

T1 _____
T2 _____
T3 _____
T4 _____

Problem Chosen

D

F1 _____
F2 _____
F3 _____
F4 _____

2017

MCM/ICM

Summary Sheet

(Your team's summary should be included as the first page of your electronic submission.)

Type a summary of your results on this page. Do not include the name of your school, advisor, or team members on this page.

Summary

Bottlenecks that passengers take time to take care of their carry-on properties before X-ray scanning and that the structure of the checkpoints is not satisfying are spotted, and detailed recommendations for the airport security management to raise throughput of the security checkpoints, to improve the **passengers' satisfaction** and to keep the cost relatively low are given with cultural factors quantified and their impacts on the models discussed.

We divided the security check process into two phases and regard the entire as two queueing models in series. By analyzing the given data, the document check is found to be a Poisson queueing ($M/M/c$) in Kendall notation, and Erlangian process ($M/E_k/c$) is concerned in the scan check process. **Numeric solution of ($M/E_k/c$)** is introduced using simulation technique.

Assumed that arrivals of passengers for a flight obey **normal distribution**, the varying passenger flow in a period of time can be generated from the real data and used for our simulation. We also believe that the mean μ and the variance σ^2 of the distribution is **culture-related**. We also find that the passenger flow can be changed by recommending the passengers their arrivals, and thus better result of the passengers' waiting time can be achieved.

We also suggested several methods to improve the design of the checkpoint, including shortening the distances between identical checkpoints and more rational **human resource allocation**.

Virtual queueing is recommended as an approach to improve passengers' experience, and modify the conventional First In First Service queueing discipline to **partial priority queueing discipline** as well. A partial priority queueing discipline is put forward to reduce the remaining time variance of the passengers and to decrease the number of passengers that missed their flights, thus better passenger satisfaction is reached.

We also introduced culture-related factors for passenger arrival recommendation and priority queueing discipline. For the latter, an "**acceptability factor**" named α is used to denote the acceptability of strict priority discipline. And the examples of different cultures are given to illustrate this idea.

Validation of each model are made in our essay to make them convincing. We later assess the models and give a complete guide for the security managers to optimize the airport security check workflow. Weaknesses and further work that is not implemented in our essay are also pointed out.

Time Counts! Less Waiting & Better Airports

1 Introduction

1.1 Background

Airport security check has been improved ever since the 911 attack. Although enhanced security means safer flights, however the complicated procedure may also increase the passengers' waiting time and add cost to the U.S. Transportation Security Agency (TSA). Under some extreme circumstances, passengers have to wait for hours (and they are often recommended to be earlier for 2–3 hours, which often lead to confusion) (Hetter, 2016). Thus, to shorten the passengers' waiting and designing a more efficient security check procedure is vitally important.

TSA is now in controversy for causing long queues waiting for security check. We, the Internal Control Management (ICM) team, trying to find a solution, faces the problems below:

1. Identify the bottlenecks of the current security check workflow.
2. Improve the process with modifications, and illustrate how the modifications work.
3. Find how to allow the modified process to be compatible with different culture backgrounds and lower the variance of the passengers' waiting time.
4. Make suggestions on the policy for the security manager, with concern of the former requirements and corresponding models.

1.2 Analysis and Approach Overview

For problem 1, we divide the security check process in two parts: Phase 1, document check; and Phase 2, luggage and body scanning. The former is a Poisson queue, while the latter concerns an Erlangian model. Simulation is practiced as a means of solving multi-server Erlangian model. By testing the total waiting time's sensitivity to changes on numbers of parallel servers in the two phases respectively, the bottlenecks of the workflow can be spotted.

To solve problem 2, considering the influential factors of a queueing process, modifications will be put forward to optimize and avoid congestions.

The current TSA recommended passengers' arrival time is used to build a model of the passenger arrival behavior at an airport, and we assume the arrivals in time for one certain flight obey normal distribution, and in a small time interval, the arrivals of all passengers for all flights obey an exponential distribution. We will modify the arrival recommendation strategy to influence passengers' arrival behavior.

Another direction to improve the current process is to provide more robust security check service with greater capacity. A few suggestions and their verification or explanation will be given.

Besides the performance, justifiability also counts. Virtual queueing under other disciplines (Zhao, et al., 2016) will be a good practice. Queueing discipline modification is culture-sensitive, and thus lead to the discussion of the third problem.

The following diagram illustrate the above ideas in a more visual way:

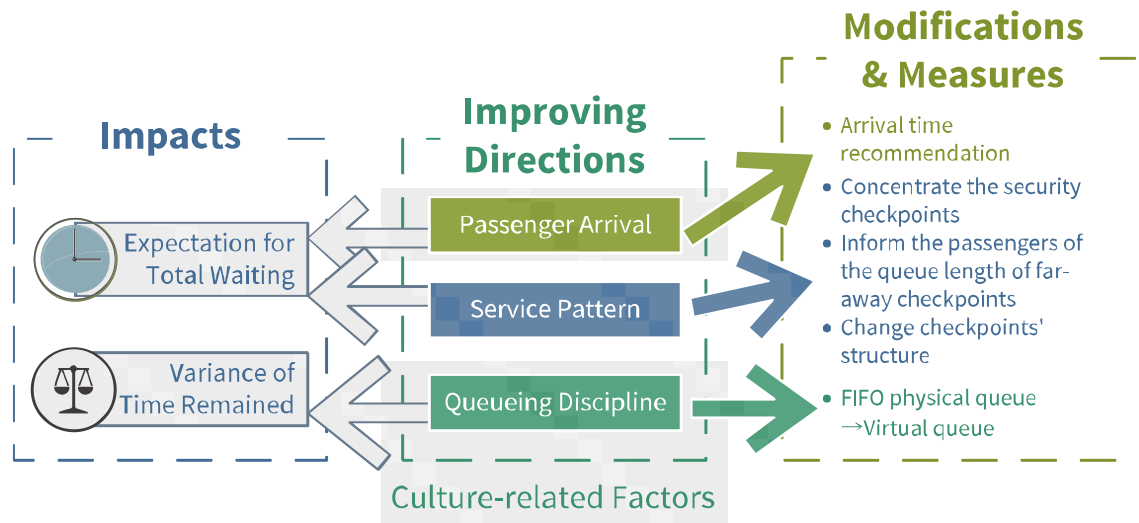


Figure 1 Optimizing directions and our modifications, where “service pattern” concerns the service rate and number of servers.

2 Assumptions

1. Individual variability is not considered for the servers, the checkpoint structures, *etc.*
2. Although the number of lanes opened in the scan check process is dynamic, we assume that the service capability is always at its maximum. That is to say, spare lanes will be open, so long as the arrival exceeds the current capability.
3. Assume that every passenger will choose to wait in the queue that minimizes their waiting time at every checkpoint.
4. Points in time that passengers arrive at the airport for a certain flight obey normal distribution.
5. TSA Pre-check won't contribute much to the congestion compared with the normal one.
6. Almost everyone would arrive at the airport for at least half an hour.

See 4.1 Security Check Process—Overview, and 4.3 Queueing Model Specification for detailed interpretation and assumptions of the given datasheet.

3 Symbols and Notations

Symbol or Notation	Specification
$(a/b/c):(d/e/f)$	Kendall notation, where a, b, c, d, e and f denote the inter-arrival-time distribution, the service time distribution, the number of parallel servers,

	queueing discipline, restriction on system capacity, and the source of the arrival (usually infinity) respectively. (Taha, 2014)
M	Denote an exponential distribution for inter-arrival time and service time, i.e. a Poisson distribution for arrival and remove rate.
E_k	Erlang distribution type k .
\hat{x}	Estimated value of x .
λ_d, μ_d	Arrival and service rates of the document check process.
λ_s, μ_s	Arrival and service rates of the scan check process.
$\Gamma(\alpha)$	$\Gamma(\alpha) = \int_0^{\infty} x^{\alpha-1} e^{-x} dx$
$N(\mu, \sigma^2)$	Normal distribution with mean μ , and variance σ^2 .

