function(C, X, Y){

  ############ PARAMETROS DA SIMULACAO ############

  #tempo entre chegadas, usado para gerar o momento das chegadas
  Tempos_entre = X

  #tempo de servicos, respeitando este tempo a fila sera formada
  servico =  Y

  #numero de servidores realizando atendimento
  servidores = C


  ############ CRIA OS "AMBIENTES" DA FILA E OS VETORES QUE ARMANEZARAO OS RESULTADOS ############

  #Cria o server, inicialmente desocupado
  #Este ojeto sera atualizado conforme a simulacao avanca
  Server = data.frame(Servidor = seq(1:servidores),
                      Ocupado = rep(0, servidores),
                      Hora_entrada= rep(0, servidores),
                      Hora_saida= rep(0, servidores))


  #Cria uma fila inicialmente vazia
  #tambem sera atualizada conforme a simulacao avanca
  fila = data.frame(NULL)


  #Vetores que irao guardar informacoes durante o processo

  #Vacancia: Numero de servers disponiveis, quando o usuário chega
  Vacancia = NULL

  #Tam_fila: Tamanho da fila a cada chegada de usuários
  Tam_fila = NULL

  #Espera: Tempo de espera dos usuários na fila
  Espera_fila = NULL

  #Hr_ini_atd: Hora que o usuário sera atendido
  Hr_ini_atd = NULL

  #Numero de usuários no sistema (Sem contar com o usuário que acabou de chegar)
  N_clientes_sistema = NULL

  #A soma do tempo entre as chegadas + hora atual resulta no momento de chegada de cada usuário
  #Neste dataframe esta contido o momento que o usuário ira chegar e seu tempo de atendimento
  #Para um dia simulado
  Clientes = data.frame(Horas_chegadas = cumsum(Tempos_entre) , Tempos_servico = servico)


  ############ SIMULACAO ############

  #iterando sobre todos os usuários,
  #de posse de seus horarios de chegada e duracao do antendimento,

  for (i in 1:nrow(Clientes)){

    #Verifica se o momento de finalizacao dos atendentes e menor que o tempo de chegada,
    #se sim eles estao livres
```

```
#o momento de referencia (atual) e o momento que chega o usuário
Server$Ocupado = ifelse(Server$Hora_saida < Clientes$Horas_chegadas[i], 0, 1)


###Calcula a vacancia: Total de servidores - total de ocupados####
v = servidores - sum(Server$Ocupado)

#o valor observado da vacancia duarante a chegada do i-esimo usuário e armazenado
Vacancia = c(Vacancia, v)


############ Cenario em que o usuário e imediatamente atendido ############

#Se houver pelo menos um servidor livre entao
if(sum(Server$Ocupado)< servidores){

  #Pega o primeiro atendente livre
  #Assume-se que todos atendentes realizam o servico no mesmo tempo,
  #logo a escolha do atendente nao importa
  Atendente = min(which(Server$Ocupado == 0))

  #Entao este atendente recebe o momento que estara ocupado e o momento que estara livre
  Server[Atendente, 2:4 ] = c(1, Clientes$Horas_chegadas[i],
                              Clientes$Horas_chegadas[i]+Clientes$Tempos_servico[i])
  #Se a um servidor livre o momento de entrada e o momento que o usuário chega
  #E o momento de finalizacao e o momento de chegada + tempo do atendimentp


  #Se tem alguem desocupado entao a fila necessariamente esta completamente vazia
  #portando o objeto da fila e zerado
  fila = data.frame(NULL)


  #Se houver um servidor livre o tempo de espera na fila e zero e o tamanho da fila tambem
  #Esses valores sao passados nos objetos que armazenam os indicadores
  Tam_fila = c(Tam_fila, 0)
  Espera_fila = c(Espera_fila, 0)

  #E por fim, o momento que o usuário sera atendido e a proprio momento de chegada
  Hr_ini_atd = c(Hr_ini_atd, Clientes$Horas_chegadas[i])

  #Numero de usuários no sistema (sem contar o usuário que acabou de chegar)
  #como nao ha fila, e dado apenas pelo numero de servidores ocupados
  N_clientes_sistema = c(N_clientes_sistema, sum(Server$Ocupado) - 1)

