National and Kapodistrian University
Of Athens
Department Of Mathematics

Ticket Queues: Modeling, Analysis and Strategic Customer Behavior

Thesis submitted to the National and Kapodistrian University of Athens, Department of Mathematics in partial fulfillment of the requirements for the MSc degree in Statistics and Operations Research

Maria Palamida

Advisor: Antonios Economou

Athens, February 2020

The present MSc thesis was completed for the fulfilment of the requirements for the MSc degree in Statistics and Operations Research at the Department of Mathematics, National and Kapodistrian University of Athens, Greece.

Approved on .... by the Evaluation Committee composed of:

Name-Surname	**Academic rank**	**Signature**
Apostolos Burnetas	Professor	....
Antonios Economou (advisor)	Professor	....
Samis Trevezas	Lecturer	....

Acknowledgments

First of all, I would like to express my special thanks to professor Antonios Economou, my advisor, for his help, advice and understanding during the creation of my thesis. I would also like to express my appreciation to professor Apostolos Burnetas and lecturer Samis Trevezas for their participation in the Evaluation Committee. I am thankful too to my brother Fotis for lending me his personal laptop to study when I needed to. Lastly, I thank the rest of my family and my close friends who were really supportive the whole time of my work.

Contents

Chapter 1

Introduction

In everyday life, we often encounter the question of entering or not a queue. In order to handle our time in the best possible way, we estimate quickly the expected time of waiting in a queue. In ordinary queues, here referred to as *physical queues*, we make the estimation according to the number of people waiting in front of us. Sometimes, seeing all the people who have joined the queue is not possible, making the actual queue "invisible". *Ticket Queues* are one of the most interesting examples of this type of queues.

Ticket Queues are systems that, upon arrival, provide customers with a ticket which has a priority number printed on it. The number of the customer who is in service is showed on a display panel. The difference between the numbers of the arriving customer and the customer in service, referred to as the *ticket position*, is only an upper bound of the actual queue length, because some of them may have balked or reneged. *Balking* means that a customer decides not to join the queue after s/he sees the ticket position while *reneging* is when a customer abandons the system after joining the queue, mostly because of impatience. An analysis of ticket queues with these two kinds of customer abandonments before service has been done successfully by Xu, Gao and Ou (2007), and Ding, Ou and Ang (2015) respectively.

Additionally, there are some other aspects which are really fascinating to explore while dealing with Ticket Queues; customers may follow a specific strategy when they face a Ticket Queue or they might just enter the queue whatever the ticket position is and obtain service. This diversity of actions leads us to research cases where we have different types of customers, so that the population may not be always homogeneous, and other occasions, where we meet strategic customers that follow a *threshold strategy*. A truly interesting case of non-homogeneous population in Ticket Queues has been studied by Hanukov, Anily and Yechiali (2019). Threshold strategies referring to when a strategic customer decides to enter the queue or not have been studied in the work of Kerner, Sherzer and Yanco (2017).

Generally, this kind of systems are widely used nowadays in a large number of various services. From their simplest forms, such as the *take-a-number* systems, frequently seen in supermarkets and banks, to more complicated forms, like tickets with letter and number combinations which indicate the service type and the service order, Ticket Queues have many advantages. They surely offer better waiting experience in many ways. One of them is that the first-in, first-out (FIFO) service rule is not violated; everyone takes his/her priority number and is served according to the arriving time. Moreover, the customers can make more productive use of their waiting time, because they can leave the queue temporally in order to grab a coffee or run several errands, and then return back to the waiting room. As a result, Ticket Queues promote customer equality.

On the other hand, there's one basic drawback; usually, customers are naive and impatient, so they overestimate their waiting time and tend to abandon the queue more frequently than in a physical queue. So, Ticket Queues often lead to a much higher balking rate compared to physical queues. This is a loss from both the customer's and business perspective as the system has a lot more capacity to process services that are not actually accomplished.

The basic disadvantage highlighted above leads to studies which aim to improve the operation of Ticket Queues. For example, it has been noted that if we offer customers more accurate information on their waiting time, the balking rate can be reduced. In general, we are interested as well in topics that can improve not only the Ticket Queues but also the strategic customer behavior in order to achieve an even better waiting experience.

Throughout this work, we offer interesting insights on Ticket Queues. Firstly, we introduce some basic mathematical tools, definitions and solution techniques that are clearly useful all over this thesis; specifically of *Matrix-analytic and Game-theoretic* character. Subsequently, we analyze two different cases of Ticket Queues, namely *Ticket Queues with Balking Customers* and *Ticket Queues with Reneging Customers.* Their analysis is supplemented with proofs and conclusions relevant to our findings.

Chapter 2

Basic Theory

2.1 Matrix Analytic Methods

It is well known that, the most fundamental queueing models are described as birth-death processes. We remind that a birth-death process is a special case of a Continuous Time Markov Chain (CTMC), where the state transitions are distinguished in two forms: birth, in which the transition increases the number of the state by one, and death, where the transition decreases the number of the state by one. M/M/1, M/M/c and M/M/1/K queues are the most basic and significant examples modeled with birth-death process. The transition diagram of a birth-death process is described by arcs showing the birth and death rates, λ_i and μ_i respectively, and is presented bellow in Figure 2.1.

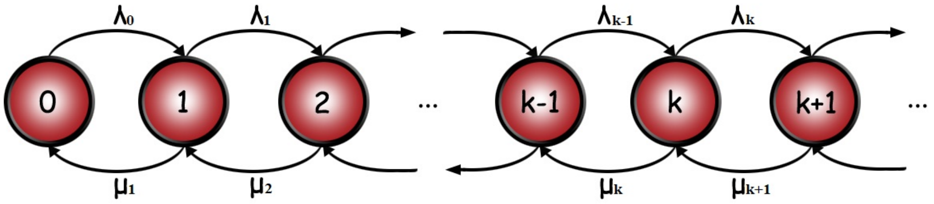


Figure 2.1 Diagram of a Birth-Death Process

However, the models that describe a Ticket Queue are much more complicated, yet with similar logic. Specifically we can model a Ticket Queue as a *quasi-birth-death process* (QBD) which actually describes a generalization of a birth-death process. The rest of this section is dedicated to QBDs and the necessary information about them.

2.1.1 Description of QBD processes

As a matter of fact, the majority of the Continuous Time Markov Chains that we encounter in introductory courses have a 1-dimensional state-space. However, there are exceptions where the states should be represented by a natural vector. For example, imagine a queueing problem where, apart of the number of customers in the system, we also need the information of whether the server is active or idle. So, if we assume that N and I are the random variables that describe the number of the customers in the queue and the condition of the server $(0 : idle, 1 : active)$ respectively, the state space will be represented by vectors (N, I).

Generally, in a vector state process, the states have the form (i, s). Commonly, these are pairs with $i = 0, 1, \ldots$ and $s = 0, 1, \ldots, m$. We shall say that i represents the *level* of such a process, while s the *phase*. As a consequence, all those states defined by $(i, 0), (i, 1), \ldots, (i, m)$ are the states at level i.

One of the most interesting facts is also that if a vector process has a transition matrix with a repetitive block structure, we can calculate the stationary probabilities using a *Matrix Geometric Method*. Note that, if the transition rate from state (i, j) to state $(i + k, j')$ is independent of the value of i for $i > i^*$, where i^* is a specific level, and $k = \pm 1$ as the transitions are only between adjacent levels, then, the process has repetitive transition structure. Such a repetitive structure on the transition rate matrix signifies that, eventually, all the matrix entries are repeated diagonally. This repetition is a substantial tool for obtaining the stationary distribution, and it is the key of the matrix-geometric method.

The process described above is actually a *quasi-birth-death process* (QBD). Recall again the birth-death process, and remember that it permits only adjacent state transitions. In correspondence, a QBD process allows state transitions only between adjacent levels or the same level. Actually, with the name *quasi*, we bring up the fact that the nearest neighbor transitions in a QBD are interpreted in terms of vectors of states while in a simple birth-death process everything is explained using scalar states.

In the following, we will show how the *Matrix Geometric Method* works through an example. Later, we will introduce the terms of *Homogeneous and Non-homogeneous* QBDs which are also remarkably useful in the rest of our work.

2.1.2 The Matrix Geometric Method

An example is the best way to see how the repetitive structure in the generator matrix helps us to determine a solution and how exactly we can work with the Matrix Geometric Method.

Let us consider a queueing system with Poisson arrivals where the rate is λ' if the system is empty, and λ otherwise. The customers, in order to obtain service, have to pass through two different exponential stages: the first one at rate μ_1 and the second at rate μ_2. Let every state be given by (i, s), $i \geq 0$, $s = 0, 1, 2$, where i is the number of the customers waiting in queue (so that the customer who receives service is not included) and s the current stage of the customer in service. By definition, we set $s = 0$ if the system is empty. The state transition diagram under these assumptions is shown in Figure 2.2.

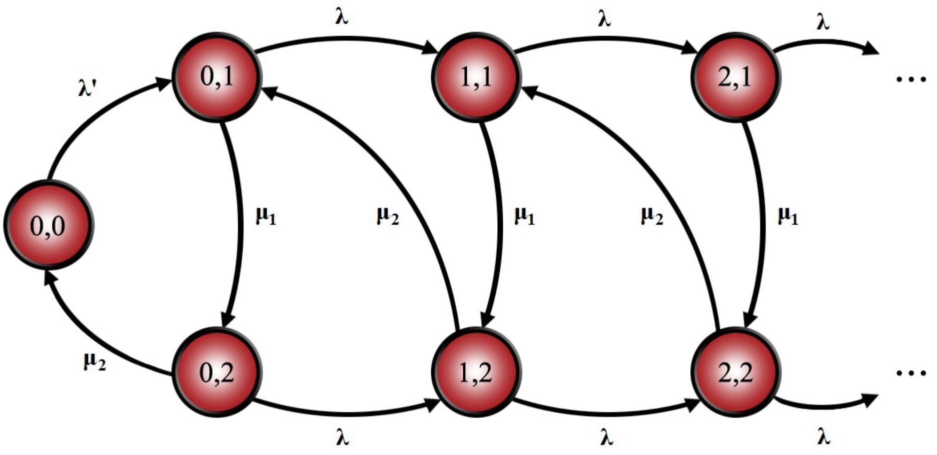


Figure 2.2 State Transition Diagram for a Vector Process

Actually, we have grouped states according to the number of customers in the queue. If we take a closer look on the diagram, we can comprehend that this model is a case of QBD process. We have two phases ($s = 1, 2$) and infinite levels, where the non-zero transition rates are only within the same level (from the first to second stage of the service) and between adjacent levels (arrivals or service completions). This fact is illustrated better in Figure 2.3.

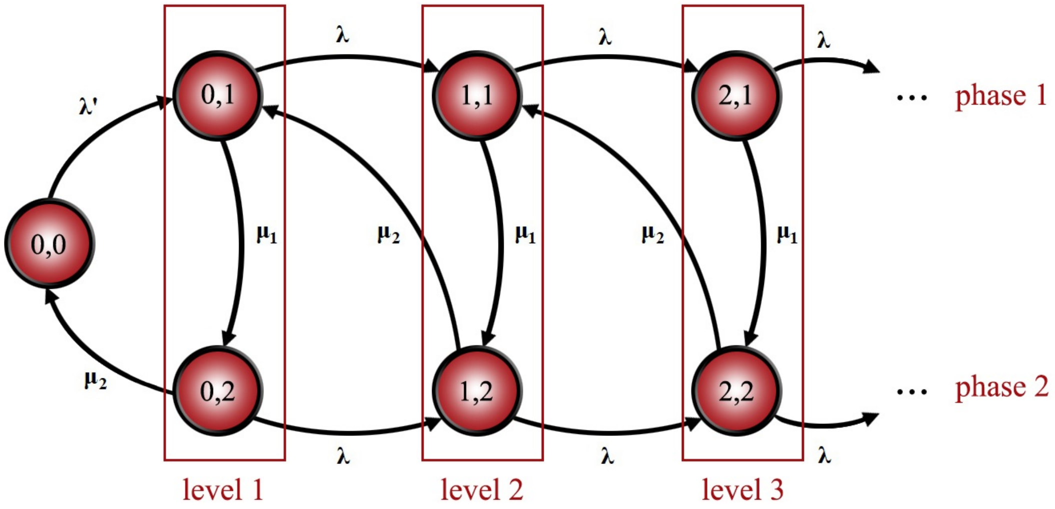


Figure 2.3 Levels and Phases in Diagram for Vector Process

At this point, we can obtain easily the generator matrix; we order states lexicographically, $(0,0),(0,1),(0,2),(1,1),(1,2),\ldots$, and let $\boldsymbol{\pi_{i,s}}$ be the stationary probability of state (i,s). So, the transition rate matrix $\mathbf{Q}$ is given by:

$$\boldsymbol{Q} = \begin{bmatrix} -\lambda' & \lambda' & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\ 0 & -\alpha_1 & \mu_1 & \lambda & 0 & 0 & 0 & 0 & 0 & \cdots \\ \mu_2 & 0 & -\alpha_2 & 0 & \lambda & 0 & 0 & 0 & 0 & \cdots \\ 0 & 0 & 0 & -\alpha_1 & \mu_1 & \lambda & 0 & 0 & 0 & \cdots \\ 0 & \mu_2 & 0 & 0 & -\alpha_2 & 0 & \lambda & 0 & 0 & \cdots \\ 0 & 0 & 0 & 0 & 0 & -\alpha_1 & \mu_1 & \lambda & 0 & \cdots \\ 0 & 0 & 0 & \mu_2 & 0 & 0 & -\alpha_2 & 0 & \lambda & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \tag{2.1}$$

where we define $\alpha_i = \lambda + \mu_i$, $i = 1, 2$

Let $\boldsymbol{\pi_i} = (\pi_{i,1}, \pi_{i,2})$ for $i \geq 1$, $\boldsymbol{\pi_0} = (\pi_{0,0}, \pi_{0,1}, \pi_{0,2})$ and $\boldsymbol{\pi} = (\boldsymbol{\pi_0}, \boldsymbol{\pi_1}, \boldsymbol{\pi_2}, \ldots)$. Furthermore, define the following matrices:

$$\boldsymbol{A_0} = \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix}, \quad \boldsymbol{A_0} = \begin{bmatrix} \alpha_1 & \mu_1 \\ 0 & \alpha_2 \end{bmatrix}, \quad \boldsymbol{A_2} = \begin{bmatrix} 0 & 0 \\ \mu_2 & 0 \end{bmatrix},$$

$$\boldsymbol{B_{1,0}} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \mu_2 & 0 \end{bmatrix}, \quad \boldsymbol{B_{0,1}} = \begin{bmatrix} 0 & 0 \\ \lambda & 0 \\ 0 & \lambda \end{bmatrix}$$

and finally,

$$\boldsymbol{B_{1,0}} = \begin{bmatrix} -\lambda' & \lambda' & 0 \\ 0 & -\alpha_1 & \mu_1 \\ \mu_2 & 0 & -\alpha_2 \end{bmatrix}$$

According to the definitions given above, we now can partition the generator matrix into blocks as follows:

$$\begin{aligned} \boldsymbol{Q} &= \left[\begin{array}{ccc|cc|cc|cc|c} -\lambda' & \lambda' & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\ 0 & -\alpha_1 & \mu_1 & \lambda & 0 & 0 & 0 & 0 & 0 & \cdots \\ \mu_2 & 0 & -\alpha_2 & 0 & \lambda & 0 & 0 & 0 & 0 & \cdots \\ \hline 0 & 0 & 0 & -\alpha_1 & \mu_1 & \lambda & 0 & 0 & 0 & \cdots \\ 0 & \mu_2 & 0 & 0 & -\alpha_2 & 0 & \lambda & 0 & 0 & \cdots \\ \hline 0 & 0 & 0 & 0 & 0 & -\alpha_1 & \mu_1 & \lambda & 0 & \cdots \\ 0 & 0 & 0 & \mu_2 & 0 & 0 & -\alpha_2 & 0 & \lambda & \cdots \\ \hline \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{array}\right] \\ &= \begin{bmatrix} \boldsymbol{B_{0,0}} & \boldsymbol{B_{0,1}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots \\ \boldsymbol{B_{1,0}} & \boldsymbol{A_1} & \boldsymbol{A_0} & \mathbf{0} & \mathbf{0} & \cdots \\ \mathbf{0} & \boldsymbol{A_2} & \boldsymbol{A_1} & \boldsymbol{A_0} & \mathbf{0} & \cdots \\ \mathbf{0} & \mathbf{0} & \boldsymbol{A_2} & \boldsymbol{A_1} & \boldsymbol{A_0} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \end{aligned} \tag{2.2}$$

Note that every entry **0** in (2.2) (and in all the following matrices) is a matrix of zeros of the appropriate dimension. We observe that **Q** is block-tridiagonal.

Let us call states $(0, 0)$, $(0, 1)$ and $(0, 2)$ the *boundary states*, and all the other states the *repeating states.* We already know how to solve the stationary probabilities for the scalar case, and here, the procedure is in the same manner. Firstly, as **Q** is a generator matrix, we have:

$$\boldsymbol{\pi Q} = \mathbf{0} \tag{2.3}$$

Hence, given in block matrix form, we obtain from (2.3):

$$\boldsymbol{\pi_{j-1} A_0} + \boldsymbol{\pi_j A_1} + \boldsymbol{\pi_{j+1} A_2} = \mathbf{0}, \quad j = 2, 3, \ldots \tag{2.4}$$

As in the scalar case, since state transitions are between nearest blocks, we discover with no surprise that the value of $\boldsymbol{\pi}_j$ is a function only of the transition rates between states with $j-1$ queued customers and states with j queued customers. Since these transition rates have no dependence on j, there must be a *constant matrix* $\mathbf{R}$ such that:

$$\boldsymbol{\pi_j} = \boldsymbol{\pi_{j-1} R} \quad j = 2, 3, \ldots \tag{2.5}$$

Hence, the values of $\boldsymbol{\pi_j}$, $j = 2, 3, \ldots$, have a *matrix geometric* form, which is calculated through *recursion*, and is:

$$\boldsymbol{\pi_j} = \boldsymbol{\pi_1 R^{j-1}} \quad j = 2, 3, \ldots \tag{2.6}$$

The basic idea is to substitute this *guess*, (2.6), into (2.4). Therefore, we get the equation:

$$\boldsymbol{\pi_1 R^{j-2} A_0} + \boldsymbol{\pi_1 R^{j-1} A_1} + \boldsymbol{\pi_1 R^j A_2} = \mathbf{0}, \quad j = 2, 3, \ldots \tag{2.7}$$

As a matter of fact, this equation must be true for all j, with $j = 2, 3, \ldots$. So, substituting $j = 2$ into (2.7) and simplifying yields gives:

$$\boldsymbol{A_0} + \boldsymbol{R A_1} + \boldsymbol{R^2 A_2} = \mathbf{0} \tag{2.8}$$

This equation is *quadratic* in the matrix $\mathbf{R}$, so typically, it can be solved numerically. However, not all the solutions of matrix $\mathbf{R}$ can satisfy the normalization condition, thus, it is important to determine an additional condition for choosing the appropriate solutions.

Recall the scalar case, where we had two possible solutions in the quadratic equation for ρ; one of these was $\rho = 1$ and it could not satisfy the normalization condition. Similarly, if the spectral radius of $\mathbf{R}$ is greater than or equal to 1, then the matrix $\mathbf{R}$ satisfies (2.8), but cannot be normalized. Thus, as in the scalar case, we pick the minimal matrix $\mathbf{R}$ which satisfies (2.8). Actually, the case where $\boldsymbol{\pi_1} \sum_{j=1}^{\infty} \boldsymbol{R^{j-1} e} < \infty$, with $\mathbf{e}$ defined as a suitably dimensioned column vector of 1s, is the one where the normalization constant is satisfied for the vector state process. Furthermore, the analogous criteria to $\rho < 1$ in the scalar case, is the fact that the *spectral radius* of $\mathbf{R}$

must be less than unity in our case. This is following from the fact that all *eigenvalues* of $\mathbf{R}$ must be less than 1 for the sum above to converge.

At this point, we remind the basic definitions of the terms denoted before. An *eigenvector* of a matrix $\mathbf{M}$ is a vector $\mathbf{x}$ for which $\mathbf{Mx}=\alpha\mathbf{x}$. The value α is the *eigenvalue* corresponding to the eigenvector. Last but not least, the *spectral radius* of matrix $\mathbf{M}$ is the magnitude of the largest eigenvalue.

Another quite interesting aspect we can note is that the matrix $\mathbf{R}$ (see [4]) has an interesting probabilistic interpretation. Recall that we are dealing with the continuous time case, so the result for $\mathbf{R}$ is actually that the entry in its (j, j') position is the expected time that the process spends to phase state j' of level $i+1$ before returning to level i, given the fact that the process is started from phase j of level i. So, have in mind that we start from a state (i, j) and the expected time in state $(i + 1, j')$ before returning to level i is $\mathbf{R}(j, j')$. Thus, this interpretation explains that the entries of the rate matrix must be non-negative.

Let us assume now that we have a solution for $\mathbf{R}$, according to what we have discussed before. The matrix $\mathbf{R}$ is termed as the *rate matrix.* As we should determine the stationary probabilities, we continue following the same process with the scalar state case. The probabilities of the boundary states of the process, are given by:

$$\begin{aligned} \boldsymbol{\pi_0 B_{0,0}} + \boldsymbol{\pi_1 B_{1,0}} \qquad &= \mathbf{0} \\ \boldsymbol{\pi_0 B_{0,1}} + \boldsymbol{\pi_1 A_1} + \boldsymbol{\pi_2 A_2} &= \mathbf{0} \end{aligned} \tag{2.9}$$

These equations can be written in matrix form, so, if we substitute the relation $\boldsymbol{\pi_2} = \boldsymbol{R\pi_1}$ in (2.9), we have the following:

$$(\boldsymbol{\pi_0}, \boldsymbol{\pi_1}) \begin{bmatrix} \boldsymbol{B_{0,0}} & \boldsymbol{B_{0,1}} \\ \boldsymbol{B_{1,0}} & \boldsymbol{A_1} + \boldsymbol{RA_2} \end{bmatrix} = \mathbf{0} \tag{2.10}$$

Similar to the scalar case, the equations in (2.10) are not sufficient to determine the probabilities $\boldsymbol{\pi_0}$ and $\boldsymbol{\pi_1}$. We also require the use of the *normalization constraint*, which is:

$$\mathbf{1} = \boldsymbol{\pi_0 e} + \boldsymbol{\pi_1} \sum_{j=1}^{\infty} \boldsymbol{R^{j-1} e} = \boldsymbol{\pi_0 e} + \boldsymbol{\pi_1 (I - R)^{-1} e} \tag{2.11}$$

In equation (2.11), we used the convergence of the infinite summation $\sum_{j=1}^{\infty} \boldsymbol{R^{j-1}}$ to $\boldsymbol{(I - R)^{-1}}$. The calculation is accomplished with the same technique that is used to close an infinite geometric series, and is showed below.

Let $\mathbf{S}$ be the matrix obtained by the infinite sum. Thus, if we assume that the sum converges, we can write:

$$\boldsymbol{S} = \sum_{j=1}^{\infty} \boldsymbol{R}^{\boldsymbol{j-1}} = \boldsymbol{I} + \boldsymbol{R} + \boldsymbol{R}^{\boldsymbol{2}} + \ldots \tag{2.12}$$

Multiplying (2.12) by $\mathbf{R}$ on the right implies:

$$\boldsymbol{S}\boldsymbol{R} = \boldsymbol{R} + \boldsymbol{R}^{\boldsymbol{2}} + \ldots$$

Then, upon subtracting the last equation from (2.12), we have:

$$\boldsymbol{S}(\boldsymbol{I} - \boldsymbol{R}) = (\boldsymbol{I} + \boldsymbol{R} + \boldsymbol{R}^{\boldsymbol{2}} + \ldots) - (\boldsymbol{R} + \boldsymbol{R}^{\boldsymbol{2}} + \ldots) = \boldsymbol{I}$$

The last step is to multiply on the right both sides of this equation by $(\boldsymbol{I} - \boldsymbol{R})^{\boldsymbol{-1}}$. Hence, we obtain the very useful result that we mentioned before:

$$\sum_{j=1}^{\infty} \boldsymbol{R}^{\boldsymbol{j-1}} = (\boldsymbol{I} - \boldsymbol{R})^{-1} \tag{2.13}$$

As a matter of fact,this derivation depends on the assumption that the infinite geometric sequence converges. Remember that, for a nonnegative scalar, $1 + x + x^2 + \ldots$ converges to $\frac{1}{1-x}$ if and only if $x < 1$. Similarly, the geometric sequence for the nonnegative matrix $\mathbf{R}$ has an analogous criterion for convergence that is the spectral radius of the matrix must be less than unity.

We close this analysis by representing a different form of (2.11), more suitable for linear equation solvers. Firstly, we define $\boldsymbol{M}^*$ to be the matrix $\boldsymbol{M}$ without its first column, and $[1, \mathbf{0}]$ as a row vector with its first component 1 followed by 4 zeros. Under these assumptions, we can determine the solution for the boundary states by solving:

$$(\boldsymbol{\pi_0}, \boldsymbol{\pi_1}) \begin{bmatrix} \boldsymbol{e} & \boldsymbol{B}^*_{\boldsymbol{0,0}} & \boldsymbol{B_{0,1}} \\ (\boldsymbol{I} - \boldsymbol{R})^{-1}\boldsymbol{e} & \boldsymbol{B}^*_{\boldsymbol{1,0}} & \boldsymbol{A_1} + \boldsymbol{R}\boldsymbol{A_2} \end{bmatrix} = [1, \mathbf{0}] \tag{2.14}$$

The equation that results by multiplying $(\boldsymbol{\pi_0}, \boldsymbol{\pi_1})$ with the first column of the matrix is actually the normalization condition, while the rest equations are given by (2.10). Hence, the linear equations given by (2.14) have a unique solution that satisfies the normalization condition too. The process has come to an end.

2.1.3 Homogeneous and non-homogeneous QBDs

Technically, now that we have seen how the Matrix Geometric Method works, we can analyze more the form of generator matrix $\mathbf{Q}$. In the previous example, the generator matrix was generally defined as follows:

$$\boldsymbol{Q} = \begin{bmatrix} \boldsymbol{B_{0,0}} & \boldsymbol{B_{0,1}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots \\ \boldsymbol{B_{1,0}} & \boldsymbol{A_1} & \boldsymbol{A_0} & \mathbf{0} & \mathbf{0} & \cdots \\ \mathbf{0} & \boldsymbol{A_2} & \boldsymbol{A_1} & \boldsymbol{A_0} & \mathbf{0} & \cdots \\ \mathbf{0} & \mathbf{0} & \boldsymbol{A_2} & \boldsymbol{A_1} & \boldsymbol{A_0} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

We notice immediately that from the third row to below, the matrices A_0, A_1 and A_2 are repeated diagonally infinitely many times. This form is incredibly simple and describes the *Homogeneous* QBDs.

Logically, the fundamental ideas for these problems can be applied beyond homogeneous QBDs and yield more complicated systems. So, the systems with increased complexity will have a more complicated generator matrix.

Actually, we refer to *non-homogeneous* QBDs. The state space is two dimensional too, with levels and phases, and the transitions are still allowed to adjacent levels and the same level only, with a basic difference; the transition probabilities out of the state (i, s) may depend on level i. This indicates that the general form of the transition matrix in a non-homogeneous QBD is:

$$\boldsymbol{Q} = \begin{bmatrix} \boldsymbol{A_1^{(0)}} & \boldsymbol{A_0^{(0)}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots \\ \boldsymbol{A_2^{(1)}} & \boldsymbol{A_1^{(1)}} & \boldsymbol{A_0^{(1)}} & \mathbf{0} & \mathbf{0} & \cdots \\ \mathbf{0} & \boldsymbol{A_2^{(2)}} & \boldsymbol{A_1^{(2)}} & \boldsymbol{A_0^{(2)}} & \mathbf{0} & \cdots \\ \mathbf{0} & \mathbf{0} & \boldsymbol{A_2^{(3)}} & \boldsymbol{A_1^{(3)}} & \boldsymbol{A_0^{(3)}} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

Note that different levels may have different numbers of phases. In this case, the blocks on the main diagonal are square matrices, but those on the secondary diagonals can be rectangular matrices of appropriate dimensions. Certainly, that depends on the system we are dealing with.

The interesting fact here is that, the method we analyzed for homogeneous QBDs can be extended instantly to the non-homogeneous case. We can follow the same steps to obtain the stationary distribution, but, as it was expected, we have many more computational issues than in homogeneous QBDs. These problems can be resolved if some additional assumptions are made about the matrix $\mathbf{Q}$ and depend again on the specific model under study.

2.2 Game Theoretic Preliminaries

In this section, we remind some of the basic concepts from *Game Theory* mainly in the form of definitions. The following are fundamental for understanding anything related to strategic behavior.

In essence, we are interested in non-cooperative games. To explain succinctly, a *non-cooperative game* is a game where the players cannot collaborate and actually everyone is obliged to choose her/his actions alone. Hence, our purpose is to give the fundamental background of such a kind of game in classical Game Theory. We let:

- $\mathcal{N} = \{1, \ldots, n\}$ be a finite set of players
- $\mathcal{A}_i$ be sets of action plans, one for each player $i = 1, \ldots, n$

As for sets $\mathcal{A}_i$, every one of them contains all actions available to player i. Thus, every element in $\mathcal{A}_i$ specifies what actions should be taken during the game, and it is referred as a *pure strategy* of i. Furthermore, a probability distribution on $\mathcal{A}_i$ is defined as a *mixed strategy* of i. Hence, the use of a mixed strategy describes the fact that a player chooses one of her/his pure strategies through the probability distribution defined by the certain mixed strategy. At this point, we also define:

- $\mathcal{S}_i$ as the set of mixed strategies for player i
- $s = (s_1, \ldots, s_n)$ as the *strategy profile*, an ordered n-tuple of strategies, one for each player, $s_i \in \mathcal{S}_i$, $i = 1, \ldots, n$
- $\mathcal{U}_i$ noted as real payoff functions, one for each player $i = 1, \ldots, n$.

We now note that s_{-i} is a (n-1)-dimensional vector which contains all the strategies of s except of the one that corresponds to player i. Hence, we can write a strategy profile as $s = (s_i, s_{-i})$.

As it was expected, the payoff functions have a central role here. Actually, a function $\mathcal{U}_i(s) = \mathcal{U}_i(s_i, s_{-i})$ determines the payoff received by player i given that the strategy profile s is adopted by the players. The function $\mathcal{U}_i(s)$ is also assumed to be linear with respect to s_i. This means that, if the strategy s_i mixes the strategies s_i^k with probabilities α_k respectively, $k = 1, 2, \ldots, r$, then we have:

$$\mathcal{U}_i(s_i, s_{-i}) = \sum_{k=1}^{r} \alpha_k \mathcal{U}_i(s_i^k, s_{-i})$$

Now, let s_i^1 and s_i^2 be both strategies of player i. We define as well that:

- Strategy s_i^1 *weakly dominates* strategy s_i^2, if for any strategy profile for the other players, s_{-i}, we have that $\mathcal{U}_i(s_i^1, s_{-i}) \geq \mathcal{U}_i(s_i^2, s_{-i})$, with the inequality strict for at least one of the strategy profiles s_{-i}.

