For office use only

T1 ________________

T2 ________________

T3 ________________

T4 ________________

Team Control Number

**65328**

Problem Chosen

**D**

For office use only

F1 ________________

F2 ________________

F3 ________________

F4 ________________

# 2017
## MCM/ICM
## Summary Sheet


In these years, the U.S. Transportation Security Agency (TSA) spares no effort to improve airport security checkpoint equipment and procedures. However, with such high throughput, it is not a rare situation when passengers have to wait in a long queue before they could have their baggage and themselves checked, which has drawn sharp social criticism to airports recently. Thus, the increasing demand of maximizing both public security and individual efficiency has been addressed to decision makers. Our goal is to asses current process for U.S. airport security checkpoint and propose solutions to improve it, or in other words, match the throughput to the demand.

We realized that the key element to solve this problem is to analyze the throughput throughly and respectively as each workstation has its own throughput. We must find the bottleneck of a queue and the variance in wait time.

First, we preprocess data to calculate mean and variance of each period of time to extract information from given data set and fit distribution to lay foundation for subsequent model assumption. Based on the data distribution and the process characters, we made the assumption that the distribution of time is subject to a Gaussian distribution.

To explore the flow of passengers through each security checkpoint, we calculate several metrics by applying Pollaczek-Khinchine formula and Little's formula to the given data set based on the assumption that each process is an M/G/k queue model, which helped us to gain a better understanding on the problem

Considering that some time data are not provided, we also construct a Monte Carlo model to simulate passengers' time distribution on each process. Then, we write a python program to retrieve simulation result on various occasions. Also, we make some extensions to our model, adding extra modifications to figure out the influence of some special properties of passengers.

Finally we reach a conclusion that bottleneck may lie in the ID check process and the millimeter scanning process. And, increasing staffing of screening is only efficient when the screening efficiency is relatively small. To reduce variance of waiting time, it is recommendable to use special lanes for slow passengers.

Based on our work, we would like to propose several recommendations to decision makers. We hope to match the throughput to the demand wisely, instead of directly increasing budget to purchase more machines or hire more people. Thus, we proposed two classes of recommendations, from macroscopic and microscopic angle respectively. To increase the bottleneck capacity long-termly, we suggest that TSA can introduce more advanced technology machines (facial recognition, QR code scanning, etc.) in ID Check process, and enhance the cooperation with international airlines. To short-termly increase the throughput, TSA could hire more part-time staffs when the passenger flow is intensive. Our simulations show that tailor people's (especially elder, infant, pregnant and disabled groups) procedure could reduce the queue waiting time variance, which means that tailoring could be a win-win solution. We suggest TSA to tailor the procedure by classifying people's background and habit, and more importantly, let travelers make their own choice.

To sum up, in our models, we apply various interdisciplinary knowledge (scientific computing, queue theory, computer simulation, operation management). The results of our two models is verified, which increases their credibility. Meanwhile, more quantitative calculation of the temporal and spatial changes of the intensity can be further improved, while some more extensions may be added to the simulation model.

# Analysis of Airport Security from Multiple Perspectives

## ICM Contest Question D

Team # 65328

January 24, 2017

## Contents

# 1 Introduction

## 1.1 The Problem

In order to protect its citizens from terrorist attacks when traveling through airlines, the U.S. Transportation Security Agency (TSA) has imposed strict rules on security inspection systems after September, 2001. It has been reported that in 2016, TSA officers screened 738,318,264 passengers in total (more than 2 million per day), which is 43,255,172 more passengers than in 2015. With such high throughput, it is not a rare situation when passengers have to wait in a long queue before they could have their baggage and themselves checked, which has drawn sharp social criticism to airports recently. Thus, the increasing demand of maximizing both public security and individual efficiency has been addressed to decision makers.

At the first glance, the mere measure of implementing increased machines and staff members seems applicable. However, given limited resources, socioeconomic efficacy has to be taken into consideration. Optimizations of the security inspection systems itself for more favorable performances may well be a reasonable alternative solution to the problem.In an attempt to increase overall performance of airports, we would like to introduce our models and offer several practical suggestions based on our results, and hopefully, to help better the current situations.

## 1.2 Our Work

First, we preprocess data to calculate mean and variance of each period of time to extract information from given data set and fit distribution to lay foundation for subsequent model assumption.

Second, we introduce M/G/k model to calculate performance metrics by applying Pollaczek-Khintchine (P-K) Formula and Little's Law. Such metrics are used to evaluate each period of time and identify bottlenecks.

Third, we build up a Monte Carlo model to simulate the whole airport security process. Results are compared to metrics calculated in M/G/k model, which further determine that bottleneck lies in Zone A. We also develop several extensions to add realistic factor into our model. Finally, we reach a conclusion and propose some suggestions based on our results and actual situation according to operational management theory.

# 2 Method

## 2.1 Data Analysis and Distribution Assumption

Due to limited size of sample data and restricted further access, we have practiced an alternative approach instead of sole analysis of the sample itself. With the data provided, we were able to find some clues on data distributions, on which preliminary model would be based. Using Monte-Carlo method, we made it possible to generate thousands of simulations with the parameters in our model, which would be subject to further analysis. By repetitive comparison of simulations and the sample data and modifications of our parameters, the model would be refined iteratively and new patterns might be discovered, with which we could make attempts to solve the problems.

### 2.1.1 Data Pre-processing

From the provided 2017_ICM_Problem_D_Data 2.xlsx we noticed that the Column A, B, E, F, G were recorded as time stamps while the C, D, H were time durations. When computing, we define the time unit as *second*(s). In the previous EXCEL file the time unit was recorded as *day*(d). Therefore, when extracting data from .xlsx to MATLAB, each vector elements should be multiplied by a constant 86,400, with an exception of Column H, where the constant should be 1,440.

