For office use only	Team Control Number	For office use only
T1 ________________	57333	F1 ________________
T2 ________________		F2 ________________
T3 ________________	Problem Chosen	F3 ________________
T4 ________________	D	F4 ________________

Summary

We use cellular automata to build models with a high degree of simulation of the flow of passengers through security check points at the airport, in order to identify the factors that affect throughput in a short time and find the optimal solution. Most people use queuing theory and simple cellular automata model to discuss such issues. But queuing theory is difficult to help us find the specific reasons for the emergence of the problem, while the simple simulation is easy to deviate from the actual situation. We hope that after the highly-simulated model completed, we can extract the data required, turn them Into relevant parameters, and make judgement on every specific situation quickly and intuitively with the help of computer.

In order to make the model fit the actual situation as far as possible,
We designed some assumptions and rules. For example, the time interval between the visitors assigned by computer in the model meet the exponential distribution ;and the model can give guidance on the direction of the passenger according to the statistics at the moment.etc.

For the first task, we define the storage capacity of the region as the difference between the number of people get out of this point with the number of people enter from former point. The model can be used to draw the curve of the storage capacity of each check point with time. We can identify the region where the problem occurs by comparing the trend of the curves and the changes of slope, and then watch the process demonstrated on computer to verify our conjecture.

For the second task, we interpret requirements for reduce the waiting time as a demand that security systems should make passengers' waiting time fluctuate in a reasonable range. We use the quotient of the total number of people in the system divided by the number of people out of the system per unit of time to find the optimal solution.

Then we find the parameter which different cultural background may influence and add a variable θ to control this parameter. By analyzing the sensitivity of the variable θ, we find that the clogging index maintains a small fluctuation while θ changes in a wide range, which prove that our model is stable.

Our model can be applied to any airport with similar security process, and we create a more flexible approach to solving the airport congestion problems. At the same time, under the background of large data, our model can make the selection of parameters closer to reality through the accumulation of data 'experience', and it is expected to get the forecast function, which can help the airport to take measures in advance. Although our model is only applicable to the airport for the time being, but the use of cellular automata simulation of the actual situation for decision-making provide a new way of thinking for human to solve more complex problem in the future.

Contents

1. Introduction

Problem background

After the events of 9·11, security checks have been largely intensified at airports around the world, which led directly to increased queuing time for passengers. Because of the strong public concern about the issue of long-term security, airlines hope to have a good way to determine which part of the checking system is the cause of delay and congestion, then find an optimal program to reduce the waiting time for passengers and keep security level at the same time. Our main tasks are as follows:

- Construct a model to find the problematic checkpoint quickly.
- Propose the optimization scheme and build a model to test it.
- Considering the differences of different cultural travelers, we improve the model and carry out the sensitivity test.
- Propose universal or differentiated schemes.

At present, it is a common way to use queuing theories for queuing problems, but the theory cannot solve multi-processes problems with complex parameters. Our goal is to simulate the process of airport security check as realistically as possible through a cellular automaton model. We can give the parameters the actual value to judge the key points in practical problems, and use the computer to build the optimization scheme. This model can present the problem in a visual form and will have a considerably wide range of applications[1].

2. Simulation Model of Cellular Automata Security process

2.1. Simulation model

Overview:

Our cellular automation consists of several parts which are Space-Matrix-Creation part, Space-Matrix-Display part, Behavior-simulation part, System-parameters, Data-statistic part, and it also has global counting vectors and global variables which build connection between different files.

Basic setting:

- If a cellular arrives in a position, the value of the position plus one, and if a cellular leaves a position, the value of the position minus one.
- The value of normal blank area is 0, the value of ID-check area is 2, the value of Check-point area is 3, the value of Bag-check area is -888, the value of bags storage area is -886, the value of Forbidden area is -900, the value of Special-blank area is -890.
- One loop of the program stands for one second.

Space-matrix-creation part:

This part divides the whole space into two parts of areas which are Pre-Check area and regular check area. Those two areas are also divided into several subareas which are Forbidden area, Special-blank area which is used as places where cellular can find the lane which contents fewer people, and Checkpoint area, bags storage area. To create the matrix, this creation-part will give value to every cell in those areas which are mentioned above.

Space-display part:

This part divides the whole matrix into two areas, ID-check area and Security-check area which is in the south of the ID-check area. It will show every cell of the Space-matrix in different colors considering of their positions and values.