- Strategy s_i^1 *strongly dominates* strategy s_i^2, if the inequality above is strict for all the strategy profiles s_{-i}.

- A strategy s_i^* is noted as a *best response* for player i against the strategy profile s_{-i} if for every strategy s_i of i we have $\mathcal{U}_i(s_i^*, s_{-i}) \geq \mathcal{U}_i(s_i, s_{-i})$. In other words, s_i^* is a best response for player i if it maximizes the function $f(s_i) = \mathcal{U}_i(s_i, s_{-i})$.

- A strategy profile $s_i^e = (s_1^e, \ldots, s_n^e)$ is a *Nash equilibrium* profile if for every $i \in \mathcal{N}$, s_i^e is a best response for player i against s_{-i}^e. That means no player has an incentive to deviate from this specific strategy profile. Note that a Nash equilibrium does not always exist.

Generally, we will deal mostly with games with an infinite number of players. Certainly, the players in this work are the strategic customers of the system. Thus, in this case, we denote the set of strategies and the payoff function by $\mathcal{S}$ and $\mathcal{U}$ respectively. We also let $\mathcal{U}(s, s')$ be the payoff function for a player who chooses strategy s while the rest of the players choose strategy s'. Then, we define the following:

- A strategy $s^e \in \mathcal{S}$ is a *symmetric Nash equilibrium* if it is a best response to itself. That is, $\mathcal{U}(s^e, s^e) \geq \mathcal{U}(s, s^e)$ with $s \in \mathcal{S}$.

We also note that, in order to comprehend the strategic customers' behavior, we should compute the payoff function $\mathcal{U}(s, s')$. It is natural to assume that the tagged customer's strategy, s, does not really make an impact on the general behavior of the system which is actually determined by the strategy s' that the other customers follow.

Additionally, we will refer to *threshold strategies*, which are included in the following of this work and generally are quite common in queueing systems. Let us assume that, upon arrival, the customer has to choose between two actions, namely A_1 and A_2. The decision has to be made according to the observation of a random variable with a non-negative integer value which describes the state of the system. The most common example of this value may be the queue length, and the action join or balk respectively. Under these assumptions, we define:

- A *pure threshold strategy with threshold n* is defined by a decision of the action A_1 for every state in $\{1, 2, \ldots, n-1\}$, while the action A_2 is decided for every other state.

Note also that there are cases where it is logical to look for a *Nash equilibrium pure threshold strategy.* However, it does not always exist.

Generally, we can face differently defined cases of a threshold strategy that can be adopted by customers, for example extensions of the definition given above. We select the appropriate one in agreement with the problem we are dealing with; the basic element is always that the decision in made through one threshold or more.

Chapter 3

Ticket Queues with balking customers

In this chapter, we are going to analyze Ticket Queues where the customers may balk if they consider that the queue length exceeds their patience. For convenience, specifically here, we will refer to Ticket Queue with balking customers simply as Ticket Queue. As mentioned before, this decision between two actions, *join* or *balk*, is made through an observed information, where, in this case is the *ticket position*. Remember that the ticket position is the difference of the number on the issued ticket with the observed number on the display panel. As a matter of fact, this is the only information customers get in this system, as the physical queue is invisible. As a consequence, they will overestimate their waiting time.

Subsequently, we will introduce an appropriate model for our analysis and will obtain the steady state probabilities. Later, we will introduce some suggestions for improvement according to this model, and we will also discuss the case of threshold strategies. Finally, we conclude with our inferences. The material we are working with here is mainly taken from the works of Xu, Gao and Ou (2007) and Kerner, Sherzer and Yanco (2017).

3.1 Description of the Model

A Ticket Queue system with balking customers can be described as follows. Imagine that a customer arrives at a single-sever system, and upon arrival, is issued with a ticket that has a number which indicates her/his position in queue. This number is increasing every time a new customer takes a ticket. The customer sees on the display panel the number of the ticket that has been issued to the customer currently under service, and decides whether s/he is going to enter the queue or not. When a service is completed, the system calls for the next number. If the next number belongs to a balking customer, the system calls the number after her/him. Otherwise, if no ticket has issued after the customer who finishes service, there are no waiting customers and the panel displays the next number so that the arriving customer who draws it can be served immediately.

To simplify the analysis and focus on the main issues, we consider the model of a single-server ticket queue where the customers arrive according to a Poisson process and they have service times with independent and identical exponential distributions. We let λ be the rate of the Poisson arrival process, and $\frac{1}{\mu}$ be the mean of the exponential service times.

We now assume that, a customer will balk if her/his ticket position is greater or equal to a threshold K, and no customer can renege if s/he has entered the queue and waits for service. This constant K is actually the tolerance level of a customer's patience. We denote the ticket position as D. We follow the assumption that an arriving customer perceives D as the actual number of customers in the system. As discussed before, D is just an upper bound of the actual queue length in the system which here is denoted as a generic random variable N and it is referred as the *queueing position*. Thus, customers join the system if $D < K$, otherwise they balk. Their decision is not related to the actual queue length N.

The general difficulty in this system is that neither D nor N separately carry all the information about the ticket queue, and this is also valid even if they are jointly given. Thus, we have to define the states with more detail in order to achieve a Markovian description of the system. More precisely, we define the following: we let state 0 denote the *empty system*, while all the other states are represented with an L-tuple vector n. L is defined as the *realization* of N, which means that it corresponds to the actual number of customers that are present in the system while being on the specific state. Hence, a state is defined by the vector $\boldsymbol{n} = (n_1, \ldots, n_L)$, with n_l representing the number of tickets issued from the arrival of the lth joining customer prior to the $(l+1)$th joining customer for $l = 1, \ldots, L$. Naturally, the first joining customer is currently in service.

With the definitions given above, it is clear that we have complete information on N and D. Specifically, in a state $\boldsymbol{n} = (n_1, \ldots, n_L)$, we have L joining customers and ticket position D equal to $\sum_{l=1}^{L} n_l$, because an arriving customer sees, according to the number on the display panel, a difference to her/his drawn ticket equal to the sum of the tickets issued from the customer in service until her/him. For instance, at a state $\boldsymbol{n} = (1, 1, 4, 2)$ we have $N = 4$ and $D = 1 + 1 + 4 + 2 = 8$.

At this point it is natural to refer to the transitions between states. We should consider the necessary condition for a customer to join the system, which is $D < K$. Thus, if $\sum_{l=1}^{L} n_l < K$, the arriving customer joins the system, and consequently the state $(n_1, \ldots, n_L)$ will change at rate λ to state $(n_1, \ldots, n_L, 1)$ which indicates that there are currently $L+1$ customers in the system and the ticket position for the next arriving customer is $\sum_{l=1}^{L+1} n_l$ with $n_{L+1} = 1$. On the other hand, if $\sum_{l=1}^{L} n_l \geq K$, the arriving customer balks and hence the state $(n_1, \ldots, n_L)$ will change at rate λ to state $(n_1, \ldots, n_L+1)$. That means we still have L joining customers to the system but the ticket position for the next arriving customer has increased by one. Additionally, if a service is completed at rate μ, the state $(n_1, n_2, \ldots, n_L)$ will change to state $(n_2, \ldots, n_L)$. The explanation here is simple too. The first joining customer left the system and so there are $L - 1$ customers left, but also releases all the balking customers between her/him and the second joining customer, because their numbers are called by the system automatically and the display panel shows the number of the next joining customer in negligible time.

Pursuant to the discussion before, the complete state space is defined by:

$$\mathcal{S} = \{0\} \cup \left\{ \boldsymbol{n} \in \mathbb{N}^L : \sum_{l=1}^{L-1} n_l < K, n_l \geq 1, l = 1, \ldots, L, L = 1, \ldots, K \right\} \quad (3.1)$$

The condition $\sum_{l=1}^{L-1} n_l < K$ in the state space definition actually ensures that every joining customer has observed a ticket position lower than her/his balking limit and hence entered the queue. Furthermore, we give two additional definitions for convenience. We note for a given state $\boldsymbol{n} = (n_1, \ldots, n_L)$ its dimension as $|\boldsymbol{n}| = L$ and its sum of components as $\|\boldsymbol{n}\| = \sum_{j=1}^{L} n_j$. In the following, we present an example of this Ticket Queue with $K = 4$ in order to make the Markovian model even more transparent and introduce briefly the steps for calculating the stationary distribution.

The illustration in Figure 3.1 shows the transition diagram for the Ticket Queue with balking limit $K = 4$. The black arrows correspond to new arrivals with transition rate λ and the red arrows to service completions with transition rate μ.

However, an observation of high interest on the transition diagram is the gray-shaded states. We firstly remark that any arriving customer in a gray-shaded state will balk, because the inequality $\sum_{l=1}^{L} n_l \geq K$ is satisfied for $K = 4$. That happens also for some states in the red-shaded zone. The actual difference though is that for every state in the gray-shaded zone we have that $n_L \geq K$. This property shows the fact that, in order to have a new joining customer in the system, we have to complete all the services of the current customers in queue. This is the only way to release n_L so that a new customer observes a ticket position lower than her/his balking limit and join the system.

As an example, we will consider two states, (22) and (25). An arriving customer to state (22) observes a ticket position of $2 + 2 = 4 = K$, so s/he balks and therefore state (22) reduces to state (23) at rate λ. Moreover, if a service completion takes place while being on state (22), we move to state (2) at rate μ. An arriving customer in (25) will also balk, as $2 + 5 = 7 > 4 = K$. So, at rate λ, state (25) jumps to state (26). A service completion in state (25) gives the transition to state (5) at rate μ as it was expected. The difference noted above here is that, from state (22) we do not need to return to empty system (0) in order to have a new joining customer, but from state (25) we need to get back to (0) for an arriving customer to join. That is why, in the gray-shaded zone all the states lead to (0) and the services of all the current customers have to be completed for a new customer to enter the queue.

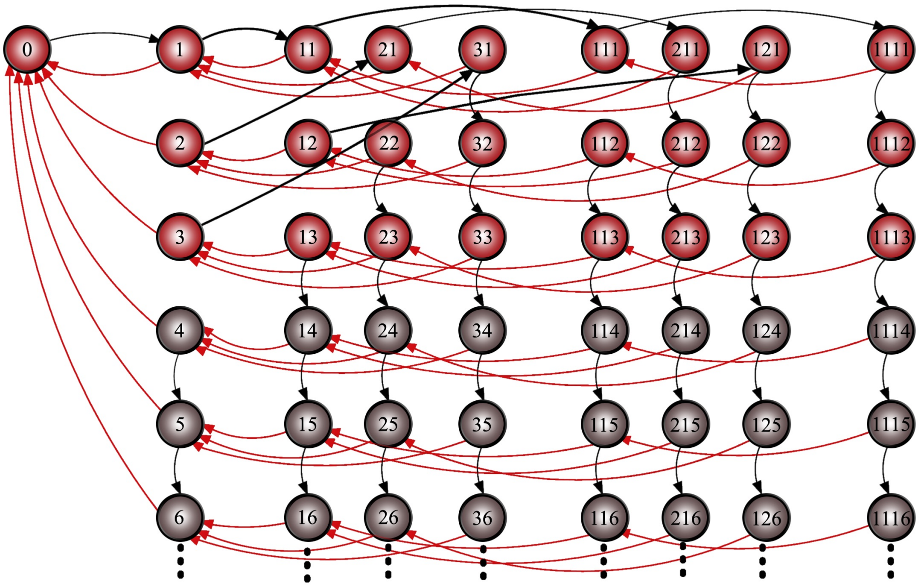


Figure 3.1 Transition Diagram of Ticket Queue with $K = 4$

3.2 Stationary distribution of the model

In the previous section, we described the model of Ticket Queue with balking customers and we also presented the case where $K = 4$. At this point, we need to obtain the steady state distribution of the system, denoted as $\{p(\boldsymbol{n}), \boldsymbol{n} \in \mathcal{S}\}$. The unpleasant situation here is that we have closed-form solution of the steady state distribution only for small values of the balking limit K. Fortunately though, we can follow a two-step procedure in order to solve for the steady state distribution for general K. We now present briefly the two steps, and the process is described in more detail in the next subsections.

- We aggregate all the states $\boldsymbol{n} \in \mathcal{S}$ with $|\boldsymbol{n}| = L$ and $n_L \geq K$ into a *super state* S_L, $L = 1, \ldots, K$, so that we have K super states. Hence, the new Markov chain, hereafter called the *aggregated Markov Chain*, has a finite state space and can be model as a QBD process.
- We disaggregate the super states and, recursively, we can obtain the steady state probabilities of the remaining states.

3.2.1 Steady State Probabilities of Single States

The process we need to follow in order to obtain the steady state probabilities starts with the *state aggregation* idea. As seen in Figure 3.1, the states in the gray zone, that is the states $\boldsymbol{n} \in \mathcal{S}$ with $n_L \geq K$, have a specific property if we group them with criterion the real number of customers in queue, which is $|\boldsymbol{n}| = L$ for each state. This property is the following:

- An arriving customer in such a state will surely balk as $\|\boldsymbol{n}\| \geq n_L \geq K$. After this event, the number of joining customers will remain L and a state transition will be accomplished towards a state of the same group; that is a state with $n_L \geq K$ and $|\boldsymbol{n}| = L$.
- If a service completion occurs, the number of the joining customers will be reduced to $L - 1$. At the same time, the Markov chain will jump to a state which has $n_{L-1} \geq K$ and $|\boldsymbol{n}| = L - 1$. Hence, the transition will be done to a state of the lower group.

As an example, in the $K = 4$ case, states (1114),(1115),(1116), …, constitute a group of infinitely many states where $n_4 \geq 4$ (have in mind that n_L is the last component of each vector state) and $|\boldsymbol{n}| = 4$. Every arrival in these states will lead to a state in the same group, and every service will lead to a state with $|\boldsymbol{n}| = 3$ and $n_3 \geq 4$. That is, the lower group.

This is exactly the aggregation idea. As the states $\boldsymbol{n} \in \mathcal{S}$ with $n_L \geq K$ and $|\boldsymbol{n}| = L$ have the same property we can aggregate all the states of the same group to a *super state*. In that way, the state space becomes finite and turns out to be a QBD process, as we will show later. We define the *super states* as:

- $S_L = \{\boldsymbol{n} \in \mathcal{S} : |\boldsymbol{n}| = L, n_L \geq K\}$, $L = 1, 2, \ldots, K-1$.
- $S_K = \{\boldsymbol{n} = (e_{K-1}, n_K) : n_K \geq K\}$,

where $e_L = (1, \ldots, 1)$ is a L-dimensional unit vector. Thus, $(e_{K-1}, n_K) = (1, \ldots, 1, n_K)$. As an example, we can see that in Figure 3.1, all the states which belong to the last column (where $L = K$) have exactly this form: $(111n_4)$ for $K = 4$.

In that way, the state space has been reduced to a finite set. Hence, the state space of the aggregated Markov chain is:

$$\begin{aligned} \mathcal{S}^* = \{0\} \cup \Bigg\{ \boldsymbol{n} \in \mathbb{N}^L : \sum_{l=1}^{L-1} n_l < K, n_L < K, n_l \geq 1, \\ l = 1, \ldots, L, L = 1, \ldots, K \Bigg\} \cup \{S_1, \ldots, S_K\} \end{aligned} \tag{3.2}$$

In order to model the aggregated Markov chain as a QBD process, we firstly need to make a partition of the state space $\mathcal{S}^*$. We divide $\mathcal{S}^*$ into $K+1$ parts, $\{K_0, K_1, \ldots, K_K\}$, where we note:

- $K_0 = \{0\}$
- $K_L = \{\boldsymbol{n} \in \mathcal{S}^* : |\boldsymbol{n}| = L, n_L < K\} \cup S_L$, $L = 1, 2, \ldots, K$

It is clear now that K_L is a collection of all the L-dimensional states which represent L joining customers in the system. This can be shown schematically in Figure 3.2 below.

We should also note that, within K_L, we order any two states *lexicographically*. Particularly, for any two states in K_L, namely $\boldsymbol{n} = (n_1, \ldots, n_L)$ and $\boldsymbol{n'} = (n'_1, \ldots, n'_L)$, $\boldsymbol{n}$ is listed ahead of $\boldsymbol{n'}$ if $n_L < n'_L$ or $n_L = n'_L$ and $\sum_{l=i}^{L-1} n_i \leq \sum_{l=i}^{L-1} n'_i$ for all $1 \leq i \leq L-1$.

For instance, the partition of $\mathcal{S}^*$ for $K = 4$, with all K_L following the lexicographic order described above is:

$$K_0 = \{0\}$$

$$K_1 = \{1, 2, 3, S_1\}$$

$$K_2 = \{11, 21, 31, 12, 22, 32, 13, 23, 33, S_2\}$$

$$K_3 = \{111, 211, 121, 112, 212, 122, 113, 213, 123, S_3\}$$

$$K_4 = \{1111, 1112, 1113, S_4\}$$

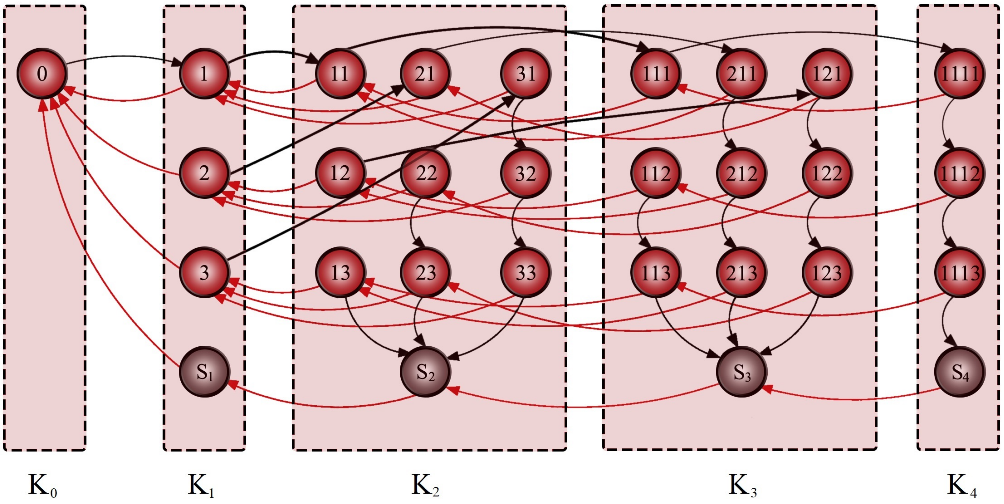


Figure 3.2 Transition Diagram of aggregated Ticket Queue with $K = 4$

Let us now treat the subsets $\{K_0, K_1, \ldots, K_K\}$ as blocks with $K+1$ levels. As we can easily see in Figure 3.2 for $K = 4$, we have $K + 1 = 5$ levels, and a block transition can change its level by at most 1. For example, from subset K_1 we can only transit to K_0 or K_2, meaning that the we have solely adjacent level transitions. Hence, for general K, if we assume that subsets K_L, $L = 0, \ldots, K$ are blocks with $|K_L|$ being the number of elements of the corresponding subset, we can treat the aggregated Markov chain as a QBD process and denote the following infinitesimal block partitioned generator matrix:

$$\boldsymbol{Q} = \begin{bmatrix} -\boldsymbol{\lambda} & \boldsymbol{A_{01}} & & & & & \\ \boldsymbol{\mu e'_K} & \boldsymbol{A_{11}} & \boldsymbol{A_{12}} & & & & \\ & \boldsymbol{A_{21}} & \boldsymbol{A_{22}} & \boldsymbol{A_{23}} & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & & \boldsymbol{A_{K-1,K-2}} & \boldsymbol{A_{K-1,K-1}} & \boldsymbol{A_{K-1,K}} \\ & & & & & \boldsymbol{A_{K,K-1}} & \boldsymbol{A_{K,K}} \end{bmatrix} \tag{3.3}$$

In matrix $\mathbf{Q}$, $\boldsymbol{e'_K}$ is the transpose of vector $\boldsymbol{e_K}$. Furthermore, $A_{L,L}$, $A_{L,L+1}$ and $A_{L+1,L}$ are $|K_L| \times |K_L|$, $|K_L| \times |K_{L+1}|$ and $|K_{L+1}| \times |K_L|$ matrices, respectively, for every $L = 1, 2, \ldots, K$.

Since we have defined the transition matrix $\mathbf{Q}$ in a block form, it is time to begin the computation of the stationary distribution. Hence, we let $\boldsymbol{p}_L$ be the steady state probabilities for states in K_L. Naturally, $\boldsymbol{p}_0 = p(0)$, as K_0 is composed by the single state (0), and $\boldsymbol{p}_L$ is a row vector of $|K_L|$ dimension. An example is that for $K = 4$, $\boldsymbol{p}_1 = \big(p(1), p(2), p(3), p(S_1)\big)$, thus $\boldsymbol{p}_1$ is a row vector of $|K_1| = 4$ dimension. Hence, we can represent the distribution $\{\boldsymbol{p}(\boldsymbol{n}), \boldsymbol{n} \in \mathcal{S}^*\}$ using the notation $\boldsymbol{p} = (\boldsymbol{p}_0, \boldsymbol{p}_1, \ldots, \boldsymbol{p}_K)$ in order to make our computations more convenient.

As it is well known, the stationary distribution can be computed by solving the balance equations: $\boldsymbol{pQ} = \mathbf{0}$. Thus, we have:

$$(\boldsymbol{p}_0, \ldots, \boldsymbol{p}_K) \begin{bmatrix} -\boldsymbol{\lambda} & \boldsymbol{A_{01}} & & & & & \\ \boldsymbol{\mu e'_K} & \boldsymbol{A_{11}} & \boldsymbol{A_{12}} & & & & \\ & \boldsymbol{A_{21}} & \boldsymbol{A_{22}} & \boldsymbol{A_{23}} & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & & \boldsymbol{A_{K-1,K-2}} & \boldsymbol{A_{K-1,K-1}} & \boldsymbol{A_{K-1,K}} \\ & & & & & \boldsymbol{A_{K,K-1}} & \boldsymbol{A_{K,K}} \end{bmatrix} = \mathbf{0}$$

By carrying out the vector-matrix multiplication we obtain:

$$-\lambda \boldsymbol{p}_0 + \mu \boldsymbol{p}_1 \boldsymbol{e'_K} = 0 \tag{3.4}$$

$$\boldsymbol{p}_{L-1} \boldsymbol{A}_{L-1,L} + \boldsymbol{p}_L \boldsymbol{A}_{L,L} + \boldsymbol{p}_{L+1} \boldsymbol{A}_{L+1,L} = \mathbf{0}, \quad L = 1, \ldots, K-1 \tag{3.5}$$

$$\boldsymbol{p}_{K-1} \boldsymbol{A}_{K-1,K} + \boldsymbol{p}_K \boldsymbol{A}_{K,K} = 0 \tag{3.6}$$

For clarification, note that the result of (3.5) is a $|K_L|$ dimensional row vector, which corresponds to the $|K_L|$ balance equations associated with the states in

K_L. As an example, (3.5) for $K = 4$ and $L = 1$ is $\boldsymbol{p}_0\boldsymbol{A}_{01}+\boldsymbol{p}_1\boldsymbol{A}_{11}+\boldsymbol{p}_2\boldsymbol{A}_{21} = \mathbf{0}$, where $\boldsymbol{A}_{01}$,$\boldsymbol{A}_{11}$ and $\boldsymbol{A}_{21}$ are 1×4, 4×4 and 10×4 matrices respectively, and $\boldsymbol{p}_0$ is a scalar, while $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$ are row vectors of dimension 4 and 10 in correspondence. Every vector-matrix multiplication in the equation results in a 1×4 matrix which is actually a row vector of $|K_1| = 4$ dimension. Thus, the resulting vector indeed corresponds to the 4 balance equations associated with the states in $K_1 = \{1, 2, 3, S_1\}$.

As things stand now, the solution to (3.4),(3.5) and (3.6) can be given through *recursion*. At first, equation (3.6) gives us after some simple matrix operations:

$$\boldsymbol{p}_K = -\boldsymbol{p}_{K-1}\boldsymbol{A}_{K-1,K}(\boldsymbol{A}_{KK})^{-1} \equiv \boldsymbol{p}_{K-1}\boldsymbol{R}_{K-1,K} \tag{3.7}$$

where $\boldsymbol{R}_{K-1,K}$ is a $|K_{K-1}| \times |K_K|$ matrix that can be found recursively using:

$$\boldsymbol{R}_{K-1,K} = -\boldsymbol{A}_{K-1,K}(\boldsymbol{A}_{KK})^{-1} \tag{3.8}$$

Following a similar process for (3.5), we can get $\boldsymbol{p}_L$. If we substitute K with L in (3.7), we obtain:

$$\boldsymbol{p}_L = \boldsymbol{p}_{L-1}\boldsymbol{R}_{L-1,L}, \quad L = 1, \dots, K-1 \tag{3.9}$$

Furthermore, if (3.9) holds for $L = 1, \dots, K-1$, then it holds for $L = L+1$ in every case. We now substitute $\boldsymbol{p}_{L+1}$ given by (3.9) to equation (3.5). Then, we have, for $L = 1, \dots, K-1$:

$$\begin{aligned} \boldsymbol{p}_{L-1}\boldsymbol{A}_{L-1,L} + \boldsymbol{p}_L\boldsymbol{A}_{L,L} + \boldsymbol{p}_L\boldsymbol{R}_{L,L+1}\boldsymbol{A}_{L+1,L} &= \mathbf{0} \\ \boldsymbol{p}_L\big(\boldsymbol{A}_{L,L} + \boldsymbol{R}_{L,L+1}\boldsymbol{A}_{L+1,L}\big) &= \boldsymbol{p}_{L-1}\big(-\boldsymbol{A}_{L-1,L}\big) \\ \boldsymbol{p}_L = \boldsymbol{p}_{L-1}\big(-\boldsymbol{A}_{L-1,L}\big)\big(\boldsymbol{A}_{L,L} + \boldsymbol{R}_{L,L+1}\boldsymbol{A}_{L+1,L}\big)^{-1} &= \boldsymbol{p}_{L-1}\boldsymbol{R}_{L-1,L} \end{aligned} \tag{3.10}$$

That means $\boldsymbol{R}_{L-1,L}$ is a $|K_{L-1}| \times |K_L|$ matrix that can be found recursively using:

$$\boldsymbol{R}_{L-1,L} = -\boldsymbol{A}_{L-1,L}\big(\boldsymbol{A}_{L,L} + \boldsymbol{R}_{L,L+1}\boldsymbol{A}_{L+1,L}\big)^{-1}, \quad L = 1, \dots, K-1 \tag{3.11}$$

We now observe that from (3.10) we can obtain $\boldsymbol{p}_L$ for $L = 1, \dots, K-1$. Recursively, we have that $\boldsymbol{p}_L = \boldsymbol{p}_{L-1}\boldsymbol{R}_{L-1,L} = \boldsymbol{p}_{L-2}\boldsymbol{R}_{L-2,L}\boldsymbol{R}_{L-1,L} = \dots = \boldsymbol{p}_0\boldsymbol{R}_{0,1}\cdots\boldsymbol{R}_{L-1,L}$. Hence, we have:

$$\boldsymbol{p}_L = \boldsymbol{p}_0 \boldsymbol{R}_L, \quad L = 1, \ldots, K \tag{3.12}$$

$$\boldsymbol{R}_L = \prod_{l=1}^{L} \boldsymbol{R}_{l-1,l}$$

where $\boldsymbol{R}_L$ is defined as a $|K_L|$ dimensional row vector. The next step to obtain the steady state probabilities, is to get $\boldsymbol{p}_0$. That is feasible, using the normalization equation, which is:

$$\boldsymbol{p}\boldsymbol{e}'_{|\mathcal{S}^*|} = \boldsymbol{p}_0 \left(1 + \sum_{L=1}^{K} \boldsymbol{R}_L \boldsymbol{e}'_{|K_L|} \right)$$

Note that $|\mathcal{S}^*|$ is defined as the total number of states in set $\mathcal{S}^*$. Thus, from normalization equation and (3.12), we obtain:

$$\boldsymbol{p}_0 = \frac{1}{1 + \sum_{L=1}^{K} \boldsymbol{R}_L \boldsymbol{e}'_{|K_L|}} \tag{3.13}$$

$$\boldsymbol{p}_L = \frac{\boldsymbol{R}_L}{1 + \sum_{L=1}^{K} \boldsymbol{R}_L \boldsymbol{e}'_{|K_L|}}, \quad L = 1, \ldots, K \tag{3.14}$$

We close this subsection with two interesting observations about the aggregated Markov chain. Initially, we note that it contains complete information about N, as $P(N = L) = \sum_{\boldsymbol{n} \in K_L} \boldsymbol{p}(\boldsymbol{n})$, $L = 1, \ldots, K$. We can also obtain from $\{\boldsymbol{p}(\boldsymbol{n}), \boldsymbol{n} \in \mathcal{S}^*\}$ other useful information, such as the customer's balking probability. However, we cannot obtain neither the complete marginal distribution of D nor the complete joint distribution of (N, D), as there is dependence on the individual probabilities of states in the super states. So, we need to disaggregate the super states for this kind of information.

The second observation is, that in order to solve the aggregated Markov chain, we could make a different partition of the state space $\mathcal{S}^*$. Specifically, the partition could be made according to the value of n_L, so that we separate the sets by rows (see Figure 3.2). Hence, we could define blocks like the following: $T_0 = \{0\}$, $T_\nu = \{\boldsymbol{n} : n_L = \nu\}$, $\nu = 1, \ldots, K-1$ and $T_K = \{S_L : L = 1, \ldots, K\}$ and continue the process. However, this partition has bigger computational complexity and that explains why we preferred the other one.

The first step has been achieved; we have the steady state probabilities of all states in $\mathcal{S}^*$. We need to obtain though the stationary distribution of all the states in $\mathcal{S}$, which means we have to find the steady state probabilities of the individual states in super states. This will be shown in the next subsection.