4 Queueing Model for Security Check Process

4.1 Security Check Process—Overview

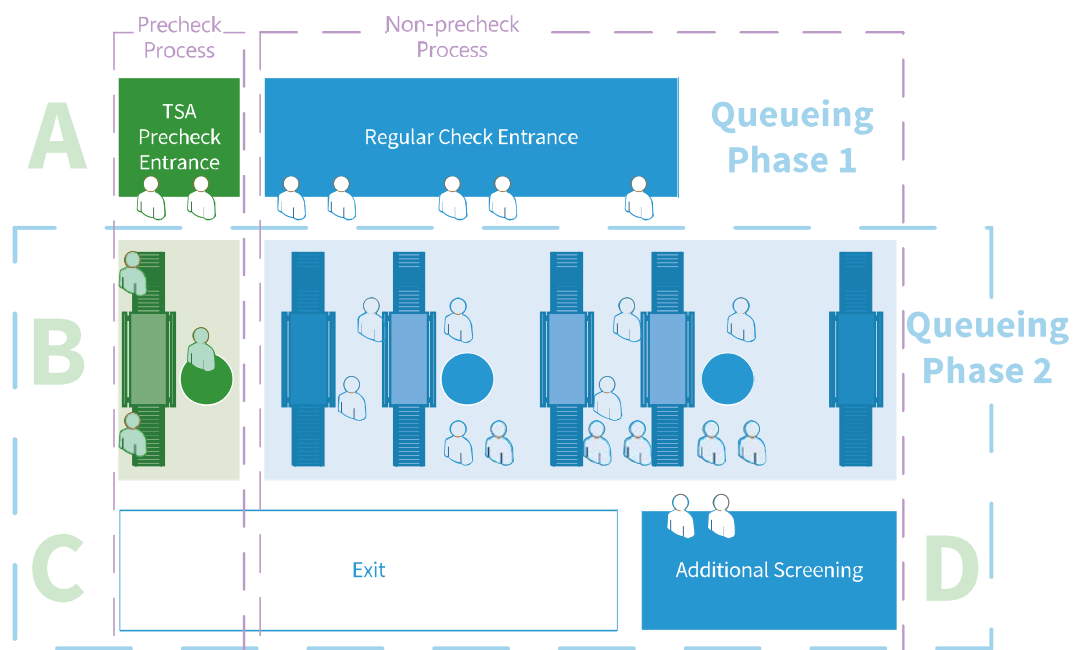


Figure 2 TSA Security checkpoint

The detailed security check process of one single checkpoint given in the diagram above. According to TSA's policy, we classify the security processes in two types (pre-check included and no pre-check concerned process), and divide the each process in two queueing phases. In Phase 1, the passengers' identity documents will be checked, after which they enter Phase 2, where luggage and body screening will be accepted. Two different types are in essence the same queueing with different parameters (number of servers, service time, *etc.*). And the entire security check process can be seen as two queueing models in series.

To better clarify the whole process, a procedure sequence diagram is given below:



Figure 3 Procedure sequence diagram

A: time waiting for the document check, **B:** document checking time

C: time waiting for sending luggage for X-ray scanning, **D:** X-ray scanning time, **E:** other possible luggage checks

F: time waiting for millimeter wave scan, **G:** body scan time, **H:** other possible body checks

The yellow shaded part is Phase 1 as we introduced in previous paragraphs, and the rest Phase 2.

The given Excel datasheet contains time records of the airport checkpoints (whose differences denote the inter-arrival time), time taken of the ID check process of previous checked passenger (*i.e.* **B** in the diagram), millimeter wave scan timestamps (differences of which are the millimeter wave scanning time, shown as **G** in the diagram), timestamps that luggage getting out of the X-ray scan (differences denote **E**), and time to get scanned property (**D**, **E**, **F**, **G**, **H**).

By analyzing the given datasheet, patterns of the airport security check behaviors, such as the distribution of passengers' inter-arrival time and service time at each section, can be discovered. Therefore a complete queueing model can be specified.

4.2 Significance Test on Two TSA Officers

The given datasheet involves timing of two different TSA officers, the significance test here is to verify that individual variability does not contribute much to the service time.

Suppose that variances of the service time of the two TSA officers are equal, $\sigma_1^2 = \sigma_2^2 = \sigma^2$.

We make the null hypothesis H_0 that service time expectations of both officers are the same, *i.e.* $\mu_1 = \mu_2$.

The combined sample variance of the two officers, $s_w^2 = 12.927$.

At 5% level, $\frac{|\bar{x}_1 - \bar{x}_2|}{s_w \sqrt{1/n_1 + 1/n_2}} = 1.375 \leq t_{0.025}(14) = 2.145$, the null hypothesis is not rejected.

Assume that the document check service time obeys the exponential distribution. The parameter of the distribution, *i.e.* service rate, is estimated, $\widehat{\mu_d} = 0.09465$.

4.3 Queueing Model Specification

The two phases of the process can be specified as a series of queues by giving the arrival and service time distribution pattern.

4.3.1 Arrival

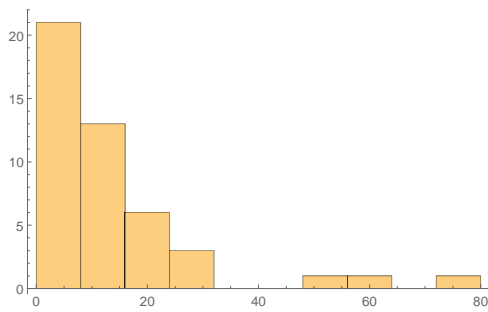


Figure 4 Frequency histogram of non-pre-check arrivals

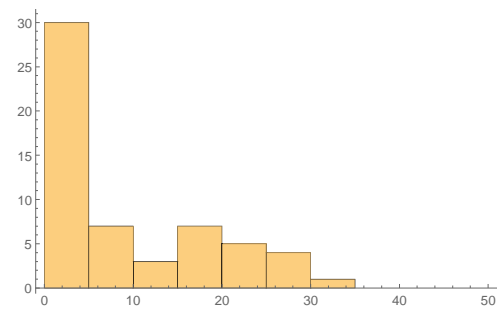


Figure 5 Frequency histogram of pre-check arrivals

According to the histograms above, we suppose that the time intervals of both the regular and pre-check arrivals obey exponential distribution $f(x) = \lambda_d e^{-\lambda_d x}$.

Maximum likelihood estimation of the parameter λ_d :

$$L(\lambda_d) = \prod_{i=1}^n f(x_i) = \prod_{i=1}^n \lambda_d e^{-\lambda_d x_i} = \lambda_d^n e^{-\lambda_d \sum_{i=1}^n x_i}$$

$$\ln L(\lambda_d) = n \ln \lambda_d - \lambda_d \sum_{i=1}^n x_i, \quad \frac{d \ln L(\lambda_d)}{d \lambda_d} = \frac{n}{\lambda_d} - \sum_{i=1}^n x_i = 0$$

Therefore $\widehat{\lambda}_d = \frac{n}{\sum_{i=1}^n x_i} = \frac{1}{\bar{x}}$, and the expectation, $E(\widehat{\lambda}_d) = E\left(\frac{1}{\bar{x}}\right) = \frac{1}{E(x)} = \lambda_d$, is an unbiased estimation.

Thus the arrival rates of the regular and pre-check procedures are estimated to be $\widehat{\lambda}_1 = 0.077244$ and $\widehat{\lambda}_2 = 0.10882016$, respectively.

Here, a goodness of fit test is done, and each of the procedures are classified into 10 classes. Let's take the regular (without pre-check) procedure as an example:

Class	0-8	8-16	16-24	24-32	32-40	40-48	48-56	56-64	64-72	72-80
True Frequency	21	13	6	3	0	0	1	1	0	1
Predicted Probability	0.461	0.248	0.134	0.072	0.039	0.021	0.011	0.006	0.003	0.002
Predicted Frequency	21.204	11.430	6.161	3.321	1.790	0.965	0.520	0.280	0.151	0.081

Table 1 Regular procedure, non-pre-check, true frequencies and predicted frequencies

$$\chi^2 = \sum_{i=1}^{10} \frac{(N_i - np_i)^2}{np_i} = 15.898 < \chi_{0.05}^2(9) = 16.919$$

Thus we consider the above as an exponential distribution.

And similarly, fit for the pre-check procedure, $\chi^2 = 15.599 < \chi_{0.05}^2(9) = 16.919$.

Since the inter-arrival time obeys exponential distribution, the number of passengers' arrival in unit time obeys Poisson distribution.