}else{

  ############ Cenario em que o usuário vai para a fila ############

  #Se nao tem servidor livre, entao vai para a fila
  Chegada_fila = Clientes$Horas_chegadas[i]

  #guarda para comparar se os outros ja sairam
  #ou seja atualiza os objetos caso ha tenha se passado o tempo necessario
  Chegada_fila_do_cliente_atual = Chegada_fila

  #O tempo de espera na fila e a diferenca entre o momento que ira finalizar
  #o primeiro atendimento e o momento de chegada,
  #para isso, analisa qual servidor ficara livre primeiro
  Espera = min(Server$Hora_saida) - Chegada_fila

  #Guarda o tempo de espera deste usuário
  Espera_fila = c(Espera_fila,Espera)

  #Saida fila, ocorre quando termina o primeiro atendimento
  Saida = min(Server$Hora_saida )
```

```
    #Inclui os dados deste usuário na fila
    fila_aux = data.frame(Chegada_fila,Saida ,Espera)

    fila = dplyr::bind_rows(fila, fila_aux)


    #O usuário que acaba de entrar na fila sera o primeiro a ser atendido
    #pois sempre o anterior ja estara considerado no server

    #Pega o atendente que ficar livre primeiro
    Atendente = which.min(Server$Hora_saida )

    #Guarda o momento que o usuário sera atendido
    Hr_ini_atd = c(Hr_ini_atd,min(Server$Hora_saida))

    #A chegada e o momento que o atendente ficou livre
    #o momento de entrada fica sendo o momento de saida do anterior
    #e o momento de saida  e o momento de saida do anterior + o tempo de servico do usuário

    Server[Atendente, 2:4 ] = c(1, min(Server$Hora_saida),
                                min(Server$Hora_saida)+Clientes$Tempos_servico[i] )


    #Caso tenha outros na fila, checa se ja sairam no momento em que o usuário chegou
    # feito para medir o tamanho da fila
    fila= fila%>%
      filter(Saida > Chegada_fila_do_cliente_atual)


    #Numero de usuários no sistema (sem contar o usuário que acabou de chegar)
    #O total e dado pela sona de usuários na fila e o total de servidores ocupados
    N_clientes_sistema = c(N_clientes_sistema, sum(Server$Ocupado) + nrow(fila) -1 )


    Tam_fila = c(Tam_fila, nrow(fila) - 1)
    # -1 pois ele ja esta sendo contado na fila

  }

}


############ ATUALIZACAO DAS INFORMACOES DOS CLIENTES APOS A SIMULACAO  ############

#Atualiza a base de usuários com as informacoes do atendimento
Clientes$Vacancia = Vacancia
Clientes$Tam_fila = Tam_fila
Clientes$Espera_fila = Espera_fila
Clientes$Hr_ini_atd= Hr_ini_atd
Clientes$N_clientes_sistema  = N_clientes_sistema


Clientes = Clientes %>%
  mutate(N_clientes_Server = servidores - Vacancia,
         Tempo_sistema = Espera_fila + Tempos_servico)


# Resultados consolidados
resultados = Clientes%>%
  dplyr::summarise(L = mean(N_clientes_sistema),
                   Lq = mean(Tam_fila),
                   r =  mean(N_clientes_Server),
                   S = mean(Tempos_servico),
                   P = sum(Vacancia>0)/length(Vacancia),
                   W = mean(Tempo_sistema),
                   Wq =  mean(Espera_fila)
            )
```

```
  #Tempo gasto na fila (dado que ha fila)
  #considera apenas os usuários que tiveram que esperar
  fila_quando_ha = Clientes%>%
    dplyr::filter(Espera_fila > 0)%>%
    dplyr::summarise(Tempo_fila = mean(Espera_fila)) %>%
    purrr::pluck("Tempo_fila")

  resultados$Wq_0 = fila_quando_ha

  resultados = resultados %>%
    select( L, Lq, W, Wq, Wq_0, S, P, r)

  #Taxas do sistema

  lambda = 1/mean(Tempos_entre)
  mu = 1/mean(servico)
  rho = lambda/(mu*C)

  resultados$lambda = lambda
  resultados$mu = mu
  resultados$rho = rho
  resultados$C = C


  return(resultados)

}
```