3.2.2 Steady State Probabilities of States in Super States

The second step of our process begins with the disaggregation of super states into individual states. Thus, we now work with the state space $\mathcal{S}$ which includes all the single states. In order to find the steady state probabilities of the remaining states, we have to repartition state space $\mathcal{S}$ into disjoint subsets, namely $\{T_\nu, \nu \geq 0\}$. They are defined as:

$$T_0 = K_0 = \{0\}$$
$$T_\nu = \{\boldsymbol{n} \in \mathcal{S} : n_L = \nu, L = 1, \ldots, K\}, \quad \nu = 1, 2, \ldots$$

In other words, T_ν contains all the states $\boldsymbol{n}$ where $n_L = \nu$, for $\nu = 1, 2, \ldots$, so that the partition is done with criterion the last component of the state to be ν. We literally treat the Markov chain like $n_L = \nu$ denotes the level and the rest of a state defines the phase. Therefore, we have to order any two states within T_ν, namely $\boldsymbol{n}$ and $\boldsymbol{n}'$, lexicographically. So $\boldsymbol{n}$ is listed ahead of $\boldsymbol{n}'$ if $|\boldsymbol{n}| < |\boldsymbol{n}'|$ or $|\boldsymbol{n}| = |\boldsymbol{n}'|$ and at the same time $\sum_{i=l}^{L-1} n_i \leq \sum_{i=l}^{L-1} n_i'$ for any l. A look at Figure 3.4 makes the partition straightforward for $K = 4$; T_ν contains all the states in the νth row of the diagram, with $\nu > 1$, and the states also appear in lexicographic order for each set. Furthermore, it can be shown that the number of states in each set T_ν is given by:

$$|T_\nu| = \sum_{L=1}^{K} \sum_{\boldsymbol{n}:n_L=\nu} n_l = 2^{K-1}, \quad \nu \geq 1 \tag{3.15}$$

The explanation of (3.15) is really simple. Have in mind that the joining customers in this kind of system are K at most. Thus, there is only possibility to have 1 or 2 or up to K joining customers in the system. As the first joining customer in every case must be in service (first position), there are $K - 1$ possible positions left for the other joining customers. That means, the problem can be modeled as *distribution of identical spheres in urns with capacity at most 1.* In order to make this straightforward, we show in the next figure the correspondence between the spheres in urns and the states in T_ν for the case of $K = 4$. Every sphere indicates a joining customer, and every urn her/his possible positions in the queue. So, for $K = 4$ we can see that $|T_\nu| = 2^{K-1} = 8$ very easily.

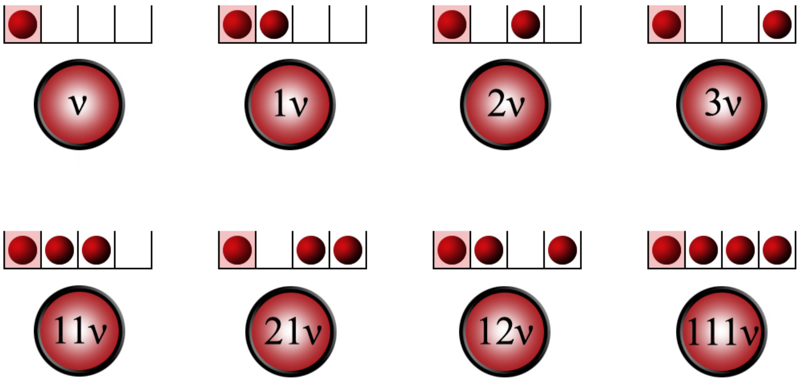


Figure 3.3 States in T_ν for $K = 4$

It is definite now that the number of states in this case is:

$$\binom{3}{0} + \binom{3}{1} + \binom{3}{2} + \binom{3}{3} = 8$$

That is, generally:

$$\binom{K-1}{0} + \binom{K-1}{1} + \ldots + \binom{K-1}{K-1} = \sum_{k=0}^{K-1} \binom{K-1}{k} = 2^{K-1}$$

The last equation is due to the well known *Binomial Theorem.*

In the previous step, we obtained the steady state probabilities of all states in the red zone, see Figure 3.1. These states, under the partition we made before, are the ones in T_ν where $\nu = K-1$. Hence, we have to compute the steady state probabilities for states in sets T_ν, with $\nu \geq K$. Clearly, the connection between the sets T_ν and super states S_L, $L = 1, \ldots, K$ is that $\bigcup_{\nu \geq K} T_\nu = \bigcup_{L=1}^{K} S_L$. So, now we have to follow a similar process and treat the sets $\{T_\nu, \nu \geq 0\}$ as blocks and $\nu \geq 1$ as level blocks. It is not difficult to realize that, from T_ν, the Markov chain can jump only to T_0 or $T_{\nu+1}$, or remain at the same level. Hence, we can presume that a portion of the block-partitioned infinitesimal generator is given by:

$$\widetilde{\boldsymbol{Q}} = \begin{bmatrix} \boldsymbol{B} & \lambda \boldsymbol{I}^0 & & \\ & \boldsymbol{B} & \lambda \boldsymbol{I} & \\ & & \boldsymbol{B} & \lambda \boldsymbol{I} \\ & & & \end{bmatrix} \tag{3.16}$$

where $\boldsymbol{I}$ is the $2^{K-1} \times 2^{K-1}$ identity matrix, $\boldsymbol{I}^0 = \boldsymbol{I}$ with the difference that its first entry on the main diagonal is zero, and $\boldsymbol{B}$ a lower triangular $2^{K-1} \times 2^{K-1}$ matrix. Also note that in (3.16), the first row with nonblank entries corresponds to the infinitesimal generator for T_{K-1}, the second row with nonblank entries to the same for T_K, and so on. All the blank entries are either equal to zero or they have nothing relevant about the computation of our desired probabilities.

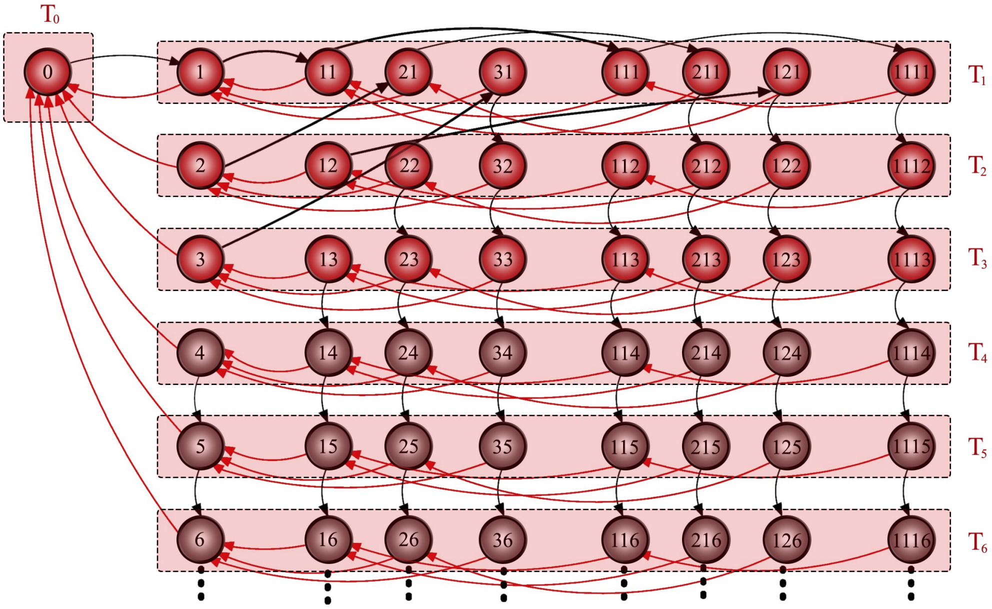


Figure 3.4 Transition Diagram of Ticket Queue with $K = 4$, Under the T_ν Partition

For the computation of the steady state probabilities, we need to obtain the matrix $\boldsymbol{B}$. That is totally achievable, once the balking limit K is specified. We give an example for $K = 4$. The sets T_ν for $\nu \geq K - 1$ are actually determined by $T_\nu = \{\nu, 1\nu, 2\nu, 3\nu, 11\nu, 21\nu, 12\nu, 111\nu\}$. Here, $|T_\nu| = 2^{K-1} = 2^3 = 8$, which means that $\boldsymbol{B}$ is a 8×8 matrix given by:

$$\boldsymbol{B}=\begin{bmatrix}-(\lambda+\mu) & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ \mu & -(\lambda+\mu) & 0 & 0 & 0 & 0 & 0 & 0\\ \mu & 0 & -(\lambda+\mu) & 0 & 0 & 0 & 0 & 0\\ \mu & 0 & 0 & -(\lambda+\mu) & 0 & 0 & 0 & 0\\ 0 & \mu & 0 & 0 & -(\lambda+\mu) & 0 & 0 & 0\\ 0 & \mu & 0 & 0 & 0 & -(\lambda+\mu) & 0 & 0\\ 0 & 0 & \mu & 0 & 0 & 0 & -(\lambda+\mu) & 0\\ 0 & 0 & 0 & 0 & \mu & 0 & 0 & -(\lambda+\mu)\end{bmatrix}$$

Regarding to the elements in the main diagonal of matrix $\boldsymbol{B}$, they are equal to $-(\lambda+\mu)$, because $\widetilde{\boldsymbol{Q}}$ is a stochastic matrix and every row sum must be equal to unity.

At this point, we let $\tilde{\boldsymbol{p}}_\nu$ be the steady state probability vector which corresponds to set T_ν. For $\nu \geq K$, we get the balance equations of $\tilde{\boldsymbol{p}}_\nu$ in a matrix form, so we have the following:

$$\lambda\tilde{\boldsymbol{p}}_{K-1}\boldsymbol{I}^0+\tilde{\boldsymbol{p}}_K\boldsymbol{B}=\mathbf{0} \tag{3.17}$$

$$\lambda\tilde{\boldsymbol{p}}_\nu+\tilde{\boldsymbol{p}}_{\nu+1}\boldsymbol{B}=\mathbf{0}, \quad \nu=K,K+1,\ldots \tag{3.18}$$

With some simple matrix operations on equation (3.17) we obtain:

$$\tilde{\boldsymbol{p}}_K=\tilde{\boldsymbol{p}}_{K-1}(-\lambda\boldsymbol{I}^0\boldsymbol{B}^{-1})=\tilde{\boldsymbol{p}}_{K-1}\widetilde{\boldsymbol{R}}_0 \tag{3.19}$$

where $\widetilde{\boldsymbol{R}}_0=-\lambda\boldsymbol{I}^0\boldsymbol{B}^{-1}$. At the same time, equation (3.18) gives us through recursion:

$$\tilde{\boldsymbol{p}}_\nu=\tilde{\boldsymbol{p}}_{\nu-1}(-\lambda\boldsymbol{B}^{-1})=\tilde{\boldsymbol{p}}_{\nu-2}(-\lambda\boldsymbol{B}^{-1})^2=\ldots=\tilde{\boldsymbol{p}}_K(-\lambda\boldsymbol{B}^{-1})^{\nu-K}$$

Furthermore, we let $\widetilde{\boldsymbol{R}}=-\lambda\boldsymbol{B}^{-1}$. Hence, combining the last equation with (3.19), we have:

$$\tilde{\boldsymbol{p}}_\nu=\tilde{\boldsymbol{p}}_{K-1}\widetilde{\boldsymbol{R}}_0(-\lambda\boldsymbol{B}^{-1})^{\nu-K}=\tilde{\boldsymbol{p}}_{K-1}\widetilde{\boldsymbol{R}}_0\widetilde{\boldsymbol{R}}^{\nu-K}, \quad \nu=K+1,K+2,\ldots \tag{3.20}$$

The steady state probabilities of the individual states in the super states can be given from (3.19) and (3.20). This is feasible as $\tilde{\boldsymbol{p}}_{K-1}$ has been computed in the previous subsection, when dealing with the aggregated Markov chain.

In order to make the computation even clearer, the next subsection is dedicated to the two step solution procedure of the stationary distribution for $K=2$.

3.2.3 Solution of Steady State Probabilities for $K = 2$

Fortunately, as mentioned before, it is totally achievable to develop the explicit solution of $\{\boldsymbol{p}(\boldsymbol{n}), \boldsymbol{n} \in \mathcal{S}\}$ for $K = 2$ with the transition diagram given below. We simply follow the two step procedure we described before.

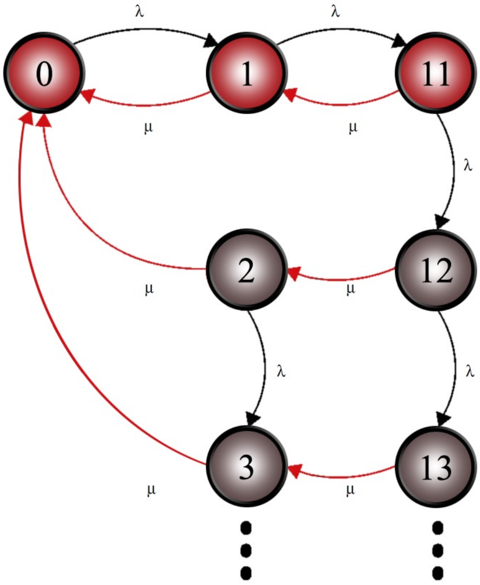


Figure 3.5 Transition Diagram of Ticket Queue with $K = 2$

The first step is to obtain the stationary distribution of the aggregated Markov chain. Thus, we have to group all the states with $n_L \geq K$ and $|\boldsymbol{n}| = L$ and to aggregate them into super states, for $K = 2$ and $L = 1, 2$. A closer look at Figure 3.5 will make it even more clear that we have now $K = 2$ super states, namely S_1 and S_2, and the rate diagram of the aggregated Markov chain is shown in Figure 3.6.

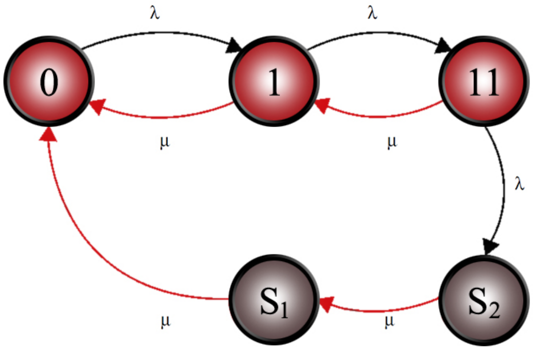


Figure 3.6 Transition Diagram of Aggregated Markov Chain with $K = 2$

According to the usual procedure, the *balance equations* and lastly the *normalization equation* are given by:

$$\lambda p(0) = \mu(p(1) + p(S_1)) \tag{3.21}$$

$$(\lambda + \mu)p(1) = \lambda p(0) + \mu p(11) \tag{3.22}$$

$$(\lambda + \mu)p(11) = \lambda p(1) \tag{3.23}$$

$$\mu p(S_1) = \mu p(S_2) \tag{3.24}$$

$$\mu p(S_2) = \lambda p(11) \tag{3.25}$$

$$\sum_{\boldsymbol{n}} p(\boldsymbol{n}) = 1 \tag{3.26}$$

Let $\rho = \lambda/\mu$. We begin solving from equation (3.23) and thus we have: $p(11) = (\lambda/(\lambda + \mu))p(1)$. Substituting $p(11)$ into (3.22), we get:

$$p(1) = \frac{\lambda(\lambda + \mu)}{\lambda^2 + \lambda\mu + \mu^2} p(0) = \rho\left(\frac{1 + \rho}{1 + \rho + \rho^2}\right) p(0) \tag{3.27}$$

Now, combining equation (3.27) with equation (3.21) gives:

$$p(S_1) = \rho\left(\frac{\rho^2}{1 + \rho + \rho^2}\right) p(0) \tag{3.28}$$

Equations (3.23) and (3.27) yield:

$$p(11) = \frac{\lambda}{\lambda + \mu} p(1) = \rho^2\left(\frac{1}{1 + \rho + \rho^2}\right) p(0) \tag{3.29}$$

Finally, combining either (3.24) with (3.28), or (3.25) with (3.29), we get:

$$p(S_2) = \rho^2 \Big(\frac{\rho}{1+\rho+\rho^2} \Big) p(0) \tag{3.30}$$

At this point, we have determined the probabilities $p(1)$, $p(11)$, $p(S_1)$ and $p(S_2)$ in terms of $p(0)$. So, we use the normalization equation to obtain $p(0)$ and then the stationary distribution of the aggregated Markov chain. Hence, we have:

$$\sum_{\boldsymbol{n}} p(\boldsymbol{n}) = p(0) + p(1) + p(11) + p(S_1) + p(S_2) = 1$$

$$p(0) = \frac{1}{1+\rho+\rho^2\big(\frac{1+\rho}{1+\rho+\rho^2}\big)} \tag{3.31}$$

Thus, we have the solution in closed form for $K = 2$, which means that step one has come to an end. We continue with the second step, which is to disaggregate the super states S_1 and S_2 and derive the probability for each state they contain.

In this case, we can simply obtain the balance equations for each state in S_1 or S_2. As we see in Figure 3.5, the states in S_1 have the form ν, for $\nu \geq 2$, while states in S_2 are denoted as (1ν), for $\nu \geq 2$. So, we get:

$$(\lambda+\mu)p(2) = \mu p(12), \quad \nu = 2 \tag{3.32}$$

$$(\lambda+\mu)p(\nu) = \lambda p(\nu-1) + \mu p(1\nu), \quad \nu > 2 \tag{3.33}$$

$$(\lambda+\mu)p(1\nu) = \lambda p(1(\nu-1)), \quad \nu \geq 2 \tag{3.34}$$

Firstly, we consider the steady state probability of of state $(1\nu) \in S_1$. Actually, equation (3.34) yields, through recursion:

$$p(1\nu) = \Big(\frac{\rho}{1+\rho}\Big) p(1(\nu-1)) = \Big(\frac{\rho}{1+\rho}\Big)^2 p(1(\nu-2)) = \ldots = \Big(\frac{\rho}{1+\rho}\Big)^{\nu-1} p(11)$$

As a matter of fact, we have already obtained $p(11)$ from the first step. Thus, we finally have:

$$p(1\nu) = \Big(\frac{\rho}{1+\rho}\Big)^{\nu-1} \frac{\rho^2\big(\frac{1}{1+\rho+\rho^2}\big)}{1+\rho+\rho^2\big(\frac{1+\rho}{1+\rho+\rho^2}\big)}, \quad \nu \geq 2 \tag{3.35}$$

Thence, we consider $p(\nu)$, $\nu \in S_1$. By combining (3.32) and (3.35), we can obtain $p(2)$. So, we get:

$$p(2) = \frac{1}{1+\rho} p(12) = \frac{\rho}{(1+\rho)^2} \frac{\rho^2 \left(\frac{1}{1+\rho+\rho^2}\right)}{1+\rho+\rho^2 \left(\frac{1+\rho}{1+\rho+\rho^2}\right)} \tag{3.36}$$

Also, for $\nu = 3$, we can get from (3.33), (3.35) and (3.36):

$$\begin{aligned} p(3) = \frac{\lambda}{\lambda+\mu} p(2) + \frac{\mu}{\lambda+\mu} p(13) &= \frac{\rho}{1+\rho} p(2) + \frac{\rho}{(1+\rho)^2} p(12) \\ &= \frac{\rho}{1+\rho} p(2) + \frac{\rho}{1+\rho} p(2) = \frac{2\rho}{1+\rho} p(2) \end{aligned} \tag{3.37}$$

Generally, we can show through *induction*, that:

$$p(\nu) = (\nu - 1) \left(\frac{\rho}{1+\rho} \right)^{\nu-1} p(2), \quad \nu \geq 3 \tag{3.38}$$

where, certainly, $p(2)$ is given by (3.36). At this point, the second step came to an end. It is obvious that this process works well for low values of K, as it has been shown in this case. The next section is devoted to some important computational benefits we have, given the stationary distribution of the Ticket Queue.

3.3 Key Performance Measures

The computation of the stationary distribution $\{\boldsymbol{p}(\boldsymbol{n}) : \boldsymbol{n} \in \mathcal{S}\}$, gives us the opportunity to obtain other significant measures, and through them we can get really important information about the Ticket Queue. In this section, we will discuss briefly the main performance measures we can get.

We surely can obtain the joint distribution of N and D, as well as their marginal distributions, which are given by:

$$P(N = 0) = p(0) \quad \textit{and} \quad P(N = L) = \sum_{\boldsymbol{n} \in K_L} p(\boldsymbol{n}), \quad L = 1, \ldots, K \tag{3.39}$$

$$P(D = 0) = p(0) \quad \textit{and} \quad P(D = d) = \sum_{\boldsymbol{n} : \|\boldsymbol{n}\| = d} p(\boldsymbol{n}), \quad d = 1, 2, \ldots \tag{3.40}$$

Naturally, for N and D, we can obtain the main performance measures such as their means. But, most importantly, we can obtain key system performance measures really useful for inferences, like the *customer balking probability*, P_b, and the *system utilization factor*, ρ_e, given bellow:

$$P_b = 1 - \sum_{\boldsymbol{n}:\|\boldsymbol{n}\|<K} p(\boldsymbol{n}) \tag{3.41}$$

$$\rho_e = \frac{\lambda}{\mu}(1 - P_b) = \rho(1 - P_b) \tag{3.42}$$

These quantities are particularly important from system's perspective. However, from the customers' side, the key information is mostly obtained by the *estimated queuing position given their ticket position*, $N|D = d$. The following equations give us the conditional distribution of $N|D = d$ and its mean respectively:

$$P(N = L|D = d) = \frac{P(N = L, D = d)}{P(D = d)} = \frac{\sum_{\boldsymbol{n}:\boldsymbol{n}\in K_L,\|\boldsymbol{n}\|=d} p(\boldsymbol{n})}{\sum_{\boldsymbol{n}:\|\boldsymbol{n}\|=d} p(\boldsymbol{n})}, \quad d = L, L{+}1, \ldots \tag{3.43}$$

$$E[N|D = d] = \sum_{L=0}^{min(d,K)} LP(N = L|D = d) = \frac{\sum_{L=0}^{min(d,K)} \sum_{\boldsymbol{n}\in K_L,\|\boldsymbol{n}\|=d} Lp(\boldsymbol{n})}{\sum_{\|\boldsymbol{n}\|=d} p(\boldsymbol{n})} \tag{3.44}$$

The equation (3.44) gives us another interesting result in order to improve the Ticket Queue for both the system and the customers. That is the *estimate of the waiting time W given the ticket position*:

$$E[W|D = d] = \frac{1}{\mu} E[N|D = d] \tag{3.45}$$

Furthermore, in case we need it, we can compute the distribution of $W|D = d$ by:

$$P(W \leq w|D = d) = \sum_{L=1}^{min(d,K)} P\Big(\sum_{l=1}^{L} Y_l \leq w\Big) P(N = L|D = d), \quad d \geq 1 \tag{3.46}$$

where Y_l are i.i.d exponential random variables with rate μ, and certainly, $P(N = L|D = d)$ is given by equation (3.43).

It seems that we managed to obtain everything we needed in order to study about the system improvement; however, there is still a case that we need to discuss. All the computational steps until this section can work numerically only for small values of K. Actually, we have managed to carry out the two step procedure just for $K \leq 9$. So, in the next section, we will explain why this happens and we will develop an approximation procedure in order to compute the stationary probabilities and everything else needed for values of K greater than 9.

3.4 The Approximation Procedure

In the last section, we mentioned that the two step procedure we have developed works for low values of K and has been numerically implemented for K up to 9. The reason which leads to inability of carrying out all the needed computations for higher values of K is the fact that *the cardinalities of K_L and T_ν grow exponentially large with K*. Thus, every set K_L or T_ν consists of such a huge number of states that we cannot handle the situation numerically.

Specifically, we have already mentioned that $|T_\nu| = 2^{K-1}$, $\nu \geq 1$, so for $K \geq 10$ we have already a prohibiting number of states. At this point, we will show what happens with $|K_L|$, the cardinality of K_L, in the proposition below.

Proposition 3.4.1. *The cardinality of K_L defined by*

$$K_L = \left\{ (n_1, \ldots, n_L) : \sum_{l=1}^{L-1} n_l < K, 1 \leq n_L \leq K-1, n_l \geq 1, \forall l \right\} \cup \{S_L\}$$

is given by:

$$|K_L| = (K-1) \cdot \binom{K-1}{L-1} + 1, \quad L = 1, \ldots, K \tag{3.47}$$

with all n_l and K positive integers.

Proof. Firstly, we note that we can partition set K_L into disjoint subsets, with the condition $n_L = \nu$, $\nu = 1, \ldots, K-1$. In that case, we write:

$$\begin{aligned} |K_L| &= \left| \bigcup_{\nu=1}^{K-1} \left\{ (n_1, \ldots, n_{L-1}, \nu) : \sum_{l=1}^{L-1} n_l < K, n_l \geq 1, \forall l \right\} \cup \{S_L\} \right| \\ &= (K-1) \left| \left\{ (n_1, \ldots, n_{L-1}, \nu) : \sum_{l=1}^{L-1} n_l < K, n_l \geq 1, \forall l \right\} \right| + 1 \end{aligned} \tag{3.48}$$

The last equality holds because $\{S_L\}$ consists only of S_L and all the disjoint subsets have the same cardinality; the same number of elements in other words. Now, we are looking for the cardinality of each set which appears in (3.48). Thus, we define:

$$\mathfrak{K}(K, L-1) = \left\{ (n_1, \ldots, n_{L-1}, \nu) : \sum_{l=1}^{L-1} n_l < K, n_l \geq 1, \forall l \right\} \tag{3.49}$$

In light of (3.47) and (3.48) it is sufficient to show that:

$$|\mathfrak{K}(K, L-1)| = \binom{K-1}{L-1}, \quad 1 \leq L \leq K \tag{3.50}$$

We use *induction* on $L-1$ to prove (3.50). Firstly, we show that the statement clearly holds for $L-1=1$ and for any $K \geq L$, because:

$$|\mathfrak{K}(K, 1)| = |\{n_1 : 1 \leq n_1 < K\}| = K - 1 = \binom{K-1}{1}$$

We continue with the inductive step; we assume that (3.47) holds for $L-2$, and we show it holds for $L-1$. Thus, with the same logic as above, we can partition set $\mathfrak{K}(K, L-1)$ into disjoint subsets. We denote the disjoint subsets according to $n_{L-1} = \nu$, and we have to determine all the possible values of $\nu \in \mathbb{N}$. We observe that the condition $n_l \geq 1 \ \forall l$ requires $\sum_{l=1}^{L-2} n_l \geq L-2$. Also, for $n_{L-1} = \nu$, we have $\nu \geq 1$ and clearly, it holds that $\sum_{l=1}^{L-1} n_l < K$. Hence, using the last two inequalities, we have:

$$\sum_{l=1}^{L-1} n_l = n_{L-1} + \sum_{l=1}^{L-2} n_l \geq \nu + L - 2$$

$$\nu \leq \sum_{l=1}^{L-1} n_l - L + 2 < K - L + 2$$

So, the condition for $n_{L-1} = \nu$ in the disjoint subsets is $1 \leq \nu \leq K - L + 1$. Thus, we can write:

$$|\mathfrak{K}(K, L-1)| = \left| \bigcup_{\nu=1}^{K-L+1} \left\{ (n_1, \ldots, n_{L-2}, \nu) : \sum_{l=1}^{L-2} n_l \geq K - \nu, n_l \geq 1, \forall l \right\} \right|$$
$$= \sum_{\nu=1}^{K-L+1} |\mathfrak{K}(K-\nu, L-2)| = \sum_{\nu=1}^{K-L+1} \binom{K-\nu-1}{L-2}$$

The last step was obtained due to hypothesis for $L-2$. Now, by adding and subtracting $L-2$, we can make a useful change of variables like this:

$$|\mathfrak{K}(K, L{-}1)| = \sum_{\nu=1}^{K-L+1} \binom{K-\nu-1}{L-2} = \sum_{\nu=1}^{K-L+1} \binom{(K-\nu-1-L+2)+(L-2)}{L-2}$$
$$= \sum_{y=1}^{K-L} \binom{y+(L-2)}{L-2}, \quad \text{where } y = K - L + 1 - \nu$$

Finally, through the use of some formulas especially seen in combinatorics, we have that:

$$\sum_{y=1}^{K-L} \binom{y+(L-2)}{L-2} = \binom{K-1}{L-1}$$

which yields (3.50) and consequently proves (3.47). □

Proposition 3.4.1 gives us the opportunity to compute the total number of states in the aggregated Markov chain. In fact, we have:

$$|\mathcal{S}^*| = |K_0| + \sum_{L=1}^{K} |K_L| = (K-1) \sum_{L=1}^{K} \binom{K-1}{L-1} + K + 1 = (K-1) \cdot 2^{K-1} + K + 1$$

Thus, the aggregated Markov chain has a huge number of states as K increases, and consequently, there is a threshold where for values equal or greater to it, we cannot carry out the computation with the two step procedure. As mentioned previously, that threshold is $K = 10$ for an ordinary

PC, where we have $max_L\{|K_L|\} = 1,135$, $|T_\nu| = 512$ and $|\mathcal{S}^*| = 4,619$. The exponential growth can be seen better if we compute these values for a specific $K > 10$, for example $K = 20$, where $max_L\{|K_L|\} = 1,755,183$, $|T_\nu| = 524,288$ and $|\mathcal{S}^*| = 9,961,493$. These numbers are insanely huge so we should handle the situation with another way. We develop an approximation procedure for the computations of the Ticket Queue. Thankfully, we will see in the following that this approximation is quite accurate; hence we can use it for values $K \geq 10$ without any problem.

The fundamental idea for the approximation procedure is to intelligently eliminate the states with negligible probabilities because we need to reduce the sizes of K_L and T_ν for large K. In order to achieve something like that, we should see the Ticket Queue differently. We observe that, if we separate the customers into two groups, the joining and the balking customers, and we only keep the total number of customers in each group, the number of states in our Markov chain will be dramatically reduced. Hence, we can consider a modification of the Ticket Queue, hereafter the *modified Ticket Queue*, where we have two separated queues, one for joining customers and another for balking customers (called joining customers' queue and balking customers' queue respectively). The description of this modification is the following; we have a higher priority of service to the joining customers' queue. So, an arriving customer draws a ticket with a number and decides if s/he is going to join or balk with the same criteria as before, because her/his ticket position corresponds to the total number of customers in both queues and s/he cannot observe the real queue length. If the customer joins, s/he is placed at the end of the joining customers' queue, and if s/he balks, s/he is placed at the end of the balking customers' queue. Exactly because we have a higher priority of service to the joining customers' queue, when this queue empties, all the customers in the balking customers' queue are released simultaneously.

Before we define the modified Ticket Queue, we should note that we expect the two systems (the *original* Ticket Queue and the modified one) to have similar stochastic behavior. This is of our concern, as there is no point in the modification then. The two systems would have similar stochastic behavior, if the balking customers in the original Ticket Queue intermix rarely with the joining customers. Meaning that, when balking customers are present, they are most likely at the end of the queue and not intermixed with joining customers. If this happens to the original Ticket Queue, it appears closer to the modified Ticket Queue, where all the balking customers are placed at the end of the queue by default (as the have lower priority from the joining customers).

