Contents

List of Figures

List of Tables

1 Survey of Quantitative Methods

The bread and butter of quantitative consulting is the ability to apply quantitative methods to business problems in order to obtain actionable insight. Clearly, it is impossible (and perhaps inadvisable, in a more general sense) for any given individual to have expertise in every field of mathematics, statistics, and computer science.

We believe that the best consulting framework is reached when a small team of consultants possesses expertise in 2 or 3 areas, as well as a decent understanding of related disciplines, and a passing knowledge in a variety of other domains: this includes keeping up with trends, implementing knowledge redundancies on the team, being conversant in non-expertise areas, and knowing where to find detailed information (online, in books, or through external resources).

In this section, we present an introduction for 9 “domains” of quantitative analysis:

- survey sampling and data collection;
- data processing;
- data visualisation;
- statistical methods;
- queueing models;
- data science and machine learning;
- simulations;
- optimisation, and
- trend extraction and forecasting;

Strictly speaking, the domains are not free of overlaps. Large swaths of data science and time series analysis methods are quite simply statistical in nature, and it’s not unusual to view optimisation methods and queueing models as sub-disciplines of operations research. Other topics could also have been included (such as Bayesian data analysis or signal processing, to name but two), and might find their way into a second edition of this book.

Our treatment of these topics, by design, is brief and incomplete. Each module is directed at students who have a background in other quantitative methods, but not necessarily in the topic under consideration. Our goal is to provide a quick “reference map” of the field, together with a general idea of its challenges and common traps, in order to highlight opportunities for application in a consulting context. These subsections are emphatically NOT meant as comprehensive surveys: they focus on the basics and talking points; perhaps more importantly, a copious number of references are also provided.

We will start by introducing a number of motivating problems, which, for the most part, we have encountered in our own practices. Some of these examples are reported on in more details in subsequent sections, accompanied with (partial) deliverables in the form of charts, case study write-ups, report extract, etc.).

As a final note, we would like to stress the following: it is **IMPERATIVE** that quantitative consultants remember that acceptable business solutions are not always optimal theoretical solutions. Rigour, while encouraged, often must take a backseat to applicability. This lesson can be difficult to accept, and has been the downfall of many a promising candidate.

1.5 Queuing Models

Queuing theory is a branch of mathematics that studies and models the act of waiting in lines. The first paper on queuing theory, "The Theory of Probabilities and Telephone Conversations" was published in 1909 by A.K. Erlang, now considered the father of the field. He pondered the problem of determining how many telephone circuits were necessary to provide phone service that would prevent customers from waiting too long for an available circuit.

In developing a solution to this problem, he began to realize that the problem of minimizing waiting time was applicable to many fields, and began developing the theory further. Erlang's **switchboard problem** laid the path for modern queuing theory.

Queueing theory boils down to answering simple questions like the following:

- How likely is it that objects/units/persons will queue up and wait in line?
- How long will the line be?
- How long will the wait be?
- How busy will the server/person/system servicing the line be?
- How much capacity is needed to meet an expected level of demand?

Knowing how to think about these kinds of questions will help clients anticipate **bottlenecks**. As a result, they will build systems and teams to be more efficient and more scalable, to have higher performance and lower costs, and to ultimately provide better service to their customers.

Queueing theory also allows for the quantitative treatment of bottlenecks and effect on performance. For instance, a question such as "how long will the wait be, on average?" will have an answer, but so will other questions concerning the variability of wait times, the distribution of wait times, and the likelihood that a customer someone will receive extremely poor service, and so on [9].

Let us consider a simple example. Suppose a grocery store has a single checkout line and a single cashier. If, on average, one shopper arrives at the line to pay for their groceries every 5 minutes and if scanning, bagging, and paying takes 4.5 minutes, on average, will customers have to wait in line? When the problem is presented this way, our intuition says that there should be no waiting in line, and that the cashier should be idle, on average, 30 seconds every 5 minutes, only being busy 90% of the time. No one ever has to wait before being served!

If you have ever been in grocery store, you know that's not really what happens in reality; many shoppers will be waiting waiting in line, and they will have to wait a long time before being processed. Fundamentally, queueing happens for three reasons:

- **irregular arrivals** – shoppers do not arrive at the checkout line on a regular schedule; they are sometimes spaced far apart and sometimes close together, so they **overlap** (an overlap automatically causes queueing and waiting);
- **irregular job sizes** – shoppers do not all get processed in 4.5 minutes; somebody shopping for a large family will require much more time, for instance (when this happens, overlap is again a problem because new shoppers will arrive and be ready to check out while the existing ones are still in progress), and

- **waste** – lost time can never be regained; shoppers overlap because the second shopper arrived before the first shopper had the time to finish, but looking at it the other way, perhaps it's not the second shopper's fault; perhaps the first shopper should have arrived earlier, but wasted time reading a gossip magazine while the cashier was idle! They missed their chance for quick service and, as a result, made the second shopper have to wait.

In general, **irregular** arrival times and job sizes are guaranteed to cause queueing. The only time there is no queueing is when the job sizes are uniform, the arrivals are timed evenly, and there is little enough work for the cashier to keep up. Even when the cashier is barely busy at all, irregular arrivals or arrivals **in bursts** will cause some queueing.

Queueing gets worse when the following is true of the system:

- **high utilisation** – the busier the cashier is, the longer it takes to recover from wasted time;
- **high variability** – the more variability in arrivals or job sizes, the more waste and the more overlap (queueing) occurs, and
- **insufficient servers** – fewer cashiers means less capacity to absorb arrival spikes, leading to more wasted time and higher utilisation.

1.5.1 Terminology

Queueing theory studies systems and processes in terms of three key concepts:

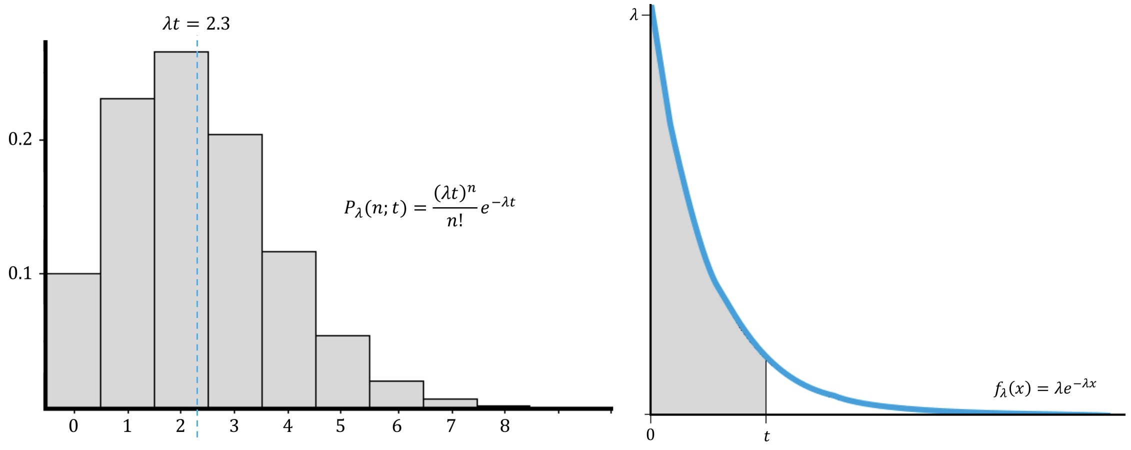
- **customers** are the units of work that the system serves – a customer can be a real person, or it can be whatever the system is supposed to process and complete: a web request, a database query, a part to be milled by a machine, etc.;
- **servers** are the objects that do the processing work – a server might be the cashier at the grocery store, a web server, a database server, a milling machine, etc., and
- **queues** are where the units of work wait if the server is busy and can not start the work as they arrive – a queue may be a physical line, or reside in memory, etc.

To begin understanding and describing queues, we must first have know and understand some useful probability distributions, as well as input and output processes.

Exponential and Poisson Probability Distributions The **Poisson** and **exponential** distributions play a prominent role in queuing theory. The Poisson distribution counts the number of discrete events occurring in a fixed time period; it is closely connected to the exponential distribution, which (among other applications) measures the time between arrivals of the events. The Poisson distribution is a discrete distribution; the random variable can only take non-negative integer values. The exponential distribution can take any (nonnegative) real value.

Consider the problem of determining the probability of n arrivals being observed during a time interval of length t, where the following assumptions are made:

- the probability that an arrival is observed during a small time interval (say of length v) is proportional to the length of interval; let the proportionality constant be λ, so that the probability is λv;



(a) Poisson distribution with $\lambda t = 2.3$. **(b)** Exponential distribution with parameter λ.

Figure 1: Poisson and exponential distributions. The shaded area (on the right) represents the probability that a customer will wait up to the length of the time interval t.

- the probability of two or more arrivals in a small interval is zero, and
- the number of arrivals in any time interval is independent of the number in non-overlapping time interval (for example, the number of arrivals occurring between times 5 and 25 does not provide information about the number of arrivals occurring between times 30 and 50).

Now, let $P(n;t)$ be the probability of observing n arrivals in a time interval of length t. Then, for some $\lambda > 0$,

$$P_\lambda(n;t) = \frac{(\lambda t)^n}{n!} e^{-\lambda t}, \ n = 0, 1, 2, \cdots$$

is the p.m.f. of the **Poisson distribution** for the discrete random variable n – the number of arrivals – for a given length of time interval t (see Figure 1a). In a queueing system, such arrivals are referred to as **Poisson arrivals**.

The time between successive arrivals is called the **inter-arrival time**. If the number of arrivals in a given time interval follows a Poisson distribution with parameter λt, the inter-arrival times follow an exponential distribution with p.d.f.

$$f_\lambda(t) = \lambda e^{-\lambda t}, \text{ for } t > 0,$$

and the probability $P(W \leq t)$ that a customer's waiting time W is smaller than the length of the time interval t is

$$P(W \leq t) = 1 - e^{-\lambda t} \quad \text{(see Figure 1b).}$$

In general, if the arrival rate is **stationary**, if **bulk** arrivals (two or more simultaneous arrivals) cannot occur, and if past arrivals do not affect future arrivals, then inter-arrival times follow an exponential distribution with parameter λ, and the number of arrivals in any interval of length t is Poisson with parameter λt.

One of the most attractive features of using the exponential distribution to model inter-arrival times is that it is **memoryless** – if a random variable X follows an exponential distribution, then for all non-negative values of t and h,

$$P(X \geq t + h | X \geq t) = P(X \geq h). \tag{1}$$

No other density function satisfies (1) [2]. The memoryless property of the exponential distribution is important because it implies that the probability distribution of the time until the next arrival is independent of the time since the last arrival – imagine if that was the case when waiting for public transportation!

For instance, if we know that at least t time units have elapsed since the last arrival, then the distribution of time h until the next arrival is independent of t. If $h = 4$, say, then (1) yields

$$P(X > 9|X > 5) = P(X > 7|X > 3) = P(X > 4|X > 0) = e^{-4\lambda}.$$

Erlang Distribution The exponential distribution is not always an appropriate model of inter-arrival times (perhaps the process should not be memoryless, say). A common alternative is to use the **Erlang** distribution $\mathcal{E}(R,k)$, a continuous random variable with **rate** and **shape** parameter $R > 0$ and $k \in \mathbb{Z}^+$, respectively, whose p.d.f. is

$$f_{R,k}(t) = \frac{R(Rt)^{k-1}e^{-Rt}}{(k-1)!},\ t \geq 0.$$

When $k = 1$, the Erlang distribution reduces to an exponential distribution with parameter R. It can be shown that if $X \sim \mathcal{E}(k\lambda, k)$, then $X \sim X_1 + X_2 + \cdots + X_k$, where each X_i is an independent exponential random variable with parameter $k\lambda$.