Behavior-simulation part:

Its basic function is to make the cellular move forward, plus, it also contains logical judgement functions and a simple function which can help cellular find the lane which contents less people and help it direct it's way to it. With the help of these functions, cellular can find the way which costs the shortest time to pass the check-point relatively and the model also can better simulate conditions happening in real airport.

Global-count vector:

They count the time that cellular has spent on waiting, they can also store the important values which are used in

Global variables:

They serve the role to transmit important parameters to functions from the main script file.

Procedure controlling the waiting-time which follows a probability distribution:

In every loop, in terms of a cellular that are waiting for the finish of a specific event, the value of one cell which belongs to a Global-count vector and counts the waiting-time of this cellular will be changed into a probability value. Then, the program will generate a random number, and if the random number is smaller than the value, we can say that the event is over and this cellular can move to the other cells. This model describes the process of passengers entering the airport, waiting in line for both security check and ID-Check. Here we distinguish the normal passage and the pre-check one.

2.2 Simulation model of process A

Main parameters:

Symbols	Definitions
T_p	The time interval of arrivals at the Pre-Check queue
T_n	The time interval of arrivals at regular check queues
N_p	Number of ID check points for Pre-Check passengers
N_n	Number of ID check points for regular passengers
T_{id}	The time required at ID-Check

Table. 1

Important assumptions and the obtaining of parameters

- Based on the existing discussion in queuing theory, the time interval of passengers arriving at the airport is usually exponentially distributed[2][3].

The probability distribution function of the exponential distribution is

$$F(x;\lambda)=\begin{cases}1-e^{-\lambda x} & ,(x\ge 0)\\ 0 & ,(x<0)\end{cases} \tag{1}$$

By collecting enough samples of the time interval between the arrival of two individuals, the parameters of the exponential distribution can be obtained by the SPSS process.

- The number of ID check points is measured according to the actual situation
- The ID-Check time is not closely around its average, because its volatility may have some impacts on clogging. In the simulation, we use the distribution which is obtained by nonparametric test of the sample to ensure that the model is closer to reality.

Rules for simulation of process A

Input rules: The number of arrivals per unit time H_p and H_n are assigned by the computer in the form of random numbers that satisfy the Poisson distribution.

Queuing rules: People arriving will be assigned to the end of a row randomly, the person will move forward until the former area is occupied.

Changing row rules: After passenger stops, if there is no one on either side of the square centered on him, the person will move randomly to one side and forward until it stops.

ID-Check rules: Checks are performed in yellow locations, where each person stays for T_{id} before proceeding. It will change to blue when someone is being checked.

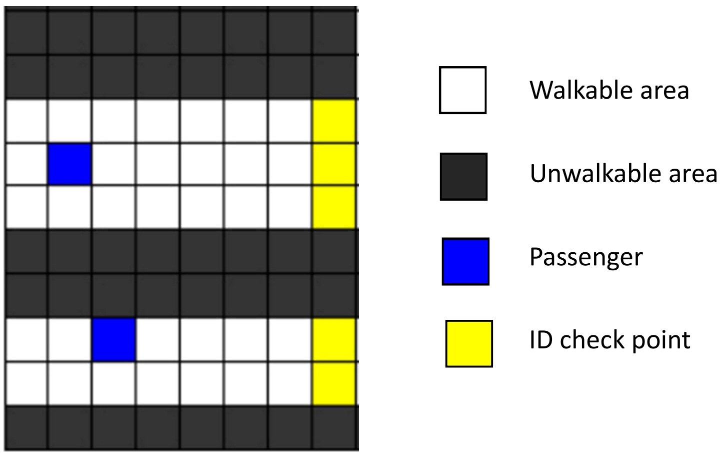

Fig.1

2.3 Simulation model of process B&C

This model is used to simulate the process passing the security area B&C after finishing process A. It mainly includes three processes: the baggage moves on the conveyor belt, the passenger goes through the millimeter wave machine, and the person gets the baggage from the conveyor belt and leaves the area.