We now model the modified Ticket Queue; to represent the states we need only an ordered pair, let (L, ν), where L with $0 \leq L \leq K$ is the length of the joining customers' queue and $\nu \geq 0$ is that of the balking customers' queue. The transitions here are made this way; with rate λ, a state (L, ν) goes to $(L+1, \nu)$ for $L+\nu < K$, otherwise goes to state $(L, \nu+1)$, $0 \leq L \leq K$, $\nu \geq 0$. Furthermore, with rate μ, a state (L, ν) goes to $(L-1, \nu)$ for $2 \leq L \leq K$ and for $L = 1$, $(1, \nu)$ goes to $(0, 0)$,$\nu \geq 0$. $(0, 0)$ represents the empty system, with no staying and balking customers at all. As an example, Figure 3.7 below shows the transition diagram of the modified Ticket Queue for $K = 4$.

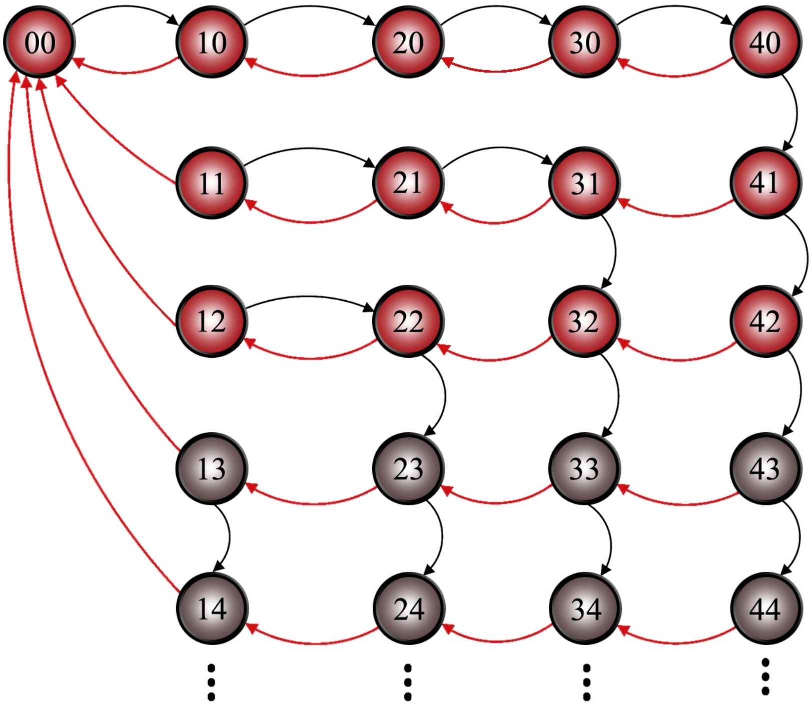


Figure 3.7 Transition Diagram of Modified Ticket Queue with $K = 4$

As a matter of fact, the solution procedure for the modified Ticket Queue is similar to the two step procedure we described before, but with a drastically reduced state space. So, we have to follow two steps in order to obtain the stationary distribution. The first step is to aggregate all states with $1 \leq L \leq K$ and $\nu \geq K-1$ into super states and model the aggregated-modified Ticket Queue as a QBD process; in the second step we disaggregate the super states and we compute the probabilities of the individual states in the super states. The description of the solution here is brief as we have seen the similar one in a more detailed way.

To begin with step 1, we redefine the super states for the aggregated-modified Ticket Queue as:

$$S_L = \{(L, \nu) : \nu \geq K - 1\}, \quad L = 1, \dots, K$$

From the definition of S_L, we can see that every arriving customer in a state in S_L would balk because $L + \nu \geq 1 + (K - 1) = K$. These are the states in the gray zone, in Figure 3.7, for $K = 4$. We group again the states that have L joining customers as K_L, so we define:

$$K_0 = \{(0, 0)\}$$
$$K_L = \{\boldsymbol{n} = (L, \nu) : 0 \leq \nu \leq K - 2\} \cup \{S_L\}, \quad L = 1, \dots, K$$

We also observe that $|K_L| = K,\ L \geq 1$. Hence, the state space of the aggregated-modified Ticket Queue is given by:

$$\boldsymbol{\mathcal{S}}^m = \{\boldsymbol{n} = (L, \nu) : 0 \leq L \leq K, 0 \leq \nu \leq K - 2\} \cup \{S_1, \dots, S_K\}$$

We can easily now see that the cardinality of $\boldsymbol{\mathcal{S}}^m$ is $|\boldsymbol{\mathcal{S}}^m| = K^2 + 1$, which increases quadratically instead of exponentially in K. For example, a simple comparison between the two models when $K = 10$ is that $|\boldsymbol{\mathcal{S}}^*| = 4,619$ while $|\boldsymbol{\mathcal{S}}^m| = 101$; and when $K = 20$, $|\boldsymbol{\mathcal{S}}^*| = 9,961,491$ and $|\boldsymbol{\mathcal{S}}^m| = 401$. These results indicate that the modified Ticket Queue has a much more reduced state space and this really simplifies the computation effort.

Continuing with the example of $K = 4$, we also represent the transition diagram of the modified Ticket Queue with super states in Figure 3.8. The diagram helps us to realize the way we will compute the stationary distribution schematically. Let $\{\boldsymbol{p}^m(\boldsymbol{n}) : \boldsymbol{n} \in \boldsymbol{\mathcal{S}}^m\}$ be the stationary distribution of the aggregated-modified Ticket Queue. We again treat $\{K_0, \dots, K_K\}$ as blocks and L as block levels, $0 \leq L \leq K$. Otherwise, we can describe L, with $0 \leq L \leq K$ as levels and note ν as phases. Both notations lead to the consideration of the aggregated-modified Ticket Queue as a QBD process, with block-partitioned infinitesimal generator $\boldsymbol{Q}$ denoted by:

$$
\boldsymbol{Q} = \begin{bmatrix}
\boldsymbol{-\lambda} & \boldsymbol{A_{01}} & & & & & \\
\boldsymbol{\mu e'_K} & \boldsymbol{A_{11}} & \boldsymbol{A_{12}} & & & & \\
 & \boldsymbol{\mu I} & \boldsymbol{A_{22}} & \boldsymbol{A_{23}} & & & \\
 & & \ddots & \ddots & \ddots & & \\
 & & & & \boldsymbol{\mu I} & \boldsymbol{A_{K-1,K-1}} & \boldsymbol{A_{K-1,K}} \\
 & & & & & \boldsymbol{\mu I} & \boldsymbol{A_{K,K}}
\end{bmatrix} \tag{3.51}
$$

Note that $\boldsymbol{A}_{01} = (\lambda, 0, \ldots, 0)$ is a matrix of dimension $1 \times K$, $\boldsymbol{A}_{L,L+1}$ is a $K \times K$ diagonal matrix with the first $K - L$ main diagonal entries equal to λ, $L = 1, \ldots, K - 1$ and zero otherwise; and finally, the matrices named as $\boldsymbol{A}_{L,L}$, $L = 1, \ldots, K$, which are upper triangular and have the following type:

$$
\boldsymbol{A}_{L,L} = \begin{bmatrix}
-(\lambda+\mu) & 0 & & & & & \\
 & \ddots & \ddots & & & & \\
 & & -(\lambda+\mu) & 0 & & & \\
 & & & -(\lambda+\mu) & \lambda & & \\
 & & & & \ddots & \ddots & \\
 & & & & & -(\lambda+\mu) & \lambda \\
 & & & & & & -\mu
\end{bmatrix} \tag{3.52}
$$

Now let $\boldsymbol{p}_0^m = p^m(0,0)$, and $\boldsymbol{p}_L^m$ be the probability vector corresponding to set K_L, $L = 1, \ldots, K$. In fact, we follow the same process as in (3.4)-(3.14), with some differentiation. We obtain again that:

$$
\boldsymbol{p}_0^m = \frac{1}{1 + \sum_{L=1}^{K} \boldsymbol{R}_L \boldsymbol{e}'_K} \tag{3.53}
$$

$$
\boldsymbol{p}_L^m = \frac{\boldsymbol{R}_L}{1 + \sum_{L=1}^{K} \boldsymbol{R}_L \boldsymbol{e}'_K}, \quad L = 1, \ldots, K \tag{3.54}
$$

where we have a difference in the notation of $\boldsymbol{R}_L$. So, we redefine $\boldsymbol{R}_L$ as:

$$
\boldsymbol{R}_L = \prod_{l=1}^{L} \boldsymbol{R}_{l-1,l} \tag{3.55}
$$

and then, with the new notation, $\boldsymbol{R}_{L-1,L}$ can be derived recursively by:

$$
\boldsymbol{R}_{K-1,K} = -\mu(\boldsymbol{A}_{K,K})^{-1} \tag{3.56}
$$

$$
\boldsymbol{R}_{L-1,L} = -\mu(\boldsymbol{A}_{L,L} - \mu \boldsymbol{R}_{L,L+1})^{-1}, \quad L = 1, \ldots, K-1 \tag{3.57}
$$

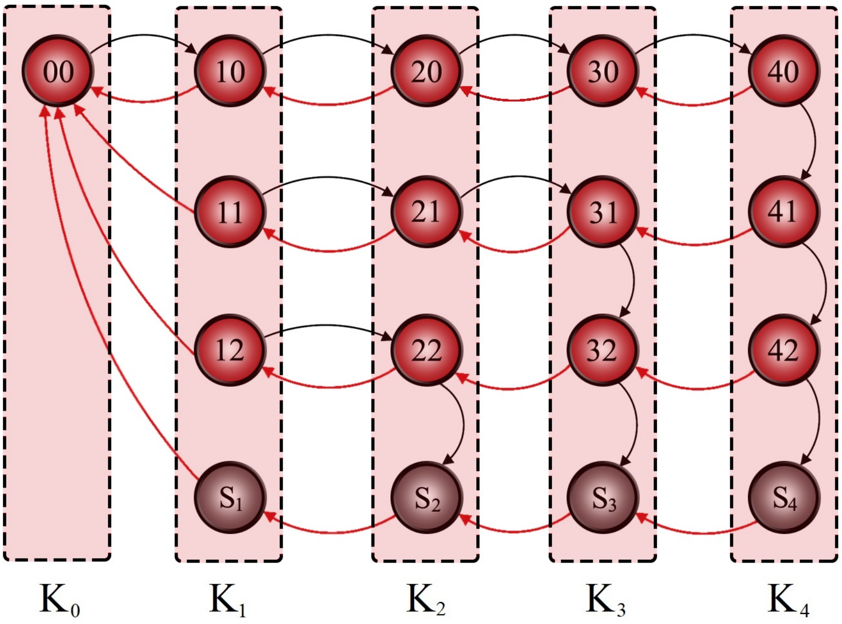


Figure 3.8 Transition Diagram of Aggregated-Modified Ticket Queue with $K = 4$

Thus, we have shown briefly the results we get after carrying out the first step. For the step 2, we need to compute the probabilities of the individual states in the super states, a process similar to the one described in subsection 3.2.2. Hence, we firstly disaggregate the super states into individual states, and we repartition the states of the modified Ticket Queue by ν which is the length of the balking customers' queue. This way, we redefine the sets T_ν as the sets of states that have ν balking customers by:

$$T_0 = \{(0,0) \cup (L,0) : L = 1,\ldots,K\}$$
$$T_\nu = \{(L,\nu) : L = 1,\ldots,K\}, \quad n = 1,2,\ldots$$

Naturally, the states in T_ν are ordered increasingly with L. Actually, each row of states in Figure 3.7 represents a state set T_ν, for $\nu \geq 0$. As for $\nu \geq 1$, note that $|T_\nu| = K$. We also observe that, the steady state probabilities of states in sets $T_0,\ldots,T_{K-2}$ are already known via (3.54).

The infinitesimal generator $\widetilde{\boldsymbol{Q}}$ in this case is similar to (3.16). So, the portion of $\widetilde{\boldsymbol{Q}}$ needed for the computation of the steady state probabilities of $\boldsymbol{n} \in T_\nu$, $\nu \geq K-1$ is given by:

$$\widetilde{\boldsymbol{Q}} = \begin{bmatrix} \boldsymbol{B} & \lambda \boldsymbol{I}^0 & & \\ & \boldsymbol{B} & \lambda \boldsymbol{I} & \\ & & \boldsymbol{B} & \lambda \boldsymbol{I} \\ & & & \end{bmatrix} \tag{3.58}$$

The differences are that now, $\boldsymbol{I}$ is the $K \times K$ identity matrix, $\boldsymbol{I}^0$ is the same as $\boldsymbol{I}$ except the fact that the first entry on its main diagonal is zero; and, finally, $\boldsymbol{B}$ is a $K \times K$ lower triangular matrix given by:

$$\boldsymbol{B} = \begin{bmatrix} -(\lambda+\mu) & & & & \\ \mu & -(\lambda+\mu) & & & \\ & & \ddots & & \\ & & \mu & -(\lambda+\mu) & \\ & & & \mu & -(\lambda+\mu) \end{bmatrix} \tag{3.59}$$

We denote as $\tilde{\boldsymbol{p}}_\nu^m$ as the probability vector corresponding to set T_ν, with $\nu \geq K-1$. We follow the similar process which led to (3.19)-(3.20), and we obtain:

$$\tilde{\boldsymbol{p}}_\nu^m = \tilde{\boldsymbol{p}}_{K-2}^m(-\lambda \boldsymbol{I}^0 \boldsymbol{B}^{-1})(-\lambda \boldsymbol{B}^{-1})^{\nu-K+1} = \tilde{\boldsymbol{p}}_{K-2}^m \widetilde{\boldsymbol{R}}^0 \widetilde{\boldsymbol{R}}^{\nu-K+1}, \quad \nu = K-1, K, \ldots \tag{3.60}$$

where $\widetilde{\boldsymbol{R}}^0 = -\lambda \boldsymbol{I}^0 \boldsymbol{B}^{-1}$, $\widetilde{\boldsymbol{R}} = -\lambda \boldsymbol{B}^{-1}$ and $\boldsymbol{B}$ is given by (3.59). Consequently, we can compute $\tilde{\boldsymbol{p}}_\nu^m$ for $\nu \geq K-1$ through (3.60), using also (3.54) in order to indicate $\tilde{\boldsymbol{p}}_{K-2}^m$. Thus, the second step is over and the steady state probabilities of the modified Ticket Queue have been obtained. Note also, that given the distribution $\{\boldsymbol{p}^m(\boldsymbol{n})\}$, we can compute the same way the performance measures we defined back in section 3.3, this time for the modified queue.

We continue this section by comparing the behavior of the original and the modified Ticket Queue. The results are based in our primary source, the work of Xu, Gao and Ou (2007), and are given through computations and simulation. The idea is that we compare various measures in a total of 130 different system parameter settings which are created from combinations of 13 different balking limits ($K = 2, 3, \ldots, 9, 10, 20, 30, 40, 50$) and 10 different traffic intensity levels ($\rho = \lambda/\mu = 0.1, 0.2, \ldots, 0.9, 1.0$). Note that, although the system is stable for any $\rho > 0$, we only test cases where $0 < \rho \leq 1$, because there they represent the most common system situations.

Furthermore, we need to highlight how actually the comparison between the original and the modified Ticket Queue works for the various values of K. For the original Ticket Queue, we carry out the computations up to $K = 9$ and for $K \geq 10$ we simulate the results as the computation is not possible due to the exponential growth of the state space. For the modified Ticket Queue we can compute the stationary distribution for all the tested values of K; so the results can be obtained completely through numerical computations. Hence, in the following, the compared measures are given just as we described before.

We should note that the comparisons are virtually shown with plots. Therefore, to quantify how close are two plotted lines, we define the following measures:

- MAD (Maximum Absolute Difference), noted as the maximum of all the absolute differences
- TAD (Total Absolute Difference), which is the sum of all the absolute differences

Table 3.1 below shows interesting statistics on MAD and TAD, on a numerical comparison of N and N^m, and D and D^m; to begin with, it reports their values of the 10 worst scenarios of TAD over the 130 scenarios tested. Additionally, the last row of Table 3.1 shows the averages of MAD and TAD over 130 cases. In brief, the inferences made from Table 3.1 are the following; the approximation is excellent, and even though it may worsen with tremendously heavy traffic intensity (e.g. $\rho = 1$), the worst values of TAD and MAD (about 0.07 and 0.0059 respectively) indicate that the modified Ticket Queue generates a good approximation in every case. Generally, the statistics obtained in Table 3.1 support the fact that the plotted lines of the distributions of N and N^m, as well as for those of D and D^m, are definitely very close over a wide range of parameter values.

At this point, we begin with the plot comparisons of different measures in order to investigate the behavior of the two tested queues; firstly we will show the plots between the distributions of N and N^m in Figure 3.9 and the distributions of D and D^m in Figure 3.10 for two specific cases. These cases are $K = 9$ or $K = 50$ with various values of ρ. As expected from the numerical statistics observed in Table 3.1, the original and the modified Ticket Queue have extremely close behavior relatively with the distributions of N, N^m, D and D^m.

		$P[N]$		$P[D]$	
K	ρ	TAD	MAD	TAD	MAD
6	1	0.02297	0.00514	0.01680	0.00514
7	1	0.02556	0.00542	0.02037	0.00542
8	1	0.03041	0.00592	0.02076	0.00592
9	0.9	0.02151	0.00445	0.02238	0.00449
9	1	0.03292	0.00588	0.02397	0.00580
10	1	0.03434	0.00591	0.02544	0.00591
20	1	0.04667	0.00534	0.03380	0.00528
30	1	0.06411	0.00453	0.04265	0.00451
40	1	0.07508	0.00402	0.05729	0.00382
50	1	0.06489	0.00307	0.04699	0.00303
Max. over 130		0.07508	0.00592	0.05729	0.00592
Average over 130		0.00812	0.00141	0.00774	0.00179

Table 3.1 Comparisons of N and N^m and D and D^m: 10 Worst Cases of TAD Over 130

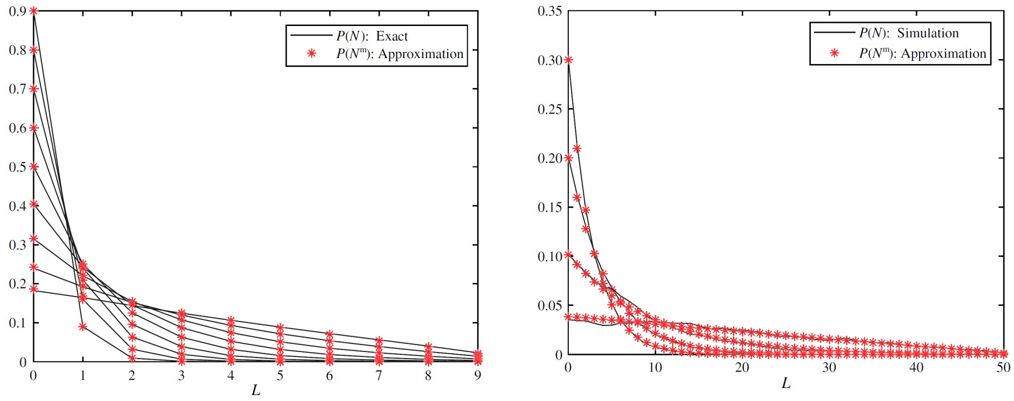


Figure 3.9 Comparisons of Distributions of N and N^m

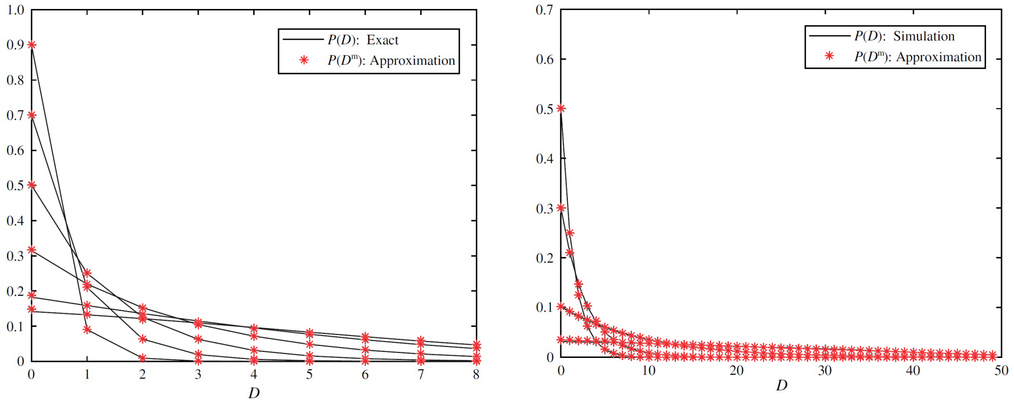


Figure 3.10 Comparisons of Distributions of D and D^m

The next measures we are going to compare are the balking probabilities of the original and the modified Ticket Queue, noted as P_b and P_b^m. The partial results for the 130 scenarios are featured in Figure 3.11.

Finally, we compare $E[N]$ and $E[N^m]$ for various values of K, ρ. In this case, we define the next measure in order to obtain the desired the results:

- APE (Absolute Percentage Errors), which can be computed as:
$$\text{APE}= 100\% \times |E[N^m] - E[N]|/E[N]$$

Note that APE is a meaningful measure to compare the relative difference of two quantities, but has the disadvantage of the instability when the denominator is really close to 0; hence it can be used properly when the probability or distribution comparisons do not approach 0 rapidly. Here, APE can be used and therefore, we have the comparisons shown in Figure 3.12.

In conclusion, we shall say that the numerical evidence we have been provided with, truly indicate the fact that the approximation of the original Ticket Queue by the modified one is of high quality in all terms. This explains our decision to define the modified Ticket Queue, and thus provide solutions despite the problem of exponential growth that appeared in the original problem.

We should also note that it is reasonable to guess that N^m is a stochastic lower bound of N. This conjecture can be confirmed when $K = 1, \ldots, 9$ and $\rho = 0.1, \ldots, 1.0$. Furthermore, the computation results show that the difference $P(N \geq \nu) - P(N^m \geq \nu)$ has a tendency to be increased in K and ρ. However, we have only computational evidence for this guess, as it was not possible for us to prove it using some formal probabilistic argument.

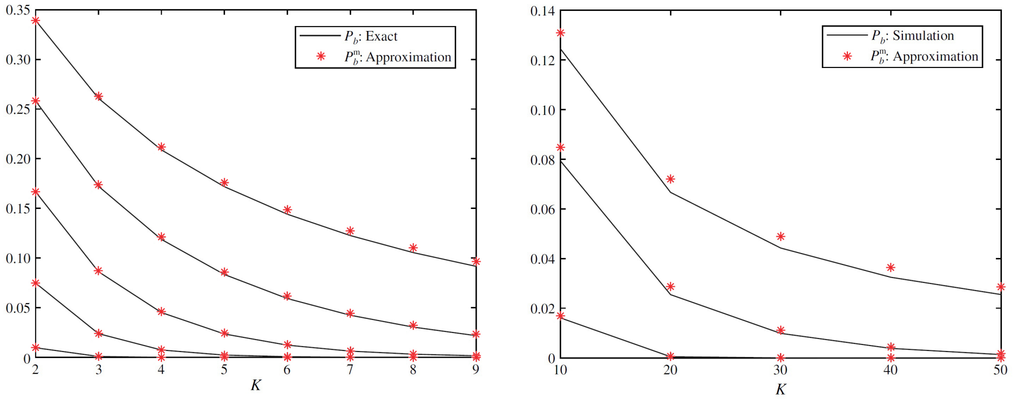


Figure 3.11 Comparisons of Balking Probabilities P_b and P_b^m

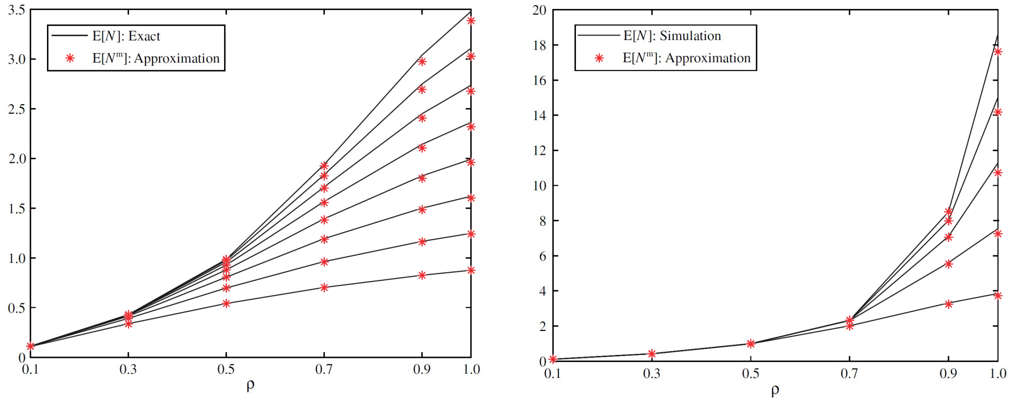


Figure 3.12 Comparisons of $E[N]$ and $E[N^m]$

Before the end of this section, we will report an interesting discovery that has been done. We have found that there is an approximately linear relationship between the queuing position and the ticket position, when K (the customers' tolerance limit) is increased. So, surprisingly, we saw that $E[N^m|D^m = cK]$, with c constant, increases approximately linearly in K.

The procedure we followed in order to reach this discovery begins with the computation of $E[N^m|D^m = cK]$. We observe in Figure 3.13 that, for the modified queue, the event of $\{D^m = d\}$ corresponds to the entries on the dth northeast to southwest diagonal. Remember that D^m, the ticket position, is computed by $D^m = L + \nu$ for every state. Hence, using the previous information, we have:

$$P(D^m = d) = \sum_{l=1}^{min(d,K)} \boldsymbol{p}^m(l, d-l) \tag{3.61}$$

$$E[N^m|D^m = K] = \frac{\sum_{l=1}^{min(d,K)} l \cdot \boldsymbol{p}^m(l, d-l)}{P(D^m = d)} \tag{3.62}$$

The next step is to use equations (3.61) and (3.62) for computing the values of $E[N^m|D^m = cK]$ for $c = 0.5, 0.6, \ldots, 1.0$ at different values of ρ and K. In Figure 3.14 we see the approximate linear relationship of $E[N^m|D^m = cK]$ in K for the previous noted values of c and $\rho = 0.9$. This also holds for any other ρ value tested.

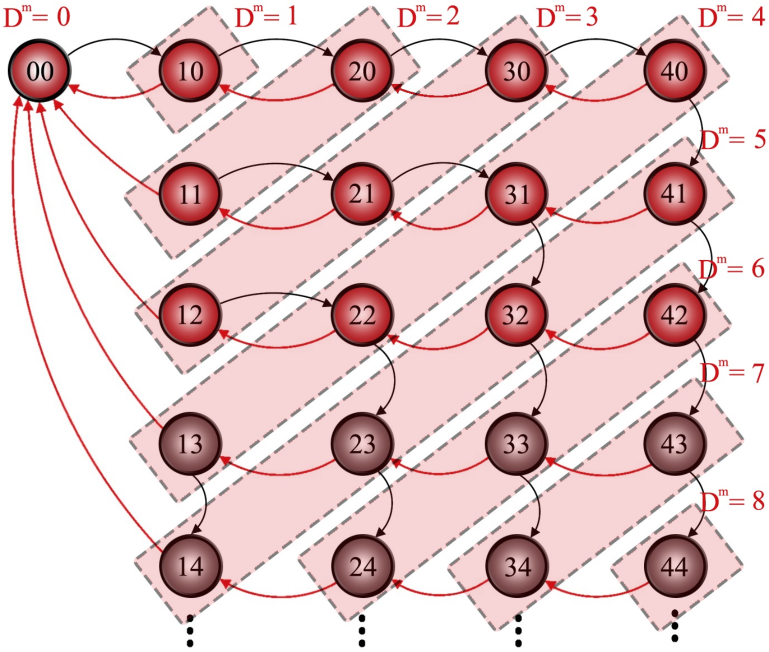


Figure 3.13 The Event of $\{D^m = d\}$ in Modified Ticket Queue

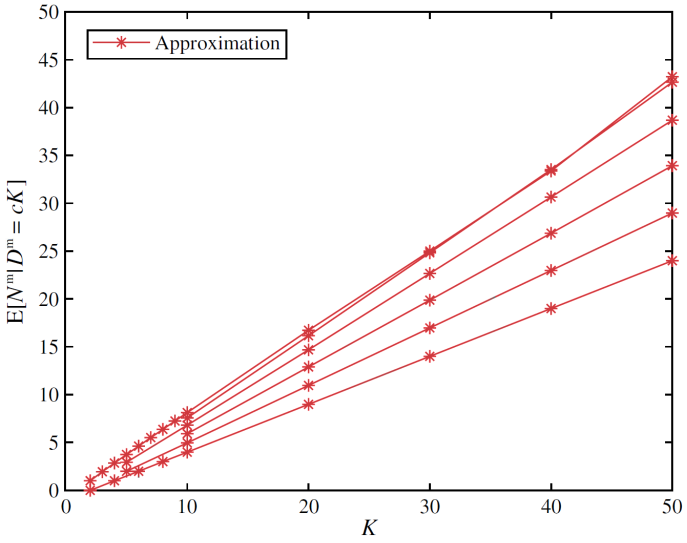


Figure 3.14 $E[N^m|D^m = cK]$ with $c = 0.5, 0.6, \ldots, 1.0$ and $\rho = 0.9$

In addition, as we wanted to establish the validity of the linear relationship, we have simulated $E[N|D = cK]$ for the original Ticket Queue, using the same c, K and ρ values. In Figure 3.15 we can see that thee approximate linear relationship is also confirmed for the original Ticket Queue.

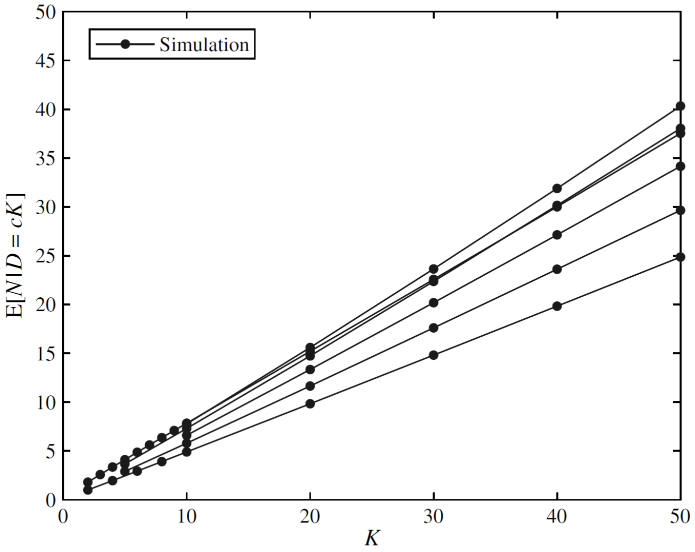


Figure 3.15 $E[N|D = cK]$ with $c = 0.5, 0.6, \ldots, 1.0$ and $\rho = 0.9$

In conclusion, we also observe the following inferences:

- ○ $E[N|D = cK]$ is decreasing in c
- ○ $E[N|D = cK] \approx cK$ for sufficiently small c
- ○ $E[N|D = cK]$ is decreasing in ρ

The second observation is made because when the ticket position is small there is a big possibility that almost all customers are present. The third one comes from the fact that a higher traffic intensity implies a higher percentage of balking customers and that means lower $E[N|D = cK]$.