### 2.1.2 Column A and B: ARRIVAL -> ID CHECK

According to queuing theory, it would be reasonable to make a basic assumption that the arrivals of passengers, either TSA Pre-check or Regular, could be regarded as Poisson Processes, which implies independent occurrences and constant average rates of arrivals.

Poisson Process is a type of random mathematical object that consists of points randomly located on a mathematical space. In a Homogeneous Poisson point process, the parameter, $\lambda$, is called the **intensity**, and related to the average number of Poisson points existing in some bounded region. The homogeneous Poisson point process can be denoted as $\{N(t), t \geqslant 0\}$, and having those following properties:

- $N(0) = 0$;
- has independent increments;

- the number of events (or points) in any interval of length $t$ is a Poisson random variable with parameter $\lambda_t$;
- the the probability of the random variable $N(t)$ being equal to n is given by:

$$P\{N(t) = n\} = \frac{(\lambda t)^n}{n!}\exp(-\lambda t).$$

In a short duration, as in the sample provided, the parameter $\lambda$ could be regarded as a constant. However, if we observe a large airport in a remarkable time scale (such as a day or even a week), we may notice that the intensity of passenger flow is no longer with constant, but subject to changes with time. Therefore the parameter $\lambda$ should be time-dependent and hence denoted as $\lambda(t)$.

With Column A and B in the data file provided, we calculated some metrics (parameters) listed as follows.

| METRIC | SYMBOL | VALUE |
|---|---|---|
| Average interval between TSA Pre-check arrivals | $1/\lambda_{tsa}$ | 9.1895(s) |
| Average interval between Regular arrivals | $1/\lambda_{reg}$ | 13.0302(s) |
| TSA Pre-check arrival intensity | $\lambda_{tsa}$ | 0.1088($s^{-1}$) |
| Regular line arrival intensity | $\lambda_{reg}$ | 0.0767($s^{-1}$) |
| Total intensity of 4 lines | $\lambda_{tot}$ | 0.3390($s^{-1}$) |
| Average intensity of 4 lines | $\lambda_{ave}$ | 0.0847($s^{-1}$) |

In a Poisson Process, time difference between consecutive events $\tau$ should be a random variable subject to an **Exponential Distribution** with its parameter $\lambda$ same as the intensity of the Process. This property gave us a hint to verify whether the arrivals are indeed a Poisson Process by means of testing whether the consecutive intervals satisfy exponential distribution, and, if so, to make preliminary estimations of the parameter $\lambda$, the intensity of the Poisson Process, since $\lambda = 1/\mu$, where $\mu$ is the average interval.

A Monte Carlo Method was applied. We generated randomized arrays using the `exprnd()` function in MATLAB with parameter $\mu = 1/\lambda$ in the function.Subsequently, we made comparisons between the original data and simulations (Figure 2.1) and find a high resemblance in distribution, indicating that the Poisson assumption was plausible. Please see the appendix for the chart.

### 2.1.3 Column C and D: ID CHECK PROCESS

We assumed that the ID Check process is subjected to a **Gaussian Distribution**. A normality test was applied with MATLAB function `lillietest()` and the result was a "fail to reject".This result should honestly not be interpreted as a "confirmed" Gaussian Distribution, however, it provided some evidence that the sample data is *not contradictory* to our assumption. Thus we would use sample mean as the preliminary (unbiased) estimation of parameter $\mu$ in the Gaussian Distribution, and sample variance as $\sigma^2$. The calculation results are listed as follows:

| METRIC | SYMBOL | VALUE |
|---|---|---|
| Average time of ID check process 1 (Column C) | $\mu_{cp1}$ | 10.1967 |
| Average time of ID check process 2 (Column D) | $\mu_{cp2}$ | 12.5514 |
| Variance of time of ID check process 1 (Column C) | $\sigma_{cp1}$ | 2.9814 |
| Variance of time of ID check process 2 (Column D) | $\sigma_{cp2}$ | 4.5448 |

### 2.1.4 Column E, F and G: BODY SCREENING & Column H: ITEMS ON THE BELT

In this part, it's difficult to find a common distribution. We have obtained a MATLAB function `allfitdist()` to fit all valid parametric probability distributions to our data. Apparently, the fitting result cannot meet our satisfaction. On one hand, the low probability density when value = 10 was neglected in the estimation; On the other hand, the parameters of suggested distributions are relatively difficult to be estimated. We conjectured that the distribution is an superposition of several distributions, or some special factors is influencing several specific interval values. As the sample size is limited, the Considering the difficulty of fitting one or more distribution properly, we would like to use a concise way to process these data. In the simulation we would **assume** that these procedures have **Gaussian Distribution**. We hold the opinion that the time for a body screening and items check procedure are constant, the random error should be subject to a Gaussian Distribution. Fitting the Gaussian Distribution to these data would surely simplify our work and the bias is acceptable. As there are merely three intervals in Column G, we would not use this column in our further modeling and simulations.

| metric | Symbol | Value |
|---|---|---|
| Average time of Millimeter Scanner check process (Column E) | $\mu_{mc}$ | 11.6372(s) |
| Average time of X-ray Scanner process 1 (Column F) | $\mu_{xc1}$ | 7.5420(s) |
| Average time of X-ray Scanner process 2 (Column G) | $\mu_{xc2}$ | 3.6700(s) |
| Average time of items on the belt interval (Column H) | $\mu_{ib}$ | 28.6207(s) |
| Standard derivation of Millimeter Scanner check process (Column E) | $\sigma_{mc}$ | 5.8688(s) |
| Standard derivation of X-ray Scanner process 1 (Column F) | $\sigma_{xc1}$ | 3.2417(s) |
| Standard derivation Scanner process 2 (Column G) | $\sigma_{xc2}$ | 3.3661(s) |
| Standard derivation on the belt interval (Column H) | $\sigma_{ib}$ | 14.0901(s) |

## 2.2 M/G/$k$ Model

We would like to introduce a M/G/$k$ model in the queueing theory to this problem.