When we model the inter-arrival process as an Erlang $\mathcal{E}(k\lambda, k)$ distribution, we are really saying that it is equivalent to customers going through k **phases** (each of which is memoryless) before being served. For this reason, the shape parameter is often referred to as the number of phases of the Erlang distribution [12].

Input/Arrival Process The input process is usually called the **arrival process**. Arrivals are called **customers**. In the models that we discuss, we assume that arrivals cannot be simultaneous (this might be unrealistic when modeling a restaurant, say). If simultaneous arrivals can occur, we say that **bulk arrivals are allowed**.

Usually, we assume that the arrival process is **unaffected by the number of customers** in the system. In the context of a bank, this would imply that whether there are 500 or 5 people at the bank, the process governing arrivals remains unchanged.

There are two common situations in which the arrival process may depend on the number of customers present. The first occurs when arrivals are drawn from a small population – the so-called **finite source models** – if all members of the populations are already in the system, there cannot be another arrival!

Another such situation arises when the rate at which customers arrive at the facility decreases when the facility becomes too crowded. For example, when customers see that a restaurant's parking lot is full, they might very well decide to go to another restaurant or forego eating out altogether. If a customer arrives but fails to enter the system, we say that the customer has **balked**.

Output/Service Process To describe the output process (often called the **service process**) of a queuing system, we usually specify a probability distribution – the **service time distribution** – which governs the customers' service time.

In most cases, we assume that the service time distribution is independent of the number of customers present in the system. This implies, for example, that the server does not work faster when more customers are present.

We can distinguish two types of servers: **in parallel** and **in series**. Servers are in parallel if they all provide the same type of service and a customer only needs to pass through one of them to complete their service. For example, the tellers in a bank are usually arranged in parallel; typically, customers only need to be serviced by one teller, and any teller can perform the desired service. Servers are in series if a customer must pass through several servers before their service is complete. An assembly line is an example of such a queuing system.

Input and output processes occur in a variety of situations:

- **situation:** purchasing Blue Jays tickets at the Rogers Centre
  input: baseball fans arrive at the ticket office
  output: tellers serve the baseball fans;
- **situation:** pizza parlour
  input: requests for pizza delivery are received; *output*: pizza parlour prepares and bakes pizzas, and sends them to be delivered;
- **situation:** government service centre
  input: citizen/residents enter the service centre
  output: receptionist assigns them to a specific queue based on their needs
    input: citizen/residents enter a specific queue based on their needs
    output: public servant addresses their needs;
- **situation:** hospital blood bank
  input: pints of blood arrive
  output: patients use up pints of blood;
- **situation:** garage
  input: cars break-down and are sent to the garage for repairs
  output: cars are repaired by mechanics and sent back on the streets.

The computations are fairly easy to execute, as the following examples demonstrate.

- On average, 4.6 customers enter a coffee shop each hour. If the arrivals follow a Poisson process, the probability that at most two customers will enter in a 30 minute period is

$$\begin{aligned} P_{\lambda=4.6}(n \leq 2; t = 0.5) &= P_{4.6}(0, 0.5) + P_{4.6}(1, 0.5) + P_{4.6}(2, 0.5) \\ &= e^{-4.6 \cdot 0.5}\left[\frac{(4.6 \cdot 0.5)^0}{0!} + \frac{(4.6 \cdot 0.5)^1}{1!} + \frac{(4.6 \cdot 0.5)^2}{2!}\right] \\ &\approx 0.5960; \end{aligned}$$

  the corresponding Poisson distribution is shown in Figure 1a.

- In a fast food restaurant, a cashier serves on average 9 customers in a one-hour time period. If the service time follows an exponential distribution, 89.5% and 2.4% of customers will be served in 15 minutes or less and after 25 minutes, respectively. Indeed,

$$P(W \leq 15/60) = 1 - e^{-9 \cdot 15/60} \approx 0.8946 \quad \text{and} \quad P(W > 25/60) = e^{-9 \cdot 25/60} \approx 0.0235.$$

Queue Discipline To describe a queuing system completely, we must also describe the **queue discipline** and the manner in which customers **join lines**. The queue discipline describes the method used to determine the order in which customers are served:

- the most common queue discipline is the **first come, first served** (FCFS) discipline, in which customers are served in the order of their arrival, as one would expect to see in an Ottawa coffee shop;
- under the **last come, first served** (LCFS) discipline, the most recent arrivals are the first to enter service; for example, if we consider exiting from an elevator to be the service, then a crowded elevator illustrates such a discipline;
- sometimes the order in which customers arrive has no effect on the order in which they are served; this would be the case if the next customer to enter service is randomly chosen from those customers waiting for service, a situation referred to as **service in random order** (SIRO) discipline; when callers to an inter-city bus company are put on hold, the luck of the draw often determines which caller will next be serviced by an operator,
- finally, **priority** discipline classifies each arrival into one of several categories, each of which is assigned a priority level (a **triage** process); within each priority level, customers enter the queue on a FCFS basis; such a discipline is often used in emergency rooms to determine the order in which customers receive treatment, and in copying and computer time-sharing facilities, where priority is usually given to jobs with shorter processing times.

Method Used by Arrivals to Join Queue Another important factor for the behaviour of the queuing system is the **method** used by customers to determine which line to join. For example, in some banks, customers must join a single line, but in other banks, customers may choose the line they want to join.

When there are several lines, customers often join the shortest line. Unfortunately, in many situations (such as at a supermarket), it is difficult to define the shortest line. If there are several lines at a queuing facility, it is important to know whether or not customers are allowed to **switch**, or jockey, between lines. In most queuing systems with multiple lines, jockeying is permitted, but jockeying at a custom inspection booth would not be recommended, for instance.

1.5.2 Queueing Theory Framework

There is a standard notation that is used to describe large families of queueing systems.

Kendall-Lee Notation Queuing systems can be described by six characteristics [8]:

$$x_1/x_2/x_3/x_4/x_5/x_6.$$

The first characteristic x_1 specifies the nature of the **arrival process**. The following standard abbreviations are used:

M = inter-arrival times are independent, identically distributed (iid) exponentials
D = inter-arrival times are iid and deterministic
E_k = inter-arrival times are iid Erlangs with shape parameter k
G = inter-arrival times are iid and governed by some general distribution.

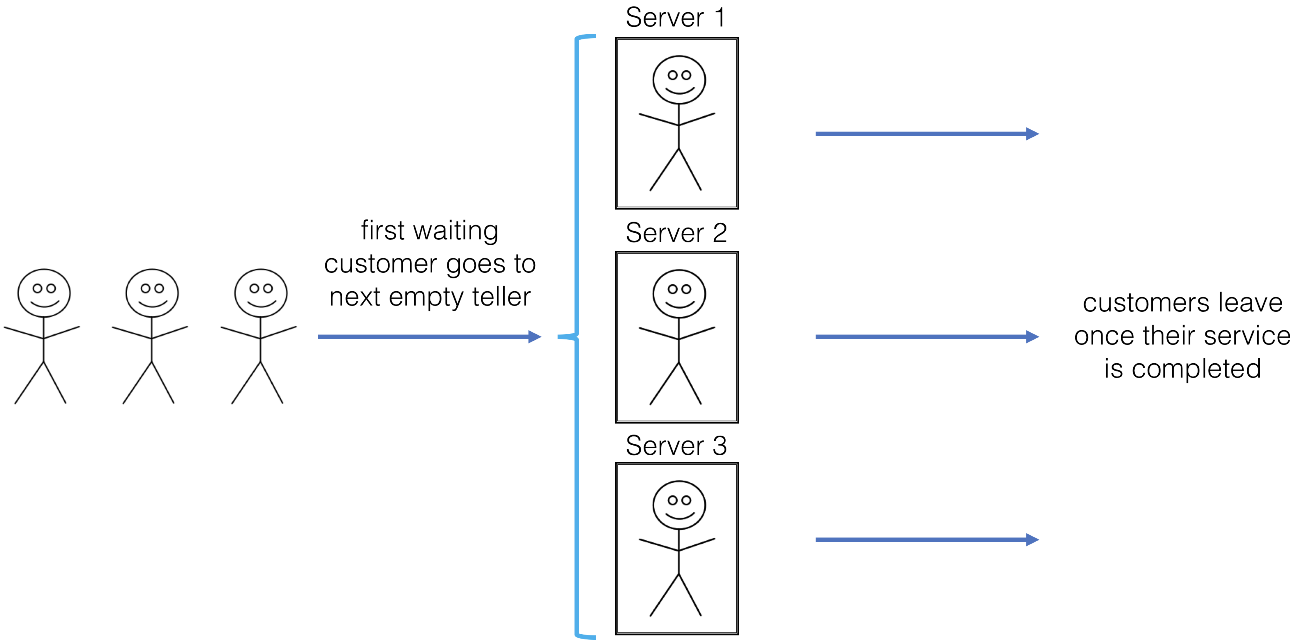

Server 1
Server 2
Server 3
first waiting customer goes to next empty teller
customers leave once their service is completed

Figure 2: Single line at bank with three tellers – $M/M/3/\text{FCFS}/20/\infty$.

The second characteristic x_2 specifies the nature of the **service times**:

$$\begin{aligned} M &= \text{service times are iid and exponential} \\ D &= \text{service times are iid and deterministic.} \\ E_k &= \text{service times are iid Erlang with shape parameter } k \\ G &= \text{service times are iid and follow some general distribution.} \end{aligned}$$

The third characteristic x_3 is the **number of parallel servers**. The fourth characteristic x_4 describes the **queue discipline**:

$$\begin{aligned} \text{FCFS} &= \text{first come, first served} \\ \text{LCFS} &= \text{last come, first served} \\ \text{SIRO} &= \text{service in random order} \\ \text{GD} &= \text{general queue discipline.} \end{aligned}$$

The fifth characteristic x_5 specifies the **maximum allowable number of customers in the system** (including customers who are waiting and customers who are in service). The sixth characteristic x_6 gives the **size of the population** from which customers are drawn. Unless the number of potential customers is of the same order of magnitude as the number of servers, the population size is considered to be infinite.

In many important models $x_4/x_5/x_6$ is $\text{GD}/\infty/\infty$; when this is the case, these characteristics are often omitted. For example, $M/M/3/\text{FCFS}/20/\infty$ could represent a bank with 3 tellers, exponential arrival times, exponential service times, a “first come, first served” queue discipline, a total capacity of 20 customers, and an infinite population pool from which to draw. The situation is partly illustrated in Figure 2.