At this point, the analysis of Ticket Queue with balking customers has been completed. In the next section, we will use this analysis and we will examine more issues in order to offer improvement suggestions for both the system and the customers in this kind of Ticket Queue.

3.5 Service Improvement

As we have already illustrated, although the Ticket Queue offers several advantages over the *physical queue* (the corresponding queue without ticket issuance; in this case M/M/1/K queue), there is a significant drawback for both management and customers. That is, no one of them has complete information about the number of customers in the system. The result is that the customers overestimate their waiting time and may abandon the queue without entering; so we have a higher balking rate and the system tends to be less productive exactly for this reason. Then, is there a way to improve the performance of the Ticket Queue in order to benefit both the customers and the system? This section is dedicated to the answer of this question.

Recall our assumption that customers are naive and they balk if their ticket position D is greater or equal to their patience limit K. Our goal is to use the comparison of this Ticket Queue and the physical queue (M/M/1/K) for obtaining results which will lead to a new, improved model, where we will virtually eliminate the performance gap between the physical and ticket queues. Thus, firstly the comparison, and secondly the improved Ticket Queue, are the subjects of our next subsections.

3.5.1 Comparison of Ticket Queue and Physical Queue

Our first step is to show that, compared with a physical queue, a Ticket Queue is “less crowded”, but still has a higher balking probability. In other words, the Ticket Queue, compared with the physical queue, even though often has less customers waiting for service, the arriving customers tend to balk more frequently because they perceive the ticket position as the real number of joining customers in the system.

Denote N, N^m and N^p as the stationary numbers of joining customers in the ticket, modified and physical queues respectively. The same goes for W, W^m and W^p, which are referred as the stationary waiting times of joining customers in each queue. Finally, we can indicate that P_b, P_b^m and P_b^p are the balking probabilities in the ticket, modified and physical queues respectively. Using this notation, we introduce the next proposition which shows exactly this fact; the modified and the original Ticket Queue are less crowded but they still have a higher balking probability than the physical queue.

Proposition 3.5.1. *Consider the ticket, modified ticket and physical queues which have the same arrival rate λ, service rate μ, and balking limit K. Then:*

- $P(N \leq n) \geq P(N^p \leq n)$ *and* $P(N^m \leq n) \geq P(N^p \leq n)$
- *As a consequence,* $E[N] \leq E[N^p]$ *and* $E[N^m] \leq E[N^p]$
- $P_b \geq P_b^p$ *and* $P_b^m \geq P_b^p$

Proof. The proof is omitted. The proposition can be shown using *coupling*. The reader is referred to the Appendix C of Xu, Gao and Ou (2007) for more information. □

As for the waiting time W, we have that $W = \sum_{l=1}^{N} Y_l$, where Y_l, $l \geq 1$, are i.i.d. exponential random variables independent of N. The same goes for W^m and W^p with $W^m = \sum_{l=1}^{N^m} Y_l$ and $W^p = \sum_{l=1}^{N^p} Y_l$ respectively. Hence, Proposition 3.5.1 implies:

$$P(W \leq w) \geq P(W^p \leq w) \quad \text{and} \quad P(W^m \leq w) \geq P(W^p \leq w) \tag{3.63}$$

$$E[W] \leq E[W^p] \quad \text{and} \quad E[W^m] \leq E[W^p] \tag{3.64}$$

It is clear now that Proposition 3.5.1 and (3.63), (3.64), can give us easily computable bounds for the key performance measures of the Ticket Queue, whose analytical solutions cannot be obtained.

We can surmise that the probability of abandonment in the system is a key measure that is essential for both the customers and management, as it is important for the two sides to lower it. In order to understand the effect of partial information on abandonments, we compute and compare the values of TAD and APE of the balking probabilities between the modified Ticket Queue and the physical queue, for all values of $0 < \rho \leq 1$ and $2 \leq K \leq 50$. At this point, we remind the meaning of the two measurements and we give their formula:

- TAD (Total Absolute Difference), defined as $P_b^m - P_b^p$
- APE (Absolute Percentage Errors), noted as $100\% \times (P_b^m - P_b^p)/P_b^p$

Thus, we now present our findings for TAD as a function of K and ρ in Figure 3.16, while Figure 3.17 shows the corresponding results of APE. Thereafter, we discuss briefly the management insights that have been obtained from our qualitative and quantitative results.

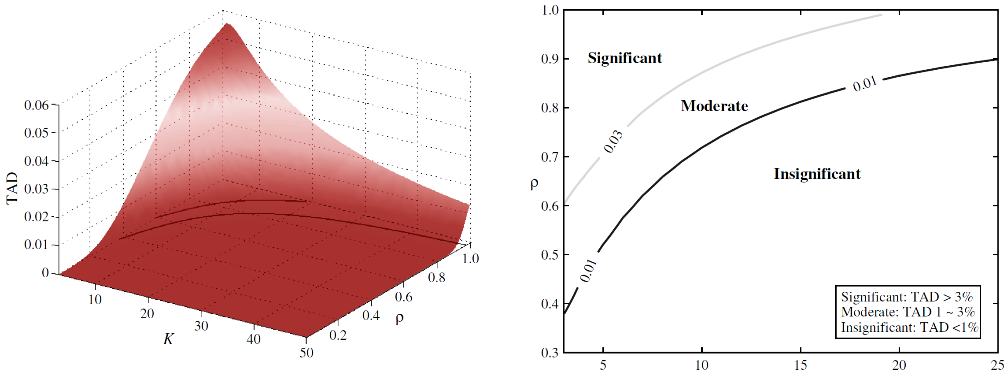


Figure 3.16 On the left: Balking Probability Difference (TAD) of Ticket Queue and M/M/1/K. On the Right: Partition of Parameter Space (K, ρ) with Different TAD Ranges.

As seen if Figure 3.16, we have partitioned the parameter space (K, ρ) into three, non-overlapping regions: *significant*, *moderate* and *insignificant*. The partition has been done in order to identify the parameter settings under which the two systems (ticket and physical queue) give different ranges of TAD. So, the significant region contains the (K, ρ) values that give TAD measurements at least 3% and up to 6%; in the moderate region, TAD varies from 1% to 3%; and finally, in the insignificant region, TAD is less than 1%.

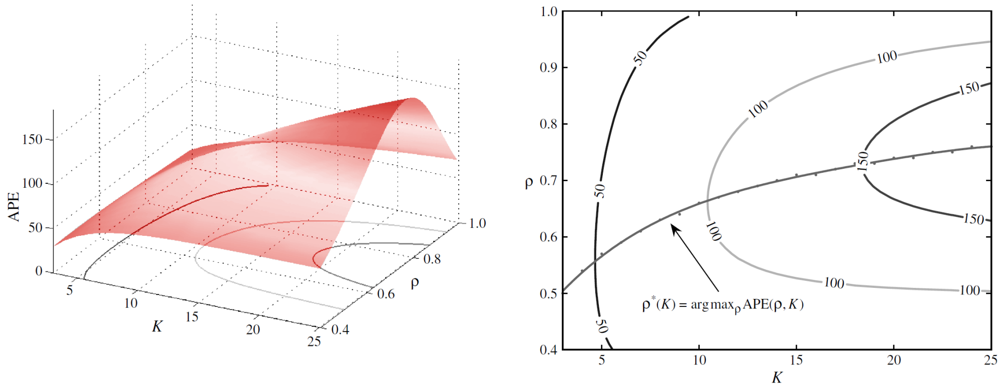


Figure 3.17 On the left: Percentage Balking Probability Difference (APE) of Ticket Queue and M/M/1/K. On the Right: Partition of Parameter Space (K, ρ) with Different APE Ranges.

We shall introduce our findings and be led to an idea of improvement:

- Firstly, we can presume from Proposition 3.5.1, that although the Ticket Queue is actually less crowed (in stochastic sense) than the physical queue, it appears busier than the later. Thus, the ticket position is the visible information and the actual position in queue is the hidden, causing naive behavior from the customer's perspective. This is a fact which can be harmful from both customers and the business. For business, it means lost sales even there was enough capacity in the system to provide service. For customers, it means unsatisfied demand even though they can tolerate the wait. So, there comes the idea to communicate to the customers the hidden information of the actual queue in a clear, quantifiable way, which will help them realize that their waiting is not so intolerable as they think.

- As seen in Figure 3.16, TAD decreases in K for fixed ρ, and also increases in ρ for fixed K, meaning that the ticket and physical queues show the most significant balking probability differences (up to 6%) for small K and large ρ values. This difference in behavior between the two queues measured by TAD is completely reasonable, as low K values mean that the customers are impatient and large ρ values that traffic intensity is high. Therefore, we should discover effective strategies for maintaining service performance especially for the Ticket Queue with impatient customers and moderate or heavy traffic. Furthermore, we observe in Figure 3.16 that as K increases and/or ρ decreases, TAD gradually decreases and eventually vanishes (for sufficiently large K and/or small ρ). Hence, these insignificant-difference cases indicate a Ticket Queue system which can offer customers a comfortable queuing environment keeping the same service level as that of a physical queue.

- One more well known thing to point out is the convexity of the balking probability of M/M/1/K queue, P_b^p, which is increasing in ρ and decreasing in K. That is, the physical system's performance worsens rapidly when K becomes smaller and ρ larger. Our findings can confirm the same pattern for P_b^m and TAD= $P_b^m - P_b^p$. So, both P_b^m and TAD deteriorate at an accelerating speed as the balking limit K decreases and the traffic intensity ρ increases. Thus, in this case, the Ticket Queue tends to exacerbate the already poor service of the physical queue. Numerically, we have carried out that in the moderate region the M/M/1/k system has an average balking probability of 5% which is worsen to 7% in the Ticket Queue. Also, the corresponding results for the significant region is that the average balking probability of the physical queue is 14.3%, which is rather high, but its performance deteriorates to 18.7% in the Ticket Queue. As a consequence we really need an effective management of the Ticket Queue when its parameters fall into the moderate or significant region.

- It can be observed that the Ticket Queue technology is applied mainly in systems with heavy traffic, such as bank offices. This is not a coincidence as it finally works better in systems with this characteristic according to our results. Additionally, note that TAD should be the primary measure for our inferences, supplemented by APE for quantifying the stochastic difference between the two systems.

Thus far, we came up with computational and numerical evidence regarding the comparison of the ticket and physical queues. The results help us to determine a Ticket Queue model, which mends the difference between the balking probabilities of the two models and takes queuing with use of tickets to another level. This model is represented in the next subsection.

3.5.2 The Ticket-Plus Queue

As seen previously, a larger performance gap takes place between the ticket and the physical queues when the customer is impatient and/or the traffic intensity is high. In order to reduce this difference, we propose a new model, namely the *Ticket-Plus Queue.*

Recall section 3.1, where we have reported an interesting finding, which is the approximately linear relationship between $E[N|D = cK]$ and K. This discovery suggests that management can perform sensitivity analysis of $E[N|D = cK]$ with various values of customer's tolerance limit K and use the result to develop a service improvement policy. For example, assume that $c = 1$. Then, $E[N|D = K]$ denotes the expected queuing position of the marginal balking customer. So, if this information can be easily computed for various K values and pass on to the customers, it can help them estimate their expected waiting time more correctly and hence reduce the balking rate. That is precisely the idea behind the recommendation of the Ticket-Plus Queue.

The Ticket Queue we are proposing has an additional element; we suggest that on the issued ticket, except of the number, there can also be printed the *expected waiting time* $(1/\mu)E[N|D = d]$ for a given ticket position d. This new information will rectify the customer's incorrect estimate of hers/his expected delay (d/μ). Furthermore, we assume that the ticket technology enables the system to keep track of ticket count d, so that it can provides the needed information dynamically. Under these assumptions, the resulting queue is the Ticket-Plus Queue. In fact, people feel better about queuing when they can estimate in advance their waiting time, hence the Ticket-Plus queue gives them the additional information they need to achieve it.

It is time to point out the computational perspective of our recommendation; thus, in order to get the expected waiting time $(1/\mu)E[N|D = d]$ printed on the ticket correctly, we assume that the system is aware that a customer will balk the queue if hers/his estimation of their waiting time is more than K/μ for an integer value of K. So, the management knows that a customer will balk, when hers/his ticket position is K and higher.

The next step is to find the smallest integer $\widetilde{K}$ such that:

$$E[N|D = \widetilde{K}] \geq K \tag{3.65}$$

where the expectation can be computed as seen in equation (3.62) of the approximative procedure in section 3.4, using $\widetilde{K}$ instead of K as the balking limit. Therefore, an upward search is applied in the finding of $\widetilde{K}$. Also, as $E[N|D = K]$ increases approximately linear in K, we can claim that $\widetilde{K}$ is well defined.

After the computation of $\widetilde{K}$, we use the result to obtain $(1/\mu)E[N|D = d]$ for ticket positions $d = 1, \ldots, \widetilde{K}$. So, we can now provide the customer with ticket position d with the *unbiased*, average waiting time $(1/\mu)E[N|D = d]$. With this modification, customers will balk if and only if their ticket position is $\widetilde{K}$ and above, so that the Ticket-Plus Queue will finally behave as the

Ticket Queue with balking limit $\widetilde{K}$. Naturally, we expect $\widetilde{K}$ to be greater than K, as the Ticket-Plus Queue implies an improvement of the initial model. Furthermore, we recommend printing the average waiting time on the ticket, because we really need to communicate this information to the customers, and with this way we can guide them appropriately to behave the same way as in the Ticket Queue with balking limit K.

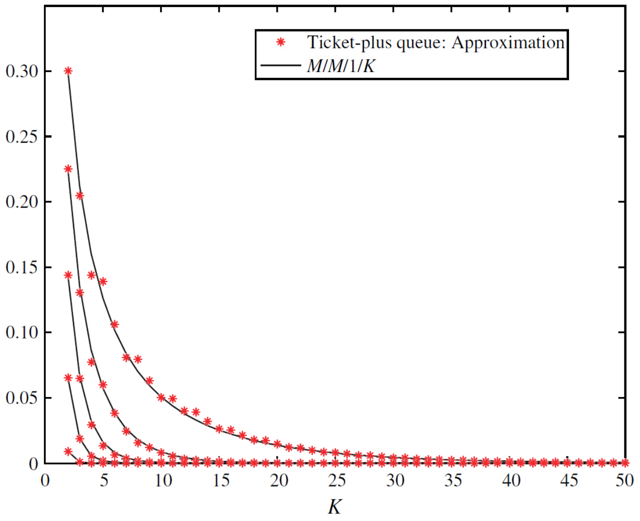


Figure 3.18 Comparison of Balking Probabilities of Ticket-Plus and Physical Queues, $\rho = 0.1, 0.3, \ldots, 0.9$

Finally, we shall see graphically in Figure 3.18 that the Ticket-Plus Queue always yields a balking probability virtually identical to that of the M/M/1/K queue. Also, note that the largest balking probability difference (TAD) is 0.009, a really low value. Additionally, we can use the relationship $\rho_e = \rho P_b^p$, where ρ_e is the system utilization factor, to show that the Ticket-Plus Queue and the physical queue have virtually the same system utilization. Lastly, our numerical results imply that the expected waiting times for those customers that do not balk in two queues is almost identical.

In conclusion, we shall say that by informing the customer of his anticipated delay based on the ticket position, management can raise the performance of the Ticket Queue to the level of the physical queue and at the same time maintain the benefits of the former. However, although the performance of both queues is virtually identical in several first moment measures like the expected waiting time, it cannot be inferred that this will happen in higher moment measures.

3.6 Threshold Strategies in Ticket Queues

In this section, we examine the same model of Ticket Queue from a different perspective. In synopsis, we have considered a Markovian queue with homogeneous customers where, each customer upon arrival is issued a ticket with an assigned number, and decides whether to join or balk according to hers/his ticket position (the observed difference of the number on hers/his ticket with the display panel number). The decision is made through a simple threshold-based strategy, to join if and only if the ticket position is below a threshold K. Hence, we keep the same assumptions and we also consider that the customers are homogeneous with respect to their payoff functions.

Now, we are going to model the system as a symmetric non-cooperative game in which the set of actions is to join or balk. So, given this simple threshold strategy we can show that it is not a Nash equilibrium strategy. Using this statement we mean that, if all the customers follow the same single threshold strategy we have described, an individual customer will prefer not to adopt it and deviate from the other customers. Consequently, our principal concern is to determinate which is the best response of an individual against the threshold strategy adopted by all the other customers, and the answer is going to be surprising.

In the following, we will reexamine the same model and we will focus at some different points in order to reach our main goal which is the determination of the best response. We will also take the previous analysis of the stationary distribution a few steps further, because we need to get scalar expressions instead of a matrix representation of the solution. Thus, in the subsequent subsection, we will remind briefly the basic aspects of the model described in section 3.1 and we will give a definition and two useful, for our next purposes, propositions.

3.6.1 Steady State Probabilities

The model we presented before works like a standard **FCFS M/M/1** queue with arrival and service rate λ and μ respectively. A customer who finds the system empty joins surely, otherwise decides to stay only if hers/his ticket position D, which is perceived as the actual queue length, is below the tolerance limit K, that is the same for all customers. Therefore, a customer joins if and only if $D < K$, otherwise balks. Also, a balking customer who is called for service, is considered to spend zero time being served.

We remind the state representation; the Markovian description is achieved with the definition of the L-dimensional vectors $\boldsymbol{n} = (n_1, n_2, \ldots, n_L)$, where L is the number of joining customers and n_l, $l = 1, \ldots, L$ is the number of tickets issued to the balking customers after the lth and prior to the $(l+1)$th joining customer. The state space is given by (3.1). We have shown the transition diagram for $K = 4$ and $K = 2$ in Figures 3.1 and 3.5 respectively, and in Figure 3.19 below we show the case of $K = 3$ to cover the three simplest cases and recall schematically the model.

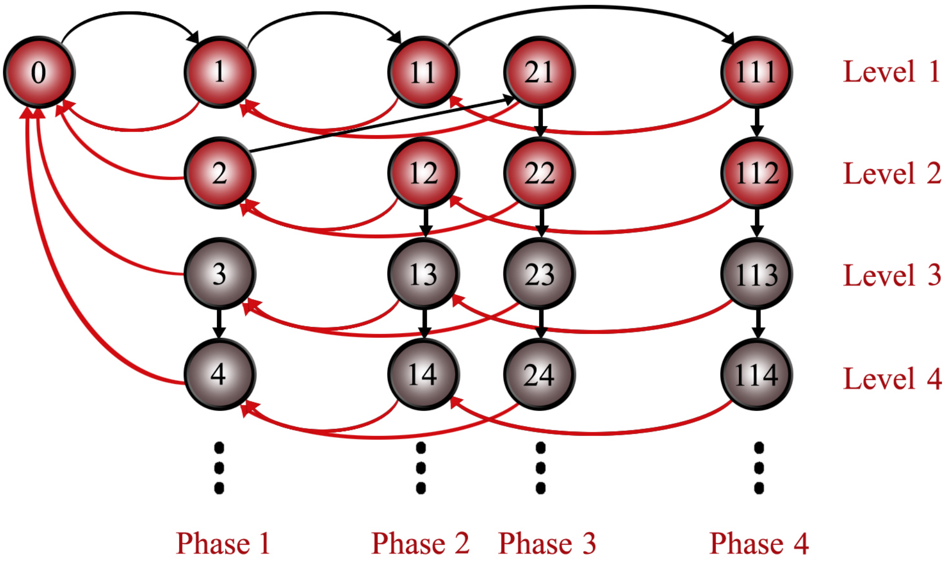


Figure 3.19 Transition Diagram of Ticket Queue with $K = 3$

In Figure 3.19, levels denote the infinite groups of states with the same n_L, where the total of 2^{K-1} phases represent state groups which have similar rest of the state, noted as $(n_1, \ldots, n_{L-1})$. This is actually the partition we made in 3.2.2 in order to model again the problem as a QBD process and it is useful to bring back that in mind.

At this point, we are going to expand the results we obtained before by providing the steady state distribution in two forms; the first one is a matrix geometric form, while the second, and the most useful, is a scalar form. For convenience, we use the same notation as in 3.2.2; so, we denote as $\tilde{\boldsymbol{p}}_\nu$ the 2^{K-1} vector of steady state probabilities associated with T_ν (as defined in 3.2.2). Hence, with the other way, if we note V as the level and H as the phase, the jth component of $\tilde{\boldsymbol{p}}_\nu$ is: $\tilde{\boldsymbol{p}}_\nu(j) = P(V = \nu, H = j)$.

Recall that, for all $\nu \geq K$ equation (3.18) gives (3.20), so that we have:

$$\tilde{\boldsymbol{p}}_\nu = \tilde{\boldsymbol{p}}_K \widetilde{\boldsymbol{R}}^{\nu-K}, \quad \widetilde{\boldsymbol{R}} = -\lambda \boldsymbol{B}^{-1}$$

Remember that $\boldsymbol{B}$ is a lower triangular matrix, placed on the diagonal of the generator matrix, thus contains the transition rates that do not change the level. Also, each element of the main diagonal of matrix $\boldsymbol{B}$ is equal to $-(\lambda+\mu)$ exactly because $\boldsymbol{B}$ appears on the diagonal of generator $\widetilde{\boldsymbol{Q}}$. So, let us now show the transition matrix $\widetilde{\boldsymbol{Q}}$ for $K=3$, which is:

$$\widetilde{\boldsymbol{Q}} = \begin{bmatrix} \boldsymbol{A_{00}} & \boldsymbol{A_{01}} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{C} & \boldsymbol{A_{11}} & \boldsymbol{A_{12}} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{C} & \boldsymbol{A_{21}} & \boldsymbol{B} & \boldsymbol{A_{23}} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{C} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{B} & \boldsymbol{\lambda I_4} & \boldsymbol{0} \\ \boldsymbol{C} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{B} & \boldsymbol{\lambda I_4} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$

Now, let $\boldsymbol{E}_{i,j}$ be a 4×4 matrix, with 1 in position (i,j) and 0 everywhere else. For instance, $\boldsymbol{E}_{12}$ is defined as:

$$\boldsymbol{E}_{12} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Furthermore, note that $\boldsymbol{I}_4$ is the 4×4 identity matrix. Hence, in $\widetilde{\boldsymbol{Q}}$ we have:

$$\begin{aligned} \boldsymbol{A}_{00} &= (-\lambda) \\ \boldsymbol{A}_{01} &= (-\lambda, 0, 0, 0) \\ \boldsymbol{A}_{11} &= \boldsymbol{B} + \lambda(\boldsymbol{E}_{12} + \boldsymbol{E}_{24}) \\ \boldsymbol{A}_{12} &= \lambda \boldsymbol{I}_4 - \lambda(\boldsymbol{E}_{11} + \boldsymbol{E}_{22}) \\ \boldsymbol{A}_{21} &= \lambda \boldsymbol{E}_{13} \\ \boldsymbol{A}_{23} &= \lambda \boldsymbol{I}_4 - \lambda \boldsymbol{E}_{11} \end{aligned}$$

The extra thing we need to obtain for our purposes in this section, is an explicit expression for $\widetilde{\boldsymbol{R}}^{\nu-K}$, for an arbitrary $\nu \geq K$. The first thought that comes in mind is, since $\widetilde{\boldsymbol{R}}$ is a lower triangular matrix, so are its powers. However, before we represent the proposition for $\widetilde{\boldsymbol{R}}^{\nu-K}$ below, we should discuss a relation between the real number of customers in the system and the phases.

Let $n(j)$ be the number of joining customers in the system associated with phase j. That is, mathematically defined:

$$n(j) = \min \left\{ k : \sum_{l=0}^{k-1} \binom{K-1}{l} \geq j \right\}$$

As an example, in the case where $K = 3$, we have that $n(1) = 1$, $n(2) = 2$, $n(3) = 2$ and $n(4) = 3$. We can see this schematically in Figure 3.15, that each column (phase) consists of states which have 1, 2, 2 and 3 components respectively. We also define the following set:

- ○ Let $\boldsymbol{\alpha} = (n_1, \dots, n_{n(j)})$ be a representative state of phase j for any given level $n_{n(j)}$. Then, $A(j)$ is the set of all phases i such that the last $n(j)$ components of their states are similar to $\boldsymbol{\alpha}$.

In other words, we can say that $A(j)$ contains all phases i which, without their $w(i,j) = n(i) - n(j)$, $w(i,j) \geq 0$ first components, represent the phase j. We will give an example to make this definition straightforward; for $K = 3$, $A(1) = \{1, 2, 3, 4\}$. This can be understandable by computing all $w(i,j)$s for $i = 1, 2, 3, 4$ (all the phases) and $j = 1$ (as we want to indicate $A(1)$). So, we have:

$$\begin{aligned} w(1,1) &= n(1) - n(1) = 0 \\ w(2,1) &= n(2) - n(1) = 1 \\ w(3,1) &= n(3) - n(1) = 1 \\ w(4,1) &= n(4) - n(1) = 2 \end{aligned}$$

Indeed, if we remove none of the first components of states in phase 1, it looks like itself. Furthermore, if we remove the first components of states in phase 2, it looks like phase 1. The same goes for phase 3, while for phase 4, if we remove the first two components from every state, it is similar to phase 1. Following this procedure, we can show also that $A(2) = \{2, 4\}$, $A(3) = \{3\}$ and $A(4) = \{4\}$.

Given this definition, we can now represent the proposition for the explicit solution of $\widetilde{\boldsymbol{R}}^{\nu-K}$. This proposition will lead us to a second one, which is about the presentation of the steady state probabilities in a scalar expression, as said before, more useful for our purpose in this section.

Proposition 3.6.1. *Consider two phases, i and j. We have:*

$$\widetilde{\boldsymbol{R}}^{\nu-K}\big(i,j\big) = \begin{cases} \binom{\nu-K+w-1}{w}\left(\frac{\mu}{\mu+\lambda}\right)^{w}\left(\frac{\lambda}{\mu+\lambda}\right)^{\nu-K}, & \text{if} \quad i \in A(j) \\ 0, & \text{otherwise} \end{cases} \tag{3.66}$$

where $w = w(i,j)$.

Proof. We show a proof with probabilistic/ combinatorial approach. Actually, in this approach, we observe the process with a discretized way; that is, we focus on an equivalent discrete time Markov chain, with transition matrix $\boldsymbol{P} = \boldsymbol{I} + \frac{1}{\lambda+\mu}\boldsymbol{Q}$.

Firstly, note that, for levels greater than or equal to K (the gray zone in diagrams), changes in phases can only be in one direction. In fact, these changes take place only after service completions and hence the result *always decreases the phase.*

In addition, note that *the levels can only be increased in this zone* ($\nu \geq K$). So, if we project the transitions only on the phase axis, there is only one trajectory of the process until reaching state 0. That means, the number of trajectories from a state until reaching another state (with a lower phase and a higher level, as there is no other possibility when navigating from state to state in gray zone, $\nu \geq K$), is finite and easy to characterize.

At this point it is time to determine (3.63); first of all, we shall say that if $i \notin A(j)$, there cannot be any transition from a state of phase i to a state of phase j, so, in that case, $\widetilde{\boldsymbol{R}}^{\nu-K}\big(i,j\big) = 0$ for all j.

Let us now consider that $i \in A(j)$. We can distinguish two cases here; the one is $i = j$ and the other is $i \neq j$.

For $i = j$, we remain in the same phase. That is, the level is increased by $\nu - K$ without any phase changes. This also means that there are $\nu - K$ arrivals without service completions. The probability of such an event is $\left(\frac{\lambda}{\mu+\lambda}\right)^{\nu-K}$, which is exactly the value of (3.36) with $w = 0$. As a matter of fact, $w = w(i,j) = n(i) - n(j)$ is equal to 0 if $i = j$. Thus, (3.36) holds in this case.

For $i \neq j$, in order to reach the higher level of $\nu - K$ units, we should jump from a phase to another phase. So, actually, we have service completions taking part here. There are $\binom{\nu-K+w-1}{w}$ trajectories that can change the level by $\nu - K$ together with w service completions.

Giving an example here, imagine that $\nu = 5 \geq K = 3$. So, we have that $\binom{\nu-K+w-1}{w} = \binom{w+1}{w} = w + 1$. That means, for $w = 0$, we have no service completions, so there is only one way to reach a level 2 units higher, and this is only by two arrivals. For $w = 1$, we have two ways to reach a level 2 units higher so that it occurs only one service completion. For instance, from state (113) there are two possible ways of reaching the fifth level with one service completion: (113)→(114)→(14)→(15) and (113)→(13)→(14)→(15).

Therefore, each event of the type described before comes with probability $\left(\frac{\mu}{\mu+\lambda}\right)^{w}\left(\frac{\lambda}{\mu+\lambda}\right)^{\nu-K}$ (w service completions, $\nu - K$ arrivals). Hence, (3.36) holds for all possible cases.

$\square$

Proposition 3.6.2. *For $1 \leq i \leq 2^{K-1}$ we have:*

$$\tilde{\boldsymbol{p}}_{\nu}(j) = \sum_{i \in A(j)} \tilde{\boldsymbol{p}}_{K}(i) \binom{\nu - K + w - 1}{w} \left(\frac{\mu}{\mu+\lambda}\right)^{w} \left(\frac{\lambda}{\mu+\lambda}\right)^{\nu-K} \tag{3.67}$$

with $\nu \geq K+1$. Additionally, note that $\tilde{\boldsymbol{p}}_{K}(i)$ can be obtained through a set of linear equations implied by the boundary condition.

Proof. Given the matrix geometric structure $\tilde{\boldsymbol{p}}_{\nu} = \tilde{\boldsymbol{p}}_{K}\widetilde{\boldsymbol{R}}^{\nu-K}$ and Proposition 3.6.1, the proof is straightforward. $\square$

Hence, the ground for obtaining an expression for the *conditional waiting time* given the total number of customers in the queue has been prepared. Generally, the results obtained from this subsection are useful to determine the best response of an individual according to the information s/he receives for the queue upon arrival, and also show that the simple threshold strategy is not a Nash equilibrium strategy. All the stuff we mentioned here is the subject of our next subsection.

3.6.2 Waiting Time Distribution and Best Response

It is time to use the results we obtained before for obtaining the waiting time distribution and the best response of an arriving customer, given the information s/he observes upon arrival. It is a good idea to recall briefly any useful notation and its meaning here. Remember that:

- D is the number of all customers that have been issued a ticket, joining and balking, in other words the ticket position.

- N is the number of joining customers in the system.

- We use in the following the two-dimensional continuous time Markov chain we studied above; that is, any pair (ν, j) determines uniquely a state of the form $(n_1, \ldots, n_{n(j)})$, where $n(j)$ is the number of joining customers of the state and for level ν holds that $\nu \geq K+1$, while for phase j we have $1 \leq j \leq 2^{K-1}$.