4.3.2 Phase 1 Queueing Specification

We also consider the service time of Phase 1 obeys exponential distribution, which is a common practice for simple cases in queueing theory, and the service rate is below.

$$\mu_d = 0.09465$$

Based on the above discussion, the Kendall notation of Phase 1 is $(M/M/c)$, where M denotes a Poisson process and c is the number of parallel servers. The arrival rate of this

queueing model depend on certain conditions, and c is given by the airport facility background. We assume that the service rate is a constant $\mu_d = 0.09465$ for each server.

4.3.3 Phase 2

Because the two queueing models are in series, we have $\lambda_s = \mu_d$. Phase 2 is also a multiple-server model, but the serving process is much more complicated, besides the histogram does not obey exponential distribution, apparently.

An Erlangian model is introduced, Erlang service is widely used to model combined serving processes, where consequent queueing models are connected in series. k identical Poisson servers in series generate an Erlang type k service. However, it is often the case that parameter k is not strictly equal to the number of phases in a server. PDF of Erlang type k distribution is given by

$$f(x) = \frac{1}{\Gamma(\alpha)\beta^\alpha} x^{\alpha-1} \exp(-x/\beta), (\alpha, \beta > 0; 0 < x < \infty)$$

with the expectation $E(x) = \alpha\beta$ and variance $Var(x) = \alpha\beta^2$. Parameter α and β is related with k and the service rate μ (Gross, et al., 1985):

$$\begin{cases} \alpha = k \\ \beta = \frac{1}{k\mu} \end{cases}$$

Fit the distribution for the total service time in Phase 2 (time to get scanned property), our results are, $k = 4, \mu_s = 0.0357$.

Therefore, the Kendall notation of Phase 2 queueing is $(M/E_k/c)$, where the input M is identical to the output of Phase 1, E_k is an Erlang type $k = 4, \mu_s = 0.0357$ distribution, and the number of parallel servers (*i.e.* the number of X-ray scanners and millimeter wave scanners) is depend on specific airport.

$(M/E_k/c)$ queueing does not have a good analytical solution; our numerical solution using simulation technique will be introduced later.

4.4 Section Summary

The airport security check process consists of two phases, first of which is an $(M/M/c)$ queueing process, and the second is an $(M/E_k/c')$ queueing. Arrival rate of the second is the service rate of the first. Parameters of the distributions can be estimated using the given data.

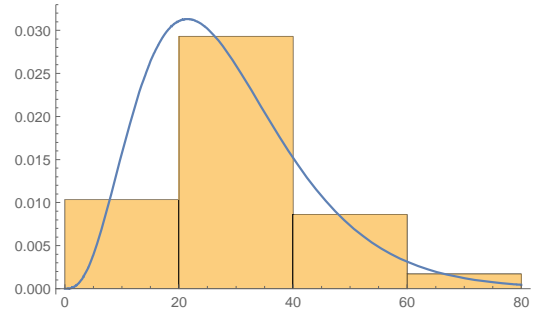


Figure 6 Histogram of the service time in Phase 2, together with a PDF plot of an Erlang distribution.

5 Queueing Simulation and Bottlenecks Spotting

5.1 Estimating Expenses

According to (Learn.org; PayScale.com), income of a security officer is \$28,624–\$58,987 and of an airline security screener is \$23,262–\$54,015. That is the human resource expenses of the security check. Costs of security devices are mostly one-off consumptions. The price of an airport X-ray baggage scanner is at \$20,000–50,000 and of a millimeter wave full body scanner is at \$100,000 (Alibaba.com). And the maintenance cost is mainly from the electricity. Another fact is that one single lane of the scan check process requires 4 security screeners on average. Therefore, we drew a conclusion that the cost of one lane (server) in Phase 2, is about the cost of 4 desks (servers) at Phase 1.

5.2 Basic Ideas

A MATLAB program is written to simulate the entire security check process. To build up the numeric solution, Markov chains are used for Erlangian queueing. (Zeng, et al., 2011)

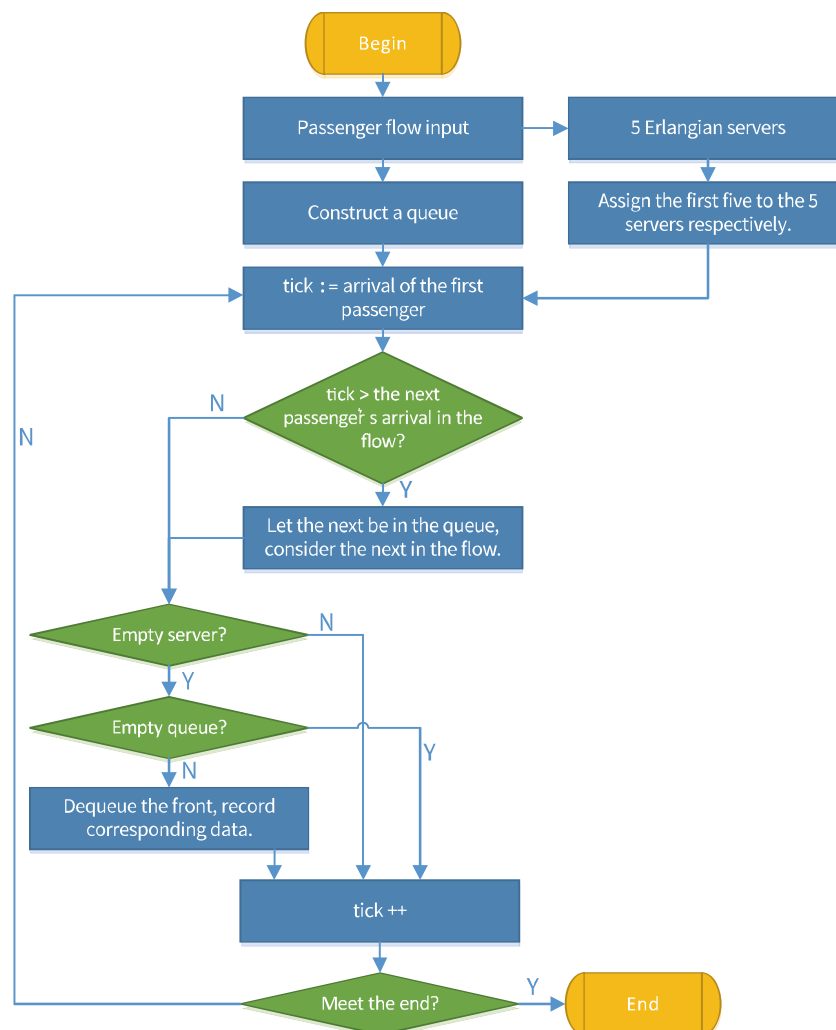


Figure 7 MATLAB simulation specification

Since there is no analytical solution of an $(M/E_k/c)$ (Phase 2) queueing model, a simulation of the entire security check process is put forward, and both Phase 1 and Phase 2 are concerned in the model in series.

Given the parameters of each probability distribution at the two phases, we generate series of random variates to simulate the process. First, random variates obeying exponential distribution is generated as people's inter-arrival time. We then calculate and store the arrival timestamps, service time (generated random variables obeying another exponential distribution) and waiting time (worked out with the previous passenger's waiting time, service time and the arrival interval) of each passenger in a table. Besides getting results of waiting time spans, the removal rate of Phase 1 is also observed, which is used as the arrival of Phase 2. After working on Phase 1, we do the same on Phase 2, but the new service is Erlangian. Finally, expectations of total waiting time are evaluated.

To spot the bottlenecks in the process, we tested the improvements of total waiting time expectations when 4 document check servers are added (row “Document Check Added” in the table below), and 1 scan check server is added (row “Scan Check Added”).

5.3 Spot the Bottlenecks!

The simulated security process has a changeable arrival rate λ_d , and both of the initial numbers of the document check servers and of the scan check servers are assigned 5, as is the case in Figure 2, which is a small scale for an airport. The service rate of one document check server is assigned $\mu_d = 0.09465$, and that of one scan check service is $\mu_s = 0.0357$. Service of Phase 2 is Erlangian type $k = 4$. These three parameters are the same to what we have mentioned before.

Below lists the expectations for total waiting time at different arrival rates. Row “Original” is the total waiting time expectation of process consist of $(M/M/5)$ and $(M/E_k/5)$. Row “Document Check Added” for the case that 4 document check servers are added, that is $(M/M/9)$ for Phase 1 and $(M/E_k/5)$ for Phase 2. For row “Scan Check Added”, 1 scan check server is added, that is $(M/M/5)$ and $(M/E_k/6)$.