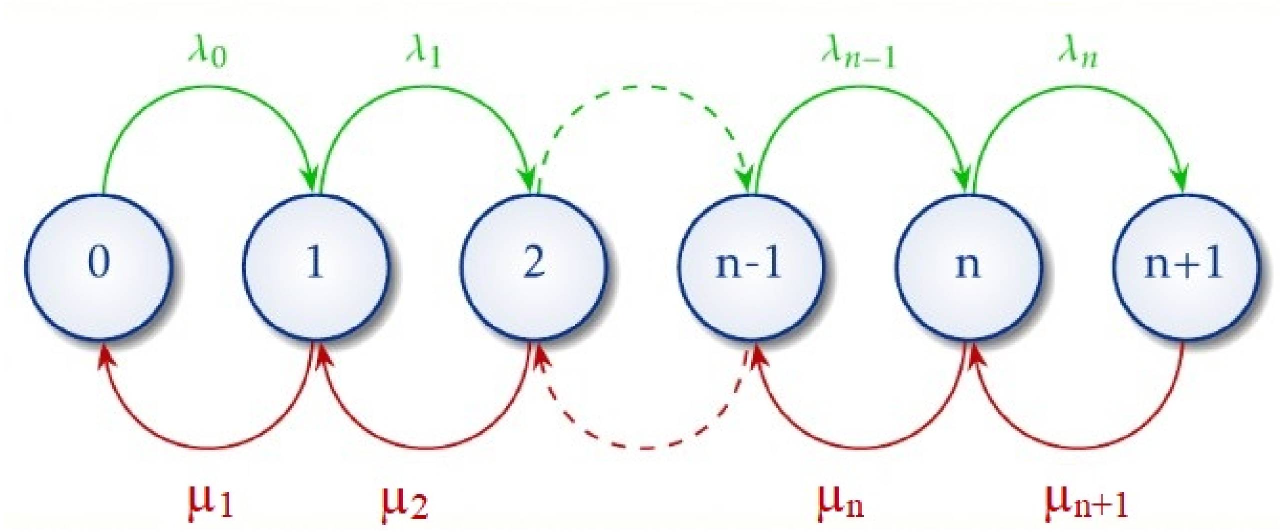


Figure 3: Birth-death Process; queueing states indexed by integers; birth rates and death rates indicated by λ_n and μ_m, respectively.

Birth-Death Processes The **state of a queueing system** at time t is defined to be the number of customers in the queuing system, either waiting in line or in service, at time t. At $t = 0$, the state of the system is the initial number of customers in the system. This state is noteworthy because it clearly affects the state at future t.

Knowing this, we define $P_{ij}(t)$ as the probability that the state at time t is j, given that the state at $t = 0$ was i. For large t, $P_{ij}(t)$ becomes independent of i and approaches a limit π_j. This limit is known as the **steady-state** of state j.

It is generally incredibly difficult to determine the steps of arrivals and services that lead to a steady-state π_j. Likewise, starting from a small t, it is difficult to determine exactly when a system will reach its steady state π_j, if such a state even exists.

For simplicity's sake, when a queuing system is studied, we begin by assuming that the steady-state has already been reached.

A **birth-death process** is a Markov process in which states are indexed by non-negative integers, and transitions are only permitted between "neighbouring" states. After a "birth", the state increases from n to $n+1$; after a "death", the state decreases from m to $m-1$. Typically, we denote the set of birth rates and death rates by λ_n and μ_m, respectively (see Figure 3). **Pure birth** processes are those for which $\mu_m = 0$ for all m; **pure death** processes those for which $\lambda_n = 0$ for all n. The **steady-state solution** of a birth-death process, i.e. the probability p_n of being in state n *can* actually be computed:

$$p_n = p_0 \frac{\lambda_0 \lambda_1 \cdots \lambda_{n-1}}{\mu_1 \mu_2 \cdots \mu_n}, \quad \text{for } n = 1, 2, \cdots, \tag{2}$$

where p_0 is the probability of being in state 0. It can further be shown [9] that:

$$p_0 = \frac{1}{1 + \sum_{n=1}^{\infty} \prod_{j=0}^{n-1} \frac{\lambda_j}{\mu_{j+1}}}.$$

Little's Queuing Formula It is often the case that a client is interested in the amount of time that a typical customer spends in the queuing system. Let W be the **expected waiting time** spent in the queuing system, including time in line plus time in service, and W_q be the **expected time a customer spends waiting in line**. Both W and W_q are computed under the assumption that the steady state has been reached. By using a powerful result known as **Little's queuing formula**, W and W_q are easily related to the number of customers in the queue and those waiting in line.

For any queuing system (or any subset of a queuing system), consider the following quantities:

- $\lambda =$ average number of arrivals entering the system per unit time;
- $L =$ average number of customers present in the queuing system;
- $L_q =$ average number of customers waiting in line;
- $L_s =$ average number of customers in service;
- $W =$ average time a customer spends in the system;
- $W_q =$ average time a customer spends in line, and
- $W_s =$ average time a customer spends in service.

But customers in the system can only either be found in the queue or in service, so that $L = L_q + L_s$ as well $W = W_q + W_s$. In these definitions, all averages are steady-state averages. For most queuing systems in which a steady-state exists, Little's queuing formula can be summarized as

$$L = \lambda W, \quad L_q = \lambda W_q, \quad \text{and} \quad L_s = \lambda W_s.$$

For instance, if 46 customers enter a restaurant each hour it is opened (on average), and if they spend 10 minutes waiting to be served (on average), then we should expect $46 \cdot 1/6 \approx 7.7$ customers in the queue at all time (on average).

1.5.3 The $M/M/1$ Queuing System

An $M/M/1/\text{GD}/\infty/\infty$ system has exponential inter-arrival times, exponential service times, and a single server. It is the simplest non-trivial queueing system to analyse as it can be modeled as a birth-death process with

$$\lambda_j = \lambda, \; j = 0, 1, 2, \ldots$$
$$\mu_0 = 0$$
$$\mu_j = \mu, \; j = 1, 2, 3, \ldots$$

Substituting these rates in (2) yields

$$\pi_j = \frac{\lambda^j \pi_0}{\mu^j} = \rho^j \pi_0,$$

where $\rho = \lambda/\mu$ is the **traffic intensity** of the system. Since the system has to be in exactly one of the states at any given moment, the sum of all probabilities is 1, or

$$\pi_0 + \pi_1 + \pi_2 + \cdots = \pi_0(1 + \rho + \rho^2 + \cdots) = 1.$$

If $0 \leq \rho < 1$ the infinite series converges to $\frac{1}{1-\rho}$ from which we derive

$$\pi_0 \cdot \frac{1}{1-\rho} = 1 \implies \pi_0 = 1-\rho \implies \pi_j = \rho^j \pi_0 = \rho^j (1-\rho)$$

as the **steady-state probability of state** j. If $\rho \geq 1$, the infinite series diverges and no steady-state exists. Intuitively, this happens when $\lambda \geq \mu$ – if the arrival rate is greater than the service rate, then the state of the system grows without bounds and the queue is never cleared.

From this point on, we assume $\rho < 1$ to guarantee that the steady-state probabilities π_j exist, from which we can determine several quantities of interest. Assuming that the steady state has been reached, it can be shown that L, L_s, and L_q are given respectively by:

$$\begin{aligned} L &= \frac{\lambda}{\mu-\lambda} \\ L_s &= \rho \\ L_q &= \frac{\rho^2}{1-\rho}. \end{aligned}$$

Using Little's queuing formula, we can also solve for W, W_s, and W_q by dividing each of the corresponding L values by λ. Notice that, as expected, both W and W_q when $\rho \to 1$. On the other hand, $W_q \to 0$ and $W \to \frac{1}{\mu}$ (the **mean service time**) as $\rho \to 0$.

(The following example is based on [1]) An average of 10 cars arrive at a single-server drive-in teller every hour. If the average customer is served in 4 minutes, service time for each customer is 4 minutes, and both inter-arrival times and service times are exponential, then:

(a) What is the probability that the teller is idle?
(b) Excluding the car that is being served, what is the average number of cars waiting in line at the teller?
(c) What is the average amount of time a drive-in customer spends in the bank parking lot (including time in service)?
(d) On average, how many customers per hour will be served by the teller?

By assumption, we are dealing with an $M/M/1/\text{GD}/\infty/\infty$ queuing system for which $\lambda = 10$ cars/hr and $\mu = 15$ cars/hr, and as such $\rho = 10/15 = 2/3$.

(a) The teller is idle one third of the time on average because $\pi_0 = 1-\rho = 1/3$.
(b) There are $L_q = \rho^2/(1-\rho) = 4/3$ cars waiting in line for the teller.
(c) We know that $L = \lambda/(\mu-\lambda) = 10/(15-10) = 2$, and so $W = L/\lambda = 0.2$ hr $= 12$ min.
(d) If the teller were always busy, it would serve an average of $\mu = 15$ customers per hour. From (a), we know that the teller is only busy two-thirds of the time, thus during each hour, the teller serves an average of $15 \cdot 2/3 = 10$ customers. This is reasonable since, in a steady-state, 10 customers are arriving each hour and 10 customers must leave the system every hour.

(This next example is based on [15]) Suppose that all car owners fill up when their tanks are exactly half full. At the present time, an average of 7.5 customers arrive every hour at a single-pump gas station. It takes an average of 4 minutes to fuel a car. Assume that inter-arrival times and service times are both exponential.

(a) What are the values of L and W in this scenario?
(b) Suppose that a gas shortage occurs and panic buying takes place. To model this phenomenon, assume that all car owners now purchase gas when their tanks are exactly three-quarters full. Since each car owner is now putting less gas into the tank during each visit to the station, we assume that the average service time has been reduced to 10/3 minutes. How has panic buying affected the values of L and W?

By assumption, we have again an $M/M/1/\text{GD}/\infty/\infty$ queuing system, this time with $\lambda = 7.5$ cars/hr and $\mu = 60/4 = 15$ cars/hr. Thus, $\rho = 7.5/15 = 1/2$.

(a) By definition, $L = \lambda/(\mu - \lambda) = 7.5/(15 - 7.5) = 1$ and $W = 1/7.5 \approx 0.13$ hr $= 7.8$ min. Hence, in this situation, everything is under control, and long lines appear to be unlikely.
(b) Under the panic buying scenario, $\lambda = 2(7.5) = 15$ cars/hr as each car owner now fills up twice as often, and $\mu = 60 \cdot 3/10 = 18$ cars/hr. Then, $\rho = \lambda/\mu = 5/6$. In that scenario,

$$L = \frac{\rho}{1-\rho} = 5 \text{ cars} \quad \text{and} \quad W = \frac{L}{\lambda} = \frac{5}{15} = \frac{1}{3} \text{ hr } = 20 \text{ min.}$$

Thus, panic buying has more than doubled the wait time in line.

In a $M/M/1$ queueing system, we have

$$L = \frac{\rho}{1-\rho} = -1 + \frac{1}{1-\rho},$$

and it is easy to see that $L \to \infty$ as $\rho \to 1$. The 5-fold increase in L when ρ jumps from 1/2 to 5/6 (with accompanying jumps in W) illustrate that fact.

Limited Capacity In the real world, queues never become infinite – they are limited due to requirements of space and/or time, or service operating policy. Such a queuing model falls under the purview of **finite queues**.

Finite queue models restrict the number of customers allowed in the service system. Let N represent the maximum allowable number of customers in the system. If the system is at **capacity**, the arrival of a $(N+1)^{\text{th}}$ customer results in a failure to enter the queue – the customer is assumed to depart without seeking service.

Finite queues can also be modeled as a birth-death process, but with a slight modification in its parameters: with these parameters:

$$\begin{aligned}
&\lambda_j = \lambda,\ j = 0,1,2,\ldots,N-1\\
&\lambda_N = 0,\ \mu_0 = 0\\
&\mu_j = \mu,\ j = 1,2,3,\ldots,N
\end{aligned}$$

The restriction $\lambda_N = 0$ is what sets this model apart from the $M/M/1/\infty$. It makes it impossible to reach a state greater than N. Because of this restriction, a steady-state always exist because even if $\lambda \geq \mu$, there can never be more than N customers in the system.