- As a consequence, $D = \sum_{l=1}^{n(j)} n_l$ and the realization of N is $n(j)$. Hence, the value of L is noted as $n(j)$.

- Last but not least, note that the queueing time of an arriving customer is $W = \sum_{l=1}^{n(j)} X_l$, where $X_l \sim Exp(\mu)$ and is independent of anything else.

Except for the reminders, we also introduce two definitions, which are useful for the following:

- Let $S_d = \bigcup_{L=1}^{K} \left\{ \boldsymbol{n} \in \mathbb{N}^L : n_l \geq 1, \sum_{l=1}^{L} n_l = d, l = 1, \ldots, L \right\}$, $d \geq K$, so that S_d is a set that contains all the states $\boldsymbol{n}$ in which we have $D = d$ for $d \geq K$.

- Let A_d be the number of real customers in the system, when for the first time a state is in S_d, in an arbitrary busy period.

So, let us recall Figure 3.19, the case of $K = 3$. Imagine that every state is replaced with their corresponding pair (ν, j) of levels and phases. Moreover, note also that we are interested in cases where $d \geq K$, that is we are looking for the behavior of the system where the customers notice a ticket position which leads them to balk according to their simple threshold strategy. We are willing to examine now if a given individual would choose to follow the threshold strategy adopted by all the others for all possible $d \geq K$ values. The answer will be revealed in the next theorem. Meanwhile, we should introduce three useful lemmas that lead us to the proof of this theorem and give interesting results generally.

Lemma 3.6.1. *For any $d \geq K$, S_d contains exactly 2^{K-1} states; each one comes from a different phase. Furthermore, the state with phase 2^{K-1} has the lowest value of n_L, which is $d - (K-1)$.*

Proof. We begin with the observation of the fact that within a phase, the values of d increase by one as the level increases by one. Consequently, the lower value of d in a phase meets in level 1. The values of d in a level are also depending on the phase, which can be any integer from 1 to K. Therefore, we get that for every value of $d \geq K$ there is exactly one state in phase, and that is the first part of the proof.

For the second part, we simply note that the state in phase 2^{K-1} has the form $(1, 1, \ldots, n_L)$ for any given level and $K \geq 1$. The number of 1s prior to n_L is $K-1$, which is the largest value of $\sum_{l=1}^{L-1} n_l$, according to the state space (3.1). Thus, for a given d, $n_L = d - \sum_{l=1}^{L-1} n_l$. So, n_L obtains its lowest value for the largest value of the sum, which is $K-1$. □

As seen, Figure 3.20 below illustrates the result obtained from Lemma 3.6.1 for the case of $K=3$.

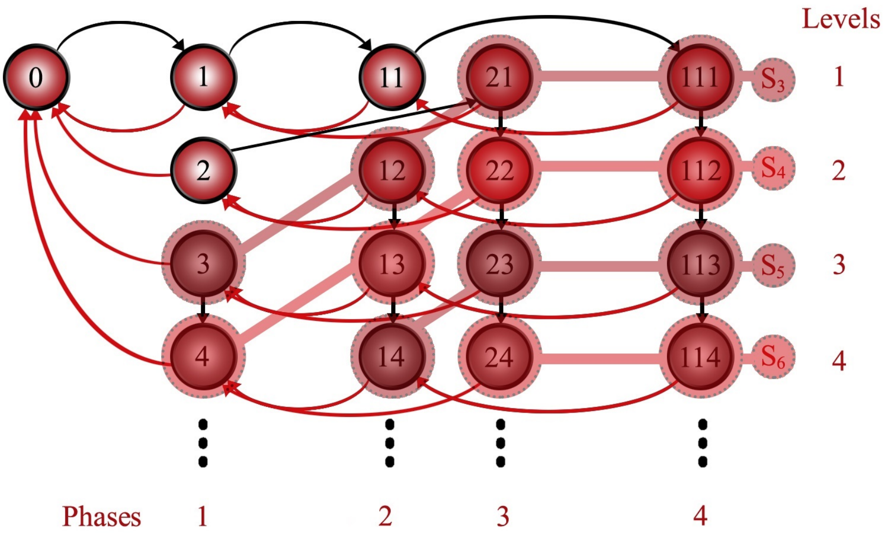


Figure 3.20 S_d Sets of Ticket Queue with $K=3$

Lemma 3.6.2. *For $d \geq 2K-1$, during a busy period, the number of joining customers in the system decreases.*

Proof. Firstly, $n_L \geq K$ is a necessary and sufficient condition such that no customer joins the queue until the system empties. That is, because it guarantees that $d \geq K$ for the rest of the busy period and as seen before, the states in the gray zone of diagrams represent this condition. Hence, no customer will join until the system empties and only service completions can be occurred. So, the number of joining customers in the system is decreased. We should now point out the value of d in which this phenomenon starts to appear; in order to achieve this, we utilize the construction of the state space (3.1). Thence, we finally have:

$$n_L = d - \sum_{l=1}^{L-1} n_l \geq (2K-1) - (K-1) = K \Rightarrow n_L \geq K$$

□

Lemma 3.6.2 shows the fact that whether d has value greater or equal than $2K-1$, the joining customers start to decrease. This outcome makes us wonder if it is truly the best policy to balk if $D \geq K$, knowing that everyone else will do the same.

Lemma 3.6.3. *For $d \geq 2K-1$, we have that $A_{d+1} \leq_{st} A_d$.*

Proof. Since there cannot be any joining customers within a busy period for $d \geq 2K-1$, we obtain a necessary condition for $\{A_{d+1} \geq n\}$, which is $\{A_d \geq n\}$. Consequently, in order to have at least n joining customers while visiting S_{d+1}, we must have at least n joining customers when visiting S_d before that. Thus, we have:

$$P(A_{d+1} \geq n) = P(A_d \geq n)P(A_{d+1} \geq n|A_d \geq n) \leq P(A_d \geq n)$$

□

We are ready to introduce the essential theorem of this subsection. The theorem will be followed by inferences and a second theorem, which makes our analysis complete, as we discuss there the *best response* of a given individual.

Theorem 3.6.4. *For $d \geq 2K-1$, the conditional distribution of $N|D=d$ is stochastically decreasing with d. Additionally, we have:*

$$\lim_{d\to\infty} P(N=1|D=d) = 1$$

Proof. Firstly, we should prove the monotonicity. The result can be shown by employing a *coupling argument.* Actually, the idea is the following: we observe that the distribution of $N|D=d+1, A_{d+1}=\alpha$ is equivalent to $N|D=d, A_d=\alpha$, where α is any integer in $\{1,\ldots,K\}$. This can be explained quite simply; to begin with, future transitions themselves do not depend on the current state. Moreover, the number of joining customers in the queue after a transition depends only of the number of joining customers present exactly before it occurred. Thus, if we add the information regarding to the number of balking customers to the given number of joining customers present, the distribution of the future does not change. So, together with Lemma 3.6.3, we shall say that $N|D=d+1$ is stochastically smaller than $N|D=d$. Hence, we just showed that the distribution of $N|D=d$ is stochastically decreasing with d.

Next, we are going to prove that, when $d\to\infty$, the limit is a unit mass at $\nu=1$. So, at this point, we make a use of Proposition 3.6.2 in order to write the joint probabilities $P(N=\nu, D=d)$ as the appropriate polynomial in d, for $d \geq 2K-1$. Thus, we have:

$$P(N=\nu, D=d) = \sum_{J:n(j)=\nu} \sum_{i \in A(j)} \tilde{\boldsymbol{p}}_K(i) \binom{\nu - K + w - 1}{w} \Big(\frac{\mu}{\mu+\lambda}\Big)^w \Big(\frac{\lambda}{\mu+\lambda}\Big)^{m-K} \tag{3.68}$$

where $m = d - \sum_{l=1}^{n(j)-1} n_l$. Substituting m, equation (3.68) can be written as:

$$P(N=\nu, D=d) =$$

$$\sum_{J:n(j)=\nu} \sum_{i \in A(j)} \tilde{\boldsymbol{p}}_K(i) \frac{(m-K)\cdots(m-K+w-1)}{w!} \Big(\frac{\mu}{\mu+\lambda}\Big)^w \Big(\frac{\lambda}{\mu+\lambda}\Big)^{d-\sum_{l=1}^{n(j)-1} n_l - K}$$

We see that each term in this sum is polynomial of rank w in d, times $\big(\frac{\lambda}{\lambda+\mu}\big)^d$. Also, note that $w = n(i) - n(j) \leq K - n(j)$, which in the above relation equals $K - \nu$. That is, the whole sum is a polynomial of rank $K - \nu$ in d, times $\big(\frac{\lambda}{\lambda+\mu}\big)^d$.

The conditional probabilities are given by:

$$P(N \geq \nu | D=d) = \frac{P(N \geq n, D=d)}{P(D=d)}, \quad n \geq 1$$

We observe that this quantity cancels the geometric factor $\big(\frac{\lambda}{\lambda+\mu}\big)^d$ and leaves us only with the fraction of polynomials. As the degree of the polynomial in the numerator is $K-\nu$, and the degree of the polynomial in the denominator is $K-1$, we have that:

$$\lim_{d \to \infty} P(N \geq \nu | D=d) = \begin{cases} 1, & \nu = 1, \\ 0, & \nu > 1. \end{cases}$$

Therefore, when $d \to \infty$, the distribution of $N|D=d$ tends to unit the mass at $\nu = 1$. □

Theorem 3.6.4 can give us explicit results for small threshold values. As an example, we show briefly how the statement of this theorem can be simplified for the case of $K=2$. Thus, as $N|D=d$ is stochastically is decreasing in d, $P(N=1|D=d)$ should be increasing in d and tend to 1 as $d \to \infty$. Hence, we get:

$$P(N=2|D=d) = \frac{\tilde{\boldsymbol{p}}_{d-1}(2)}{\tilde{\boldsymbol{p}}_d(1) + \tilde{\boldsymbol{p}}_{d-1}(2)} = \frac{(1+\rho)^3}{1+\rho(3+\rho(2+d+\rho))} \underset{d\to\infty}{\longrightarrow} 0$$

Have in mind that the values of $\tilde{\boldsymbol{p}}_d(1)$ and $\tilde{\boldsymbol{p}}_{d-1}(2)$ are obtained directly from subsection 3.2.3.

The main thing we shall comment about Theorem 3.6.4 is the intuition behind it. Remember, that an arriving customer perceives the ticket position $D = d$ as the real queue length in the system. Thus, we realize that the larger the total queue length observed by an arriving customer, the longer s/he estimates the elapsed time since the arrival of the last who joined. That is, the number of the customers who joined back then and are still queuing is likely to be *small*. So, the realization obtained by this theorem leads us to express the best action of an arriving customer given what he observes upon arrival.

Before we state the next theorem, that gives us the best response, we define the following:

- Let U be the utility from being served.
- Let $C(t)$ be waiting cost function, while t is the waiting time. Moreover, assume that $C(t)$ is any monotone increasing function of t.
- As a result, $U - C(E[W|D = d])$ is the expected utility for an individual who observed ticket position d.

Theorem 3.6.5. *Consider the M/M/1 Ticket Queue. Assume that customers are homogeneous with respect to the service value and the waiting cost. Furthermore, suppose also that the service value is large enough such that for lower values of d, $U - C(E[W|D = d]) \geq 0$. Then, we have:*

- *For any waiting cost function $C(t)$ that is increasing with the waiting time, a threshold strategy is not a symmetric Nash equilibrium strategy.*
- *If all customers adopt the same threshold strategy, then the individual's best response is to follow a double threshold strategy (K_1, K_2) where $K_1 < K_2$: join if and only if $D < K_1$ or $D > K_2$.*

Proof. Let us consider that all adopt some threshold strategy K. As for a customer upon arrival, we assume firstly that the observed queue length D is relatively small. In that case, the arriving customer gets a positive expected utility from joining which is $U - C(E[W|D = d]) \geq 0$, exactly what we have presumed for lower values of d.

As a second case, we present the one where an arriving customer observes an extreme large queue length. Due to Theorem 3.6.4, the customer concludes that for big values of d (from $2K - 1$ and up), the actual queue length which describes the number of joining customers in the system is extremely small and hence, his expected utility from joining is positive as well.

Thus, the best response of a given arriving customer is a double threshold strategy (K_1, K_2) with $K_1 < K_2$ and the logic behind it is to join if and only if $D < K_1$ or $D > K_2$. So, this given individual deviates from the strategy chosen by all others, and as a consequence, the simple threshold strategy cannot be a Nash equilibrium strategy. □

Theorem 3.6.5 brought to the surface the best response of a given individual if all the other customers follow the simple single threshold strategy; s/he will want to join the queue even if the observed queue length is enormous. At first thought it would seem odd, however, after our analysis, it appears pretty logical. So, a great way to close this subsection is to give a plain example with numbers on this best response.

Assume that the customers balk if they observe a ticket position greater or equal than 10. Now, consider an individual who observes a ticket position equal to 50. Due to the strategy of all others, s/he knows that the last 40 tickets issued belong to customers who actually balked upon arrival. So, it is more likely that, since the arrival of the last joining customer, there were more than 40 arrivals who balked and also a few service completions taking place. Thus, this individual customer can conclude that the actual number of customers currently in the system is really small and hence, s/he will want to join the queue.

3.6.3 Summary and Extensions

In the current section, we showed that although in real life customers can adopt simple one threshold strategies, these cannot be Nash equilibrium strategies. That happens because given that all others follow them, a specific individual prefers to deviate and join even if the queue is long. We are closing the section by introducing some extensions and thoughts on this subject.

An interesting question at this point is whether the double threshold strategy we presented is a candidate for Nash equilibrium. However, it is clear that this option is rejected; since, if all adopt it, then the tagged individual who observes a really long queue knows that the queue tail and more must be constructed of joining customers. Thus, s/he prefers not to join, because hers/his waiting time distribution is relatively large in the stochastic sense. So, the double threshold strategy cannot be a Nash equilibrium strategy.

A next appealing topic for discussion is about the validity of our results if we change some of our model assumptions. Are these assumptions necessary for obtaining the same result about the threshold strategy, that it cannot be a Nash equilibrium? We end this subsection referring to different generalizations of the system, and whether the result we get changes.

The first issue we should refer to is about the set of actions. The assumption we made has two possible actions: to join or balk upon arrival. Yet, in many systems, a customer joins upon arrival but reneges after a while. If we expand the set of actions in this model accordingly, then our main result still holds. A tagged individual will still prefer to adopt the double threshold strategy, as we have proved it is a better response against the simple one. On the other hand, if we want to examine whether the double threshold strategy is the *best response* against the threshold strategy, the answer depends on the properties of the cost function. Thus, if the waiting cost function is concave with time, then if one decides to join, s/he would never leave. However, if the waiting cost function is convex with time, then one may abandon the queue at some point; exactly when the waiting would have become more expensive than it was while joining the queue.

One more generalization we are going to report is about multiple servers; so that we are dealing with a M/M/c queue rather than a M/M/1 queue. In this case, if there is an idle server, customers will join for reasonable parameter values. Additionally, if we assume that all servers are busy, the behavior of a customer depends on the number of servers only through the total service rate, something that does not affect our results; hence all of them still hold.

There are far too many generalizations that can be discussed, however they need more analysis in order to reach a proper answer about the issue we are interested in. Hopefully, this work could widen the horizons about the wealth of topics we shall discover about Ticket Queues and the benefits we can derive by them.

Chapter 4

Ticket Queues with Reneging Customers

In the current chapter, we examine a Ticket Queue under different assumptions; that is, we focus on the customers who entered the queue, lost their patience and eventually left the system without obtaining service, the so-called *reneging* customers. Our analysis relies on the fine work of Ding, Ou and Ang (2015).

As in the previous model, a customer estimates her/his expected waiting time through the difference between the displayed service number and her/his ticket number, the ticket position. So, if that difference is too large, s/he may balk. However, even after joining the queue, s/he may renege at any moment. In this chapter, we assume that the customers' tendency to renege depends dynamically on the ticket position.

Once more, we need to point out that the ticket position is not a precise description of the actual number of queuing customers in between. Hence, we consider such type of Ticket Queue with customer reneging; yet because of the difficulty we face due to the exponential growth of the state space, we should analyze this queue through *approximation.* Therefore, we will call *R-Ticket Queue* the Ticket Queue with reneging customers, and we shall introduce the appropriate modification later.

In the following, we represent this model's formulation. We are also specifically interested in obtaining the total reneging percentage, as we would like to reduce it in order to improve the R-Ticket Queue model. Last but not least, we will briefly present an extension of this model regarding to the R-Ticket Queue with multi-servers.

4.1 Model Formulation and Definitions

We begin with the description of the Ticket Queue with reneging customers, the one we call the R-Ticket Queue. This model has the same basic elements as the Ticket Queue with balking customers, so, it is plausible to recall them and point out the differences.

As expected, the customers are issued a ticket upon their arrival, which represents their waiting position in the queue. The number of the customer being served is broadcast on display panels; so that the only queuing information which the server and the customer have, is the difference between their ticket number and the number being served, the well known ticket position we denote generically as D.

It is known as well, that the Markovian model of the R-Ticket Queue assumes Poisson arrival process with rate λ and service times identically exponentially distributed with rate μ. Additionally, the time to call up a customer to service is negligible, so every time a reneged customer is called for service, the server moves to the next number with no delay.

We should now introduce our assumptions about the customers' reneging behavior; we presume that it is also Markovian, and, at the time when a customer observes a ticket position $D = d$, then the rate of reneging is τ_d. As expected, τ_d gets higher as d increases and we have a faster increase of τ_d for a larger d. This can be written formally as:

$$\tau_d \geq 0, \quad \tau_{d+1} \geq \tau_d, \quad \tau_{d+2} - \tau_{d+1} \geq \tau_{d+1} - \tau_d, \quad d \in 0, 1, \ldots \tag{4.1}$$

Note that $\tau_0 = 0$. That means, the customer is in service and therefore, s/he will never renege. Apparently, the stability of the R-Ticket Queue is guaranteed; that is a consequence of such reneging, because it does not allow the number of customers to be infinite.

Furthermore, we need to define the state notation for R-Ticket Queue, which is actually similar to the one we introduced in Chapter 3. We denote the empty system as $\boldsymbol{n} = (0)$ and the non-empty system using a vector $\boldsymbol{n} = (n_1, \ldots, n_L)$, where L is the number of customers who are still in the system, and positive integers n_l, $l \in \{1, \ldots, L-1\}$ represent the difference of ticket numbers issued to the lth and $(l+1)$th customers who are waiting in queue. Thus, n_l actually shows how many customers reneged between the lth and the $(l+1)$th customers in the system. Moreover, n_L represents the difference of ticket numbers issued to the last customer who is still in the system and the next forthcoming customer. In Figure 4.1 below, we can see an example on this model. It is also observable that the notation is similar to the Ticket Queue of the previous chapter, with the difference that n_l, $l \in \{1, \ldots, L\}$ represent reneging customers and not balking.

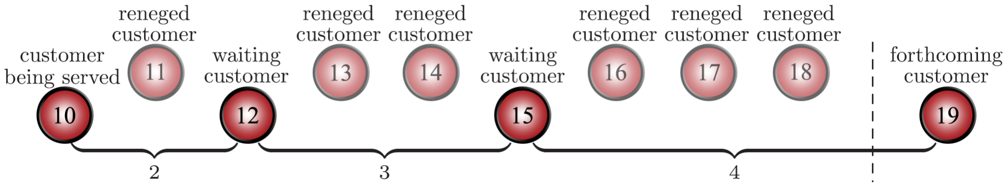


Figure 4.1 An example for recording the system state of the single-server R-Ticket Queue. Customers with tickets 10,12 and 15 are waiting in the queue, while the others have reneged from the system. Such state can be written as $\boldsymbol{n} = (2, 3, 4)$

The Markovian model we build in order to describe the R-Ticket Queue is illustrated in Figure 4.2. We use again black arrows for new arrivals, red arrows for service completions and we add gray arrows which represent the reneging situation. There, we can notice that the state space explodes up really fast. Accurately, for a given l, there are 2^l possible states if the ticket position of the next arriving customer is lower or equal than l, equivalently $\sum_{k=1}^{L} n_k \leq l$. For instance, when $l = 3$, there exist $2^3 = 8$ possible states which are (0), (1), (2), (3), (1,1), (1,2), (2,1) and (1,1,1).

Moreover, we can observe in Figure 4.2 a partition of the states. Actually, system states in the dash-and-dot rectangles are grouped into *super states*. We give a formal definition; let super state S_L represent the collection of all states in which there are L customers in the system, waiting or being served. As we can see, super state S_0 contains only state (0), while super state S_1 includes states (1),(2),(3),... and so on.

We define as $p(S_L)$ the steady probability for super state S_L in the single-server R-Ticket Queue. In the following, we will introduce two interesting propositions; the first one provides us with tight lower and upper bounds for the steady probabilities of super states in the R-Ticket Queue. The second, gives us also an useful outcome which is about the total reneging percentage of the single-server R-Ticket Queue; in fact it provides us with a tight lower bound of it.

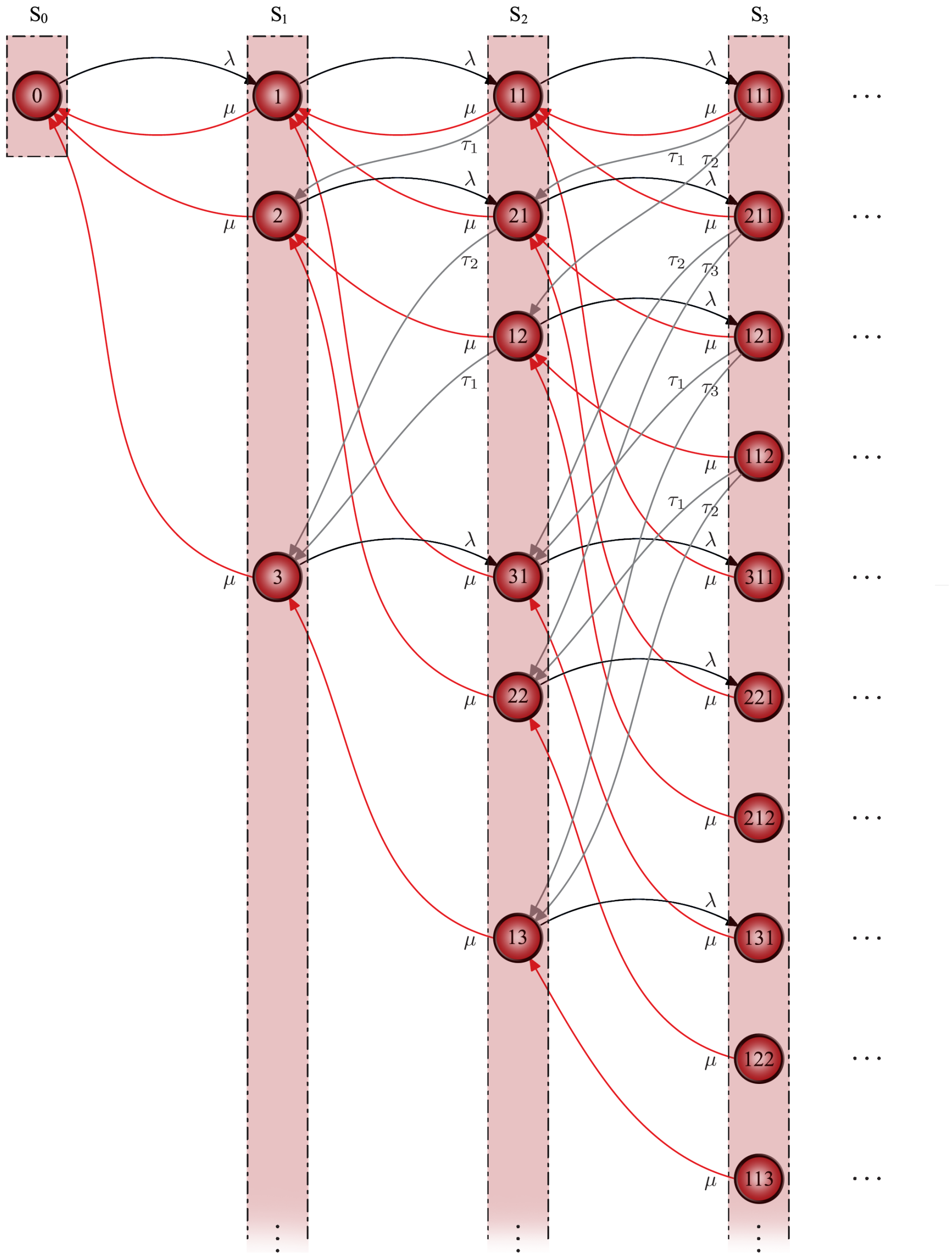


Figure 4.2 Transition Diagram for Single-server R-Ticket Queue

Proposition 4.1.1. *For a single-server R-Ticket Queue, there exists an integer $m \geq 1$, such that:*

- *If $0 \leq \boldsymbol{n} \leq m$, we have:*

$$p(S_L) \geq \frac{\prod_{k=1}^{\boldsymbol{n}} \frac{\lambda}{\mu+\sum_{d=1}^{k-1}\tau_d}}{1+\sum_{j=1}^{\infty}\left(\prod_{k=1}^{j}\frac{\lambda}{\mu+\sum_{d=1}^{k-1}\tau_d}\right)}$$

- *If $\boldsymbol{n} \geq m$, then:*

$$p(S_L) \leq \frac{\prod_{k=1}^{\boldsymbol{n}} \frac{\lambda}{\mu+\sum_{d=1}^{k-1}\tau_d}}{1+\sum_{j=1}^{\infty}\left(\prod_{k=1}^{j}\frac{\lambda}{\mu+\sum_{d=1}^{k-1}\tau_d}\right)}$$

The lower and upper bounds are tight when all τ_d are same, or when λ/μ approaches to zero or infinity.

Proof. In order to obtain the desired result, we firstly consider the corresponding to the R-Ticket Queue *physical queue*. That is, a normal queue with reneging customers, with no tickets. The transition diagram of such a queue appears in Figure 4.3.

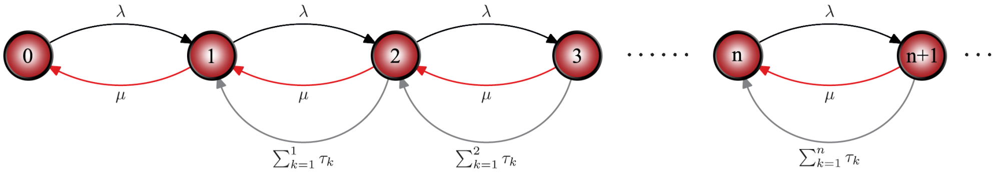


Figure 4.3 Transition Diagram for Single-server Physical Queue

We assume that a non-negative integer n denotes the system state, so that n describes the number of customers who is still in the system. Additionally, we denote as $p^p(n)$ the steady state probability for state n in the physical queue.

From the balance equations of states (0) and (0), we obtain:

$$\lambda p^p(0) = \mu p^p(1) \quad \text{or} \quad p^p(1) = \frac{\lambda}{\mu}p^p(0) \tag{4.2}$$

$$\lambda p^p(0) + \left(\mu + \sum_{d=1}^{1} \tau_d\right) p^p(2) = (\mu + \lambda) p^p(1) \tag{4.3}$$

Substituting (4.2) into (4.3), we get:

$$p^p(2) = \frac{\lambda}{\mu + \sum_{d=1}^{1} \tau_d} p^p(1) = \frac{\lambda}{\mu + \sum_{d=1}^{1} \tau_d} \frac{\lambda}{\mu} p^p(0)$$

Following a similar, procedure, we have:

$$p^p(n) = \frac{\lambda}{\mu + \sum_{d=1}^{n-1} \tau_d} p^p(n-1) = \prod_{k=1}^{n} \frac{\lambda}{\mu + \sum_{d=1}^{k-1} \tau_d} p^p(0) \tag{4.4}$$

Hence, using the normalization equation, we get:

$$1 = \sum_{n=0}^{\infty} p^p(n) = \left(1 + \sum_{n=1}^{\infty} \left(\prod_{k=1}^{n} \frac{\lambda}{\mu + \sum_{d=1}^{k-1} \tau_d} \right)\right) p^p(0) \tag{4.5}$$

From (4.5), we obtain $p^p(0)$. Thus, substituting this result to (4.4), we have:

$$p^p(n) = \frac{\prod_{k=1}^{n} \frac{\lambda}{\mu + \sum_{d=1}^{k-1} \tau_d}}{1 + \sum_{n=1}^{\infty} \left(\prod_{k=1}^{n} \frac{\lambda}{\mu + \sum_{d=1}^{k-1} \tau_d} \right)}, \quad n = 0, 1, \ldots \tag{4.6}$$

So far, we have obtained the steady state probabilities for each state in the corresponding physical queue of R-Ticket Queue. Now, we consider the R-Ticket Queue, where we will work with the super states we have defined. Hence, primarily, from the balance of super state S_0 we get:

$$\lambda p(S_0) = \mu p(S_1) \quad \text{or} \quad p(S_1) = \frac{\lambda}{\mu} p(S_0) \tag{4.7}$$

We shall take a look in Figure 4.2 and consider the balance of super state S_1. Equation (4.7) shows that the transition from super state S_0 to super state S_1 is balanced with those from super state S_1 to super state S_0. Therefore, the transitions from super state S_1 to super state S_2 will also be balanced with those from super state S_2 to super state S_1. This leads to the equation:

$$\left(\mu + \sum_{d=1}^{\infty} \tau_d\right) p(S_2) = \lambda p(S_1)$$

Furthermore, notice that each state in super state S_2, transfers to a state in super state S_1 at reneging rate $\tau_1, \tau_2, \ldots$, which are all no less than τ_1. Mathematically, we can write $\mu + \sum_{d=1}^{1} \tau_d \leq \mu + \sum_{d=1}^{\infty} \tau_d$. That is, using the last equation, we obtain:

$$\left(\mu + \sum_{d=1}^{1} \tau_d\right) p(S_2) \leq \lambda p(S_1), \quad or$$

$$p(S_2) \leq \frac{\lambda}{\mu + \sum_{d=1}^{1} \tau_d} p(S_1) = \frac{\lambda}{\mu + \sum_{d=1}^{1} \tau_d} \frac{\lambda}{\mu} p(S_0)$$

Following similar procedures, we get:

$$p(S_n) \leq \frac{\lambda}{\mu + \sum_{d=1}^{n-1} \tau_d} p(S_{n-1}) \leq \prod_{k=1}^{n} \frac{\lambda}{\mu + \sum_{d=1}^{k-1} \tau_d} p(S_0) \tag{4.8}$$

We now combine (4.5) and (4.8) and hence, we obtain:

$$\left(1 + \sum_{n=1}^{\infty} \left(\prod_{k=1}^{n} \frac{\lambda}{\mu + \sum_{d=1}^{k-1} \tau_d}\right)\right) p^p(0) = \sum_{n=0}^{\infty} p^p(n) = 1$$
$$= \sum_{n=0}^{\infty} p(S_n) \leq \left(1 + \sum_{n=1}^{\infty} \left(\prod_{k=1}^{n} \frac{\lambda}{\mu + \sum_{d=1}^{k-1} \tau_d}\right)\right) p(S_0)$$

That is, we have: $p(S_0) \geq p^p(0)$. Using (4.2) and (4.7), we get in addition that $p(S_1) \geq p^p(1)$. However, since $\sum_{n=0}^{\infty} p(S_n) = \sum_{n=0}^{\infty} p^p(n) = 1$, we cannot have $p(S_n) > p^p(n)$ for all n. Thus, there is some m, where $p(S_m) \leq p^p(m)$, and then, from (4.4) and (4.8), we obtain:

$$p(S_{m+1}) \leq \frac{\lambda}{\mu + \sum_{d=1}^{m} \tau_d} p(S_m) \leq \frac{\lambda}{\mu + \sum_{d=1}^{m} \tau_d} p^p(m) = p^p(m+1)$$

This leads to the inference that $p(S_n) \leq p^p(n)$ for all $n \geq m$. Finally, the tightness of the bounds can be easily verified. □

An observation we can make about Proposition 4.1.1, is that the threshold m highly depends on the reneging rate series τ_d, as well as the system traffic low λ/μ.