λ_d	3.247	3.710	4.329	4.478	4.638	4.810	5.194
Original	31349	22632	12781	10807	8988	7391	3668
Document Check Added	31092	21878	11832	10671	8925	7215	3291
Scan Check Added	18258	10643	3045	1680	271	88	24

Table 2 Simulation results of the total waiting time (in seconds)

We find that, the document check process is not the bottleneck of the process, but the scan check. By adding scan check lanes, service capability of the whole system is significantly increased. We can even observe that some divergent queueing cases converge (*e.g.* $\lambda_d = 5.194$).

5.4 Comments

Based on the above analysis, bottlenecks restricting the queueing capacity and efficiency comes from the scan process Phase 2. From the given data, we see that the X-ray scan time is usually fast and steady, and is not likely to be the bottleneck. However, time taken for the passenger to take off clothes and to take care of their carry-on properties contribute much to the next passenger's waiting (time C in Figure 3), especially when someone is heavy with carry-on properties or not experienced in air travelling.

We would recommend the airport to allow the passengers to take a bin for their to-be-scanned properties before queueing, and thus waiting time in Phase 2 will be reduced.

6 Comments on the Safety

Firstly, on the stand of the airport, any security accident, including terrorist attacks, hijacking or aircraft destroying, if happens, will cause unbearable blame from the public opinion, which is unacceptable for the airport. Thus it does not sound like a good idea to reduce the links in security check procedure to fasten check-in and to lower the expenses.

A fundamental principle of our later discussion is that, no procedure in the security check chain is omitted. Means like optimizing queueing processes, arranging human resources, guiding passengers' arrival can be implemented as ways to improve the passenger experience.

7 Passenger Arrival Behavior Modeling

7.1 Passenger Flow Generating

Flight records of O'Hare International Airport on Wednesday, Jan. 18, 2017 were found on (Flight Stats), including the departure time and information of the plane models. Most of the airlines are Boeing 737-800 whose capacity is usually 104–189 passengers.

To simulate the passenger arrival behavior, *i.e.* the varying passenger flow in a time period, we assume that the capacity of every flight is about 189, and the points of the passengers' arrival time obey normal distribution. Since TSA recommends passengers to be at the airport 2 hours earlier before the flight for domestic departures, we model the passengers' arrival time for each flight to obey normal distribution $N(\mu, \sigma^2)$, and let μ be 2 hours before the departure and $\mu + 3\sigma$ be the point in time that there is half an hour remained.

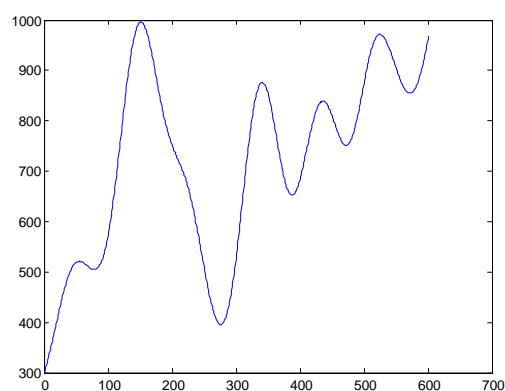


Figure 8 Varying passenger flow in a time period, from 6:30 a.m. to 16:40 p.m., O'Hare International Airport, Jan. 18, 2017

We then add these PDFs together and get the dynamic passenger arrival data. The shown plotted graph (Figure 8) is such a practice on timespan between 6:30 a.m. to 16:40 p.m.

We can see a peak at 9:30 a.m. and at around 11:00, the passenger flow get down to a relatively low level (this is probably because of massive departures).

Based on the above passenger flow distribution, we will simulate the security check queuing process at a domestic departure checkpoint later.

7.2 Arrival Time Recommendation

We tend to believe that the passenger arrival behavior can be changed by recommending their arrival time, however this is often much more complex due to airport location or cultural norm of a certain nation.

We calculated the mean waiting time of the passengers by simulating using the varying passenger flow introduced in the last subsection, at different normal distributions as we mentioned above. Only one checkpoint is taken into account in the simulation, and the flow is generated with domestic departure flights. When the average arrival of the passengers for a certain flight is 120 minutes earlier, its mean waiting time in the security check process is about 26 minutes as shown in the chart below.

Difference of μ and the departure time (min)	90	105	120	150	180
σ (Ensured that $\mu + 3\sigma$ is at 30 min earlier the departure)	20	25	30	40	50
Passenger flow per hour	491	498	502	503	498
Mean waiting at Phase 1 (sec)	0.0953	0.0633	0.0781	0.0654	0.0631
Mean waiting at Phase 2 (sec)	82.9986	56.146	26.0189	21.7186	16.9306

Table 3 Result for waiting time at different recommended time

From the chart, we curiously found that the earlier the passengers arrive, the less the mean waiting time will be. This is quite expected, because earlier arrivals and more uniformly distributed arrival intervals can reduce the “strikes” of the passenger flow.

Cultural norm impact on the arrival rate distribution (“varying passenger flow”, in other words) will be discussed later. Measures like sending text messages to passengers for recommending their arrivals based on real-time data as well as their cultural background will help.

8 Checkpoint Redesign

8.1 Concentrating the Checkpoints

8.1.1 Basic Ideas

In queueing theory, it is found that $(M/M/c)$ queueing has a better performance than c parallel $(M/M/1)$ queueing processes, with shorter mean queue length and better mean waiting time. (See Appendix 13.1 for Proof) Simulation of a more complex system, as in the security check process is made and the results are consistent. According to our assumption, so long as

the passengers could see the servers, they will choose to wait in the queue that minimizes their waiting time. Thus making the passengers know the queueing conditions at checkpoints works.

8.1.2 Measures to Take

Security checkpoints to terminals are shown in the O'Hare terminal map. (Chicago Department of Aviation, 2012). We would recommend the airport to reduce the distances between each checkpoints at Terminal 1 and 3. This will help the passengers to choose a checkpoint minimizing their waiting among the equivalent checkpoints at each terminal.

Another possible solution which is equivalent to the former is to monitor the real-time queue length of each checkpoint at Terminal 1 and 3 and install LED boards to show these information. This also helps the passengers to choose a proper checkpoint.

The above measures may both avoid some queue's getting too long and cause a congestion.

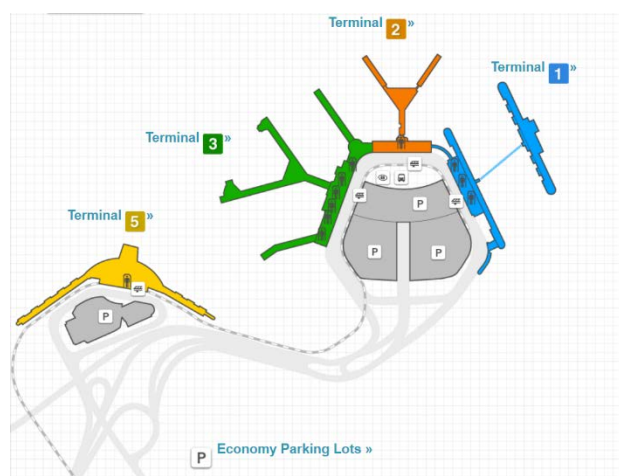


Figure 9 O'Hare Terminal Map, © 2012 Chicago Department of Aviation, URL: <http://www.flychicago.com/OHare/EN/AtAirport/map/default.aspx> (Chicago Department of Aviation, 2012)

8.2 Checkpoint Structure

8.2.1 Distribution of Human Resource between the 2 Phases

This suggestion on checkpoint redesign is a continued section of Section 5.3, Spotting Bottlenecks. We know that the cost of a lane in Phase 2 is 4 times more than that of a desk at Phase 1. Optimizing the checkpoint structure and keep the cost relatively low can balance the service rates of the document check process and the scan check process, as well as saving money.

This is a programming problem in operation research. Assume the number of document check servers is k_1 , and that of the scan check is k_2 .

$$\begin{aligned} & \text{Minimize } k_1 + k_2, \\ & s. t. \begin{cases} 0 < k_1, k_2 \leq 5 \\ k_1 \mu_d > \lambda_d \\ k_2 \mu_s > \lambda_s \end{cases} \end{aligned}$$

We trial the number of servers with our simulation to find the appropriate ratio. Our result is, when there are 5 lanes of scan check servers, 3 document check servers are enough for most common cases. Cut on document check service desks won't cause bottlenecks, and according to our previous study, adding new lanes will greatly improve the security check service.