Mathematically, this has the effect of replacing the infinite series linking the π_j's by a finite geometric series, which always converges:

$$\pi_0 + \pi_1 + + \cdots + \pi_N = \pi_0(1 + \rho + \cdots + \rho^N) = 1,$$

from which we can derive

$$\pi_0 \cdot \frac{1-\rho^{N+1}}{1-\rho} = 1 \implies \pi_0 = \frac{1-\rho}{1-\rho^{N+1}} \implies \pi_j = \begin{cases} \rho^j \frac{1-\rho}{1-\rho^{N+1}} & \text{for } j = 0, \ldots, N \\ 0 & \text{for } j > N \end{cases}$$

Since $L = \sum_{j=0}^{N} j \cdot \pi_j$,

$$L = \frac{\rho[1 + N\rho^{N+1} - (N+1)\rho^N]}{(1-\rho)(1-\rho^{N+1})}$$

when $\lambda \neq \mu$.

As in the $M/M/1/\infty$ queue, $L_s = 1 - \pi_0$, and $L_q = L - L_s$. It is somewhat trickier to compute W and W_q because, in a finite capacity model, only $\lambda - \lambda\pi_N = \lambda(1 - \pi_N)$ arrivals per unit time actually enter the system on average (λ arrive, but $\lambda\pi_N$ find the system full). With this fact,

$$W = \frac{L}{\lambda(1-\pi_N)} \quad \text{and} \quad W_q = \frac{L_q}{\lambda(1-\pi_N)}.$$

What does that look like in practice? Consider a one-man barber shop with a total of 10 seats. Assume, as has always been the case so far but need not be, that inter-arrival times are exponentially distributed with an average of 20 prospective customers arriving each hour at the shop. Those customers who find the shop full do not enter – perhaps they do not like standing? The barber takes an average of 12 minutes to cut each customer's hair; assume that haircut times are also exponentially distributed.

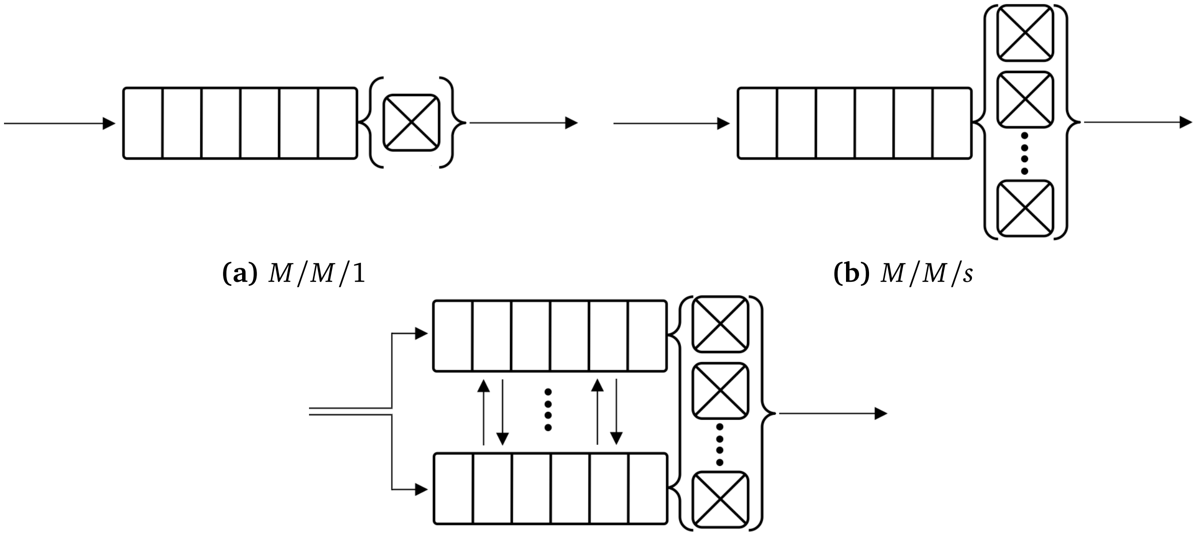
(a) On average, how many haircuts per hour will the barber complete?
(b) On average, how much time will be spent in the shop by a customer who enters?

There is not much to say. Let's dive in!

(a) A fraction π_{10} of all arrivals will find the shop is full. Thus, an average of $\lambda(1 - \pi_{10})$ will actually enter the shop each hour. All entering customers receive a haircut, so the barber will give an average of $\lambda(1 - \pi_{10})$ haircuts per hour. In this scenario, $N = 10$, $\lambda = 20$ customers/hr, and $\mu = 60/12 = 5$ customers/hr. Thus $\rho = 20/5 = 4$ and we have

$$\pi_0 = \frac{1-\rho}{1-\rho^{N+1}} = \frac{1-4}{1-4^{11}} \quad \text{and} \quad \pi_{10} = 4^{10}\pi_0 = \frac{3}{4}.$$

Thus, an average of $20(1 - 3/4) = 5$ customers per hour will receive haircuts. This means that an average of $20 - 5 = 15$ prospective customers per hour will not enter the shop.



(a) $M/M/1$

(b) $M/M/s$

(c) Tandem

Figure 4: Schematics of various queueing systems; customers arrive from the left, enter the queue and progress through it until they are served, at which point the exit the queue.

(b) To determine W, we must first compute

$$L = \frac{4[1+(10)4^{11}-(11)4^{10}]}{(1-4)(1-4^{11})} = 9.67.$$

Using the formulas described above, we obtain

$$W = \frac{L}{\lambda(1-\pi_{10})} = \frac{9.67}{5} = 1.93 \text{ hr}.$$

This barber shop is crowded – the barber would be well-advised to hire at least one more barber!

But what *would* be the effect of hiring a second barber? In order to answer this question, let us study $M/M/s$ queueing systems.

1.5.4 The $M/M/c$ Queuing System

An $M/M/c/\text{GD}/\infty$ queueing system also has exponential inter-arrival and service times, with rates λ and μ, respectively. What sets this system apart is that there are now c servers willing to serve from a single line of customers, perhaps like one would find in a bank (see Figure 4).

If $j \leq c$ customers are present in the system, then every customer is being served and there is no wait time; if $j > c$ customers are in the system, then c customers are being served and the remaining $j-c$ customers are waiting in the line.

To model this as a birth-death process, we have to observe that the death rate is dependent on how many servers are actually being used.

ρ	$c=2$	$c=3$	$c=4$	$c=5$	$c=6$	$c=7$
.10	.02	.00	.00	.00	.00	.00
.20	.07	.02	.00	.00	.00	.00
.30	.14	.07	.04	.02	.01	.00
.40	.23	.14	.09	.06	.04	.03
.50	.33	.24	.17	.13	.10	.08
.55	.39	.29	.23	.18	.14	.11
.60	.45	.35	.29	.24	.20	.17
.65	.51	.42	.35	.30	.26	.21
.70	.57	.51	.43	.38	.34	.30
.75	.64	.57	.51	.46	.42	.39
.80	.71	.65	.60	.55	.52	.49
.85	.78	.73	.69	.65	.62	.60
.90	.85	.83	.79	.76	.74	.72
.95	.92	.91	.89	.88	.87	.85

Table 1: Probabilities $P(n \geq c)$ that all servers are busy in an $M/M/c$ system for $c = 2, \ldots, 7$ and values of ρ between 0.1 and 0.95. [1, p.1088].

If each server completes service at a rate of μ (which may not be the case in practice as there might be variations in servers... at least for human servers), then the **actual death rate** is $\mu\times$ the number of customers actually being served. The parameters for this process are

$$\begin{aligned} \lambda_n &= \lambda,\ n = 0,1,2,\ldots \\ \mu_n &= n\mu,\ n = 0,1,2,\ldots,c \\ \mu_n &= c\mu,\ n = c+1, c+2,\ldots \end{aligned}$$

The traffic intensity for the $M/M/c$ system is $\rho = \lambda/(c\mu)$ and the steady-state solution is

$$\pi_n = \begin{cases} \frac{(c\rho)^n}{n!}\pi_0 & , 1 \leq n \leq c \\ \frac{c^c\rho^n}{c!}\pi_0 & , n \geq c \end{cases} \qquad \text{where } \pi_0 = \left[1 + \frac{(c\rho)^c}{c!(1-\rho)} + \sum_{n=1}^{c-1} \frac{c\rho^n}{n!}\right]^{-1}.$$

Note that, as was the case in a $M/M/1$ system, if $\rho \geq 1$, there can be no steady state – in other words, if the arrival rate is at least as large as the maximum possible service rate $\lambda \geq c\mu$, then the system "blows up".

From the client's point of view, there might be a desire to ensure that customers do not wait in line an inordinate amount of time, but there might also be a desire to minimise the amount of time for which at least one of the server is idle. In a $M/M/c$ queueing system, this steady-state probability is given by

$$P(n \geq c) = \frac{(c\rho)^c}{c!(1-\rho)}\pi_0.$$

Table 1 shows $P(n \geq c)$ for a variety of situations depending on s and ρ. Cumbersome calculations, using $W_s = \frac{1}{\mu}$, yield

$$L_q = P(n \geq c)\frac{\rho}{1-\rho}, \quad W_q = \frac{L_q}{\lambda}, \quad W = \frac{1}{\mu} + W_q, \quad \text{and} \quad L = \frac{\lambda}{\mu} + L_q.$$

Consider, for instance, a bank with two tellers. An average of 80 customers arrive at the bank each hour and wait in a single line for an idle teller. For this specific bank, the average service is 1.2 minutes. Assume that inter-arrival times and service times are exponential. Determine:

(a) The expected number of customers in the bank.
(b) The expected length of time a customer spends in the bank.
(c) The fraction of time that a particular teller is idle.

We are dealing with an $M/M/2$ system with $\lambda = 80$ customers/hr and $\mu = 50$ customers/hr. Thus, $\rho = \frac{80}{2\cdot 50} = 0.80 < 1$ and the steady-state exists.

(a) From the above table, $P(n \geq 2) = 0.71$, from which we compute

$$L_q = P(n \geq 2) \cdot \frac{.8}{1-.8} = 2.84 \text{ customers}$$
$$L = \frac{80}{50} + L_q = 4.44 \text{ customers.}$$

(b) We know that $W = \frac{L}{\lambda} = \frac{4.44}{80} = 0.055$ hr $= 3.3$ min.

(c) To determine the fraction of time that a particular server is idle, note that tellers are idle during all moments when $n = 0$, and half the time (by symmetry) when $n = 1$. The probability that a server is idle is thus given by $\pi_0 + 0.5\pi_1$. But

$$\pi_0 = \left[1 + \frac{(2\cdot .8)^2}{2!(1-.8)} + \sum_{n=1}^{2-1} \frac{2\cdot .8^n}{n!}\right]^{-1} = \frac{1}{9} \quad \text{and} \quad \pi_1 = \frac{1.6}{1!}\pi_0 = 0.176$$

and so the probability that particular teller is idle is $0.11 + 0.5(0.176) = 0.198$.

IMPORTANT NOTE: other queueing models are not understood to the same extent, and their given performance measurements may only be approximate and highly-dependent on the specifics of the problem at hand. For this reason, $M/M/c$ models are sometimes used even when their use is not supported by the data (the situation is not unlike the wide use of the normal distribution). In various applications, the empirical distributions of arrivals and service times are nearly Poisson and exponential, respectively, so that the assumption is not entirely missing the mark, but numerical simulations should not be eschewed when departures from the $M/M/c$ model are too pronounced.

