A Cellular-automaton Model for Throughput Optimization at Airport Security Checkpoint

Hong-qing ZHOU*, Cui-hua LU and Shu-hu WANG
Naval Aeronautical University, Yantai China 264001
*Corresponding author

Keywords: Cellular-automaton model, Throughput optimization, Security checking process.

Abstract. The purpose of throughput optimization is to optimize the passenger throughput at airport security checkpoint while still maintain the level of passenger safety. Queuing theory is frequently used to optimize the queuing efficiency. While in this paper, we apply cellular-automaton theory to simulate the passenger flow in the process and evaluate the performance of the airport security checkpoint. The simulation results show the validation of the cellular-automaton model.

Introduction

Security check at airport is of vital importance to guarantee passengers’ individual safety; however, it always receives criticism for extremely long lines at security checkpoints. Tons of passengers are suffering a lot waiting too long for security check. It’s hard to find a perfect balance between safety and security-check efficiency. We believe there exists the opportunity to relieve this problem.

We first try to build a mathematical model that can analyze the passenger flow through the airport security checkpoint and assess the performance, consisting of throughput and wait time, of the airport security checkpoint. With that model we can find out the area with the lowest process rate and thus explain what mainly contributes to the congestion in the airport. Furthermore, we can propose some modifications to increase the throughput and lower the difference among passengers’ wait time while the safety standard is maintained.

Bottleneck-finding Method

The security check process for a typical airport security checkpoint is shown in Figure 1[1].

![Figure 1. The Security Checking Process.](image)

In the security checking process, passengers first enter checkpoint at Zone A and queue up waiting for processing ID Check Step. After ID check, they enter Zone B and choose one line to wait for screening. When in the top of the queue, the passenger will remove their belongings and get ready for Millimetre Scan for physical body as well as X-ray Scan for luggage. If the officer suspects there is something forbidden for boarding, passenger will have an additional check at Zone D.
The real Security Checkpoint Wait Times data is available on data.gov [2]. By grey system theory and exclusion, we have found that belonging-removal step is the bottleneck. So we emphasize on applying a cellular-automaton-based approach [3] to simulate the security check process and get throughput of passenger flow.

Waiting-time-finding Model

In order to understand how the security checking procedure influence throughput and variance in wait time, we analyze the behavior of passengers and find out the cycle time in the airport at any specific time. The problem can be considered as stochastic agent-based model. A cellular automaton is a discrete model that describes the time development of a system. It is referred to as a discrete model because it treats time as discrete variable. The model requires an initial configuration and a set of fixed laws that determine how the system develops. At every time-step, the cellular automaton will advance incrementally and the laws will implement [4].

Model Assumptions

- The number of passengers’ arrivals obeys a Poisson process.
- Time of each checking step is determined by normal distribution.
- Approximately 45% of pre-check passengers are considered in this model [5].
- The rate of passengers with additional scanning is around 9% [6].
- The time of each significant step is the average from the data published by TSA. For missing data, we use empirical time instead.
- The probability of failing to identify dangerous items in regular-check is 0.5% and 1.5% in pre-check, both of which are empirical numbers.
- The probability of regular-check passengers taking dangerous items is 0.03% and 0.01% for pre-check passengers.

Cellular-automaton-based Modeling

To simplify the simulation, we mesh the movement of the whole checking process, as shown in Figure 2. As depicted, C1-C5 represent the belonging-removal area and D1-D5 represent the body scanning area. Also, for pre-check passengers, there are two entrances in Zone A and one lane for scanning in Zone B. While for regular-check passengers, the corresponding numbers are 5 in Zone A, and 3 when free and 5 when busy in Zone B. Additional-check passengers will go to Zone D in the dashed way.

Figure 2. Checkpoint Layout for Simulation.

Poisson Process of the Arrival. The passengers’ arrival at the airport could be determined by a Poisson distribution [7]. Hence, the probability of n arrivals at time t could be described as:

$$P(N(t) = n) = e^{-\lambda t} \frac{\lambda^t}{n!} (N(0) = 0)$$

(1)

$$\forall s, t, P(N(t + s) - N(s) = n) = e^{-\lambda t} \frac{\lambda^t}{n!}$$

(2)
Where N(t) represents the cumulative arrivals at time t.
Hence we can deduce:

\[ P_{w_n}(t) = \lambda e^{-\lambda t} \frac{\lambda t}{(n-1)!} \quad (t \geq 0) \]  

(3)

Where \( w_n \) represents the cumulative time of n arrivals.
Define \( T_n \) as the time interval n-1 and n arrival, thus

\[ T_n = W_n - W_{n-1} \]  

(4)

\( T_n \) is determined by exponential distribution, so its density function is:

\[ P_{T_n}(t) = \lambda e^{-\lambda t} \quad (t \geq 0) \]  

(5)

From where we could deduce that:

\[ E(T_n) = \frac{1}{\lambda} \]  

(6)

From the published data, we are able to estimate the average time interval \( E(T_n) \). Hence \( \lambda \) can be concluded and the distribution of arrival can be solved.