<i>Time since the initial (min)</i>	<i>391-450</i>	<i>451-510</i>	<i>511-570</i>	<i>571-630</i>	<i>631-690</i>	<i>691-750</i>	<i>751-810</i>	<i>811-870</i>	<i>871-930</i>	<i>931-990</i>
<i>Passenger flow /h</i>	563	433	277	473	430	484	550	541	662	617
<i>Positions needed for Phase 1</i>	2	2	1	2	2	2	2	2	3	3
<i>Mean waiting time at Phase 1</i>	28.7135	7.6859	103.3296	12.2084	7.2299	11.2017	21.9008	17.9357	4.1698	3.188
<i>Positions needed for Phase 2</i>	5	4	3	5	4	5	5	5	5	5
<i>Mean waiting time at Phase 2</i>	27.3362	31.1131	14.1786	7.986	26.8053	7.2921	19.351	15.4442	6172	96.1114

Figure 10 Numbers of positions needed at Phase 1 and Phase 2 to make the queue length converge

8.2.2 Dynamic Assignment of Duty in a Time Span

Based on our previous work on passenger flow distribution in a period of time and the programming in the above subsubsection, we evaluate the duty assignment at each checkpoint every hour.

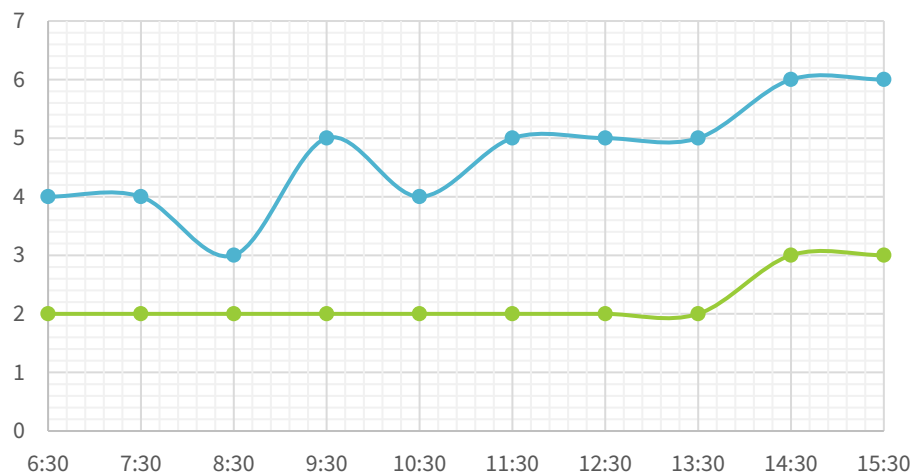


Figure 11 Dynamic assignment of duty with Phase 1 in blue and Phase 2 in green

Allocating fewer staff at the troughs, and more at the peaks, will help the security check system running properly at relatively low cost.

And for the specific problem we are now considering, we practice this method and plan to allocate the duty like this:

—	8:30 a.m.	2 for document check and 4 for scan check
8:30 a.m.	— 13:30 p.m.	2 for document check and 5 for scan check
13:30 p.m.	— 16:30 p.m.	3 for document check and 6 for scan check

Table 4 Dynamic duty allocation

8.2.3 Sensitivity Test

A sensitivity test is made for our model. The test is done via simulation.

	Original	Decreased flow	Increased service ability
<i>Passenger flow</i>	502	452	502
<i>Serve ability (Phase 2)</i>	0.036	0.036	0.040
<i>Mean waiting time</i>	41.70	12.57	7.834

Table 5 Sensitivity test result, simulations for each case are done for 5 times

We find that when the passenger flow decreases by 10%, the mean waiting time decreases by 69.9%. When the document check service ability is raised by 10%, the mean waiting time almost remains the same. And if the service ability at Phase 2 is increased by 10%, the mean waiting time drops as much as 81.2%!

From the above test, we drew a conclusion that the queueing system is sensitive to the scan check service ability, while a low sensitivity is observed for changes of the document check ability. This again verifies the bottlenecks that we found earlier, and optimization can be given around the found bottlenecks.

9 Virtual Queueing

Waiting room can be useful to improve passengers' experience. What's even better is that the queueing discipline can be modified with this method. The conventional physical queue method is a strict FIFO (First In First Served), while virtual queueing can be more flexible. With the popularization of smartphones, new queueing methods that basically share the same process as virtual queueing, such as mobile queueing, but are more convenient and more sufficient are introduced, which will also provide us more possibilities to modify the queueing process. (Wikipedia; Zhao, et al., 2016)

9.1 Strict Priority Queue

Passengers' satisfaction can be quantified by the number of passengers that missed their flights, mean remaining time after security check and the variant of the remaining time. Although FIFO sounds like an absolutely equitable practice, it is frustrating under some extreme circumstances. Imagine an "early bird" comes to the airport 6 hours in advance of his or her departure. If an unfortunate congestion occurs, the early passenger would hold up the passengers behind whose flight might be to take off. This is not acceptable.

Consider a strict priority queue, where the priority is strictly based on the departure time of each passenger. We now prove that this approach achieves better passenger satisfaction with our simulation.



Figure 12 Passenger satisfaction

Our approach is to change the queueing discipline in the simulation, and we do find a strict priority queue performs better in the above aspects. The testing data involves a passenger flow of 723 passengers per hour. Our results are as follows:

	1	2	3	4	5	6	7	8	9	10	Mean
<i>FIFS</i>	231	292	233	280	346	298	333	316	209	274	281.2
<i>Priority</i>	19	0	0	35	0	31	0	2	7	0	9.4

Figure 13 Numbers of passenger that missed their flights. 10 experiments are done here.

And the means and the variances of remaining time:

	Mean	Variance
<i>FIFS</i>	6918	6.8787×10^6
<i>Priority</i>	6864	3.9684×10^6

Figure 14 Means and variances of the remaining time

Means are almost the same (this is expected, because expectation for the waiting time won't vary with queueing discipline, which is a theorem in queueing theory. The slight difference is caused by the generation of random variates) while variance of queueing where priority is considered seems to be better.

9.2 Virtual Queueing Concerning Partial Priority Discipline

However, a strict priority queue does not sounds rational to a lot of people. (And of course, it is culture dependent.) A better queueing discipline is introduced here to be a fairer and more efficient option, and above all, cultural adaptable.

We introduce an “acceptability factor” α ($0 \leq \alpha \leq 1$) denote the acceptance of the strict priority discipline. We then use a partial priority discipline, which is a mixture of strict FIFS and strict priority discipline.

Let t_a denote a passenger's arrival time, and t_d denote his or her departure time. We use

$$t_p = \alpha t_d + (1 - \alpha)t_a$$

as the priority of the passengers, and sort the waiting queue in this discipline.

We tested $\alpha = 0.5$ for a passenger flow at 721 passengers/h, which is quite an extreme case for a single checkpoint. Below is our result.

	Passengers Missed Flight	Variance
<i>FIFS</i>	281.2	6.8787×10^6
<i>Partial Priority</i>	71.3	5.1711×10^6
<i>Strict Priority</i>	9.4	3.9684×10^6

Table 6 Number of passengers that missed the flight (10 hours) and variance of remaining time, $\alpha = 0.5$

We find the partial priority discipline is also effective in raising passenger satisfaction.

10 Inter-cultural Applicability

10.1 Cultural Impacts on Punctuality

Different cultural norms have an impact on passengers' punctuality.

People in the United States, Canada and many northern European countries have a linear view of time, and are more likely to be punctual. This is also true to East Asian countries, like China and Japan, where social efficiency is seen important. In these countries, people tend to have a keen sense of time, and would usually be early for an event. On the contrast, Spaniards or people from Southern Europe are not so punctual. Passengers from such nations often hurry to board. (Business Insider)

Viewing the differences of varied cultural norms with our passenger arrival behavior model (see Section 7), we can say that the passenger flow in countries where people are more punctual is often more smooth and steady. More mathematically, the mean arrival time μ is often far earlier than the passengers' flight departures, and the variance of their arrival distribution σ^2 is relatively small. Case for countries that are not so punctual is just opposite, as shown in the following plotted graph.