1.5.5 Case Study: Wait Time Impact Model at Canadian Airports

By providing efficient and effective **pre-board screening** (PBS), the *Canadian Air Transport Security Authority* (CATSA) ensures the safety of all passengers and crew aboard flights departing Canadian airports while maintaining an appropriate balance between staffing and the wait time experienced by passengers.

The number of active screening stations and the number of passengers affect the wait times, and, as a result, budget cuts have a strong impact on the system, both in Canada and in the United States.

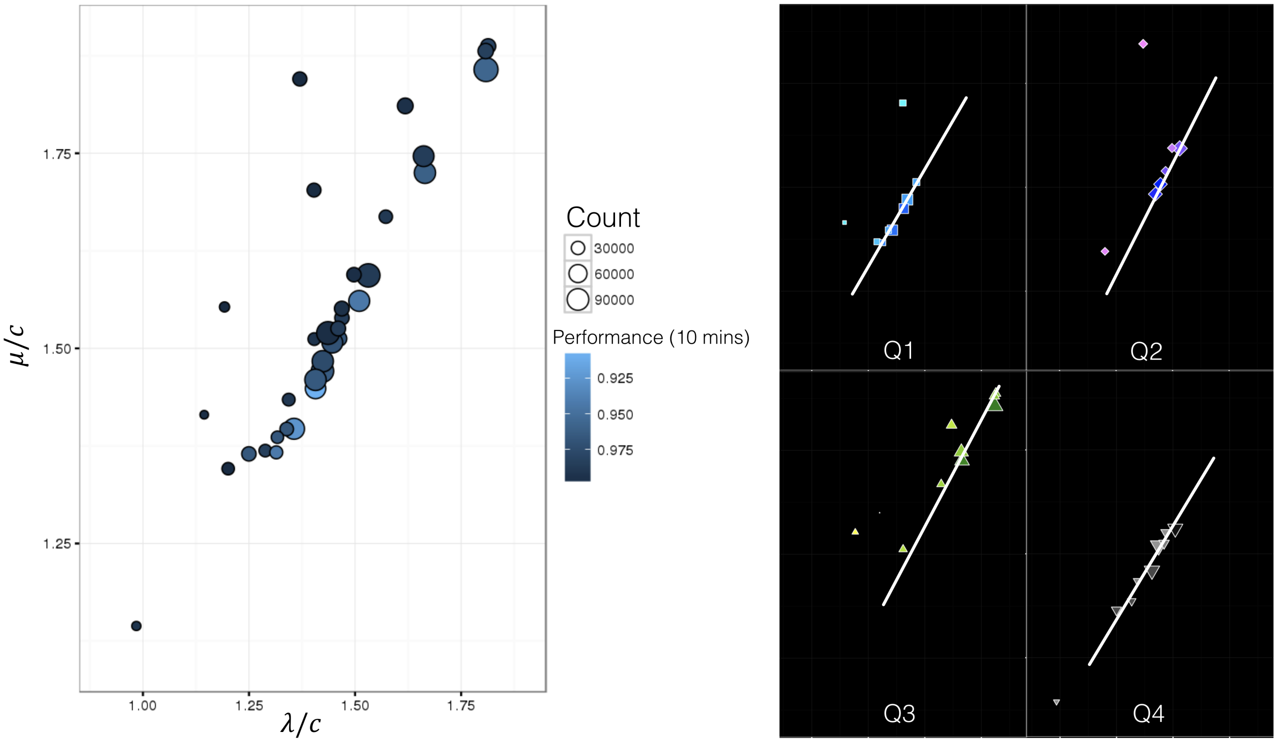


Figure 5: Visualisation of a specific checkpoint's queueing parameters – λ, μ, $\bar{c}$, passenger count, and performance (percentage of travellers waiting less than 15 minutes to be screened); the relationship between $\lambda/\bar{c}$ and $\mu/\bar{c}$ is practically linear (left), which is easier to see at the quarter level (right).

Numerous factors influence the wait time at pre-board screening checkpoints at Canadian airports: the schedule intensity of departing flights, the volume of passengers on these flights, the number of servers and processing rates at a given checkpoint, etc.

One of CATSA's goals is to ensure that the pre-board screening experience at Canadian airports is made as efficient as possible by minimizing the waiting time at checkpoints. With this in mind, the **Wait-Time Impact Model** (WTIM) was designed to achieve the following tasks:

1. provide estimates of the passenger arrival rates λ, the processing rates μ and the number of servers c at each checkpoints, using available field data;
2. calculate the Quality of Service (QoS) level (p_x, x) and determine what service level can be achieved at each checkpoint (i.e. the percentage p of passengers which will wait less than x minutes, for x fixed) for a given arrival rate λ, processing rate μ, number of servers c;
3. provide the average number of servers c^* required to achieve a prescribed QoS level (p_x, x), given an arrival profile λ^*;
4. provide quality of service (QoS) level curves $(p_x(x), x)$ (i.e. cumulative distribution curves) under various arrival rate and number of active servers for each checkpoint (where x is allowed to vary).

The queueing structure leads to some interesting insights (see Figure 5). The prediction's quality are seen in Figure 6. Details are available in the Project Summary, as well as in the Final Report (extract).

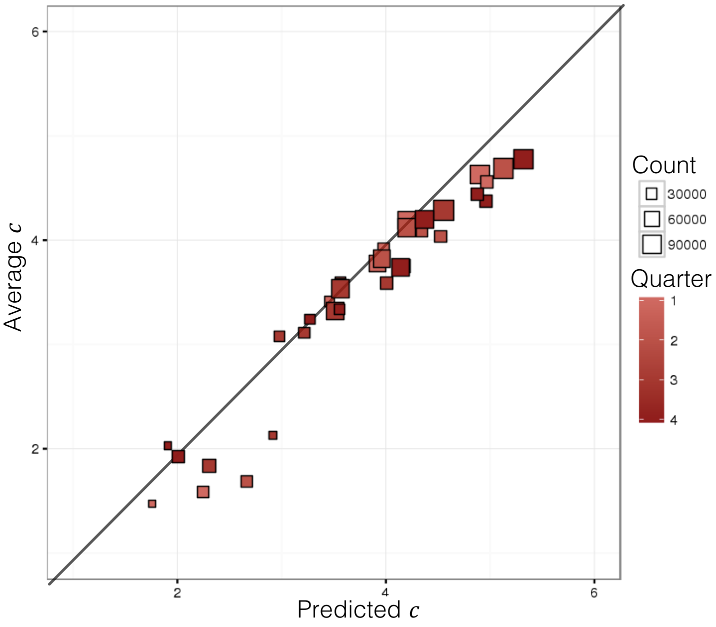


Figure 6: Predicted average number of server against actual number of server required to maintain prescribed performance, with passenger count, by quarter. The perfect prediction line is added for ease of comparison.

References

[1] Winston, W.L. [2004], Operations Research: Applications and Algorithms, 2nd ed., PWS-Kent Publishing, Boston.

[2] Ross, S.M. [2014], Introduction to Probability Models, 11th ed., Academic Press.

[3] **https://nptel.ac.in/courses/110106046/Module%209/Lecture%204.pdf**

[4] **https://nptel.ac.in/courses/110106046/Module%209/Lecture%205.pdf**

[5] **http://web.engr.illinois.edu/ dmnicol/ece541/slides/queueing.pdf**

[6] **https://www.wisdomjobs.com/e-university/quantitative-techniques-for-management-tutorial-297/poisson-and-exponential-distributions-9931.html**

[7] **https://www.percona.com/live/17/sites/default/files/the-essential-guide-to-queueing-theory.pdf**

[8] Kendall, D.G. [1953], "Stochastic Processes Occurring in the Theory of Queues and their Analysis by the Method of the Imbedded Markov Chain", Ann.Math.Stat. 24 (3): 338.

[9] Kleinrock, L. [1975], Queueing Systems, vol. 1, Wiley.

[10] Gross, D., Shortle, J.F., Thompson, J.M., Harris, C.M. [2008], Fundamentals of Queueing Theory, 4th ed., Wiley.

[11] Burke, P.J. [1956], The Output of a Queueing System, Operations Reserach vol 4 (6): 699704.

[12] Newell, G.F. [1971], Applications of Queueing Theory, Chapman and Hall.

[13] Walrand, J. [1983], A probabilistic look at networks of quasi-reversible queues, IEEE Transactions on Information Theory, vol 29 (6): 825831.

[14] **https://en.wikipedia.org/wiki/Queueing_theory**

[15] Erickson, W. [1973], Management Science and the Gas Shortage, Interfaces 4:47âĂŞ51.

Project Summary – CATSA

Wait Time Impact Model at Pre-Board Screening Checkpoints for Canadian Airports

by Yiqiang Zhao, Patrick Boily, Wenzhe Ye
Centre for Quantitative Analysis and Decision Support, Carleton University

CLIENT ORGANIZATION

The Canadian Air Transport Security Authority (CATSA) ensures the safety and well-being of passengers as they board their flights at Canadian airports each year. A federal crown corporation founded in the aftermath of the September 11, 2001 terrorist attacks on American soil, CATSA protects the public by efficiently screening air travellers and their baggage at designated airports.

CATSA offers a number of security-related services including pre-board screening, where passengers and their belongings are searched for prohibited and potentially dangerous items such as knives, firearms and explosives. CATSA also conducts hold-baggage screening – where checked baggage is screened using explosive detection equipment – and non-passenger screening where airport workers who have access to restricted areas are searched. Finally, CATSA also administers and maintains the Restricted Area Identity Card (RAIC) program.

PROJECT INTENT, SCOPE, AND OBJECTIVES

Numerous factors influence the wait time at pre-board screening checkpoints at Canadian airports: the schedule intensity of departing flights, the volume of passengers on these flights, the number of servers and processing rates at a given checkpoint, etc.

One of CATSA's goals is to ensure that the pre-board screening experience at Canadian airports is made as efficient as possible by minimizing the waiting time at checkpoints. In order to help CATSA gain a better understanding of the waiting process, CQADS developed a Wait Time Impact (WTIM) model using a Queueing Theory approach.

The original scope of the project consisted of:

1. Provide estimates of the passenger arrival rates λ, the processing rates μ and the number of servers c at each checkpoints, using the available field data
2. Calculate the Quality of Service (QoS) level (p_x, x) and determine what service level can be achieved at each checkpoint (i.e. the percentage p of passengers which will wait less than x minutes, for x fixed) for a given arrival rate λ, processing rate μ and number of servers c.
3. Provide the average number of servers c^* required to achieve a prescribed QoS level (p_x, x), given an arrival profile λ^*.
4. Implement the WTIM on a SAS platform to allow for the analysis of various scenarios (such as passenger growth, for instance) via the tweaking of a small number of parameters and whenever the available data is updated.

Upon satisfactory completion of these objectives, a second phase was initiated at CATSA's behest, with the intent of extending their implementation of the WTIM. This second phase's scope consisted of three objectives:

5. Provide quality of service (QoS) level curves $(p_x(x), x)$ (i.e. cumulative distribution curves) under various arrival rate and number of active servers for each checkpoint (where x is allowed to vary).
6. Seamlessly integrate the WTIM with CATSA's scheduling optimizer, in order to implement a one-click SAS program.
7. Provide validation and modeling analytics support for the integrated WTIM.