### 2.2.1 Introduction to M/G/$k$ model

An M/G/$k$ queue consists of $k$ single servers and queue with Poisson passengers arrivals, where the service time of a job has a general distribution. In the case of airport security checkpoint, the service time is not a negative exponential distribution, thus we cannot simply apply M/M/k model. Suppose the arrival rate is $\lambda$, the mean and the variance of service time is

$1/\mu$ and $\sigma^2$. When

$$\rho = \frac{\lambda}{\mu} < 1,$$

the system will reach a long-term steady state. We will use following traditional performance metrics:

| METRIC | SYMBOL |
|---|---|
| Mean number in system | $L$ |
| Mean number in queue | $L_q$ |
| Mean response time | $W$ |
| Mean waiting time | $W_q$ |

Please note that the metrics $L$, $L_q$,$W$, $W_q$only depend on $\rho$ and $\sigma^2$ and have no relation to the type of distribution.

#### 2.2.2 Pollaczek-Khintchine Method

We apply two Formulas to our modeling theory. Based on the two following formulas,

- Pollaczek-Khintchine (P-K) Formula:

$$L = \rho + \frac{\rho^2 + \lambda^2\sigma^2}{2(1-\rho)}$$

- Little's Law:

$$L = \lambda W$$

#### 2.2.3 Matrix Analytic Method

Another method worth introduced is the Matrix Analytic Method.

A Markov transition matrix is a square matrix describing the probabilities of moving from one state to another in a dynamic system. In each row are the probabilities of moving from the state represented by that row, to the other states. Thus the rows of a Markov transition matrix each add to one. An M/G/1-type stochastic matrix is one of the form

$$P = \begin{bmatrix} b_0 & b_1 & b_2 & b_3 & \cdots \\ a_0 & a_1 & a_2 & a_3 & \cdots \\ 0 & a_0 & a_1 & a_2 & \cdots \\ 0 & 0 & a_0 & a_1 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

where $b_i$ and $a_i$ are $k \times k$ matrices. Such a matrix describes the embedded Markov chain in an M/G/1 queue.

#### 2.2.4 Assumption and Modeling

Obviously, the first queue system (consisting the arrival before id-check and id-check process) is an M/G/1 queue. For the following queues, we would like to apply **Gaussian Distribution**. Since passengers would not skip the processes, we can simplify the matrix analytical method by dividing it into several steps, each step process time can be fitted into a Gaussian Distribution. To simplify our calculation and obtain a concise result, the process of the items to be scanned has been neglected. In other words, we assume that the body scanning process takes longer than the items scanning process. We also assume that passengers don't need to wait in line when picking up items. We assume that each scanning system consists a millimeter wave scanner, an x-ray scanner and an items scanner. According to the M/G/k theory, the service time in three stages are: ID check process, Items placing process and body scanning process. Thus, there three waiting queues to be calculated: the line in front of ID check counter, the belt, and the body scanner. We also assume that people will randomly choose a line at the first two stages. So the intensity $\lambda$ is dependent on the number of counters/lines distribution. When it comes to the scanner process, we assume that there is a probability $P$ influencing a person's choice. This probability can be affected by the personal character of the person (culture background, health condition, etc.)

### 2.3 Monte Carlo Model

The aforementioned M/G/$k$ model is a simplified model based on excessive assumptions. Therefore, we build up a model using Monte Carlo method to simulate the whole airport security process. Monte Carlo method is a class of computational algorithm that rely on repeated random sampling to obtain numerical results. The probabilistic interpretation of our problem makes it natural to apply this method.

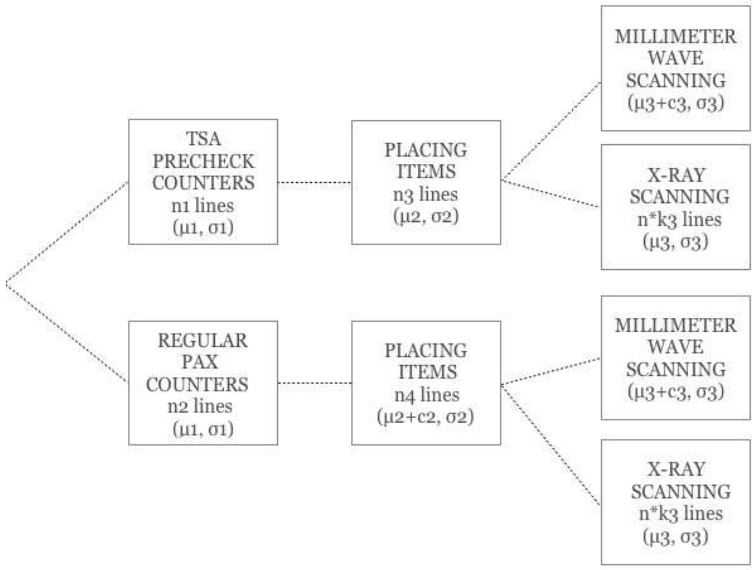


Figure 1: The Flow Chart of M/G/$k$ Model

#### 2.3.1 Assumptions and Parameters

The assumptions of our model on the security check procedure on which subsequent simulations would be based are as follows.

Passengers must go through a series of four procedures listed below before they could finish the whole security inspection process.

1. In Zone A, passengers arrive in a random manner. Upon arrival, TSA pre-check passengers wait in one queue and regular pax in the other three queues until ID check.

2. In Zone B, both TSA Pre-check and regular pax passengers wait in queues until they reach the lane and prepare all their belongings for X-ray screening. Since items regular pax passengers are required to remove should typically be more than their privileged counterparts, it is predictable that they would basically spend more time on belongings preparation before putting items on conveyor belts.