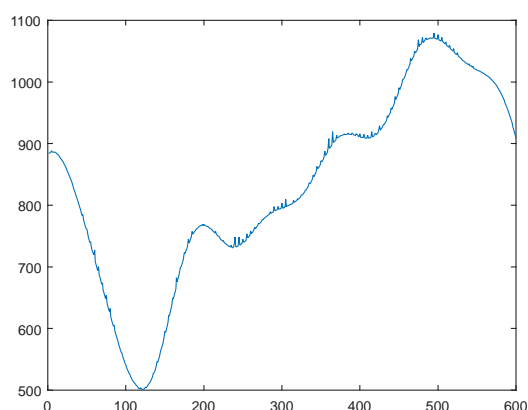


Figure 15 Smooth passenger flow, μ is 150 minutes earlier the departure, and $\sigma = 40$.

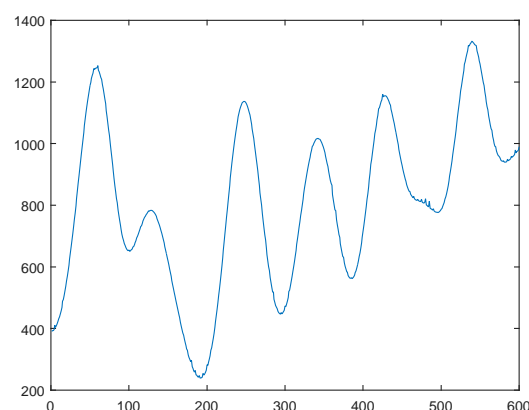


Figure 16 Rough passenger flow, where μ is 90 minutes earlier, and $\sigma = 20$.

Airports with more smooth and steady passenger flow does not requires too much human resource, and its assignment can be more flexible, for passenger flow peaks are seldom observed. Scale effect that adopt $(M/M/c)$ rather than c parallel $(M/M/1)$ also tend to work. Instead, it is not this satisfying in a less punctual nation. More peaks will be found, and massive human resource is needed to deal with the sudden congestions.

As we have mentioned before, sending text messages in advance to recommend passengers' check-in time personally based on the current passenger flow at the airport is a very good practice. But when different cultural norms are considered, this is not that easy, for people's punctuality must be taken into account. Since there is no such practice now, we cannot determine the μ and σ^2 of people's arrival behavior in different countries, yet we can prove it certainly works.

10.2 Cultural Impacts in Queueing Process

Another cultural impact on our model is the acceptability of priority queueing in different nations. People in the United States respect others' privacy as well as they hate other people cutting in line. Swiss are punctual and take collective efficiency as the most important. While in China, jumping queues is usual because individual efficiency is respected, and people tend to forgive such behaviors if the queue jumper has reasonable excuses.

The above varied thinking patterns may lead to different acceptability in different nations. Our acceptability factor α (See 9.2 for definition), enlighten the culture-related factor in our model.

For example, passengers from the United States tend to think the new queueing discipline is not acceptable, because it damages the FIFO principle. However, Swiss might be in favor of our method because it dramatically raises the collective efficiency, and thus a greater α could be assigned. Chinese would probably think a neutral α was a good practice.

11 Discussion and Conclusions

11.1 Policy Recommendations

We here give a complete guide on security check modifications and policy making for the TSA security manager based on our previous model, as our solution to problem 4.

To Security Managers

To solve the airport security check congestion, and to optimize the throughput at the airport security checkpoints, we give you the following recommendations on policy making based on our investigation.

On **human resource distribution**, we would recommend fewer document check servers, while more for scan check. Bottlenecks of the security check workflow are at the scan check phase, and we have proved that 2 or 3 desks for document check is enough for a 5-lane scan check area. Dynamic allocation of document check servers and lanes opened should be reserved as a way to reduce unnecessary cost, besides, our approach to solve this has been put forward in our paper.

On **facilities**, waiting rooms should be built for virtual queueing. If possible, adding lanes for scan check at each checkpoint will dramatically increase the passenger throughput. Shorten the distances between identical checkpoints in the terminal will help passengers to choose one that minimizes their waiting. LED boards providing information of queue lengths at each checkpoint works in the same way. If possible, we also recommend a mobile queue system to be built, together with a queue management system, these facilities, thanks to the new information technology, will enhance the efficiency encouragingly. Providing passengers with bins for

their to-be-scanned properties in advance will also shorten their waiting during the scan check process and avoid bottlenecks.

On **safety and the cost**, no reduction of links on the current security check procedure is recommended, since the risks cannot be assessed in value. Dynamic assignment of the duties is a possible solution as we have just mentioned above. Opening new lanes for checkpoints will also cost money, but money saved by good human resource allocation can be used for throughput enhancement.

We would like to highlight our approach to **influence the passengers' arrival**. TSA has already made recommendations of time-in-advance for the passengers to avoid missing flights. Through our assumptions and modeling, we are curious to find that it is possible to affect the passenger flow by informing passengers of the arrival time ahead of their departure personally and dynamically based on the real-time statistics of the passenger flow at the airport. And the influenced passenger flow may reduce the occurrence of congestions at security checkpoints.

Passengers' satisfaction is considered and quantified. With the help of virtual queueing mechanism, the departure priority queueing discipline can be practiced, which will reduce the variant of the passengers' remaining time (thus raises the fairness) and more importantly, reduce the missed flights of the passengers.

Cultural norm impact on our model is also taken into consideration. Passengers' cultural background may influence their punctuality of arrival and acceptability of our partial priority queueing discipline. Punctuality of a nation has an impact on the passenger flow at an airport, and acceptability of priority queueing should also be taken into account to avoid cultural taboos. This requires adaptation of our models in different countries. Changing the assumed normal distribution for passenger arrival and the acceptability factor α (See 9.2 for definition) will help. For example, for countries where people are not punctual, we can recommend them to be earlier for the flights. Another example is to illustrate the acceptability of our partial priority queueing discipline. In countries where jumping queue is extremely disagreeable, we can set the queueing to be more FIFO. On contrary, in countries where collective efficiency is emphasized, we adjust our factor α to add the priority components in the queueing process.

We hereby sincerely hope the above recommendations will work.

Internal Control Management (ICM) team

11.2 Model Assessment

Model validation has been implemented in each section of the above modeling. Here, we assess the strengths, weaknesses and possible future work in the section.

11.2.1 Strengths

- Full assumptions and validations are made for passengers' arrival distribution and security check service time models. Bottlenecks are spotted and thorough analysis is made.
- Our simulation of one day's passenger flow is based on the real data of O'Hare International Airport, which is convincing.
- Safety is taken as a fundamental principle. All optimization is done based on the principle.
- Recommendations on passenger arrival guidance, checkpoint redesign, change of the queueing discipline, allocation of human resources, virtual queueing are given. Passengers' satisfactory are considered, and the passenger's flight experience will be improved.
- Cultural differences are quantified and fully assessed. Sensitivity to these cultural factors of the passenger flow and acceptability to our priority queueing discipline.

11.2.2 Weaknesses

- TSA Pre-check is not investigated thoroughly.
- Lack of further consideration on safety.

11.3 Conclusions

For problem 1, we use the given data and find the distribution pattern of passenger flow and service time, then queueing models are built for the security check process at each checkpoint. We see the security check process as a $(M/M/c)$ queueing and a $(M/E_k/c)$ process in series. Evaluation of passenger flow behavior in a period of time is made. We use normal distribution to simulate the varying passenger flow.

For problem 2, we put forward several possible approaches to improve passenger throughput and raise the passengers' satisfaction, including checkpoint redesign and virtual queueing where a partial priority queueing discipline is introduced.

Cultural norm impacts are also taken into account. For problem 3, we discussed the cultural impact on passenger flow and acceptability of the priority discipline, and recommendations are given to adapt our model to different cultures.

A detailed guide for policy making is made as our solution of problem 4, again, cultural factors are considered and recommendations concerning cultural difference are made. The recommendations we gave are all proved to be valid in previous discussions.

11.4 Further Work

- Discuss the Pre-check modification. Levels of security check based on the passengers' credit rating, or more detailed background check can be implemented.

- Investigation on how cultural norm differences impact the punctuality and attitude towards the priority queueing need to be done. The correlations between the mathematical parameters and factors in reality should be revealed.
- Detailed research on raising the check-in experience of the disabled and the senior should be made.
- Effect caused by waiting based on Lorenz curve.