METHODOLOGY

In order to complete the assignment, CQADS used the following methodological steps:

1. *Exploration of available data*, in order to identify any underlying patterns and essential characteristics.
2. *Understanding the conceptual model*: including document review pertaining to CATSA's existing framework to gain a full understanding of the structure of its queueing system.
3. *Estimation of model parameters*, which required: making appropriate assumptions to simulate the processes in the queueing system according to the knowledge gained through data exploration; selecting appropriate parameter estimation methods, using the appropriate statistical inference and/or numerical method, based on the completeness and characteristics of the existing data; and conducting parameter estimation accordingly.
4. *Implementation of the conceptual $M/M/c$ model* on a SAS platform, which allowed for the discovery of the importance of certain notions whose importance only emerged after running some early scenarios through the modification of a small number of parameters (arrival profile, service time distribution, number of servers, service level, etc.), in particular when it came to vacation policy regarding the number of lines, which lead to a switch to a generalized $M/M/1$ model.
5. *Validation of the generalized $M/M/1$ model*, by comparing the estimated characteristics of the prototype queueing model (e.g. inter-arrival and service time distributions, average idle time per server, etc.) with their empirical counterparts to determine the validity of the conceptual model. The conceptual model was found to be mostly invalid until a key link between the average arrival rate, the processing rate and the number of lines was established. This combined generalized $M/M/1$ and Regression model produced good results in most cases, but in certain instances, a departure from the empirical data could still be identified. Further analysis lead to a breakthrough and the introduction of a Departure parameter. The final model, then, combined the $M/M/1$, Regression and Departure hypotheses.
6. *Performance evaluation of the final model* was achieved in two ways. A preliminary performance evaluation pitted the model favourably against historical data, but the ultimate test came once predictions were compared to data that were collected after the final model was delivered, again very favourably.
7. *Documentation of the final model*: a technical report providing an overview of the model, as well as describing and justifying the various assumptions, was written and delivered to CATSA stakeholders.
8. *Knowledge transfer* was achieved through meetings (in person or by phone) and email exchanges detailing the progress, increasing in frequency as the deadline approached.
9. *Provision of on-going support* to CATSA's model users allowed for a number of improvements, both in scope and in implementation.

PROJECT SUMMARY

The available data covered 26 checkpoints, at 8 Canadian airports. At each checkpoint, the pre-board screening process is structurally similar: passengers arriving at the beginning of the main queue may have their boarding passes scanned at the S_1 position (the start of the waiting queue), but they are always scanned at the S_2 position (as they are being processed).

For each checkpoint, 3 datasets were available for each year:

- the *Raw Data* which contains – for each passenger reaching the end of the queue at S_2 – the date, scan time at S_1, scan time at S_2 and the wait time between S_1 and S_2;
- the *Checkpoint Utilization Report* which records – for each day of the year and each non-overlapping 15-minute block – the maximum number of open processing lines, and

- the *Waiting Time Report* which consists of the subset of the *Raw Data* for which S_1 and S_2 are both available (and for which observations with anomalous and/or outlying wait time behaviour have been removed by CATSA).

The data was then grouped into meaningful clusters exhibiting properties that can be characterized by the same Poisson process, which allows for proper estimation of queueing model parameters, under the assumption that the queueing model $M/M/c$ model was valid.

[One difficulty with this approach is that, in practice, the number of servers c varies with time, according to a vacation policy which depends on a variety of factors. As such, it is extremely difficult to model. This is problematic since the sought QoS level (p_x, x) depends not only on the arrival rates, but also on the processing rates, which themselves depend, among other things, on the number of open servers. Switching to a generalized server (behind which the actual servers are hidden) circumvents this issue, but at the cost of not immediately being able to retrieve the number of servers c from the generalized $M/M/1$ model.]

The average arrival rates λ for each cluster were computed from the *Raw Data* using Burke's Theorem and were shown to indeed follow a Poisson process as the inter-arrival times between consecutive S_2 events were i.i.d. exponential random variables with parameters λ, lending support to the generalized $M/M/1$ hypothesis. The average wait times $\overline{W}_q$were then estimated using the *Wait Time Report*.

[An analysis of the reasons for the omission of those observations without an S_1 scan from the *Wait Time Report* suggests that using the latter to estimate the cluster average wait times $\overline{W}_q$ is likely to affect the predicted QoS levels, especially in the small wait time regime. However, since short wait times are not likely to cause consternation among the general public, this issue may not arise in practice and can be side-stepped in the estimation phase.]

The estimated processing rates $\hat{\mu}_M$ and QoS levels $(\hat{p}_M, x)$ were easily recovered from the relationships

$$\overline{W}_q = \frac{\hat{\rho}_M}{\hat{\mu}_M - \lambda}, \quad \hat{p}_M = 1 - \hat{\rho}_M e^{-(\hat{\mu}_M - \lambda)x},$$

where $\hat{\rho}_M = \lambda/\hat{\mu}_M$ represents the estimated traffic intensity.

Since these relations do not hold if the generalized $M/M/1$ hypothesis fails, the need to validate it became more pressing. The simplest way to do so was to compare the wait times generated by the model to those of the empirical data: were the estimated QoS curves $\hat{p}_M(x)$ "close to" the empirical QoS curves $p(x)$? Using two different metrics (largest relative difference ratio, largest area ratio), we showed that the generalized $M/M/1$ assumption, while not exact, is a reasonable one to make at the checkpoint level.

[This result is achieved without explicitly invoking the number of open servers c. Granted, that number is implicitly involved in the determination of the average wait times $\overline{W}_q$, but it does not change the fact that it cannot be recovered using solely the tools provided by queueing theory. The Regression assumption asserts that, on a quarterly level, the cluster processing rates $\mu = \mu(c, \lambda)$ is a function of the number of active servers c (hidden behind the generalized server) and the arrival rates λ, and that this functional relationship is the same for all regression clusters making-up a given quarter.]

Using the *Checkpoint Utilization Report*, the average service rates per line $\hat{\mu}_M/c$ and average arrival rates per line λ/c were estimated for each checkpoint, quarter, and cluster, and then regressed against one another to determine the optimal regression parameters $\hat{a}, \hat{b}$, yielding new estimates $\hat{\mu}_R = \hat{a}c + \hat{b}\lambda$ for the cluster processing rates. Thus, estimates for the QoS level $(\hat{p}_R, x)$ were easily computed, without explicitly referring to processing rates, using

$$\hat{p}_R = 1 - \frac{\lambda}{\hat{a}c + \hat{b}\lambda} e^{-(\hat{a}c+\hat{b}\lambda-\lambda)x},$$

which held as a direct consequence of the combined $M/M/1$ and Regression assumptions.

Using the two validation metrics introduced above, it was shown that the combined assumptions, while proving slightly less valid than the $M/M/1$ hypothesis on its own, still provided reasonably close QoS estimates at the quarter and checkpoint levels.

[This lessened validity should come as no surprise, as there is no way to extract the number of clusters c without postulating an external relationship of the form $\mu = \mu(c, \lambda)$, and that the simple linear regression form used necessarily introduces some uncertainty. Some of that uncertainty might decrease using a more complex regression function.]

In order to predict the number of servers required to meet a given QoS level (p, x) at a given checkpoint during a given quarter (i.e. for a given pair of regression parameters a, b), for a given arrival profile λ, it then sufficed to solve for c, yielding

$$c_R = \frac{1}{ax}\left[\mathrm{W}_0\left(\frac{\lambda x}{1-p} e^{\lambda x}\right) - b\lambda x\right],$$

where W_0 is the main branch of the Lambert W-function.

[Unfortunately, W_0 cannot be evaluated by elementary needs except at special values and so one has to depend on efficient numerical algorithms to recover c_R. As SAS does not lend itself particularly well to repeated algorithmic computations, it becomes imperative to find a quick and relatively accurate alternative approach, such as the following approximation, which can be implemented in SAS:

$$c_R \approx \frac{1}{ax}\left[\ln\left(\frac{\lambda x}{1-p} e^{\lambda x}\right) + 1.031 \ln\left(\ln\left(\frac{\lambda x}{1-p} e^{\lambda x}\right)\right) + 0.207 - b\lambda x\right],$$

valid for $e \leq \frac{\lambda x}{1-p} e^{\lambda x} \leq e^{1000}$.]

For any given checkpoint, quarter, and cluster, it was thus possible to compare the actual number of open servers c (given by the *Checkpoint Utilization Report*), and the estimated value c_R given the actual arrival rate λ and the actual QoS level (p, x). Plotting c_R against c for all clusters strongly suggested that the prediction and the actual values were linked at the checkpoint level according to $c = \hat{d} \cdot c_R$, for some checkpoint departure parameter $\hat{d}$. Computed values of d near 1 for nearly all checkpoints further validated the combined model. The final prediction for the number of servers was further refined by setting $c_D = \hat{d} \cdot c_R$.

In theory, it is thus possible to forecast the number of servers c_D for a cluster using only its regression parameters a, b, its departure parameter d, an arrival rate λ, and a QoS level (p, x). The validation procedure in this case is slightly different: it makes little sense to compare the predicted value c_D with the actual number of servers c found in the historical data as the prediction depends not only on the forecasted arrival rate (which is likely to be different from the historical rate), but also on the attained QoS level (for which an independent forecast is unavailable).

The best validation alternative, then, is to wait for new data to be collected, to determine the actual cluster arrival rate and QoS level to be used in the forecast in order to provide a prediction c_D, and to compare it with the actual c recorded over the data collection period.

DIFFICULTIES AND ADDITIONAL TASKS

Apart from the usual issues surrounding the transfer of large datasets, the specific issue of the lack of algorithmic computability of the Lambert W-function in SAS, and the technically difficult modeling problem, additional tasks were requested by CATSA and the scope of the project was accordingly extended.

These tasks included:

- the modification of the originally requested $M/M/c$ queueing model to a $M/M/1$ queueing model;
- the introduction of the combined $M/M/1$ and regression hypothesis to recover the number of servers c;
- the identification and clean-up of data integrity issues for three airports (YUL, YYC, YVR), and
- the conversion of a MATLAB non-linear solver for the computation of the Lambert W-function to an approximation which could be implemented in SAS.

RESULTS AND RELEVANCE

While CQADS has not been made privy to all details of the validation work conducted by CATSA on 2013 data (especially when it comes to exact accuracy figures), the CATSA Project Authority has let it be known informally that the predictions and QoS level curves which were generated by the WTIM were found to be quite in agreement with the actual data: it is CQADS' understanding that the model is currently in use within Operations Reporting and Analysis at CATSA.

PROJECT LOGISITCS

Timeline

Phase I	Contractual Tasks	14-Jun-14 31-Aug-14	to
	Additional Tasks (at CATSA's request)	01-Sep-14 30-Sep-14	to
Phase II	Contractual Tasks	11-Oct-14 11-Nov-14	To

Resources/Personnel

Yiqiang Q. Zhao, Ph.D.
Professor, School of Mathematics and Statistics
Carleton University
Subject Matter Expert (Queueing Theory)

Patrick Boily, Ph.D.
Managing Consultant, CQADS
Project Lead / Senior Analyst

Wenzhe Ye
Consultant, CQADS
Analyst (Queueing Theory)

Jun Gao
Junior Consultant, CQADS
Analyst

Total Effort Level

580 hours (estimate)

		Zhao	Boily	Ye	Gao
Phase I	Contractual Tasks	60	100	120	50
	Additional Tasks (at CATSA's request)		100		
Phase II	Contractual Tasks		150		
	Total:	**60**	**350**	**120**	**50**

Dollar Value

Phase I	Contractual Tasks	$22,000.00	
	Additional Tasks (at CATSA's request)	$22,000.00	
Phase II	Contractual Tasks	$10.000.00	
	Total:	$54,000.00	(+ HST)

CATSA Project Authority

Maryam Haghighi, Ph.D.
Senior Advisor

Operations Reporting and Analysis
99 Bank Street
Ottawa, Ontario K1P 6B9
Canada

613 993-7094
maryam.haghighi@catsa.gc.ca

Extract from the report

Wait Time Impact Model at Pre-Board Screening Checkpoints for Canadian Airports

by Yiqiang Zhao, Patrick Boily, Wenzhe Ye

[…]

1.2 Model Outline

The model establishes a relationship between the arrival rates, the service rates, the number of servers and the service levels. Basic concepts, process descriptions, and limitations are provided in §2.