Main parameters:

Symbols	Definitions
N_m	The number of the millimeter wave machines
N_b	The number of the baggage screening machines
T_m	The time it takes for passengers to pass the millimeter wave machine
T_b	The time it takes for items to pass the baggage screening machines
T_p	The total time Pre-check passengers spend in process B&C
T_n	The total time regular passengers spend in process B&C

Table. 2

Important assumptions and the obtaining of parameters:

- It is assumed that the time that an item passes through the scanning machine is changeless and is determined only by the speed of the conveyor.

According to the information on the website, generally, the scanning part of the scanner is 1.2 meters, usually at the speed of 0.2 m / s[4]. Therefore, without changing the speed of the conveyor, we set T_b for a fixed value. As is shown below

Fig.2 The schematic diagram and relevant figure of a normal baggage scanner

- Assuming that T_m is the shortest time people spend in millimeter wave machine, we believe that the time spent over the shortest one is affected by many factors. It is difficult to obtain the factors directly through the data measurement, so the variation of this step will be adjusted with the total time T
- The total time T is obtained by directly measuring the period from putting items to taking items. In addition to the time required for the inspection, and the time required to wait and undress etc.

$$T = max\left(T_m, T_b\right) + T_o \tag{2}$$

And we assume that T_o satisfies a certain distribution。

- We assume that the time to wear and take off clothing is equal

Rules for Simulation of process B&C:

Input rules: Each person entering the B process places one luggage on the gray box beside, from where the person and baggage proceed respectively. If one's baggage cannot be placed in the gray box, he cannot move forward.

Queuing rules: Passengers entering the B process can only proceed in the middle area between the black boxes on both sides until there is one person in front of them

Millimeter wave machine rules: The person arriving at this point is going stay for ‘t’, and then directly fell to the bottom (If there are people in front, the person will then stop).

For person in Pre-Check lanes

$$t = T_m + \left[T_p - max\left(T_m, T_b \right) \right] / 2 \tag{3}$$

For person in regular lanes

$$t = T_m + \left[T_n - max\left(T_m, T_b \right) \right] / 2 \tag{4}$$

Baggage screening rules: The item leaves this position after T_b and then falls directly to the bottom (If there are items in front, the item will then stop).

Rules of taking luggage: When the items and people arrive at the bottom, we start timing, and both of them go out from the system after $T_o/2$.

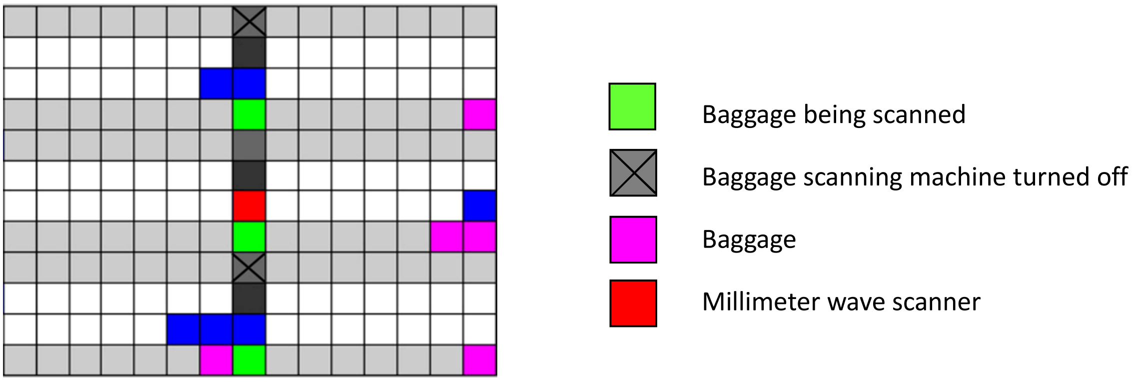

Fig. 3

Simulation of the complete process of security inspection

Connecting the A process and the B process together, we can simulate the entire security process by changing the parameters flexibly.