12 References

- [1] **Alibaba.com.** Airport security device x-ray baggage scanner. *Alibaba*. [Online] [Cited: 1 22, 2017.]
https://www.alibaba.com/product-detail/Airport-security-device-x-ray-baggage_60381309000.html.
- [2] —. High Quality Product Millimeter Wave Full Body Scanner Price. *Alibaba*. [Online] [Cited: 1 22, 2017.]
https://www.alibaba.com/product-detail/High-Quality-Product-Millimeter-Wave-Full_60569216900.html?spm=a2700.7724838.0.0.uHVhJp.
- [3] **Business Insider.** How Different Cultures Understand Time. *Business Insider*. [Online] [Cited: 1 24, 2017.]
<http://www.businessinsider.com/how-different-cultures-understand-time-2014-5?IR=T>.
- [4] **Chicago Department of Aviation. 2012.** O'Hare Map. *FlyChicago*. [Online] Chicago Department of Aviation, 2012. [Cited: 1 23, 2017.]
<http://www.flychicago.com/pages/landingpage.aspx>.
- [5] **Flight Stats.** Flight Stats Information. *Flight Stats*. [Online] Flight Stats. [Cited: 1 23, 2017.] <http://www.flightstats.com/go/FlightStatus/flightStatusByFlight.do>.
- [6] **Gross, Donald and Harris, Carl M. 1985.** *Fundamentals of Queueing Theory*. s.l. : Wiley, 1985. pp. 170-175. ISBN 0-471-89067-7.
- [7] **Hetter, Katia. 2016.** How long will you wait in the TSA security line? *CNN*. [Online] 6 9, 2016. [Cited: 1 20, 2017.] <http://edition.cnn.com/2016/06/09/travel/tsa-security-line-wait-times-how-long/>.
- [8] **Learn.org.** How Much Do Airport Security Jobs Pay? *Learn.org*. [Online] [Cited: 1 22, 2017.] http://learn.org/articles/How_Much_Do_Airport_Security_Jobs_Pay.html.
- [9] **PayScale.com.** Security Guard Salary (United States). *PayScale*. [Online] [Cited: 1 22, 2017.] http://www.payscale.com/research/US/Job=Security_Guard/Hourly_Rate.
- [10] **Taha, Hamdy A. 2014.** *Operations Research: Pearson New International Edition: An Introduction, 9/E*. s.l. : Pearson, 2014. ISBN-10: 129202433X.
- [11] **Wikipedia.** Queue area. *Wikipedia*. [Online] [Cited: 1 23, 2017.]
https://en.wikipedia.org/wiki/Queue_area.
- [12] **Zeng, Yong, Dong, Lihua and Ma, Jianfeng. 2011.** *Modeling, Analysis and Simulation in Queueing Phenomena*. Xi'an : Xidian University Press, 2011. pp. 92-96. ISBN 978-7-5606-2672-7.
- [13] **Zhao, Zhenwu and Ma, Jianjun. 2016.** Tendency and Research on the Queue System of Airport Security Check. *China Transportation Review*. 10 2016, Vol. 38, 10, pp. 64-69.

13 Appendix

13.1 Proof

Theorem: Mean waiting time of a $(M/M/k)$ queueing is shorter than that of k parallel $(M/M/1)$ queueing processes (Zeng, et al., 2011; Taha, 2014).

Proof: Consider k parallel $(M/M/1)$ processes, input of each server is Poisson λ/k . The mean queue length,

$$L_{q,1} = \frac{\rho_1}{(1 - \rho_1)^2} \cdot P_{0,1}, \text{ where } \rho_1 = \frac{\lambda}{k\mu}$$

$$P_{0,1} = \left(1 + \frac{\rho_1}{1 - \rho_1}\right)^{-1} = 1 - \rho_1$$

Thus,

$$L_{q,1} = \frac{\rho_1^2}{1 - \rho_1}$$

And the mean waiting time (Little's formula),

$$W_{q,1} = \frac{L_{q,1}}{\lambda/k} = \frac{\rho^2}{k\lambda(1 - \rho/k)}, \text{ where } \rho = \frac{\lambda}{\mu}$$

As for the $(M/M/k)$ process, the input is Poisson λ .

$$L_{q,2} = \frac{\rho^{k+1}}{(k-1)!(k-\rho)^2} \cdot P_{0,2}$$

$$P_{0,2} = \left(\sum_{n=0}^{k-1} \frac{\rho^n}{n!} + \frac{\rho^k}{k!} \cdot \frac{1}{1 - \rho/k}\right)^{-1}$$

And thus,

$$L_{q,2} = \frac{\rho^{k+1}}{(k-1)!(k-\rho)^2} \cdot \frac{1}{\sum_{n=0}^{k-1} \frac{\rho^n}{n!} + \frac{\rho^k}{k!} \cdot \frac{1}{1 - \rho/k}}$$

The mean waiting time is given by

$$W_{q,2} = \frac{L_{q,2}}{\lambda} = \frac{1}{\lambda} \cdot \frac{\rho^{k+1}}{k^2(k-1)!(1 - \rho/k)^2} \cdot \frac{1}{\sum_{n=0}^{k-1} \frac{\rho^n}{n!} + \frac{\rho^k}{k!} \cdot \frac{1}{1 - \rho/k}}$$

To prove that $W_{q,2} < W_{q,1}$,

$$\begin{aligned} &\Leftrightarrow \frac{1}{\lambda} \cdot \frac{\rho^{k+1}}{k^2(k-1)!(1 - \rho/k)^2} \cdot \frac{1}{\sum_{n=0}^{k-1} \frac{\rho^n}{n!} + \frac{\rho^k}{k!} \cdot \frac{1}{1 - \rho/k}} < \frac{\rho^2}{k\lambda(1 - \rho/k)} \\ &\Leftrightarrow \frac{\rho^{k-1}}{k! \left(\sum_{n=0}^{k-1} \frac{\rho^n}{n!} \left(1 - \frac{\rho}{k}\right) + \frac{\rho^k}{k!}\right)} < 1 \end{aligned}$$

$$\Leftarrow \frac{\rho^{k-1}}{k! \left(0 + \frac{\rho^k}{k!}\right)} = \frac{1}{\rho} < 1$$

Since $\rho = \lambda/\mu$, generally we have $1 < \rho < k$. It is true that $W_{q,2} < W_{q,1}$ ■

13.2 MATLAB Source Code for Simulation

13.2.1 simulation_simple.m

```

1  clc;
2  clear all;
3  load leave_time_ID;
4  load peo_num_ID
5  % load peo_num;%change comments
6  i=1;
7  x_k = 4;
8  x_mu = 1/28;
9  machine_number = 5;
10 % mean_arr = 12.9870/2.7;%0.077/10 %just influence the people number
11 % final_span = 3600*10;
12
13 % peo_num = floor(span/mean_arr);%change comments
14 peo_num = peo_num_ID;
15
16 % % generate the arrival intervals of people at airport randomly
17 % peo_in = exprnd(mean_arr,1,peo_num);
18 % actual_span = sum(peo_in);
19 % peo_in = peo_in.*(span/actual_span);
20 % peo_arr = cumsum(peo_in);
21
22 % record_arr =
23 [leave_time_ID(1,:),leave_time_ID(2,:),leave_time_ID(3,:),leave_time_ID(4,:),leave_time_ID(5,:)];
24 record_arr = (leave_time_ID(:))';
25 record_arr(find(record_arr == 0)) = [];
26 record_arr = sort(record_arr);
27 peo_arr = record_arr;
28
29 % %generate the service time of people at airport
30 % peo_serv = gamrnd(x_k,1/(x_k*x_mu),1,peo_num);
31
32 %record the state of each security check point

```

```

33 state = zeros(3*machine_number,peo_num);
34
35 %initialize the time and intervals to leave in each array
36 leave_time = zeros(machine_number,peo_num);
37 leave_interval = zeros(machine_number,peo_num);%the time each person in arrays
38
39 %initialize the number of people waited in each array
40 count = ones(1,machine_number);
41
42 %initialize the person's data
43 for i = 1:machine_number
44     state(3*(i-1)+1,1) = peo_arr(i);
45     state(3*(i-1)+2,:) = gamrnd(x_k,1/(x_k*x_mu),1,peo_num);
46     state(3*i,1) = 0;
47
48     leave_time(i,1) = sum(state(3*(i-1)+1 : 3*i , 1));%generate the leaving time of the first
49     person in each array
50     leave_interval(i,1) = sum(state(3*(i-1)+3:3*i,1));%generate the leaving intervals of the first
51     person in each array
52 end
53
54 record_min = zeros(1,machine_number);
55 for i = machine_number+1 : peo_num
56
57     for j = 1:machine_number
58         record_min(j) = leave_time(j,count(j));
59     end
60     [temp,min_array] = min(record_min);
61
62     count(min_array) = count(min_array) + 1;
63     state(3*(min_array-1)+1 , count(min_array) ) = peo_arr(i);%let the people in the min_ array
64
65     if state(3*(min_array-1)+1 , count(min_array) ) < state(3*(min_array-1)+2 , count(min_array)-
66     1 )...
67         +state(3*(min_array-1)+1 ,
68 count(min_array)-1 )...
69         +state(3*(min_array-1)+3 ,
70 count(min_array)-1 )
71
72     state(3*(min_array) , count(min_array) ) = state(3*(min_array-1)+2 , count(min_array)-
73     1 )...