3. Afterwards, they wait to process through either a millimeter wave scanner or metal detector, while their belongings are transported by conveyor belt through an X-ray machine. The **maximum makespan**

of baggage and body screening is to be regarded as the total scanning time in Zone B.

4. From Zone C, passengers collect all their belongings to depart, providing that they need no further check in Zone D. We assert that the collecting time equals the previous preparation time in Zone B.

The assumptions regarding distributions and parameters of the random variables are listed below:

- Passenger arrivals could be regarded as a Poisson flow with a intensity of $\lambda_{tot}$.
- Time consumption of ID check procedure is subject to a Gaussian distribution with mean $\mu_{cp1}$ and variance $\sigma^2_{cp1}$.
- Preparation times of TSA Pre-check follows a Gaussian distribution with mean $5.0s$ and standard deviation $1.0s$;
- Preparation times of Regular Pax follows a Gaussian distribution with mean $10.0s$ and standard deviation $1.0s$;
- Time consumption of millimeter wave scan is regarded as a Gaussian distribution with mean $\mu_{mc}$ and variance $\sigma^2_{mc}$;
- Time consumption of millimeter wave scan is regarded as a Gaussian distribution with mean $\mu_{xc1}$ and variance $\sigma^2_{xc1}$;

Meanwhile, we define several parameters for our model:

- The probability of choosing the millimeter $P$;
- The number of TSA Pre-check screening lane $n_1$;
- The number of regular screening lane $n_2$.

### 2.3.2 Extensions

In addition, following special occasions are considered.

- Some passengers may spend more times in the process due to age or health problems. We can measure the influence of those people in our model;

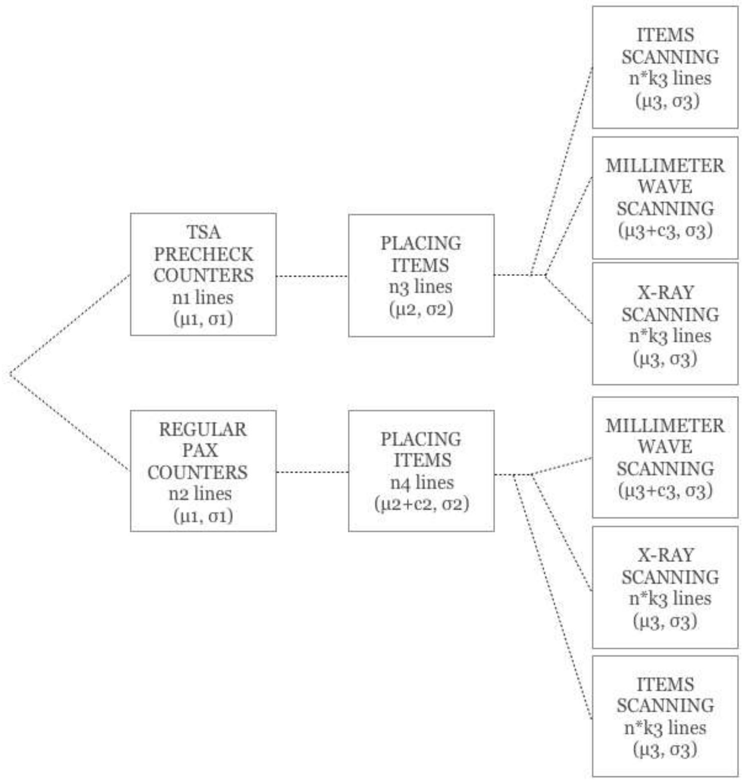


Figure 2: The Flow Chart of Monte Carlo Model

- Different people may take all kinds of baggage. The amount of baggage may affect the time related to baggage.
- Since more people will take TSA Pre-check rather than taking regular check, a time-sensitive man may go to regular check queues though he is enrolled in the Pre-check program.

# 3 Result

## 3.1 M/G/$k$ Model

### 3.1.1 Analytical Solution

The analytical solution of the M/G/$k$ model is derived from our illustrated parameters. The illustration is shown in the appendix.

Remarks: in the foot notation $\rho_{11}$, the first number means the first stage, the second number mean a single randomized line of the first category.

From Pollaczek-Khintchine (P-K) Formula, we can obtain the following results (for example, $L_q$)

$$Lq_{11} = -\frac{\dfrac{\lambda_0^2\mu_1^2}{n_1^2} + \dfrac{\lambda_0^2\sigma_1^2}{n_1^2}}{\dfrac{2\lambda_0\mu_1}{n_1} - 2}$$

$$Lq_{12} = -\frac{\dfrac{\lambda_0^2\mu_1^2}{n_2^2} + \dfrac{\lambda_0^2 + \sigma_1^2}{n_2^2}}{\dfrac{2\lambda_0\mu_1}{n_2} - 2}$$

$$Lq_{21} = -\frac{\dfrac{\lambda_0^2(c_2+\mu_2)^2}{n_1^2n_3^2} + \dfrac{\lambda_0^2\sigma_2^2}{n_1^2n_3^2}}{\dfrac{2\lambda_0(c_2+\mu_2)}{n_1n_3} - 2}$$

$$Lq_{22} = -\frac{\dfrac{\lambda_0^2\mu_2^2}{n_2^2n_4^2} + \dfrac{\lambda_0^2\sigma_2^2}{n_2^2n_4^2}}{\dfrac{2\lambda_0\mu_2}{n_2n_4} - 2}$$

$$Lq_{311} = -\frac{\frac{\lambda_0^2 p^2 (c_3+\mu_3)^2}{n_1^2 n_3^2} + \frac{\lambda_0^2 p^2 \sigma_3^2}{n_1^2 n_3^2}}{\frac{2\lambda_0 p (c_3+\mu_3)}{n_1 n_3} - 2}$$