Proposition 4.1.2. *For single-server R-Ticket Queues,*

$$1 - \left(1 - \frac{1}{1 + \sum_{j=1}^{\infty}\left(\prod_{k=1}^{j} \frac{\lambda}{\mu + \sum_{d=1}^{k-1} \tau_d}\right)}\right) \cdot \frac{\mu}{\lambda}$$

is a lower bound for its total reneging percentage. The bound is tight when all τ_d are same, or when λ/μ approaches zero or infinity.

Proof. This is a straightforward corollary of Proposition 4.1.1. More precisely, let us examine the single server R-Ticket Queue when the system is in *steady state*. Then, the average number of customers in the queue increases by 1 with arrival rate λ, while it decreases by 1 with both the service rate, $(1-p(S_0))\mu$, and the average reneging rate. Consequently, the total reneging percentage is:

$$\frac{\lambda - (1 - p(S_0))\mu}{\lambda} = 1 - (1 - p(S_0))\frac{\mu}{\lambda}$$

Proposition 4.1.1 gives us that $p(S_0) \geq p^p(0)$, and $p^p(0)$ is given by (4.6). Therefore, the lower value of the total reneging percentage can be given easily by substituting $p^p(0)$ into it. Hence, we obtained simply the lower bound of total reneging percentage for the single-server R-Ticket Queue. □

4.1.1 The Modification of R-Ticket Queue

We have considered during the proof of Proposition 4.4.1 the physical R-Ticket Queue, which is the normal queue with reneging customers. We shall make a comparison with the R-Ticket Queue in order to investigate their relation in respect of their total reneging percentage.

As a matter of fact, the customers in the R-Ticket Queue renege at rates depending on their ticket position, which are higher than those in the corresponding physical queue. That is because, τ_d increases faster for a large d, and generally, the ticket position is higher than the number of real customers in the system. Hence, intuitively, we can say that the total reneging percentage of the physical queue is a lower bound for that of the R-Ticket Queue.

In addition, note that, when all τ_d are the same, the reneging rates are independent of queuing positions and as a consequence, the R-Ticket Queue and the corresponding physical are identical. Moreover, when λ/μ approaches infinity, that is the service rate is relatively negligible, the total reneging rates tend to be 1 in both systems. There is also the case when λ/μ approaches zero, which means the arrival rate is relatively negligible, and hence the total reneging rates tend to be 0 in both systems.

In order to investigate further the total reneging percentage of the R-Ticket Queue, we have performed simulation studies with various parameters. Especially, we normalized the customers' arrival rate, $\lambda = 1$, and we varied the service rate μ in interval $[0.5, 2]$; so that the traffic load ranges from 50% to 200%.

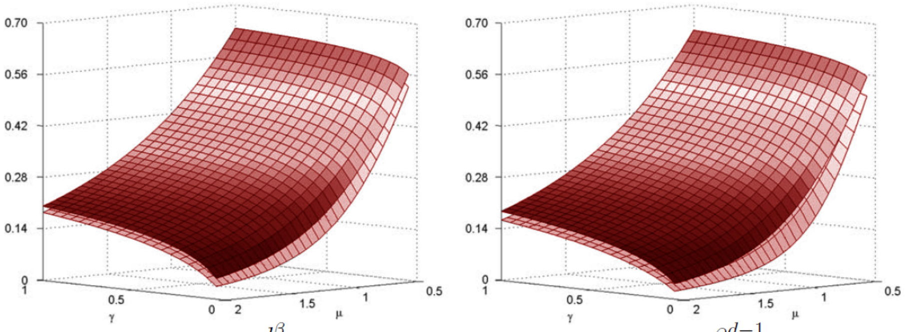
We assume that there are two general types of reneging rates in the simulation; so, for $d \in \{1, 2, \ldots\}$ we have:

$$\tau_d = \gamma \cdot d^{\beta} \tag{4.9}$$

$$\tau_d = \gamma \cdot \beta^{d-1} \tag{4.10}$$

Have in mind that in both cases, $\gamma = \tau_1 \geq 0$ is the reneging rate for the waiting customer with $d = 1$ and $\beta \geq 1$ is required to satisfy (4.1). As for a special case, we have $\beta = 1$, so that (4.9) becomes linear and (4.10) a constant reneging rate. We cover a large variety of possibilities by randomly varying $\gamma \in [0, 1]$ and $\beta \in [1, 2]$ in the simulation study.

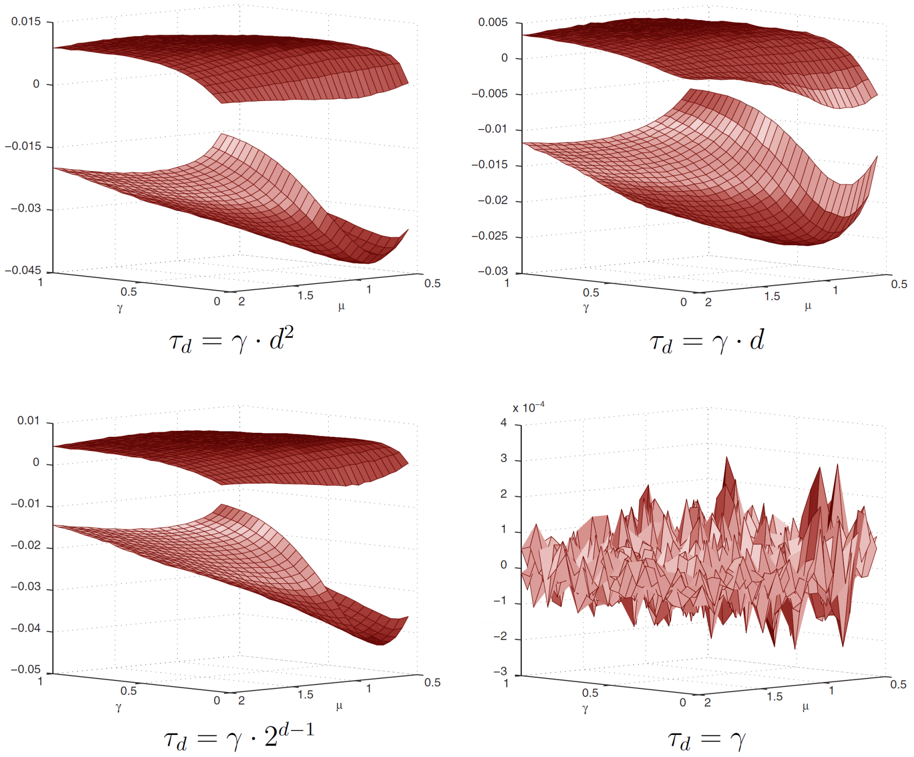
In Figure 4.4, we present graphically the total reneging percentage according to our simulation; note that, for each combination choice of μ, τ_d series, γ and β, the R-Ticket Queue is simulated 1000 times and each time, 1 billion customers are served. Intuitively, we expect that as γ and β increase, so does the total reneging percentage, while for higher values of μ we have the opposite. In the worst cases of our simulation study, there are more than 60% of customers who leave the system without obtaining service.



$\tau_d = \gamma \cdot d^{\beta}$ $\tau_d = \gamma \cdot \beta^{d-1}$

Figure 4.4 Simulated Total Reneging Percentage for the Single-server R-Ticket Queue. We only show the cases of $\beta = 1$ and $\beta = 2$ (lower and upper surfaces respectively).

As mentioned before, the extremely large state space of R-Ticket Queue causes difficulties. The reason why the state space is so huge, lies on the fact that reneging customers can be anywhere in the queue. Thus, if all reneging customers were packed together in the queuing positions, there could be a significant reduce of the state space. This idea led to experiments with two options to pack all the reneging customers together: one with all of them packed at the head of the queue, and another at the tail. We simulated both of them for the same range of parameters as for the R-Ticket Queue. In Figure 4.5, we can see the results that show which one of these queues is closer to the R-Ticket Queue. The upper surfaces show the total reneging percentage differences between the R-Ticket Queue and the queue with the reneging customers packed all at the head, while the lower surfaces indicate the same for R-Ticket Queue and the queue with all reneging customers packed together at the tail. The last graphic shows that in the case of $\tau_d = \gamma$, the differences are negligible. That is because the customer reneging rate does not depend on the queueing position.



$\tau_d = \gamma \cdot d^2$

$\tau_d = \gamma \cdot d$

$\tau_d = \gamma \cdot 2^{d-1}$

$\tau_d = \gamma$

Figure 4.5 Total Reneging Percentage Differences Between R-Ticket Queue and Two Candidate Models for its Approximation

As we can see, the winning model of our simulation is the one where all reneging customers are packed at the head of the queue. In fact, the differences between the total reneging percentages of these two queues lie in a tight range of -0.0065 to 0.0114. As for the model with all reneging customers packed at the tail, the differences are much wider, with the largest more than 0.0424. This corresponds to a relative difference of 10.5%.

As a consequence, we choose the first candidate model in order to approximate the R-Ticket Queue. That is, thereafter, we will call the queue with all reneging customers packed at the head the *modified R-Ticket Queue*. The important outcome we should highlight is that the total reneging percentage of single-server R-Ticket Queue can be approximated by that of single-server modified R-Ticket Queue with the same parameters. Thus, we shall analyze the modified R-Ticket Queue in the next section, because it is a model we can work with.

4.2 Analysis of the Modified R-Ticket Queue

In the first place, we should describe the form of the states we have in modified R-Ticket Queue; since the modification we made is to put all reneging customers at the head of the queue, we can represent its state by a two-dimensional vector $\boldsymbol{n} = (L, \nu)$, where L represents the number of customers in the system and ν is the number of customers that have reneged but their tickets are still counted in the queue. Note that, L customers in the system means 1 in service and $L-1$ waiting in queue.

The state transitions in this kind of system are achieved with three ways; assume that we are in state (L, ν). Then, if a customer completes hers/his service, this state changes to $(L-1, 0)$, so that we have one less customer waiting and at the same time all reneging customers' tickets are released. Additionally, an arrival of a customer will change the state to $(L+1, \nu)$, while after a customer's reneging we transit to $(L-1, \nu+1)$ (one less customer waiting, one more ticket counted at the head which will be deleted after the next service completion). One more information we obtain from the state (L, ν) is that the corresponding ticket position equals to $L + \nu$. In Figure 4.6 below, we can see the state transition diagram of the modified R-Ticket Queue.

We were interested in making a modification which decreases significantly the state space. In the modified R-Ticket Queue, we observe that for a given l, there are $(l^2 + l + 2)/2$ possible states if the ticket position of the next arrival customer is no more than l, that is if $L + \nu \leq l$. For example, when

$l = 3$, there are $(3^2 + 3 + 2)/2 = 7$ possible states in the queue, (0,0), (1,0), (1,1), (1,2), (2,0), (2,1) and (3,0). If we compare this number with the 2^l possible states of the R-Ticket Queue, then we see that we have achieved our purpose; it is an important reduce especially when l is not too small.

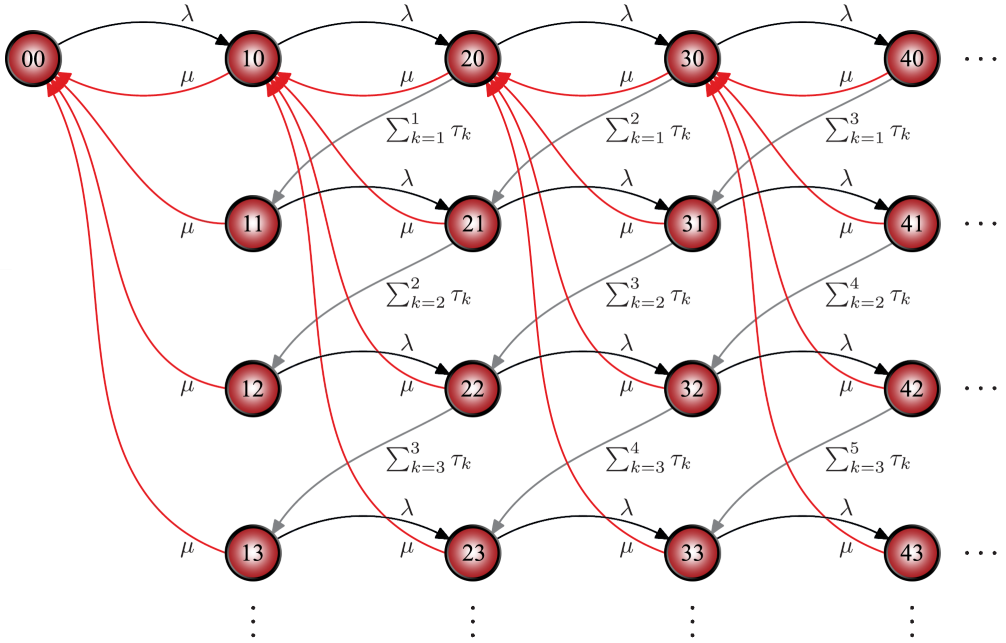


Figure 4.6 Transition Diagram of the Modified R-Ticket Queue

Let us denote as $p^m(\boldsymbol{n})$ the steady probability for state $\boldsymbol{n}$ of the modified R-Ticket Queue. We can obtain the balance equations quite easily:

$$p^m(0,0) \cdot \lambda = \left(\sum_{j=0}^{+\infty} p^m(1,j) \right) \cdot \mu$$

$$p^m(L,0) \cdot \left(\lambda + \mu + \sum_{k=1}^{L-1} \tau_k \right) = p^m(L-1,0) \cdot \lambda + \left(\sum_{j=0}^{+\infty} p^m(L+1,j) \right) \cdot \mu, \quad L = 1, 2, \ldots$$

$$p^m(1,\nu) \cdot (\lambda + \mu) = p^m(2,\nu-1) \cdot \tau_\nu, \quad \nu = 1, 2, \ldots$$

$$p^m(L,\nu)\cdot\left(\lambda+\mu+\sum_{k=\nu+1}^{L+\nu-1}\tau_k\right)=p^m(L-1,\nu)\cdot\lambda+p^m(L+1,\nu-1)\cdot\sum_{k=\nu}^{L+\nu-1}\tau_k,$$
$$L=2,3,\ldots,\quad \nu=1,2,\ldots$$

As a matter of fact, these infinitely many linear equations can be solved approximately if we truncate their system into a finite number equations. Naturally, the results can be more accurate by keeping more equations in this new system, which, thereafter is called the *truncated R-Ticket Queue.*

The truncation we are about to make constitutes a reasonable approximation of the original model; indeed, the reneging rate τ_d can seen to be infinitely large when d is larger from a point and on. Thus, there is a level V where all customers renege immediately when they observe a ticket position no less than V. So, we cutoff the queue at this level. We also partition the state space into blocks, namely K_i, where $i = 0,\ldots,V$, and each of them represents the number of states that have exactly i customers in the system.

However, there are still infinitely many equations even after the truncation. Every K_i consists of infinitely many states, so we additionally aggregate in a super state, denoted as S_V, all the states where the ticket position is more than V. Therefore, S_V includes the states $\{(1,V-1),(1,V),\ldots\}$ from K_1, the states $\{(2,V-1),(2,V),\ldots\}$ from K_2 and so on. That is, the super state S_V will remain unchanged when a new customer comes. In Figure 4.7 we can see the illustration of the truncated R-Ticket Queue model.

Clearly, at this point we have a reduced finite set of system states, that is partitioned into $V+1$ blocks $K_0,\ldots,K_V$ as follows:

$$\begin{aligned}K_0&=\{(0,0)\},\\K_1&=\{(1,0),(1,1),\ldots,(1,V-2),S_V\},\\&\vdots\\K_l&=\{(l,0),(l,1),\ldots,(l,V-l)\},\\&\vdots\\K_V&=\{(V,0)\}\end{aligned}$$

According to the previous description, the transition rate matrix $\boldsymbol{Q}^t$ of the truncated R-Ticket Queue can be written as bellow:

$$\boldsymbol{Q}^t = \begin{bmatrix} -\boldsymbol{\lambda} & & & & & \\ \boldsymbol{\mu e'_V} & \boldsymbol{A_{11}} & \boldsymbol{A_{12}} & & & \\ & \boldsymbol{A_{21}} & \boldsymbol{A_{22}} & \boldsymbol{A_{23}} & & \\ & \vdots & \ddots & \ddots & \ddots & \\ & \boldsymbol{A_{V-1,1}} & & \boldsymbol{A_{V-1,V-2}} & \boldsymbol{A_{V-1,V-1}} & \boldsymbol{A_{V-1,V}} \\ & \boldsymbol{A_{V,1}} & & & \boldsymbol{A_{V,V-1}} & \boldsymbol{A_{V,V}} \end{bmatrix} \tag{4.11}$$

The matrix $\boldsymbol{Q}^t$ is of dimensions $\left(\frac{V^2+V+2}{2}\right) \times \left(\frac{V^2+V+2}{2}\right)$. Furthermore, $\boldsymbol{e}_V$ is the V-dimensional unit vector and $\boldsymbol{e}'_V$ its transpose. We also define the following matrices: $\boldsymbol{A_{01}}$, $\boldsymbol{A_{11}}$, $\boldsymbol{A_{j,1}}$, $\boldsymbol{A_{j,j+1}}$, $\boldsymbol{A_{j,j-1}}$, $\boldsymbol{A_{j,j}}$ of dimensions $1 \times V$, $V \times V$, $(V-j+1) \times V$,$(V-j+1) \times (V-j)$, $(V-j+1) \times (V-j+2)$ and $(V-j+1) \times (V-j+1)$ respectively.

$$\boldsymbol{A_{01}} = (\lambda, 0, \ldots, 0),$$

$$\boldsymbol{A_{11}} = \begin{bmatrix} -\lambda-\mu & & & \\ & \ddots & & \\ & & -\lambda-\mu & \\ & & & -\mu \end{bmatrix},$$

$$\boldsymbol{A_{j,1}} = \begin{bmatrix} 0 & \ldots & 0 & 0 \\ \vdots & & \vdots & \vdots \\ 0 & \ldots & 0 & 0 \\ 0 & \ldots & 0 & \lambda \end{bmatrix}, \quad \boldsymbol{A_{j,j+1}} = \begin{bmatrix} \lambda & & \\ & \ddots & \\ & & \lambda \\ 0 & \ldots & 0 \end{bmatrix},$$

$$\boldsymbol{A_{j,j-1}} = \begin{bmatrix} \mu & \sum_{k=1}^{j-1} \tau_k & & & \\ \vdots & & \ddots & & \\ \mu & & & \sum_{k=i}^{j-2+i} \tau_k & \\ \vdots & & & & \ddots \\ \mu & & & & & \sum_{k=V-j+1}^{V-1} \tau_k \end{bmatrix},$$

$$\boldsymbol{A_{j,j}} = \begin{bmatrix} -\lambda-\mu-\sum_{k=1}^{j-1} \tau_k & & & & \\ & \ddots & & & \\ & & -\lambda-\mu-\sum_{k=i}^{j-2+i} \tau_k & & \\ & & & \ddots & \\ & & & & -\lambda-\mu-\sum_{k=V-j+1}^{V-1} \tau_k \end{bmatrix}$$

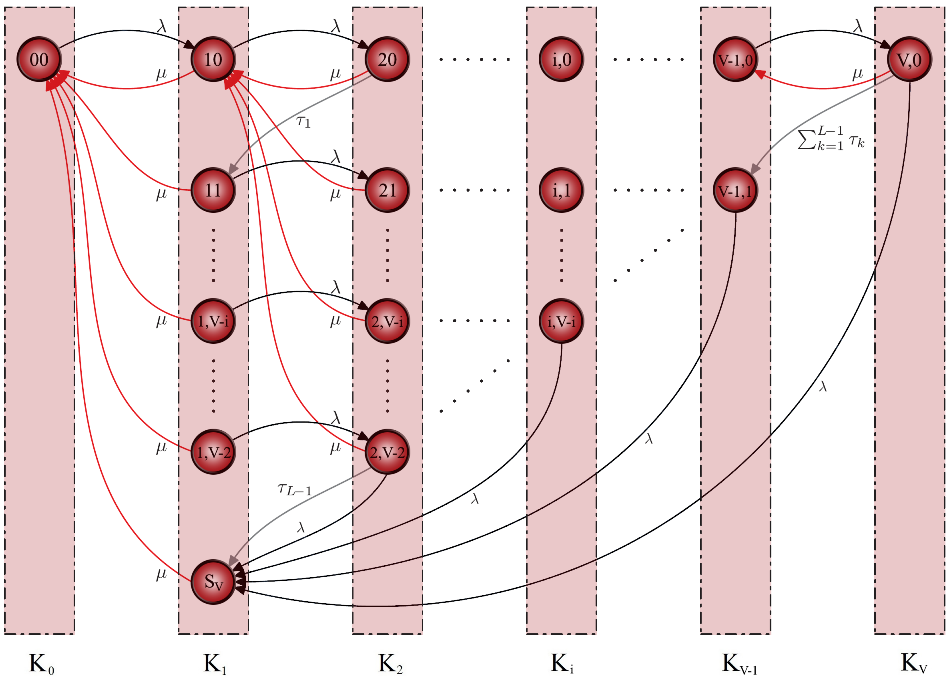


Figure 4.7 Transition Diagram of the Truncated R-Ticket Queue

The problem can be solved now as a QBD process; let the steady state probabilities for the states in block K_i be $\boldsymbol{p}_i^t$, with $i = 0, 1, \ldots, V$. So, generally, the steady state probabilities are described by vector $\boldsymbol{p}^t = (\boldsymbol{p}_0^t, \ldots, \boldsymbol{p}_V^t)$. In order to obtain $\boldsymbol{p}^t$, we start the process by writing down the balance equations, $\boldsymbol{p}^t\boldsymbol{Q}^t = \boldsymbol{0}$, right below:

$$-\lambda\boldsymbol{p}_0^t + \mu\boldsymbol{p}_1^t\boldsymbol{e}_V' = 0 \tag{4.12}$$

$$\sum_{i=0}^{V}\boldsymbol{p}_i^t\boldsymbol{A}_{i,1} = 0 \tag{4.13}$$

$$\boldsymbol{p}_{i-1}^t\boldsymbol{A}_{i-1,i} + \boldsymbol{p}_i^t\boldsymbol{A}_{i,i} + \boldsymbol{p}_{i+1}^t\boldsymbol{A}_{i+1,i}, \quad i = 2, \ldots, V-1 \tag{4.14}$$

$$\boldsymbol{p}_{V-1}^t\boldsymbol{A}_{V-1,V} + \boldsymbol{p}_V^t\boldsymbol{A}_{V,V} = 0 \tag{4.15}$$

The next phase of the process requires to define the following:

$$\boldsymbol{R}_{V-1,V} = -\boldsymbol{A}_{V-1,V}(\boldsymbol{A}_{V,V})^{-1} \tag{4.16}$$

$$\boldsymbol{R}_{i-1,i} = -\boldsymbol{A}_{i-1,i}(\boldsymbol{A}_{i,i} + \boldsymbol{R}_{i,i}\boldsymbol{A}_{i+1,i})^{-1}, \quad i = 2, \ldots, V-1 \tag{4.17}$$

$$\boldsymbol{R}_{1,i} = \prod_{j=1}^{i-1} \boldsymbol{R}_{j,j+1}, \quad i = 2, \ldots, V \tag{4.18}$$

$$\boldsymbol{R}_{0,1} = -\boldsymbol{A}_{0,1}\left(\boldsymbol{A}_{1,1} + \sum_{j=2}^{V} \boldsymbol{R}_{1,j}\boldsymbol{A}_{j,1}\right)^{-1} \tag{4.19}$$

$$\boldsymbol{R}_{i} = \prod_{j=1}^{i-1} \boldsymbol{R}_{j,j+1} = \boldsymbol{R}_{0,1}\boldsymbol{R}_{1,i}, \quad i = 2, \ldots, V \tag{4.20}$$

Then, based on (4.15) and (4.16), we get:

$$\boldsymbol{p}_V^t = -\boldsymbol{p}_{V-1}^t\boldsymbol{A}_{V-1,V}(\boldsymbol{A}_{V,V})^{-1} = \boldsymbol{p}_{V-1}^t\boldsymbol{R}_{V-1,V} \tag{4.21}$$

That is, we have $\boldsymbol{p}_{i+1}^t = \boldsymbol{p}_i^t\boldsymbol{R}_{i,i+1}$, $i \in \{2, 3, \ldots, V-1\}$. Now, according to (4.14):

$$\begin{aligned}\boldsymbol{p}_{i-1}^t\boldsymbol{A}_{i-1,i} + \boldsymbol{p}_i^t(\boldsymbol{A}_{i,i} + \boldsymbol{R}_{i,i+1}\boldsymbol{A}_{i+1,i}) &= 0\\ \boldsymbol{p}_i^t = -\boldsymbol{p}_{i-1}^t\boldsymbol{A}_{i-1,i}(\boldsymbol{A}_{i,i} + \boldsymbol{R}_{i,i+1}\boldsymbol{A}_{i+1,i})^{-1}&\end{aligned}$$

Thus, using also (4.17), we obtain $\boldsymbol{p}_i^t = \boldsymbol{p}_{i-1}^t\boldsymbol{R}_{i-1,i}$, that can be written better as:

$$\boldsymbol{p}_{i+1}^t = \boldsymbol{p}_i^t\boldsymbol{R}_{i,i+1}, \quad i = 2, 3, \ldots, V-1 \tag{4.22}$$

We can obtain from (4.22) that $\boldsymbol{R}_{i,i+1} = (\boldsymbol{p}_i^t)^{-1}\boldsymbol{p}_{i+1}^t$. Substituting this result into (4.18), we get:

$$\boldsymbol{p}_i^t = \boldsymbol{p}_1^t\prod_{j=1}^{i-1}\boldsymbol{R}_{j,j+1} = \boldsymbol{p}_1^t\boldsymbol{R}_{1,i}, \quad i = 2, 3, \ldots, V-1 \tag{4.23}$$

Based on (4.13) and (4.23), we have consecutively:

$$\boldsymbol{p}_0^t\boldsymbol{A}_{0,1} + \boldsymbol{p}_1^t\boldsymbol{A}_{1,1} + \sum_{i=2}^{V}\boldsymbol{p}_i^t\boldsymbol{A}_{i,1} = 0$$

$$\boldsymbol{p}_0^t\boldsymbol{A}_{0,1} + \boldsymbol{p}_1^t\boldsymbol{A}_{1,1} + \sum_{i=2}^{V}\boldsymbol{p}_1^t\boldsymbol{R}_{1,i}\boldsymbol{A}_{i,1} = 0$$

$$\boldsymbol{p}_0^t\boldsymbol{A}_{0,1} + \boldsymbol{p}_1^t(\boldsymbol{A}_{1,1} + \boldsymbol{R}_{1,2}\boldsymbol{A}_{2,1} + \ldots + \boldsymbol{R}_{1,V}\boldsymbol{A}_{V,1}) = 0$$

These equations, together with (4.19), lead us to obtain $\boldsymbol{p}_1^t$. Hence, we have:

$$\boldsymbol{p}_1^t = -\boldsymbol{p}_0^t\boldsymbol{A}_{0,1}\left(A_{1,1} + \sum_{j=2}^{V}\boldsymbol{R}_{1,j}\boldsymbol{A}_{j,1}\right)^{-1} = \boldsymbol{p}_0^t\boldsymbol{R}_{0,1} \tag{4.24}$$

At this point, (4.20) gives us that $\boldsymbol{R}_{1,i} = \boldsymbol{R}_{0,1}^{-1}\boldsymbol{R}_i$ and thus, (4.23) becomes $\boldsymbol{p}_i^t = \boldsymbol{p}_1^t\boldsymbol{R}_{0,1}^{-1}\boldsymbol{R}_i$. Substituting into the new form of (4.23) $\boldsymbol{p}_1^t$ from (4.24), we get:

$$\boldsymbol{p}_i^t = \boldsymbol{p}_0^t\prod_{j=0}^{i-1}\boldsymbol{R}_{j,j+1} = \boldsymbol{p}_0^t\boldsymbol{R}_i, \quad i = 2, 3, \ldots, V-1 \tag{4.25}$$

The final step for obtaining the steady state probabilities in truncated R-Ticket Queue is the normalization equation:

$$\boldsymbol{p}_0^t + \boldsymbol{p}_1^t\boldsymbol{e}_V' + \ldots + \boldsymbol{p}_V^t\boldsymbol{e}_1' = 1$$

Based on (4.25), we have:

$$\boldsymbol{p}_0^t\left(1 + \sum_{i=1}^{V}\boldsymbol{R}_i\boldsymbol{e}_{V+1-i}'\right) = 1$$

Therefore, we obtain straightforward from the last equation and (4.25):

$$\boldsymbol{p}_0^t = \frac{1}{1 + \sum_{i=1}^{V}\boldsymbol{R}_i\boldsymbol{e}_{V+1-i}'} \tag{4.26}$$

$$\boldsymbol{p}_i^t = \frac{\boldsymbol{R}_i}{1 + \sum_{i=1}^{V}\boldsymbol{R}_i\boldsymbol{e}_{V+1-i}'}, \quad i = 1, \ldots, V \tag{4.27}$$

Equations (4.26) and (4.27) give the steady state probabilities of the truncated R-Ticket Queue. This finding also provides us with the *total reneging percentage of the truncated model*, that is: $1 - (1 - \boldsymbol{p}_0^t)\mu/\lambda$.