The WTIM is best described via the flow chart of Figure 1 on the next page (the various concepts will be defined as they arise in the corresponding section):

1. computation of the arrival rates λ from the raw data (§3.2);
2. computation of the distribution of the number of servers c from the checkpoint utilization reports (§3.3);
3. computation of the waiting time distribution W_q from the waiting time report (§3.4);
4. computation of the QoS levels (p,x) from the waiting time report (§3.4);
5. computation of the estimated QoS levels $(\hat{p}_M,x)$ under the $M/M/1$ assumption (§3.5);
6. validation of the $M/M/1$ assumption based on a comparison of $(\hat{p}_M,x)$ and (p,x) (§3.6);
7. computation of the estimated service rates $\hat{\mu}_M$ under the $M/M/1$ assumption (§3.5);
8. computation of the seasonal checkpoint regression parameters a,b under the combined $M/M/1$ and $Regression$ assumptions (§4.1);
9. computation of the estimated QoS levels $(\hat{p}_R,x)$ under the combined $M/M/1$ and $Regression$ assumptions (§4.2);
10. validation of the combined $M/M/1$ and $Regression$ assumptions based on a comparison of $(\hat{p}_R,x)$, $(\hat{p}_M,x)$ and (p,x) (§4.3);
11. prediction of the number of servers c_R under the combined $M/M/1$ and $Regression$ assumptions (§5);
12. validation of the combined $M/M/1$ and $Regression$ assumptions based on a comparison of c_R and c (§5.3);
13. computation of the checkpoint departure parameters d under the combined $M/M/1$, $Regression$ and $Departure$ assumptions (§5.3);
14. computation of the estimated QoS levels $(\hat{p}_D,x)$ for various projected arrival growth rates λ^* under the combined $M/M/1$, $Regression$ and $Departure$ assumptions (§6.2);
15. prediction of the number of servers c_D for various projected arrival growth rates λ^* under the combined $M/M/1$, $Regression$ and $Departure$ assumptions (§6);
16. final validation of the combined $M/M/1$, $Regression$ and $Departure$ assumptions based on a comparison of $(\hat{p}_D,x)$ and c_D with empirical data (§6.3).

In order to illustrate the WTIM process, the details are worked out on a step-by-step basis for the Domestic/International Checkpoint at the Edmonton International Airport (YEG), based on 2012 data. The results are shown at the end of each section. A summary of results for all checkpoints is also provided, as well as recommendations and suggested next steps.

[…]

2.1 Definitions

The various mathematical concepts to which the report will refer are described below:

- An $M/M/c$ **queueing model** describes a system where arrivals form a single queue and are governed by a Poisson process (the first M), units arriving are processed by c servers and service times are exponentially distributed (the second M).
- A **Poisson process** is a stochastic process where the time between any two consecutive event has an exponential distribution with parameter λ.
- The **arrival rate** is the rate at which passengers arrive for PBS (i.e. passengers per minute), the **service rate** is the processing rate at a screening line (i.e. maximal potential throughput), the **number of servers** is the number of screening lines and the **service level** is the percentage of people waiting less than a given number of minutes at a checkpoint.

2.2 Description of the PBS Process

At each checkpoint, the PBS process is structurally similar: passengers arriving at the beginning of the main queue may have their boarding passes scanned at the S_1 position, but they are always scanned at the S_2 position (see Figure 2).

[…]

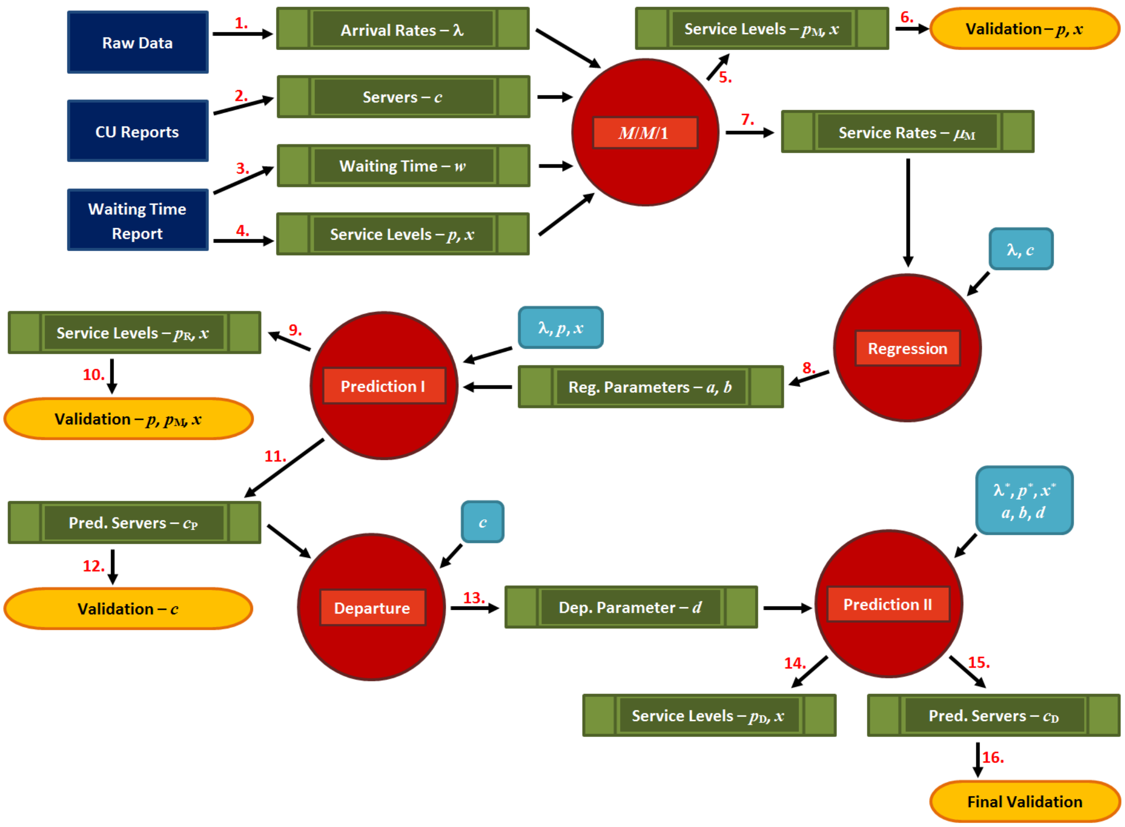


Figure 1 – **WTIM flow.** The dark blue rectangles are CATSA-provided data inputs; the green boxes indicate computed and derived values; the red circles are conceptual nodes; the light blue boxes represent carry-over values, and the orange cells are validation steps.

3. $M/M/1$ Queueing Model

One of the difficulties for the situation under consideration is that the number of servers varies with time, according to different factors: there are times when all servers are busy, others when a number of open servers are idle, and the number of open servers changes according to some vacation policy which it is difficult to model. This is problematic when using an $M/M/c$ model as service rate estimates depend, amongst other things, on the number of open servers.

It is possible to circumvent this issue altogether, without invoking Vacation Models, by noticing that an $M/M/c$ queueing system may be viewed as an $M/M/1$ queueing system where the servers are hidden behind a generalized server (see Figure 3, on the next page). Under that interpretation, the service rates can be estimated independently of the number of servers. Furthermore, not only do $M/M/c$ results still hold for $M/M/1$ (simply by setting $c = 1$ in the appropriate theorems), but the quantities to be computed tend to be simpler in this case.

While this conceptual simplification has removed some of the difficulties associated to server vacation, there remains another problem: the theory of $M/M/1$ systems, alone, is not sufficient to recover (and later predict) the actual (and hidden) number of servers for the checkpoint. This situation can be addressed by finding another way to link the arrival rates, the estimated service rates and the number of servers (see §4.1 for more details).

3.1 Clustering

In order to better predict the average behaviour of a system and its possible outcomes, a wide range of typical patterns must be considered. When analyzing the behaviour of queues, it may become necessary to group the data into meaningful clusters exhibiting similar properties (for example, properties that can be characterized by the same Poisson process).

This approach allows for proper estimation of queuing model parameters (arrival rates, processing rates, etc.), which in turn yields the most reliable results. The selection of the appropriate cluster size relies on finding a balancing point between two extremes:

- In order to properly define the stochastic process, a minimum amount of data with similar properties is required. If clustering is not performed (i.e., if the clusters are too large), the data may present different characteristics which cannot be represented by a single Poisson process.
- On the other hand, if the clusters are too small, they may not contain enough data to capture the underlying properties. More importantly, clusters that cover too short a period are unlikely to exhibit the statistical behaviour of the process.

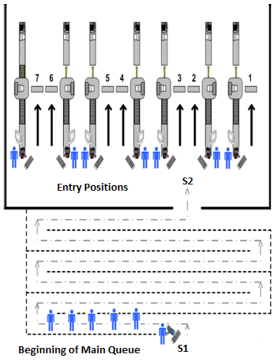


Figure 2 – **Schematics of pre-board screening (PBS).** Passengers enter the main queue, where their boarding pass may be screened at S_1. Once they reach the end of the main queue, their boarding pass is screened at S_2 and they are sent to one of the active lines for processing (image provided by CATSA). In practice, it may happen that only the S_2 reading is available.

A preliminary analysis of the model's accuracy was assessed based on the following criteria: *Checkpoint*, *Weekly patterns* (day of week vs weekday/weekend), *Seasonal patterns* (season vs month) and *Daily patterns* (2-hour period vs 4-hour period). The cluster combination that produced the most encouraging queueing results when compared against actual reports was: checkpoint, weekday/weekend, season, 4 hour-period. Clustering also plays a role in the Regression stage of the model (see §4 for details), but the optimal regression cluster combination need not be the same as the queueing cluster combination.

3.2 Computing the Average Arrival Rate

Since not all boarding passes are scanned at S_1, the Wait Time report (S_1 data) cannot be used to derive the cluster arrival rates. The $S_1 - S_2$ line-up (main queue) is a birth-death process (i.e. a reversible one-dimensional Markov chain). In particular, the forward chain $S_1 - S_2$ and its reverse are stochastically identical and the arrival epochs of the reversed chain are the departure epochs of the forward chain. We can then use Burke's Theorem for $M/M/c$ queues.

Theorem 1 (BURKE'S THEOREM, [1]) *Consider an $M/M/c$ queue in the steady state with arrivals modeled by a homogeneous Poisson process with rate parameter λ. Then the departure process is also a homogeneous Poisson process with rate parameter λ.*

This does not rule out the possibility that, at a particular time, the arrivals at S_1 could be greater than the departures at S_2, due to the inherent randomness of Poisson processes. But all S_1 arrivals will eventually leave at S_2 and thus the fluctuations at S_2 follow the same statistical property governing arrivals to the queue. Therefore, the arrival rates can be estimated by using data readings at S_2 within a given cluster.