$$Lq_{312} = -\frac{\frac{\lambda_0^2 (p-1)^2 \mu_3^2}{n_1^2 n_3^2} + \frac{\lambda_0^2 (p-1)^2 \sigma_3^2}{n_1^2 n_3^2}}{\frac{2\lambda_0 (p-1) \mu_3}{n_1 n_3} - 2}$$

$$Lq_{321} = -\frac{\frac{\lambda_0^2 p^2 (c_3+\mu_3)^2}{n_2^2 n_4^2} + \frac{\lambda_0^2 p^2 \sigma_3^2}{n_2^2 n_4^2}}{\frac{2\lambda_0 p (c_3+\mu_3)}{n_2 n_4} - 2}$$

$$Lq_{312} = -\frac{\frac{\lambda_0^2 (p-1)^2 \mu_3^2}{n_2^2 n_4^2} + \frac{\lambda_0^2 (p-1)^2 \sigma_3^2}{n_2^2 n_4^2}}{\frac{2\lambda_0 (p-1) \mu_3}{n_2 n_4} - 2}$$

We can then calculate the waiting time from Little's law.

### 3.1.2 Model Testing

We would like to give a trivial examination of the initial M/G/1 system. We assume that a group of people was passing through the id-check queue. The group's $E(X)$ and $Var$ is subject to an unbiased estimation. A trivial examine is using the calculated $\lambda$, $\sigma$ and $\mu$ in the Primary Data Analysis. As the sample size in Column D was limited, we decided to use the Column C data. In this process we assume that the time duration for an TSA officer to check the TSA Pre-check passengers and regular passengers are the same. For the Regular queue in the existing data sample, we obtained that

$$\rho = 0.7825 < 1$$

, which means that the system is in a long-term steady state.

| Metrics | Value |
|---|---|
| $L^{(reg)}$ | 2.3109(persons) |

| Metrics | Value |
|---|---|
| $L_q^{(reg)}$ | 1.5284(persons) |
| $W^{(reg)}$ | 30.1118(s) |
| $W_q^{(reg)}$ | 19.9151(s) |

For the TSA Precheck queue, the $\rho$ was larger than 1, so that the further calculation on $L$ and $W$ would be meaningless.

## 3.2 Monte Carlo Model

We use python to run a Monte Carlo simulation of $100,000$ passengers.

### 3.2.1 ID Check Process

First we examine waiting time of the ID check process, the result is shown as follow:

| Check Type | Mean(s) | Standard Deviation(s) |
|---|---|---|
| Pre-check | $> 1000$ | $> 1000$ |
| Regular | 20.05 | 23.59 |

The result of mean waiting time is similar to the result $W_q^{(reg)} = 19.9151$ of the M/G/$k$ model, showing that our models is consistent and believable.

### 3.2.2 Screening Process

Since one can choose either millimeter wave or X-ray scanning, thus we set $P = 0.5$. For different $n_1$ and $n_2$, we examine the waiting time in preparation and screening phase. The result is shown as follow:

For TSA Pre-check, we have

| | Preparation | | Millimeter | | X-ray | |
|---|---|---|---|---|---|---|
| $n_1$ | Mean(s) | SD(s) | Mean(s) | SD(s) | Mean(s) | SD(s) |
| 1 | 0.07 | 0.40 | 4.12 | 7.12 | 0.53 | 1.61 |
| 2 | 0.03 | 0.27 | 1.37 | 3.76 | 0.21 | 0.97 |
| 3 | 0.02 | 0.22 | 0.84 | 2.84 | 0.15 | 0.86 |

For regular pax, we have

| $n_2$ | Preparation Mean(s) | SD(s) | Millimeter Mean(s) | SD(s) | X-ray Mean(s) | SD(s) |
|---|---|---|---|---|---|---|
| 3 | 13.49 | 16.04 | 2.69 | 5.39 | 0.22 | 0.91 |
| 4 | 5.28 | 7.78 | 1.70 | 4.08 | 0.15 | 0.70 |
| 5 | 3.34 | 5.62 | 1.41 | 3.77 | 0.13 | 0.67 |
| 6 | 2.38 | 4.44 | 1.04 | 3.01 | 0.09 | 0.57 |

#### 3.2.3 Extension

Suppose that there are 5 percent of passengers which cost twice more time. Then, when $n_2 = 5$, we runs the simulation:

| Preparation Mean(s) | SD(s) | Millimeter Mean(s) | SD(s) | X-ray Mean(s) | SD(s) |
|---|---|---|---|---|---|
| 5.10 | 8.48 | 2.75 | 7.59 | 0.55 | 2.69 |

It is shown that, slow passengers will largely increase the variance of waiting time, thus, it is better to use a special lane for those passengers.

When all passengers are greedy, the arrivals will be equally distributed among the ID check officers. The results of the ID check are:

| Check Type | Mean(s) | Standard Deviation(s) |
|---|---|---|
| Pre-check | 109.94 | 103.16 |
| Regular | 109.74 | 101.90 |

We can find that, the waiting time of TSA Pre-check is reduced while that of regular check is increased. The greedy behavior will reduce the overall variance of the waiting time of ID check process.

# 4 Conclusion

## 4.1 Concepts from Operation Management

To get problem solved and offer better suggestions, we would like to introduce several concepts of Operation Management.