```



```
74                                     +state(3*(min_array-1)+1 ,
75 count(min_array)-1 )...
76                                     +state(3*(min_array-1)+3 ,
77 count(min_array)-1 )...
78                                     -state(3*(min_array-1)+1 ,
79 count(min_array) );
80     else
81         state(3*(min_array) , count(min_array) ) = 0;
82     end
83
84     leave_time(min_array,count(min_array)) = sum(state(3*(min_array-1)+1:3*min_array ,
85 count(min_array) ) );
86     leave_interval(min_array,count(min_array)) = sum(state(3*(min_array-1)+3:3*min_array ,
87 count(min_array) ) );
88 end
89
90 total = 0;
91 for i = 1:size(leave_interval,1)
92     for j = 1:size(leave_interval,2)
93         if leave_interval(i,j) ~= 0
94             total = total + leave_interval(i,j);
95         end
96     end
97 end
98 means = total/peo_num;
```

13.2.2 simulation_tick.m

```
1  clc;
2  clear all;
3  load change2;
4
5  x_k = 4;
6  x_mu = 1/28;
7
8  mean_serv = 10.5652;
9  machine_number = 5;
10
11  peo_arr = change2(2,:).*60;
12  peo_fli = change2(1,:).*60;
13  peo_num = size(change2,2);
```

```
14
15 state = zeros(4*machine_number,peo_num);
16 %the first row is the flight time of each passenger
17 %the second row is the arrival time of each passenger
18 %the third row is the service time of each passenger
19 %the fourth row is the waiting time of each passenger
20
21 leave_time = zeros(machine_number,peo_num);
22 wait_time = zeros(machine_number,peo_num);
23
24 %initialize the state matrix
25 for i = 1:machine_number
26     state(4*(i-1)+1,1) = peo_fli(i);
27     state(4*(i-1)+2,1) = peo_arr(i);
28 %     state(4*(i-1)+3,:) = exprnd(mean_serv,1,peo_num);%service times of each machine in
29 % exponential distribution
30     state(4*(i-1)+3,:) = gamrnd(x_k,1/(x_k*x_mu),1,peo_num);
31     state(4*(i-1)+4,1) = 0;
32
33     leave_time(i,1) = sum(state(4*(i-1)+2 : 4*i ,1));
34     wait_time(i,1) = state(4*i,1);
35 end
36
37 count = ones(1,machine_number);%record the array length of each machine
38
39 queue = zeros(1,peo_num);
40 front = 1;
41 rear = 1;
42
43 scanning = machine_number+1;
44
45 record_min = zeros(1,machine_number);
46 for j = 1:machine_number
47     record_min(j) = leave_time(j,count(j));
48 end
49 [min_time,min_array] = min(record_min);
50
51 % %push the next person into queue
52 % queue(rear) = scanning;
53 % rear = rear + 1;
54 % scanning = scanning+1;
```

```

55
56 for tick = floor(peo_arr(1)):floor(peo_fli(length(peo_fli)))
57     if tick > peo_arr(scanning) && scanning < peo_num%push the latest person into queue
58         queue(rear) = scanning;
59         rear = rear + 1;
60         %         if rear > front
61         %             temp_queue = queue(front:rear - 1);
62         %             [temp_record,temp_order] = sort(peo_fli(temp_queue));
63         %             queue(front:rear-1) = temp_queue(temp_order);
64         %         end
65
66         scanning = scanning + 1;
67     end
68
69     %     if peo_arr(queue(front)) < min_time
70     if tick > min_time
71         if rear > front % the queue is not empty
72             count(min_array) = count(min_array) + 1;
73             state(4*(min_array-1)+1 , count(min_array) ) = peo_fli(queue(front));
74             state(4*(min_array-1)+2 , count(min_array) ) = peo_arr(queue(front));
75             state(4*(min_array-1)+4 , count(min_array) ) = max( min_time - peo_arr(queue(front)) ,
76 0);
77
78             leave_time( min_array,count(min_array) ) = sum(state(4*(min_array-1)+2 :
79 4*(min_array) ,count(min_array)));
80             wait_time( min_array,count(min_array) ) =
81 state(4*(min_array),count(min_array));%update the leave time
82
83             front = front + 1;%pop the prior person to serve
84
85             for j = 1:machine_number
86                 record_min(j) = leave_time(j,count(j));
87             end
88             [min_time,min_array] = min(record_min);%update the min_time and the min_array
89
90         end
91
92     end
93
94 end
95

```

```
96 total = 0;
97 flag = 0;
98 for i = 1:machine_number
99     for j = 1:count(i)
100         total = total + state(4*(i-1)+1,j)-state(4*(i-1)+2,j)-state(4*(i-1)+3,j)-state(4*i,j);
101         if state(4*(i-1)+1,j)-state(4*(i-1)+2,j)-state(4*(i-1)+3,j)-state(4*i,j) < 0
102             flag = flag + 1;
103         end
104     end
105 end
106 means = total / peo_num;
107
108 summation = 0;
109 for i = 1:machine_number
110     for j = 1:count(i)
111         summation = summation + (state(4*(i-1)+1,j)-state(4*(i-1)+2,j)-state(4*(i-1)+3,j)-
112 state(4*i,j)-means)^2;
113     end
114 end
115 variance = summation / peo_num;
```

13.2.3 stream.m

```
1 function [ total ] = stream( pretime, variance,people)
2 %UNTITLED
3 % original:pretime=120,variance=30,people=189
4 a=[-3*variance:1:3*variance];
5 a=[a;a];
6 a(2,1)=normcdf(a(1,1),0,variance)*people;
7 for i=2:6*variance+1
8     a(2,i)=normcdf(a(1,i),0,variance)-normcdf(a(1,i-1),0,variance);
9     a(2,i)=a(2,i)*people;
10 end
11 data=xlsread('data',2);
12 siz=size(data);
13 dataset=[];
14 data=data*24*60;
15 for i=1:siz(1,1)
16     for j=1:siz(1,2)
17         if data(i,j)>0
18             dataset=[dataset,data(i,j)];
```

```
19     end
20     end
21 end
22 siz=size(dataset);
23 total=zeros(1,1440);
24 for i=1:siz(1,2)
25     for j=1:variance*6+1
26         if dataset(1,i)-pretime+a(1,j)>=1
27             total(1,round(dataset(1,i)-pretime+a(1,j)))=total(1,round(dataset(1,i)-
28 pretime+a(1,j)))+a(2,j);
29         end
30     end
31 end
32 end
```

13.2.4 stream2.m

```
1 function [ total ] = stream( pretime, variance,people)
2 %UNTITLED
3 % original:pretime=120,variance=30,people=189
4 a=[-3*variance:1:3*variance];
5 a=[a;a];
6 a(2,1)=normcdf(a(1,1),0,variance)*people;
7 for i=2:6*variance+1
8     a(2,i)=normcdf(a(1,i),0,variance)-normcdf(a(1,i-1),0,variance);
9     a(2,i)=a(2,i)*people;
10 end
11 data=xlsread('data',2);
12 siz=size(data);
13 dataset=[];
14 data=data*24*60;
15 for i=1:siz(1,1)
16     for j=1:siz(1,2)
17         if data(i,j)>0
18             dataset=[dataset,data(i,j)];
19         end
20     end
21 end
22 siz=size(dataset);
23 total=zeros(1,1440);
24 for i=1:siz(1,2)
```

```
25     for j=1:variance*6+1
26         if dataset(1,i)-pretime+a(1,j)>=1
27             total(1,round(dataset(1,i)-pretime+a(1,j)))=total(1,round(dataset(1,i)-
28 pretime+a(1,j)))+a(2,j);
29         end
30     end
31 end
32 end
```