**Rules & Simulation.** The following rules are set based on the assumptions for the model.

- We use a fixed time interval for the state change of every cell/passenger. For the first round of simulation, suppose it is 10 seconds.
- We assume nobody is at the airport at the beginning of simulation, and every interval we generate a random number by Poisson distribution to simulate the passengers’ arrivals in a certain period of time.
- For every cell/passenger at the airport, it should be in any of the six states described in Table 2. The corresponding process of each state is also depicted in Figure 3.

<table>
<thead>
<tr>
<th>State</th>
<th>Description</th>
<th>End Condition</th>
<th>Time Distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>( S_0 )</td>
<td>Queuing for ID check</td>
<td>Reach the head</td>
<td>Gaussian</td>
</tr>
<tr>
<td>( S_1 )</td>
<td>ID checking</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( S_2 )</td>
<td>Queuing for scanning</td>
<td>Reach the head</td>
<td>Gaussian</td>
</tr>
<tr>
<td>( S_3 )</td>
<td>Removing belongings</td>
<td></td>
<td>Gaussian</td>
</tr>
<tr>
<td>( S_4 )</td>
<td>Millimeter scanning</td>
<td></td>
<td>Gaussian</td>
</tr>
<tr>
<td>( S_5 )</td>
<td>Additional check</td>
<td></td>
<td>Gaussian</td>
</tr>
</tbody>
</table>

Table 2. Cell States Description.

Figure 3. Cell States in Checkpoint Layout.

Through published wait time data, we can get the estimation of expectation and standard deviation (to represent variance) of state \( S_1, S_4, S_5 \) and estimate \( S_3 \) with empirical knowledge. Here comes the last problem for simulation, which is the decision making process of each cell. For clarity, we use a flow chart which could be found in Figure 4 to describe this process. After the first round of simulation, we get the result shown in Table 4.
Optimization

In the previous conclusion, we found that Belongings-removal step is the bottleneck, thus subsequently, our modifications for optimizing passenger throughput mainly focus on this step. Our goal is to raise the passenger throughput as well as reduce the waiting time variance with the constraint of not lowering passenger safety. Here we define $P(S)$ as safety level, which is the probability of event $S$ occurs that passengers with dangerous items are not checked out. Obviously, $P(S)$ could be decomposed into the multiplying of two independent probabilities $P(A)$ and $P(B)$, where $P(A)$ represents the occurrence probability that passengers take dangerous items and $P(B)$ represents the occurrence probability that dangerous items are not detected by TSA officer. Hence we got:

$$P(S) = P(A)P(B)$$

So our constraint is translated to at least maintain $P(S)$ and don’t raise it while optimizing.

Modification 1: Increase Types of Security Check

Currently there are only two types of passengers for security checking, that are pre-check and regular-check. It’s intuitive that there exists the chance to let some passengers with regular type go into another type which doesn’t need as much screening as regular-check needs. Here we call it medium-check. Main differences between the three types are described in Table 5.

The simulation result with the additional passenger type is shown in Table 6. For safety evaluation, we calculate $P(S)$ with a series of probability assumption. As a result, the safety levels are extremely close that the differences could be ignored. Hence our modification is validated.
Table 6. Simulation results with additional passenger type.

<table>
<thead>
<tr>
<th></th>
<th>Throughput</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Before modification</td>
<td>0.37</td>
<td>430.01</td>
</tr>
<tr>
<td>After modification</td>
<td>0.51</td>
<td>420.68</td>
</tr>
</tbody>
</table>

**Modification 2: Adjust the Proportion of Lanes**

Considering the current percentage of pre-check is up to 45%, so maybe switching some regular-check lanes to pre-check will help raise the checking efficiency. To determine the switch number, we conduct a sensitivity analysis as shown in Figure 5. We could see from the chart that when number of pre-check lanes is over 3, the variance remains steady, while the throughput reaches the peak when it’s 3. In addition, this modification doesn’t have any impact on safety, so we finally choose 3 pre-check lanes for optimization.

![Figure 5. Deviation of waiting time.](image)

**Summary**

In this paper, based on the bottleneck-finding model of airport security check step, we apply a cellular-automaton-based algorithm to maximum simulate the actual checking process. Then we propose significant modifications to optimizing the throughput and waiting time variance. With the assistance of cellular-automaton-based modeling, we are able to validate our modifications.

**Reference**