The explanation here is not complicated. When the system is in steady state, the average number of customers in the queue increases by 1 with every arrival at rate λ. At the same time, it decreases by 1 with both the service rate $(1 - \boldsymbol{p}_0^t)\mu$ and the average reneging rate. Thus, based on the system balance, the average reneging rate is $\lambda - (1 - \boldsymbol{p}_0^t)\mu$. So, the reneging percentage formula for the truncated R-Ticket Queue is given by:

$$1 - (1 - \boldsymbol{p}_0^t)\frac{\mu}{\lambda}$$

As mentioned before, the total reneging percentage of single-server R-Ticket Queue can be approximated by that of single-server modified R-Ticket Queue with same parameters. Truncated R-Ticket Queue is actually the modified one, but with a moderate number of equations so that can be solved accurately. Formulas (4.26) and (4.27), in combination with the total reneging formula can lead efficiently to computational results. We executed calculations for the same range of parameters as used in the simulation. Especially, when the customer arrival rate λ is fixed at 1, the service rate μ is varied from 0.5 to 2, and the reneging rate is computed as in (4.9) and (4.10), with a γ variation of 0 to 1 and β varying from 1 to 2. We also need to select the cutoff level V for each set of parameters, in a way that the calculated total reneging percentage becomes stable. Indeed, the calculation results of total reneging percentage are very close to the simulation results, as the differences are all within ± 0.00007. This indicates that the formulas (4.26) and (4.27) are reliable.

The basic analysis for single-server R-Ticket Queue has been done. However, it is also interesting to study a generalization of this queue with many servers. Therefore, the next section is dedicated to an extension of the queue we analyzed, the multi-server R-Ticket Queue.

4.3 The Multi-server R-Ticket Queue

In this section, we describe the multi-server R-Ticket Queue in a quite brief way. Firstly, we shall introduce the form of the states. We assume that there are s servers. Thus, if L is the number of customers in the system, the state vector $\boldsymbol{n'}$ can be defined as following:

- If $L \leq s - 1$, then $\boldsymbol{n'} = (L)$;
- If $L = s$, then $\boldsymbol{n'} = (n'_1)$, where $n'_1 - s$ is the number of customers who have reneged since the last customer starts to receive service;
- If $L > s - 1$, then $\boldsymbol{n'} = (n'_1, \ldots, n'_{L-s+1})$, where: $n'_1 - s + 1 \geq 1$ is the ticket position of the first waiting customer, n'_i, $i = 2, \ldots, L - s$ is a positive integer which represents the ticket number difference between

the $(i-1)$th and the ith waiting customers and finally, n'_{L-s+1} denotes the difference between the ticket number of the last waiting customer and that to be issued to the next arriving customer.

We give an example for clarity; let $s = 4$ and $\boldsymbol{n'} = (5, 3, 4)$ a state in the multi-server R-Ticket Queue. Then, we can derive the subsequent information:

- There are 4 customers being served and 2 more customers waiting in queue. The number of waiting customers is verified by the number of components that the vector state has.
- The ticket position of the first waiting customer is $5 - 4 + 1 = 2$. That means, 1 customer before her/him has reneged.
- The ticket number difference between the first and the second waiting customers is 3. That is, 2 customers between them have reneged.
- For a new arriving customer, the ticket number difference between her/him and the second waiting customer is 4, which means that three customers after the second waiting customer have reneged.

The transition diagram of the multi-server R-Ticket Queue can be seen in Figure 4.8. We observe that it is quite similar to that of the single-server R-Ticket Queue.

We continue by defining super states for the multi-server R-Ticket Queue. Let super state $\widetilde{S}_L$, $L = 0, 1, \ldots$ represent the collection of all states in which there exist L customers in the system, waiting or being served. Then, when $L = \{0, 1, \ldots, s-1\}$, the corresponding super states $\widetilde{S}_0, \widetilde{S}_1, \ldots, \widetilde{S}_L$ contain only the state (L). On the other hand, $\widetilde{S}_L$ with $L = \{s, s+1, \ldots\}$ contains infinite states. As an example, we can see in Figure 4.8 that super state $\widetilde{S}_s$ contains states such as (s), $(s+1)$, ..., so we have an infinite amount of states.

We denote as $p^s(\widetilde{S}_L)$, $L = 0, 1, \ldots$ the steady state probability for each super state in the multi-server R-Ticket Queue. In the following, we represent two propositions which are similar to 4.1.1 and 4.1.2. Then, we show the way that the steady state probabilities of the multi-server R-Ticket Queue can be derived.

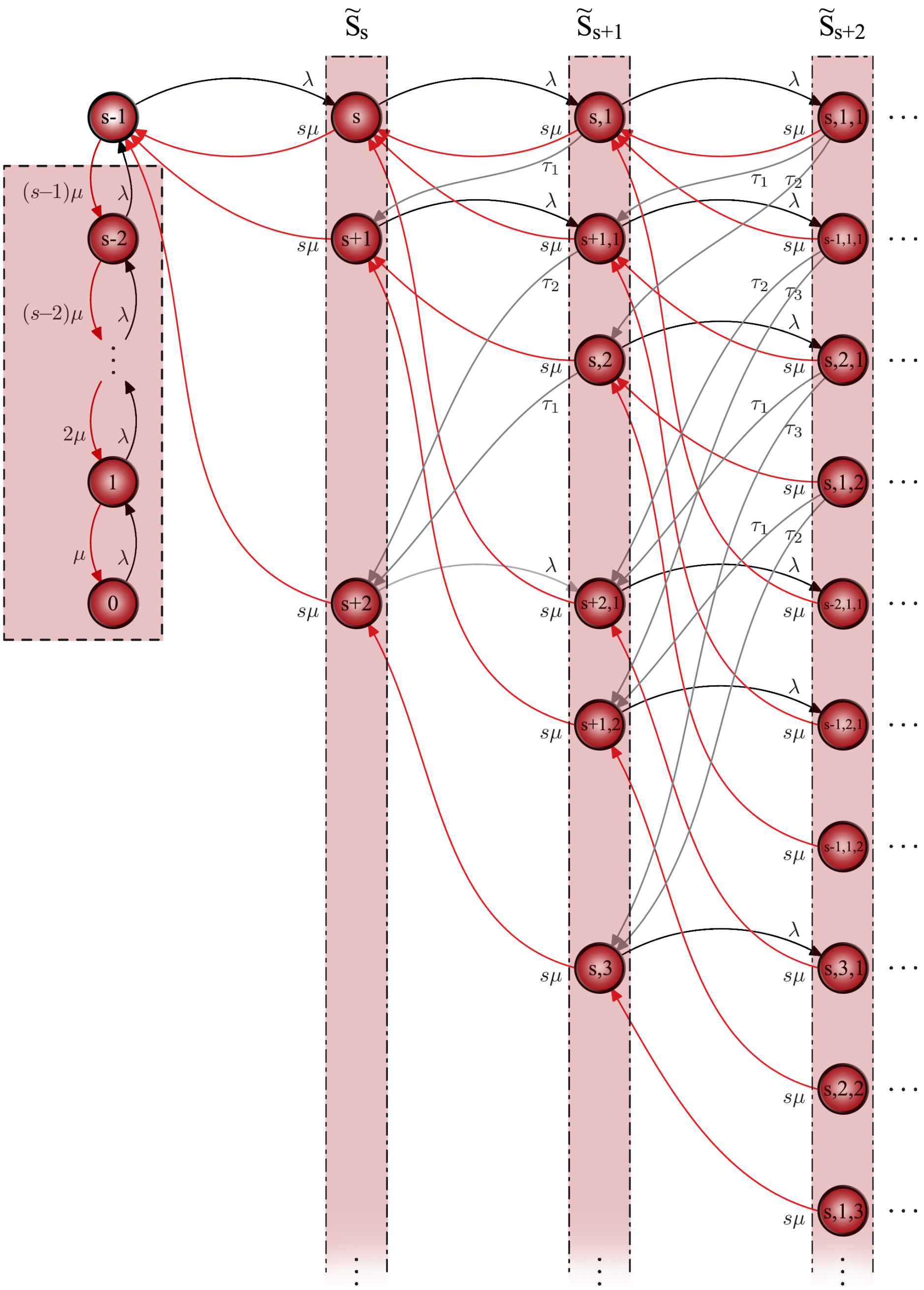


Figure 4.8 Transition Diagram for Multi-server R-Ticket Queue

Proposition 4.3.1. *For a multi-server R-Ticket Queue, there exists an integer $m \geq s$, such that:*

- *If $0 \leq \boldsymbol{n}' \leq s-1$, we have:*

$$p^s(\widetilde{S}_L) \geq \frac{\prod_{k=1}^{\boldsymbol{n}'} \frac{\lambda}{k\mu}}{1+\sum_{j=1}^{s}\left(\prod_{k=1}^{j}\frac{\lambda}{k\mu}\right)+\left(\prod_{k=1}^{s}\frac{\lambda}{k\mu}\right)\cdot\sum_{j=s+1}^{\infty}\left(\prod_{k=1}^{j-s}\frac{\lambda}{s\mu+\sum_{d=1}^{k}\tau_d}\right)}$$

- *If $s \leq \boldsymbol{n}' \leq m$, then:*

$$p^s(\widetilde{S}_L) \geq \frac{\prod_{k=1}^{\boldsymbol{n}'} \frac{\lambda}{k\mu}\cdot\left(\prod_{k=1}^{\boldsymbol{n}-s}\frac{\lambda}{s\mu+\sum_{d=1}^{k}\tau_d}\right)}{1+\sum_{j=1}^{s}\left(\prod_{k=1}^{j}\frac{\lambda}{k\mu}\right)+\left(\prod_{k=1}^{s}\frac{\lambda}{k\mu}\right)\cdot\sum_{j=s+1}^{\infty}\left(\prod_{k=1}^{j-s}\frac{\lambda}{s\mu+\sum_{d=1}^{k}\tau_d}\right)}$$

- *If $\boldsymbol{n}' > m$, we have:*

$$p^s(\widetilde{S}_L) \leq \frac{\prod_{k=1}^{\boldsymbol{n}'} \frac{\lambda}{k\mu}\cdot\left(\prod_{k=1}^{\boldsymbol{n}-s}\frac{\lambda}{s\mu+\sum_{d=1}^{k}\tau_d}\right)}{1+\sum_{j=1}^{s}\left(\prod_{k=1}^{j}\frac{\lambda}{k\mu}\right)+\left(\prod_{k=1}^{s}\frac{\lambda}{k\mu}\right)\cdot\sum_{j=s+1}^{\infty}\left(\prod_{k=1}^{j-s}\frac{\lambda}{s\mu+\sum_{d=1}^{k}\tau_d}\right)}$$

Proof. We will present a brief version of the proof, as it is quite similar to that in Proposition 4.1.1. Firstly, consider the corresponding to the multi-server R-Ticket Queue *physical queue*; that is, a normal queue with reneging customers and s servers. We assume that n stands for the state that there exist n customers in the system, waiting or being served. Moreover, we denote as $p^{mp}(n)$ the steady state probability of state n in the multi-server physical queue. From the state balance, we get:

$$\lambda p^{mp}(n) = (n+1)\mu p^{mp}(n+1), \quad n = 0, \ldots, s-2$$

$$\lambda p^{mp}(n) = \left(s\mu + \sum_{d=1}^{n-s+1} \tau_d\right) p^{mp}(n+1), \quad x = s-1, s, \ldots$$

Using the normalization equation, $\sum_{n=0}^{\infty} p^{mp}(n) = 1$, we can compute the steady probabilities for all states, in the same manner as before.

Following the same thoughts as in Proposition 4.1.1, from the balance of states and super states, and the fact that $\tau_1 \leq \tau_2 \leq \ldots$, we obtain for the multi-server R-Ticket Queue the following equations:

$$\lambda p^s(\widetilde{S}_n) = (n+1)\mu p^s(\widetilde{S}_{n+1}), \quad n = 0, \ldots, s-2$$

$$\lambda p^s(\widetilde{S}_n) \geq \left(s\mu + \sum_{d=1}^{n-s+1} \tau_d \right) p^s(\widetilde{S}_{n+1}), \quad n = s-1, s, \ldots$$

With a very similar deduction as seen in the proof of Proposition 4.1.1, we can obtain the lower and upper bounds for each state or super state in the multi-server R-Ticket Queue. □

Proposition 4.3.2. *For multi-server R-Ticket Queues,*

$$\frac{\left(\prod_{k=1}^{s} \frac{\lambda}{k\mu} \cdot \left(1 + \left(1 - \frac{s\mu}{\lambda}\right) \right) \cdot \sum_{j=s+1}^{\infty} \left(\prod_{k=1}^{j-s} \frac{\lambda}{s\mu + \sum_{d=1}^{k}} \tau_d \right) \right)}{1 + \sum_{j=1}^{s} \left(\prod_{k=1}^{j} \frac{\lambda}{k\mu} \right) + \left(\prod_{k=1}^{s} \frac{\lambda}{k\mu} \right) \cdot \sum_{j=s+1}^{\infty} \left(\prod_{k=1}^{j-s} \frac{\lambda}{s\mu + \sum_{d=1}^{\tau_d}} \right)} \tag{4.28}$$

is a lower bound for its total reneging percentage. The bound is tight when all τ_d are same, or when λ/μ approaches zero or infinity.

Proof. As expected, this is a straightforward consequence from Proposition 4.3.2. When the multi-server R-Ticket Queue system is in steady state, the average number of customers in the queue increases by 1 with any arriving customer at rate λ, and decreases by 1 with both the average service rate and the average reneging rate. As there are s servers in the system, the average service rate is given by:

$$\mu p^s(\widetilde{S}_1) + 2\mu p^s(\widetilde{S}_2) + \ldots + s\mu p^s(\widetilde{S}_s) + s\mu\big(p^s(\widetilde{S}_{s+1}) + p^s(\widetilde{S}_{s+2}) + \ldots\big)$$

Hence, the total reneging percentage is given by:

$$\frac{\lambda - \sum_{n=1}^{s}(n\mu p^s(\widetilde{S}_n)) - s\mu \sum_{n=s+1}^{\infty}(p^s(\widetilde{S}_n))}{\lambda}$$

Using Proposition 4.3.1 and deriving some computations, we have the lower bound shown in (4.28). The tightness of the bound can be easily verified. □

At this point, we search for the steady probabilities of states $\boldsymbol{n}'$ in the multi-server R-Ticket Queue. Let $p^s(\boldsymbol{n}')$ denote this probability for every state $\boldsymbol{n}'$. We begin with the states (0),. . . ,(s-2). As seen in Figure 4.8, they are the states in the dashed box. The derivation of their steady probabilities goes as follows;

- ∘ From the balance of state (0) we get: $\lambda p^s(0) = \mu p^s(1)$
- ∘ The balance of state (1) gives: $\lambda p^s(0) + 2\mu p^s(2) = (\lambda + \mu)p^s(1)$
- ∘ The two previous equations give: $\lambda p^s(1) = 2\mu p^s(2)$

With very similar procedures, we assume the balance of (2),(3),..., and therefore, we obtain:

$$p^s(k) \cdot \lambda = p^s(k+1) \cdot (k+1)\mu, \quad k = 0, \ldots, s-2 \tag{4.29}$$

We derive an interesting consequence from (4.29); that is, if we discard the states (0),...,$(s-2)$, the remaining system is still in balance. An even more appealing observation is that, if we look carefully both figures 4.2 and 4.8, we can see that there is a *bijective mapping* between multi-server system state $\boldsymbol{n'} = (n'_1, \ldots, n'_L)$ outside the dashed box in Figure 4.8 and the single-server system $\boldsymbol{n} = (n_1, \ldots, n_L) = (n'_1 - s + 1, n'_2, \ldots, n'_L)$ in Figure 4.2. Undoubtedly, if we cover the dashed box in Figure 4.8, the diagram looks identical to that of Figure 4.2. Thus, this bijective mapping gives us the opportunity to derive the steady state probabilities of the multi-server R-Ticket Queue directly from those of the single-server R-Ticket Queue.

For this purpose, we denote as $p^*(\boldsymbol{n})$ the steady state probabilities of state $\boldsymbol{n}$ of the single-server R-Ticket Queue with service rate $s\mu$. Additionally, for all the states outside the dashed box in Figure 4.8, we define:

$$p^s(n'_1, n'_2, \ldots, n'_L) = \xi \cdot p^*(n'_1 - s + 1, n'_2, \ldots, n'_L), \tag{4.30}$$

where $0 < \xi < 1$. That is, the total probability of these states is ξ. As a consequence, the total probability of the states inside the dashed box in Figure 4.8 is $1 - \xi$. Thus, equation (4.29) gives:

$$p^s(k) = p^s(s-1) \cdot \frac{(s-1)!}{k!} \cdot \left(\frac{\mu}{\lambda}\right)^{s-1-k} = \xi p^*(0) \cdot \frac{(s-1)!}{k!} \cdot \left(\frac{\mu}{\lambda}\right)^{s-1-k}, \tag{4.31}$$

$$k = 0, \ldots, s-2$$

Therefore, we get from (4.31):

$$\sum_{k=0}^{s-2} p^s(k) = \xi p^*(0) \cdot (s-1)! \cdot \sum_{k=0}^{s-2} \left[\frac{1}{k!} \cdot \left(\frac{\mu}{\lambda}\right)^{s-1-k}\right] = 1 - \xi \tag{4.32}$$

Equation (4.32) provides us with the value of ξ, that is finally given by:

$$\xi = \frac{1}{1 + p^*(0) \cdot (s-1)! \cdot \sum_{k=0}^{s-2} \left[\frac{1}{k!} \cdot \left(\frac{\mu}{\lambda}\right)^{s-1-k}\right]} \tag{4.33}$$

At this point, we introduce the next proposition, which gives the steady state probabilities and the total reneging percentage of the multi-server R-Ticket Queue.

Proposition 4.3.3. *The steady state probabilities of the multi-server R-Ticket Queue can be derived directly using the formulas (4.30), (4.31) and (4.33). Moreover, its total reneging percentage is given by:*

$$\xi\big(1 - \big(1 - p^*(0)\big)s\mu/\lambda\big) = \frac{1 - \big(1 - p^*(0)\big)s\mu/\lambda}{1 + p^*(0) \cdot (s-1)! \cdot \sum_{k=0}^{s-2} \left[\frac{1}{k!} \cdot \left(\frac{\mu}{\lambda}\right)^{s-1-k}\right]} \tag{4.34}$$

Recall that $p^*(0)$ is the steady probability of state (0) in the single-server R-Ticket Queue with rate $s\mu$; hence, it can be derived too.

Proof. We have already analyzed the derivation of equations (4.30), (4.31) and (4.33). We shall present the computations that lead to the total reneging percentage of the multi-server R-Ticket Queue.

Therefore, when the system is in steady state, the average number of customers in the queue increases by one with each arrival at rate λ. On the other hand, it decreases by 1 at rate $k\mu$ with probability $p^s(k)$, where $k = 1, \dots, s-1$, at rate $s\mu$ with probability $\xi - p^s(s-1)$, and at the average reneging rate. Hence, the total reneging percentage is derived as:

$$\begin{aligned}
&\frac{\lambda - \sum_{k=1}^{s-1}\big(k\mu \cdot p^s(k)\big) - s\mu \cdot \big(\xi - p^s(s-1)\big)}{\lambda} \\
&= \frac{\lambda - \sum_{k=1}^{s-1}\big(\lambda \cdot p^s(k-1)\big) - s\mu \cdot \big(\xi - \xi \cdot p^*(0)\big)}{\lambda} \\
&= \frac{\lambda - \lambda \cdot (1-\xi) - s\mu \cdot (\xi - \xi \cdot p^*(0))}{\lambda} \\
&= \xi\left(1 - (1 - p^*(0)) \cdot \frac{s\mu}{\lambda}\right),
\end{aligned}$$

where the first equation is based on formulas (4.30) and (4.31), and the second one has been derived by formula (4.33). □

We conclude this section with some more details about the results we obtained for the multi-server R-Ticket Queue. At first, note that the total reneging percentage lower bound given by Proposition 4.3.2 is compatible with that of single server R-Ticket Queue in Proposition 4.1.2. That is because the later is a special case of the former with $s = 1$ and $\xi = 1$.

We should also highlight that we cannot obtain an accurate value of $p^*(0)$, as it is approximated by the total reneging percentage of the modified R-Ticket Queue. So, if we use this approximated value in equation (4.33), the total reneging percentage of the multi-server R-Ticket Queue will also be an approximated result.

By carrying out some experiments, we can see that generally, the multi-server total reneging percentage formula leads to approximated results with reasonable errors. Specifically, we apply the formula (4.33) with both true total reneging percentage and approximated total reneging percentage at $e = 5\%$ error. In Figure 4.9 below, we can see the relative errors for cases of $s = 10$ and $s = 50$ (the upper and lower surfaces respectively).

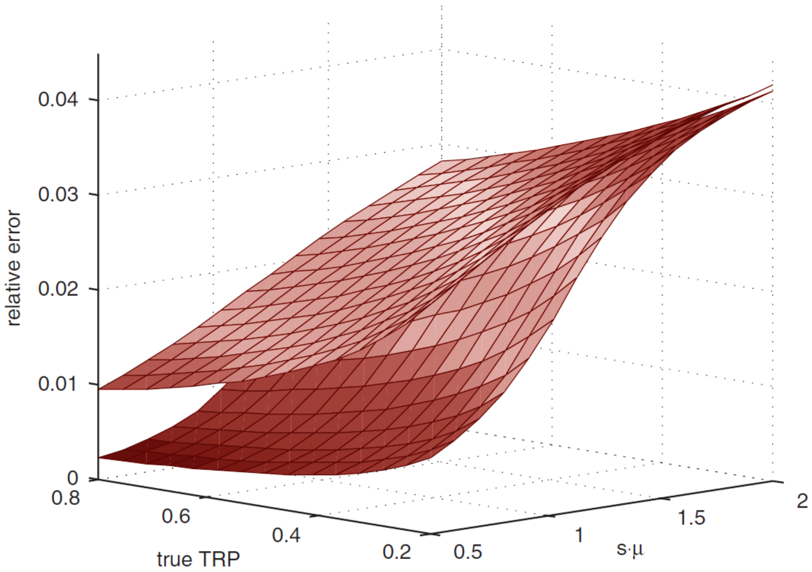


Figure 4.9 Relative Errors of Total Reneging Percentages

We observe in Figure 4.9 that the relative error is smaller when s is higher and μ is lower, for given total service rate $s\mu$. Moreover, the approximation is better with higher true total reneging percentages.

The analysis of the multi-server R-Ticket Queue has been completed. The next section concludes our findings with a brief and interesting discussion about how we can use the obtained results for our advantage.

4.4 Total Reneging Percentage Improvement

In this chapter, we have studied the R-Ticket Queue model, a Markovian model of Ticket Queue with reneging customers. Additionally, we developed an approximation procedure to obtain numerically the steady state probabilities and calculate the total reneging percentage. The approximation procedure turned out to be largely efficient on an extensive set of numerical examples. Therefore, through this procedure, we are able to estimate the actual queue length for a ticket position, and thus potentially we can reduce the total reneging percentage by offering more information to the customers upon arrival.

For this purpose, we can calculate the expected queue length when an arriving customer observes a ticket position of d_A. We assume that n_A is the number of customers who are actually in the queue, being served or waiting, except of the newly arrived customer at that time. Apparently, we have $n_A \leq d_A$ and when $d_A = 0$, then $n_A = 0$. Same goes when $d_A = 1$, where $n_A = 1$. For $d_A \geq 1$, we can compute the conditional expected queue length $E[n_A|d_A]$ as follows:

$$E[n_A|d_A] = \sum_{k=1}^{d_A} [k \cdot P(n_A = k|d_A)] = \frac{\sum_{k=1}^{d_A}[k \cdot P(n_A = k, d_A)]}{P(d_A)}$$

$$E[n_A|d_A] = \frac{\sum_{k=1}^{d_A}[k \cdot P(k, d_A - k)]}{\sum_{k=1}^{d_A} P(k, d_A - k)} \tag{4.35}$$

Note that $P(n_A = k, d_A)$ is the probability that there are k customers already in the queue, given the fact that a customer upon her/his arrival observes ticket position equal to d_A. Furthermore, $P(k, d_A - k)$ is the probability of the state with 1 customer in service, $k - 1$ customers waiting in queue and $d_A - k$ reneging customers with their tickets still counted in the queue. We can easily comprehend that these two probabilities are equal.

Generally, we observe that $E[n_A|d_A]$ is smaller than d_A, because some of the customers may have reneged and hence, it reflects with more accuracy the real queue length. So, there is the idea to provide the customers with this information upon their arrival; this action could make them discount the actual queue length by a proportion of $E[n_A|d_A]/d_A$. If we assume as well that the discounting is kept through their stay in the queue, the immediate reneging rate will be adjusted to $\tau_{[d \cdot E(n_A|d_A)/d_A]}$ when later the ticket position changes to d. We refer to the resulting queue as the R-Ticket-Plus Queue, that is a queue where the customers conscientiously discount the queue length as indicated by the ticket position d, and the discounting factor is given upon arrival.

For the R-Ticket-Plus Queue, we compute the total reneging percentage as in Section 4.1. It actually turns out much smaller than the R-Ticket Queue, but still larger than the corresponding physical queue. So, providing information on $E[n_A|d_A]$, helps us to reduce the difference of the total reneging percentage between th R-Ticket Queue and the physical one.

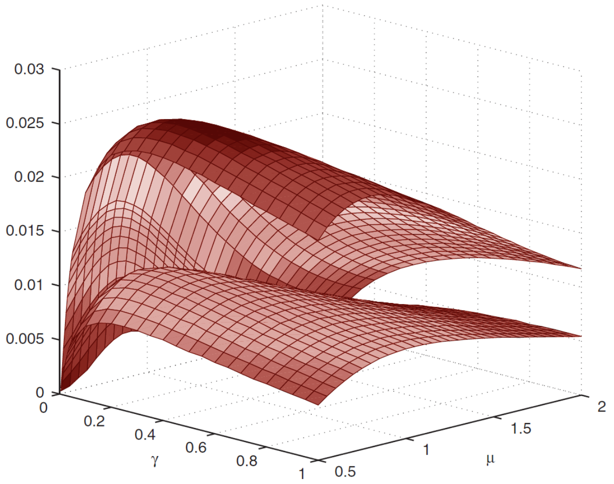


Figure 4.10 Total Reneging Percentage Differences

We supplement the intuitive arguments with numerical examples, using the same set of parameters as before. The simulation shows that the reduction can be as high as 65%, which is an impressive result. In Figure 4.10 we can see the differences in total reneging percentage between the R-Ticket Queue and the corresponding physical (upper surface) and between the R-Ticket-Plus Queue and the physical one (lower surface). It is clear that the latter have smaller differences than the former. That is, the R-Ticket-Plus Queue always improves R-Ticket Queue in terms of the total reneging percentage, and we can say that generally, performs very close to the physical queue with reneging customers. Finally, it is important that the improvement over the R-Ticket Queue is mostly pronounced when its performance is the worst relative to the physical queue; this makes the R-Ticket Queue a really good modification for improvement.

In summary, providing the customers upon arrival with $E[n_A|d_A]$ seems an efficient idea that may reduce the total reneging percentage by 65%. However, there are so many open problems in Ticket Queues yet, that this analysis and the modification idea are only a starting point for more exploration in this kind of queues.

Chapter 5

Conclusion

Throughout this work, we gained insights about Ticket Queues that are actually very interesting; we studied two different models of Ticket Queues, one with balking and the other with reneging customers. We developed the Markovian representation of these models, we introduced methods to find their steady state probabilities and showed efficient evaluation tools for useful service performance measures. This analysis led to suggestions for improvement, as the obtained results can help management to predict customers' behavior quite accurately; and hence provide them with information that can reduce either the balking or the reneging percentage.

We should highlight that this work is an analysis of fundamental Ticket Queue models and is merely one quite informative and useful starting point for even more exploration in this topic. In fact, these Markov models can definitely serve as the starting points, from which we could study more general or complex systems. For example, a project which could be studied is a Ticket Queue with both balking and reneging customers, or customers that are not strategically homogeneous. As mentioned in the beginning, a recent work of Hanukov, Anily and Yechiali (2019), shows an appealing, really complex system where we have strategic and non-strategic customers. The latter, called as regular customers, appear to be those who join the queue whatever their ticket position is, while strategic customers follow a double threshold strategy. Imagine we have two thresholds, m and n, where $m < n$. Then, we let d be the ticket position, so that a strategic customer joins if $d \leq m$. While $m < d < n$, a strategic customer becomes an orbit customer; that is, s/he takes the ticket and leaves temporarily the system in order to run some other errands and then return, hoping s/he would not lost her/his position in the queue. Finally, when $d \geq n$, the customer balks. Such complex and difficult Ticket Queue systems have been researched according to the primal work that is presented here, and there are plenty of options that can be explored in the future.

We should also mention that in this work we assumed as a basic reason of balking or reneging the expected delay of a customer in the system. Thus, a customer makes a decision by estimating the time s/he needs to wait in order to be served. However, in reality, a customer's decision may depend on other attributes than her/his anticipated delay. Understanding the customers' psychology when waiting in a queue is a substantial tool for improvement. Even though researches are made for customers' psychology in physical queues, their behavior may differ when dealing with a Ticket Queue. So, it is important to understand deeper the customers' behavior in Ticket Queue systems to develop better models that truly reflect their needs and make them more satisfied. This benefits the management as well. Consequently, with different reasons of customers' dissatisfaction, we should develop efficient methods to estimate different performance measures and follow other ways to communicate this information to the customers. For example, some companies use handhold electronic devices to communicate with their customers. There are plenty of ways and policies that a company can apply to achieve the goal of improvement.

As the time goes by, Ticket Queues are systems that we are dealing with more and more in everyday life. As a consequence, it is a subject of growing interest, and we expect a big amount of additional works to be developed in the future.

Bibliography

[1] Xu, S.H., Gao, L. and Ou, J. (2007) Service performance analysis and improvement for a ticket queue with balking customers. Management Science 53, 971-990.

[2] Kerner, Y., Sherzer, E. and Yanco, M.A. (2017) On non-equilibria threshold strategies in ticket queues. Queueing Systems 86, 419-431.

[3] Hassin, R. and Haviv, M. (2003) To queue or not to queue: Equilibrium behavior in queueing systems. Kluwer.

[4] Nelson, R. (1995) Probability, Stochastic Processes and Queueing Theory. Springer.

[5] Ding, D., Ou, J. and Ang, J. (2015) Analysis of ticket queues with reneging customers. Journal of the Operational Research Society 66, 231-246.

[6] Hanukov, G., Anily, S., Yechiali, U. (2019) Ticket Queues with Regular and Strategic Customers. To Appear in Queueing Systems.

[7] Latouche, G., Ramaswami, V. (1999) Introduction Matrix Analytic Methods in Stochastic Modeling. ASA-SIAM.

[8] Economou, A. (2020) The impact of information structure on strategic behavior in queueing systems. In Anisimov, V. and Limnios, N. eds. Queueing Theory 2, Advanced Trends. Wiley/ISTE.