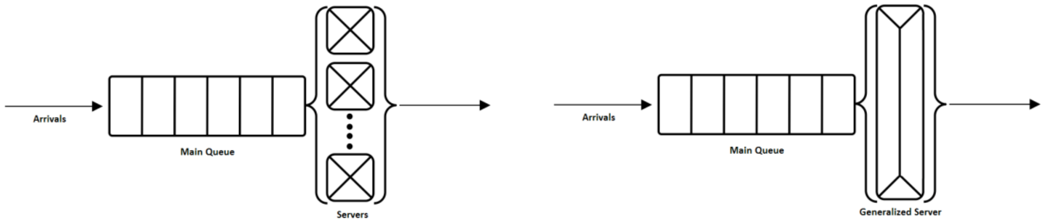


Figure 3 – **Queuing systems.** Conceptual visualization of an $M/M/c$ queue (on the left) as an $M/M/\mathbf{1}$ queue (on the right); the c servers can be considered as 1 generalized server.

It remains only to show that arrivals follow a homogeneous Poisson process in each cluster (this is a common hypothesis). To do so, one must show, assuming that the number of arrivals in the cluster by time t is denoted by $N(t)$, that (see [3, 4] for details)

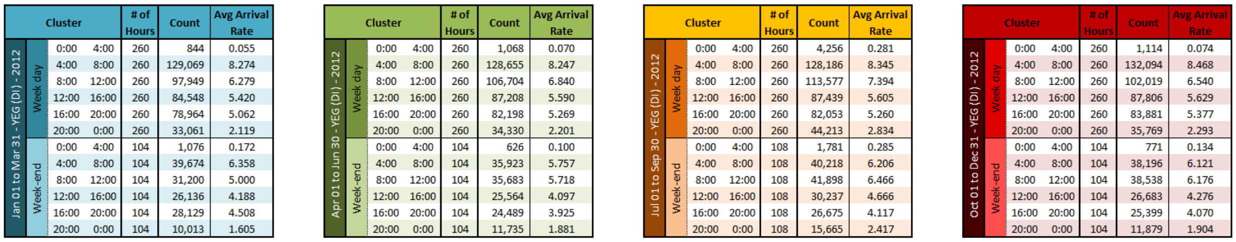
1. $N(t)$ is a counting process with independent and stationary increments, and
2. the number of arrivals in any time interval of length t is Poisson-distributed with mean λt, i.e. for all $s, t \geq 0$,

$$P[N(t+s) - N(s) = n] = e^{-\lambda t}\frac{(\lambda t)^n}{n!}, \quad n = 0,1,\ldots$$

The first assumption is satisfied with the introduction of clusters. The third assumption holds if the **inter-arrival times** (the times between consecutive events) are independent and identically distributed (i.i.d.) exponential random variables with the same rate λ: analysis of S_2 in the raw data suggests that this is the case.

The total counts of arrivals for each cluster at YEG's D/I checkpoint based on 2012 data are shown in Table 1. Note that the arrival rate λ is simply calculated by dividing the count in each cluster by the number of minutes in each cluster, independently of the open status of the checkpoint during the period spanned by the cluster. A low arrival rate may thus indicate either that checkpoint traffic was low or intermittent for the cluster, or that it was closed for some or all of the period that it spans.

[…]



Cluster				# of Hours	Count	Avg Arrival Rate
Jan 01 to Mar 31 - YEG (DI) - 2012	Week day	0:00	4:00	260	844	0.055
		4:00	8:00	260	129,069	8.274
		8:00	12:00	260	97,949	6.279
		12:00	16:00	260	84,548	5.420
		16:00	20:00	260	78,964	5.062
		20:00	0:00	260	33,061	2.119
	Week-end	0:00	4:00	104	1,076	0.172
		4:00	8:00	104	39,674	6.358
		8:00	12:00	104	31,200	5.000
		12:00	16:00	104	26,136	4.188
		16:00	20:00	104	28,129	4.508
		20:00	0:00	104	10,013	1.605

Cluster				# of Hours	Count	Avg Arrival Rate
Apr 01 to Jun 30 - YEG (DI) - 2012	Week day	0:00	4:00	260	1,068	0.070
		4:00	8:00	260	128,655	8.247
		8:00	12:00	260	106,704	6.840
		12:00	16:00	260	87,208	5.590
		16:00	20:00	260	82,198	5.269
		20:00	0:00	260	34,330	2.201
	Week-end	0:00	4:00	104	626	0.100
		4:00	8:00	104	35,923	5.757
		8:00	12:00	104	35,683	5.718
		12:00	16:00	104	25,564	4.097
		16:00	20:00	104	24,489	3.925
		20:00	0:00	104	11,735	1.881

Cluster				# of Hours	Count	Avg Arrival Rate
Jul 01 to Sep 30 - YEG (DI) - 2012	Week day	0:00	4:00	260	4,256	0.281
		4:00	8:00	260	128,186	8.345
		8:00	12:00	260	113,577	7.394
		12:00	16:00	260	87,439	5.605
		16:00	20:00	260	82,053	5.260
		20:00	0:00	260	44,213	2.834
	Week-end	0:00	4:00	108	1,781	0.285
		4:00	8:00	108	40,218	6.206
		8:00	12:00	108	41,898	6.466
		12:00	16:00	108	30,237	4.666
		16:00	20:00	108	26,675	4.117
		20:00	0:00	108	15,665	2.417

Cluster				# of Hours	Count	Avg Arrival Rate
Oct 01 to Dec 31 - YEG (DI) - 2012	Week day	0:00	4:00	260	1,114	0.074
		4:00	8:00	260	132,094	8.468
		8:00	12:00	260	102,019	6.540
		12:00	16:00	260	87,806	5.629
		16:00	20:00	260	83,881	5.377
		20:00	0:00	260	35,769	2.293
	Week-end	0:00	4:00	104	771	0.134
		4:00	8:00	104	38,196	6.121
		8:00	12:00	104	38,538	6.176
		12:00	16:00	104	26,683	4.276
		16:00	20:00	104	25,399	4.070
		20:00	0:00	104	11,879	1.904

Table 1 – **YEG DI 2012 totals.** Number of hours, count of arrivals and average arrival rates, per cluster, per quarter (1st – teal, 2nd – green, 3rd – yellow, 4th – red).

5.3 Validating the Combined Model Using Departure at the Checkpoint Level

The relative accuracy of the formula

$$c_R \approx \frac{1}{ax}\left[\ln\left(\frac{\lambda x}{1-p}e^{\lambda x}\right) + 1.031\ln\left(\ln\left(\frac{\lambda x}{1-p}e^{\lambda x}\right)\right) + 0.207 - b\lambda x\right], \quad \text{when } e \leq \frac{\lambda x}{1-p}e^{\lambda x} \leq e^{1000},$$

used to estimate the average number of servers c_R required to reach the QoS level (p, x) at a checkpoint with regression parameters a, b and average arrival rate λ, suggests another method to validate the combined model.

For any given checkpoint, the plot of c_R against the actual c strongly suggests that the variables are linked according to $c_R = d \cdot c$, for some d.

Linear regressions once again determine the optimal $\hat{d}$ for each checkpoint. The **departure parameter** $\hat{d}$, then, serves as a measure of the predictive model's departure from reality. If $\hat{d} \approx 1$ (i.e. if $c_R \approx c$), then the assumptions that go into the combined model are justified *a posteriori*, in the context of predicting the average number of active servers. The modified predictions $c_D = c_R/\hat{d}$ for a checkpoint where $\hat{d}$ is large or close to 0 (i.e c_R is a poor approximation of c) may still end up being accurate (i.e $c_D \approx c$), but in that case a careful analysis should be undertaken to understand whether any anomalous activity is in play.

The regression of c_R against c for YEG's D/I checkpoint (based on 2012 data) is shown in Figure 6 (on the next page). The departure parameter for this checkpoint is $\hat{d} \approx 1.0161$, which is reflected by the tight linear fit of the two variables. As such, the original prediction c_R requires only a very slight modification.

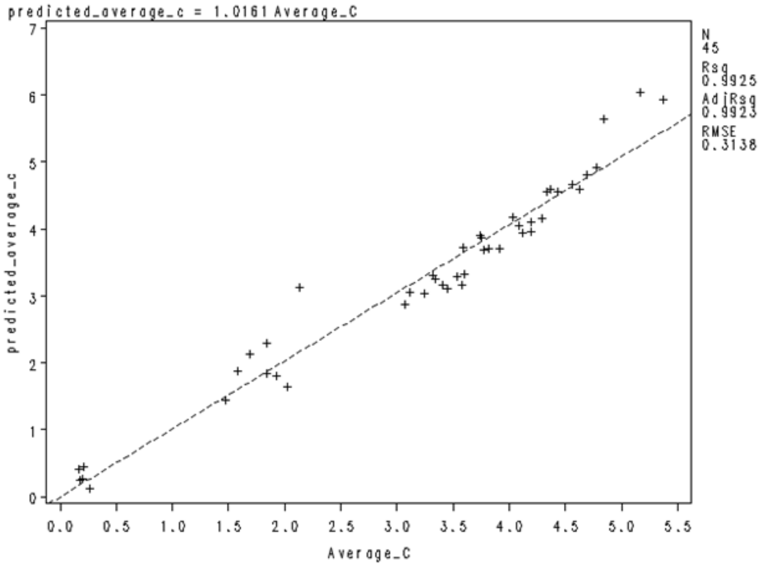


Figure 6 – Departure Regression. Regression of the predicted average number of servers against the actual number of servers.

[…]

8. Recommendations and Final Comments

Perhaps the foremost conclusion is that the $M/M/1$ model on its own provides the best QoS levels estimates, while the best estimates for the average number of active servers are provided by the Departure model.

This discrepancy may be partly explained by the fact that, in any modeling endeavour, some loss of information is inevitable due to the necessity of making simplification assumptions. Below is a list of possible issues which could affect the WTIM's accuracy:

1. The underlying arrival processes are roughly Poisson, and the wait time distributions are roughly conditionally exponential for each cluster; depending on the distance between the theoretical process and the empirical data, the $M/M/1$ assumption may be inappropriate.
2. The wait time distribution may be seriously biased as not every boarding pass has been scanned at S_1, and there is no easy way to verify how representative the subset of those for which wait time data is available actually is.
3. The server vacation policy is unknown, and may not be uniformly adhered to (if one even exists).
4. The actual number of active servers is only crudely approximated by the maximum number of active lines within a 15 minute block.

5. The service rate seems to depend on factors other than the number of active servers and the arrival rate, leading to wildly different outputs for similar inputs and contributing to the lessened accuracy of the regression model when estimating QoS levels.
6. Different checkpoints might require different optimal clustering strategies.

It might be possible to minimize some of that information loss simply by selecting a slightly more sophisticated regression functional form linking the average arrival rate per line and the average service rate per line. Preliminary analysis suggests that the choice

$$\mu = \mu(c, \lambda) = ac + fc^2 + b\lambda$$

may provide better QoS results. Further analysis is needed in that regard, as it is clear that other factors need to be included in order to get the best possible fit and to minimize the number of clusters which become unstable as a result.

Finally, it is conceivable that while adding more historical data to the model could have a useful effect, going too far back into the past may bias the results if policy changes have led to characteristically distinct underlying data over the years. It seems clear that at least one year's worth of data is needed, but, as the datasets only contained trustworthy data for the year 2012, it is still too early to get a definitive answer on this topic.

[…]

References

[3] Newell, G.F. [1971], *Applications of Queuing Theory*, Chapman and Hall.

[4] Ross, S.M. [2010], *Introduction to Probability Models*, 10th ed., Academic Press.

(End of extract)