Capacity is the throughput, or number of units a facility can hold, receive, store or produce in a period of time. The Capacity planning can be viewed in three time horizons: Long-range ($>$ 1 year)Adding facilities and long lead-time equipment Intermediate-range (318 months)Aggregate planning tasks, including adding equipment, personnel, and shifts; subcontracting; and building or using inventory Short-range ($<$ 3 months)Scheduling jobs and people, and allocating machinery

The bottleneck is the limiting factor or constraint in a problem. The bottle neck time is the longest process. In the bottleneck theory, we obtained four principles:

- Release work orders to the system at the pace set by the bottlenecks capacity
- Lost time at the bottleneck represents lost capacity for the whole system
- Increasing the capacity of a non-bottleneck station is a mirage
- Increasing the capacity of the bottleneck increases capacity for the whole system

We fully understand that it is such a waste that if we put too much cost directly. Our goal is to match the capacity to the demand, which means, to use the budget wisely. The concepts above implies that, the key of capacity analysis is the role bottleneck. We will firstly analyze the bottleneck.

## 4.2 Bottleneck Analysis

Due to the lack of consideration of the scanning time of the baggage in the M/G/$k$ model, we use the data from the simulation to analyze the bottle neck. From the result, we find that the ID check process has the most waiting time consumption. Thus, the ID check process can be regarded as the bottleneck.

In the screening process, the millimeter cost the most time and have most variance. Increasing the number of lanes to $n_1 = 2$ and $n_2 = 5$ will well reduce the mean time and variance, while further increasing show no obvious improvement on perfomance.

From the result of the extension, we know that it is recommended to perform special treatment to special people, which will largely reduce the variance.

## 4.3 Suggestions

We would like to give the following suggestions:

#### 4.3.1 Globally Applicable Recommendations

- Long range strategy To increase the capacity of Bottlenecks

  Advanced technology in ID Check Process. We have learnt that the ID Check Process could be the bottleneck. With the rapid development of information science and technology, we suggest using more advanced technology such as facial recognition, QR code scanning.

  Enhance the cooperation with international airlines. TSA Pre-check is now cooperating with 19 airlines and reaching nationwide. In our model, cooperating with more airlines can separate the queue system and reducing the regular pax intensity(lambda).

- Short range strategy

  Adjusting the employment by analyzing throughput change - release work orders to the system at the pace set by the bottlenecks capacity. From our simulation, the average time for a person go through every process is about 3 mins. However, from MyTSA app, we found that (data collected on Jan. 22nd )

  At Boston Logan International Airport, 7 days ago on 7:12 A.M. EST, the average security check duration is about 21- 30 mins

  At Chicago OHare Airport, on 5:10 A.M. EST, the average security check duration is about 46- 60 mins At Los Angeles airport, on 11: 39 A.M. EST, the average security check duration is about 30 - 45 mins

  We recommend that Big airport could increase the capacity bottleneck by hiring more half-time staffs when the intensity is high. The intensity could be inferred/calculated from the flight distribution.

#### 4.3.2 Tailored Recommendations

Let traveler choose. By giving TSA Pre check members more choices, TSA will not only won passengers heart, but also reducing the variance of each line (See our simulated results).

- Customize services for elder persons, infants, pregnant ladies, disabled persons and family traveling. We have learnt that if a patient cannot go screening due to some health reasons, he or she may provide the officer with the TSA notification card or other medical documentation to describe his condition. We think this kind of TSA notification card can be promoted to elder persons, infants, pregnant ladies, disabled persons and families. Travelers with TSA notification card could ask for an extra/independent line without waiting in the queue.

- Set different TSA Pre-check price levels for students and soldiers.
- Investigate the culture influence. Send social investigations request to different people from different background via e-mail. In a multicultural country like U.S., we are sure that TSA can find an appropriate balance for making every passenger comfortable with Security Check.

# 5 Strengths and Weakness

## 5.1 Strength

- Various means and interdisciplinary knowledge are applied to analyze the problem, including scientific computing, queueing theory, computer simulation and operational management.
- Two models are developed in our paper, one theoretical and one simulative. Their result are verified that they produce similar output, which largely increase the credibility of our models.
- Large amount of data about airports in US is extracted, which makes the models well fit the reality of US airports.

## 5.2 Weakness

- Quantitative calculation of the temporal change of the arrival intensity needs to be further explored.
- The spatial difference of the intensity and the difference of domestic and international flight should also be considered in the calculation.
- The simulation model can be extended to characterize the flow control in the system.

# 6 Future Works

We hope to further examine the temporal fluctuation of arrival intensity by more mathematical techniques, such as partial differential equations.

# 7 Reference

Slack N, Chambers S, Johnston R. Operations management[M]. Pearson education, 2010.
Render B, Heizer J H. Principles of operations management: building and managing world-class operations[M]. Allyn and Bacon, 1994.
Harchol-Balter M. Performance modeling and design of computer systems: queueing theory in action[M]. Cambridge University Press, 2013.
Tijms H C. A first course in stochastic models[M]. John Wiley and sons, 2003.
Kingman J F C. Poisson processes[M]. John Wiley & Sons, Ltd, 1993.
Bolch G, Greiner S, de Meer H, et al. Queueing networks and Markov chains: modeling and performance evaluation with computer science applications[M]. John Wiley & Sons, 2006.
http://economics.about.com/od/termsbeginningwithm/g/markov_transition_matrix.htm
http://www.mathworks.com/matlabcentral/fileexchange/34943-fit-all-valid-parametric-probability-distributions-to-data
DHS/TSA/PIA-032(d) Advanced Imaging Technology, December 18, 2015
http://www.oregon.gov/OBMI/docs/TSA-AIT_ScannerFactSheet.pdf
https://www.tsa.gov
http://apps.tsa.dhs.gov/mytsa/index.aspx

# 8 Appendices

## 8.1 Data Analysis

The charts of data analysis is shown below:

## 8.2 Parameter Illustration of the Analytic Methods

| Symbol | metric | notation meaning |
|---|---|---|
| $\lambda_0$ | the coming intensity | |
| $n_1$ | the number of counters of TSA Precheck ID check | |

| Symbol | metric | notation meaning |
|---|---|---|
| $n_2$ | the number of counters of Regular Pax ID check | |
| $n_3$ | the number of scanning systems for TSA Precheck members | |
| $n_4$ | the number of scanning systems for Regular Pax members | |
| $\rho_{11}$ | parameter defining the line steady degree | one line before TSA Precheck ID check counter |
| $\rho_{12}$ | parameter defining the line steady degree | one line before Regular Pax ID check counter |
| $\rho_{21}$ | parameter defining the line steady degree | one line before scanning systems for TSA Precheck members |
| $\rho_{22}$ | parameter defining the line steady degree | one line before scanning systems for Regular Pax members |
| $\rho_{311}$ | parameter defining the line steady degree | one line before millimeter scanner for TSA Precheck members |

| Symbol | metric | notation meaning |
|---|---|---|
| $\rho_{312}$ | parameter defining the line steady degree | one line before x-ray scanner for TSA Precheck members |
| $\rho_{321}$ | parameter defining the line steady degree | one line before millimeter scanner for Regular Pax members |
| $\rho_{322}$ | parameter defining the line steady degree | one line before x-ray scanner for Regular Pax members |
| $\lambda_n$ | the coming intensity | infront of line, same as $\rho$ |
| $L_{qn}$ | the queue line length | notation same as $\rho$ notation |
| $L_n$ | the system line length | same as $\rho$ notation |
| $M_{qn}$ | the queue waiting time | same as $\rho$ notation |
| $M_n$ | the full processing time | same as $\rho$ notation |

| Symbol | metric | notation meaning |
|---|---|---|
| $\mu_n$ | the average processing time | n means stage number |
| $\sigma_n$ | the standard derivation processing time | n means stage number |
| $c_2$ | the extra time constant of regular pax passenger spend | on the second stage, dropping shoes.etc |
| $c_3$ | the extra time constant of millimeter checked passengers spend | on the third stage, wait for the millimeter machine |

### 8.3 Source Code of the Monte Carlo Method

```
import random
import numpy

#s = 10.0

special_speed = 2.0 #time multiplier for special passengers
```