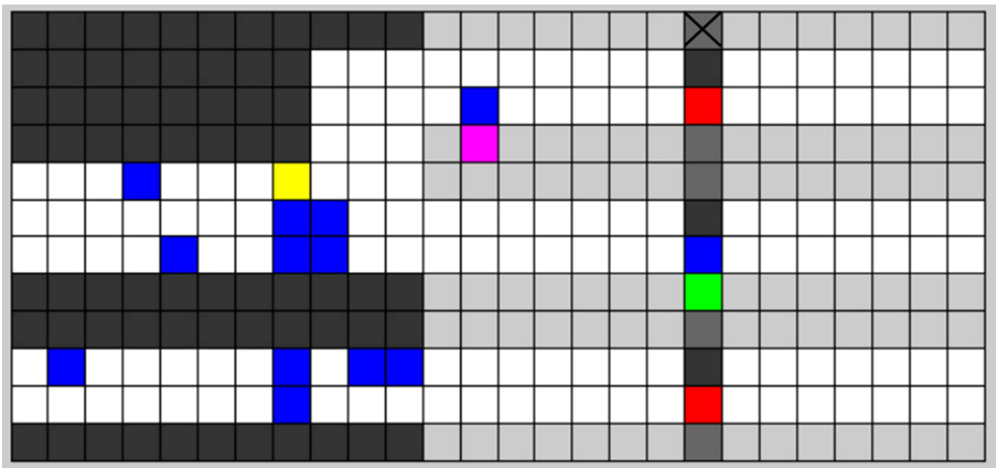

Fig. 4

2.4 Model evaluation index

- Determine the problematic checkpoint

First of all, through the simulation model, we can visually observe the block situations of the lanes to judge the problematic checkpoint, but we also need to explain based on some certain data. We use computers to record the total number of people entering and passing each security check area. The gap between them represents the number of people strand in the link. By comparing the people in each link, we can find the bottleneck of the system, through which we can adjust and optimize the relevant parameters.

- Determine the optimization method

In this paper, we will not consider completely reforming the security process, but by changing the number of open nodes and the proportion of them to improve efficiency.
Obviously, if more channels, including ID-check points, are opened, the goal of reducing passenger waiting time will certainly be achieved, but the airport will also have to control manpower and equipment costs to meet their operational needs. Therefore, we interpret requirements for reduce the waiting time as a demand that security systems should make passengers' waiting time fluctuate in a reasonable range.

$$T_{average} = N_1 / N_2 \tag{5}$$

$$N_1 = N_{in} - N_{out} \tag{6}$$

$$N_2 = N_{out} / T_{total} \tag{7}$$

$T_{average}$:Average waiting time
N_1: The number of people in the system
N_2: The average number of people that pass through the security check point in per unit of time
N_{in}: The number of people going into the system
N_{out}: The number of people leaving the system
T_{total}: Total time of the simulation with the given parameters

We can get the time-varying images of the number of people leaving entering the system. In that way we can judge if the system meet the requirements.

If it does not meet the requirements, we can adjust the parameters, including the number of the lines and the duration of the every link. If some parameters need to be adjusted, we can use the model to find the optimal solution of which the cost is minimum.

3. Verify the model with the given data

3.1 Task one: Identify the problematic checkpoint

Analyze the data to obtain the required parameter values for the model

- The number of machines at checkpoints: The data in the table did not present it. We assume that our model has the same distribution and the numbers of machine with the picture given in the problem. The values of them are as follows:

$$N_p=2 \quad N_n=3 \quad N_m=3 \quad N_b=4$$

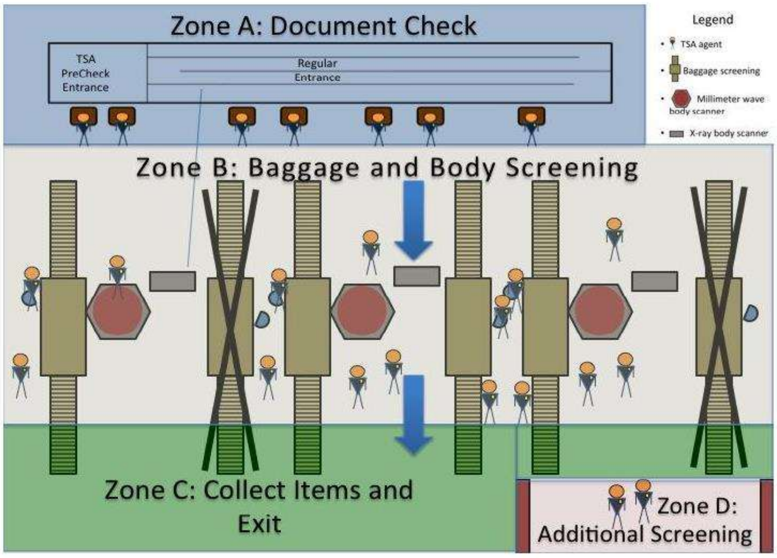

Fig. 5 The picture given in the problem

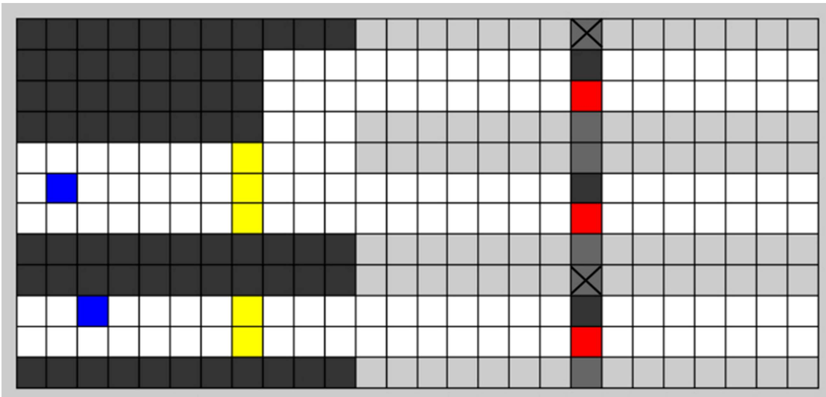

Fig .6 The structure of our model

- T_n is obtained from the data in column B of the table(Regular Pax Arrival Times). For the selection of zero point is uncertain, we find the gap between the latter one and the former one (The time interval for a person to reach the regular ID-check window).

One-Sample Kolmogorov-Smirnov Test

		VAR00001
N		45
Normal Parameters[a,,b]	Mean	11.6978
Most Extreme Differences	Absolute	.156
	Positive	.156
	Negative	-.082
Kolmogorov-Smirnov Z		1.048
Asymp.Sig.(2-tailed)		.222

Asymp.Sig.(2-tailed)=0.222>0.05

Thus it satisfies the exponential distribution, $\lambda=1/11.6978$.

- T_p is obtained from the analysis of column A(TSA Pre-Check Arrival Times) which is the same as that of column B. Nonparametric statistics are shown in the following figure:

One-Sample Kolmogorov-Smirnov Test

		VAR00001
N		56
Normal Parameters[a,,b]	Mean	9.1554
Most Extreme Differences	Absolute	.188
	Positive	.188
	Negative	-.118
Kolmogorov-Smirnov Z		1.406
Asymp.Sig.(2-tailed)		.038

a. Test distribution is Normal.

b. Calculated from data.

Asymp.Sig.(2-tailed)=0.038<0.05

It does not satisfy the exponential distribution, but we still believe that it is a problem caused by data collection. We see it as the exponential distribution with the parameter of $\lambda=1/9.1554$.

- The obtainment of T_{id}: The data in column C and column D is time interval which can be seen as Non-discriminatory data. The two samples were combined for nonparametric testing. Statistical results are shown in the following figure:

One-Sample Kolmogorov-Smirnov Test

		VAR00001
N		16
Normal Parameters[a,b]	Mean	11.2125
	Absolute	3.79260
Most Extreme Differences	Positive	.116
	Negative	.116
	Mean	-.101
Kolmogorov-Smirnov Z		.462
Asymp.Sig.(2-tailed)		.983

a. Test distribution is Normal

b. Calculated from data

- T_m is the minimum value of the differences between the neighboring data, which is 3.5s.
- T_b is the period for scanning when conveyor is at the regular speed. According to the materials, T_b is 6s.

The data in the column of F and G is the time stamp as bags exited the x-ray screening. This set of data does not reflect the effective information directly, so it is not used in our model.

- As for data in column H, we first sorted it and found that there were several small numbers. With our reasonable speculation, we assume that some of these data sets belong to Pre-Check lanes and the others belong to regular lanes. The scatter plot is as follows:

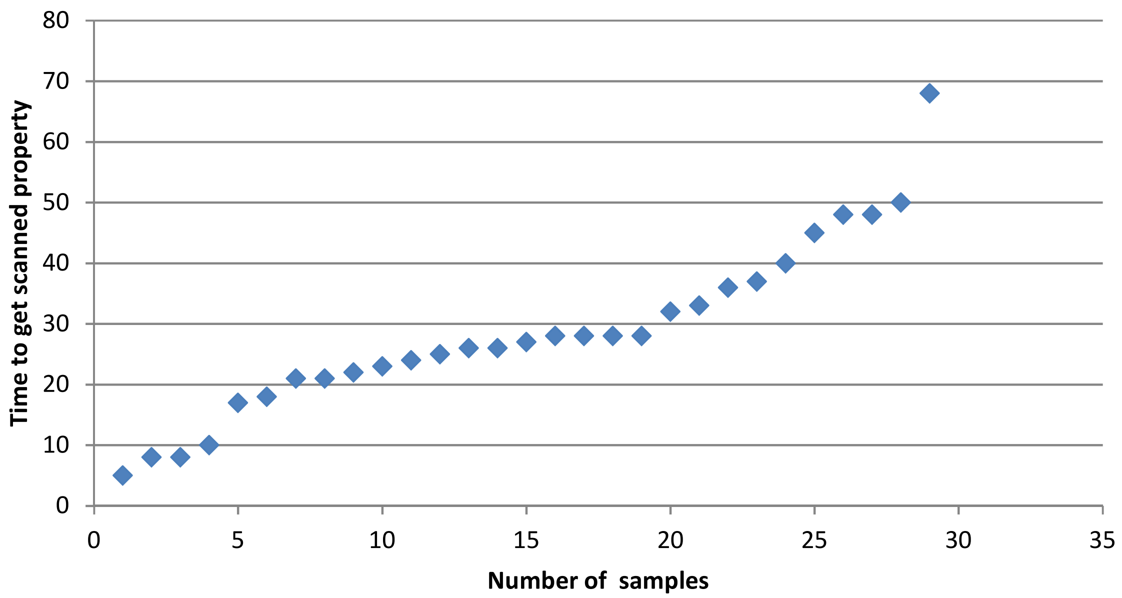

Chart. 1 The scatter plot of Time to get scanned property

T_p=15，T_n=35.

Run the program and find the problem

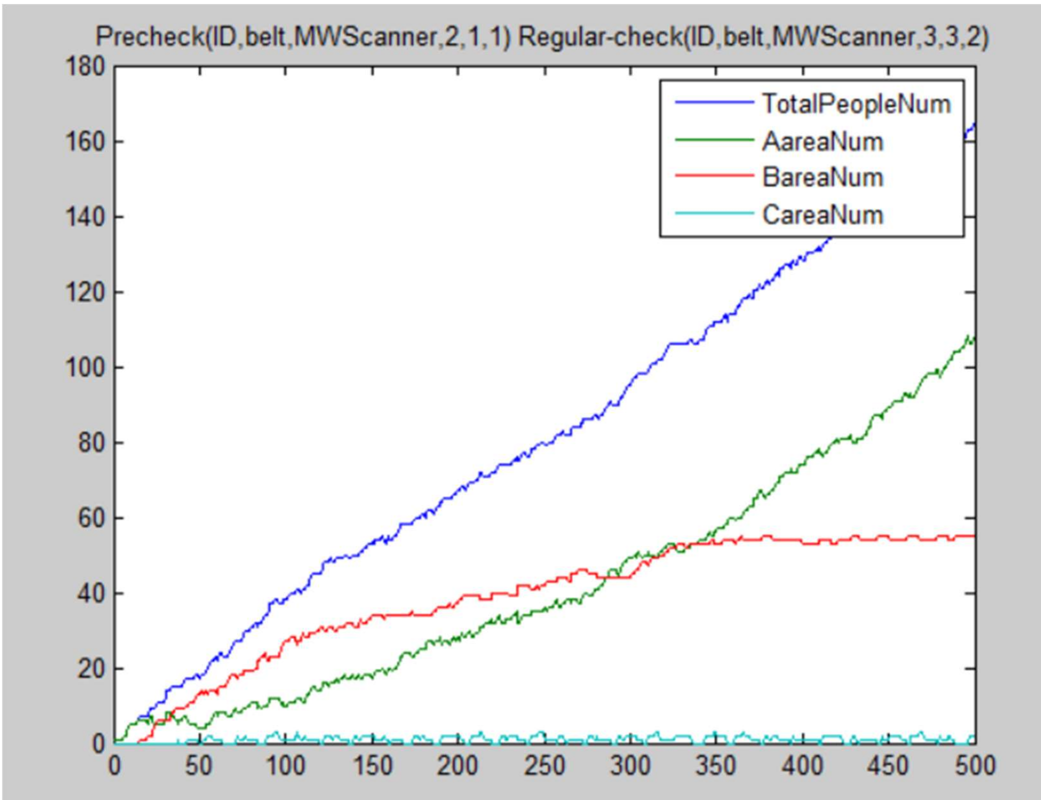

Chart.2 Real-time number of people in each area

The abscissa of this chart is time, and the ordinate is the number of people. Each line reflects the real-time number of each node.