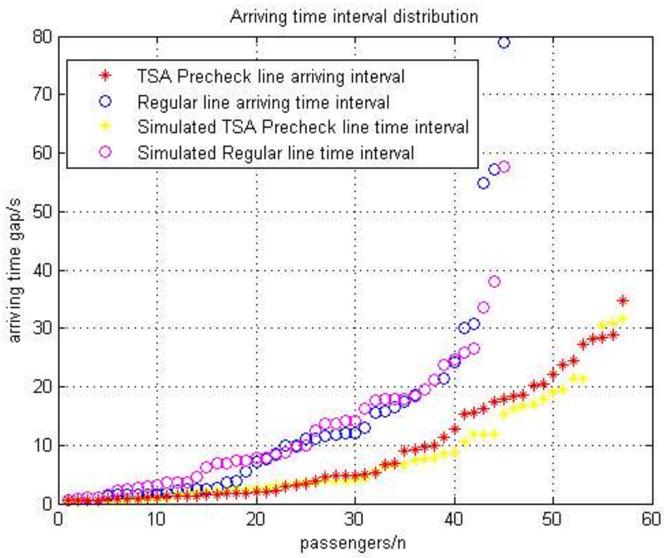


Figure 3: Data Analysis of Column A & B

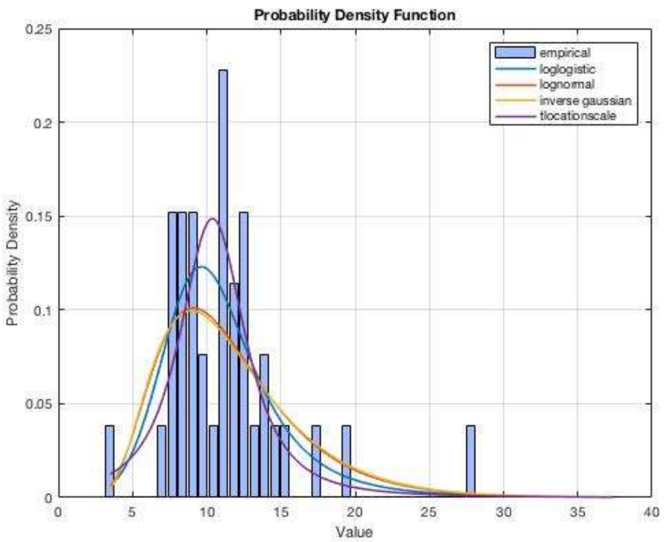


Figure 4: Fitting of Column E

```
tsa_percent = 32 #percent of tsa pre-check passengers
special_percent = 0 #percent of secial passengers
baggage_type_0 = 0 #percent of passengers with no baggage
baggage_type_01 = 100 #percent of passengers with no or small baggage

passenger_number = 100000#number of passengers
l = 0.3390 #average number of passengers per second

tsa_id_check_number = 1#numbers of id checking officers for pre-check
reg_id_check_number = 3#... for regular pax

tsa_screening_number = 3#numbers of screening lanes
reg_screening_number = 6

p = 50


class passenger:
    def __init__(self, pid, time, tsa, baggage_type, group_size, special):
        self.pid = pid
        self.arrival_time = time
        self.tsa = tsa
        self.baggage_type = baggage_type
        self.group_size = group_size
        self.special = special

    def __(self):
        return "pass #%d" % self.pid

    def move(self):
        return passenger(self.pid, self.exit_time, self.tsa, self.baggage_type, se

    def show(self):
        pass
    def show2(self):
        pass
        #print "#%d: (tsa %d, bg type%d, group %d, spec %d), arrive %.2f, service

def truncatednormal(mu, sigma, x):#normal distribution with mu and sigma truncated
```

```
        t = random.normalvariate(mu, sigma)
        while t < x:
            t = random.normalvariate(mu, sigma)
        return t

def dtm(mu, sigma, x):#returns a function output a truncated normal distribution
    return lambda t: truncatednormal(mu, sigma, x)



class queue:#service with queueing
    def __init__(self, rand):
        self.random_service_time = rand #a function returns service time
        self.previous_exit_time = 0
        self.pa_list = [] #list of passengers ever in this queue

    def enqueue(self, pa):
        pa.service_time =
        self.random_service_time(pa) * pa.group_size * (1.0 + pa.special * special
        pa.service_start = max(pa.arrival_time, self.previous_exit_time)
        pa.exit_time = pa.service_start + pa.service_time
        self.previous_exit_time = pa.exit_time
        self.pa_list.append(pa)

    def mean_waiting_time(self):
        return
        numpy.mean(map(lambda x: x.service_start - x.arrival_time, self.pa_list))

    def std_waiting_time(self):
        return
        numpy.std(map(lambda x: x.service_start - x.arrival_time, self.pa_list))

    def arrival_rate(self):
        return
        (len(self.pa_list) - 1)/(self.pa_list[-1].arrival_time - self.pa_list[0].a

    def exit_rate(self):
        return
        (len(self.pa_list) - 1)/(self.pa_list[-1].exit_time - self.pa_list[0].exit
```

```
class service:#parallel service
    def __init__(self, rand):
        self.random_service_time = rand

    def serve(self, pa):
        pa.service_time = self.random_service_time(pa)
        pa.service_start = pa.arrival_time
        pa.exit_time = pa.service_start + pa.service_time



def random_passenger(pid, time):# generate a random passenger
    tsa_seed = random.randint(0,100)
    tsa = 1 if tsa_seed < tsa_percent else 0

    special_seed = random.randint(0,100)
    special = 1 if special_seed < special_percent else 0

    baggage_seed = random.randint(0,100)
    baggage = 0 if baggage_seed < baggage_type_0 else
        (1 if baggage_seed < baggage_type_01 else 2)

    return passenger(pid, time, tsa, baggage, 1, special)



t = 0.0

passenger_list = []
for i in range(passenger_number):#generate passengers
    t += random.expovariate(l)
    pa = random_passenger(i, t)
    passenger_list.append(pa)

#step one: id check

tsa_a_list = []
for i in range(tsa_id_check_number):
    tsa_a_list.append(queue(dtm(10.19767,2.9814,0)))
```

```
reg_a_list = []
for i in range(reg_id_check_number):
    reg_a_list.append(queue(dtm(10.19767,2.9814,0)))

for i in range(passenger_number):
    pa = passenger_list[i]
    if pa.tsa:
        random.choice(tsa_a_list).enqueue(pa)
        pa.show2()
    else:
        random.choice(reg_a_list).enqueue(pa)
    pa.show()

print "============"

tsa_b_list = [] # step 2: preparing for screening
reg_b_list = []
for pa in passenger_list:
    if pa.tsa:
        tsa_b_list.append(pa.move())
    else:
        reg_b_list.append(pa.move())

tsa_b_list.sort(key = lambda x: x.arrival_time) #
reg_b_list.sort(key = lambda x: x.arrival_time)


tsa_check = []
for i in range(tsa_screening_number):
    tsa_check.append(queue(dtm(5,1,0)))

reg_check = []
for i in range(reg_screening_number):
    reg_check.append(queue(dtm(10,1,0)))

for pa in tsa_b_list:
    random.choice(tsa_check).enqueue(pa) #passengers move to a random lane
    pa.show2()
```

```
for pa in reg_b_list:
    random.choice(reg_check).enqueue(pa)
    pa.show()

print "============"

tsa_scan = []
tsa_scan_x = []
for i in range(tsa_screening_number):
    tsa_scan.append(queue(dtm(11.6372,5.8688,0)))
    tsa_scan_x.append(queue(dtm(7.5420,3.2417,0)))

reg_scan = []
reg_scan_x = []
for i in range(reg_screening_number):
    reg_scan.append(queue(dtm(11.6372,5.8688,0)))
    reg_scan_x.append(queue(dtm(7.5420,3.2417,0)))

# step 3: screening

x_ray_time = 5.0# time for baggage is constant

tsa_c_list = []
reg_c_list = []
for i in range(len(tsa_check)):
    print "in lane #%d:" % i #each lane in step 2 output to a lane in step 3
    for pa in tsa_check[i].pa_list:
        new_pa = pa.move()
        tmp = random.randint(0,100)
        if tmp < p:
            tsa_scan[i].enqueue(new_pa)
        else:
            tsa_scan_x[i].enqueue(new_pa)
        new_pa.exit_time =
            max(new_pa.exit_time, new_pa.arrival_time + x_ray_time * pa.baggage_ty
        tsa_c_list.append(new_pa)
        new_pa.show()

for i in range(len(reg_check)):
    print "in lane #%d:" % i
```

```
    for pa in reg_check[i].pa_list:
        new_pa = pa.move()
        tmp = random.randint(0,100)
        if tmp < p:
            reg_scan[i].enqueue(new_pa)
        else:
            reg_scan_x[i].enqueue(new_pa)
        new_pa.exit_time =
            max(new_pa.exit_time, new_pa.arrival_time + x_ray_time * pa.baggage_ty
        reg_c_list.append(new_pa)
        new_pa.show()


print "============"

tsa_collecting = service(dtm(10,5,0))
reg_collecting = service(dtm(10,5,0))

for pa in tsa_c_list:# step 4: collecting
    new_pa = pa.move()
    tsa_collecting.serve(new_pa)
    new_pa.show()
    #collecting is a parallel process

for pa in reg_c_list:
    new_pa = pa.move()
    reg_collecting.serve(new_pa)
    new_pa.show()
```