According to the figure, at the beginning, the number of people in area A and area B increase from the start, but the area B increases faster. Due to the capacity of area B is limited, it gradually reach the upper limit and the curve gradually become moderate. B area reached the upper limit caused that the passengers stranded in area A. Therefore, the block in area A becomes more severe. After that, and the speed of forming blockage is also accelerated.

- Deduction 1: There is a problem in one part of the B area.
- Deduction 2: A small increase in the capacity of B area cannot solve the congestion problem fundamentally. It only postpones the congestion.

Further observing the process of computer simulation, we find that the area B of Pre-Check passage appears to be blocked first. And it is indeed the congestion of B area that leads to the congestion of area A. So deduction 1 is established. In addition, the congestion of zone B is caused by the longtime taking off clothes before the passengers entering millimeter wave body scanner.

3.2 Task two: Propose optimization and simulation

Optimization:

Step one: Despite the fact that the passengers passing the pre-check lane are not required to remove shoes, belts, light jackets, as well as their computers when proceeding through area B, the degree of congestion is still higher than expected. We think it is because an increasing number of people choose the Pre-Check lane but the number of Pre-Check lanes is only equivalent to one-third of the regular ones. So we try to increase the number of Pre-Check lanes to two.

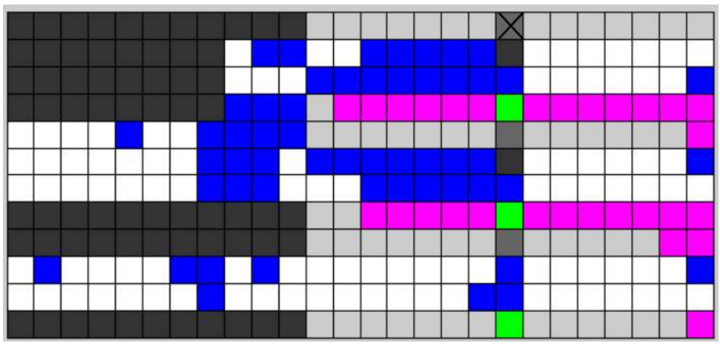

Fig.7 System simulation when time is 1000s

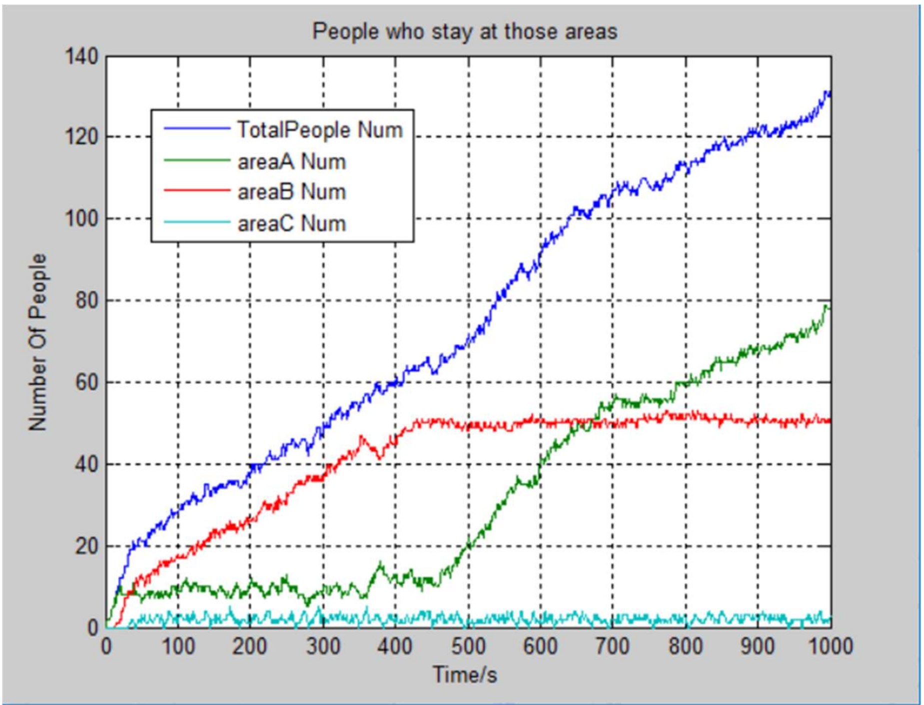

Chart.3 Real-time number of passengers in every area after step one

After being optimized, Pre-Check lane is no longer crowded. The next step for us is adjusting the regular ones.

Step two: To alleviate crowding problem of regular ones, we can consider reducing the cost of qualification for Pre-Check service as well as intensifying propaganda. Passengers are encouraged to select the pre-check channel to reduce the pressure on the normal passage. For example, increase the number of passengers entering the Pre-Check queue ($1/T_{p}$) from 1/9.16 per second to 1/7.50, meanwhile, reduce the number of passengers entering the regular queue ($1/T_n$) from 1/11.70 per second to 1/16.31.(1/9.16+1/11.70 = 1/7.50+1/16.31 The total number of arrivals per unit time remains unchanged) As can be seen from the following figure, the above measures played a role in extending the time against the congestion. However, because the input speed of area B is greater than the output speed, the system can not fully carry the passengers shown by the data in long terms.

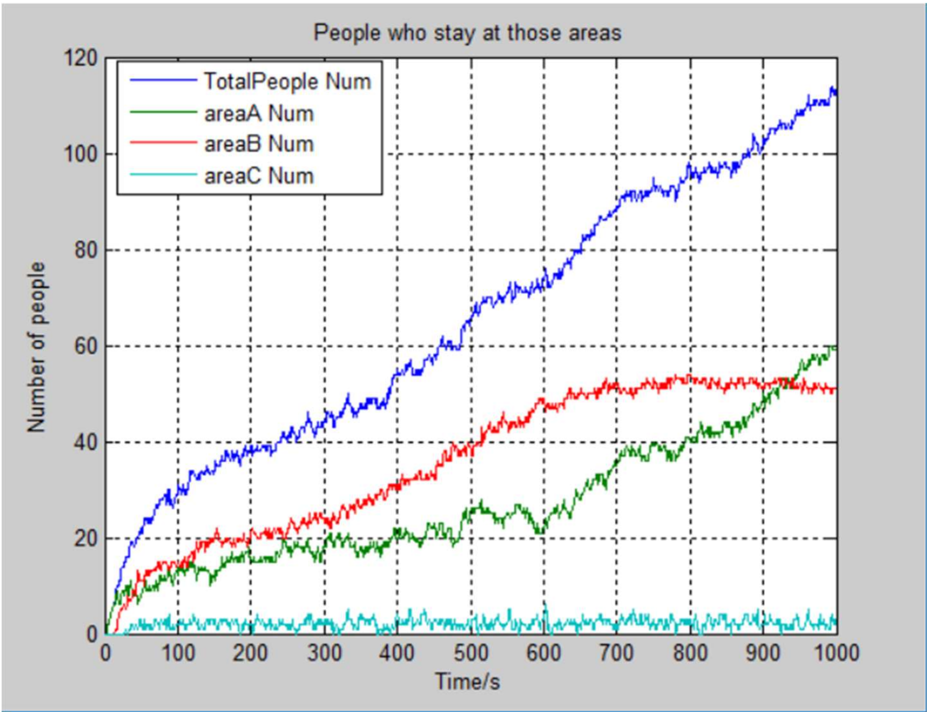

Chart.4 Real-time number of passengers in every area after step two

Step three: To Eliminate bottlenecks in the area B of normal lanes， we can consider the same way used for the pre-check problem. That is, increasing the number of lanes. But this will inevitably increase the cost. Observing several previous simulations, we notice that time spent on transporting and inspecting is not the short board of the system. On the contrary, there exists situation that the passengers are still being checked while their suitcases have been examined and can’t be took away timely. In view of this situation, we chose to close one of the conveyor belts and add a millimeter wave inspection device. In practice, adding TSA agents with metal detectors is also feasible.

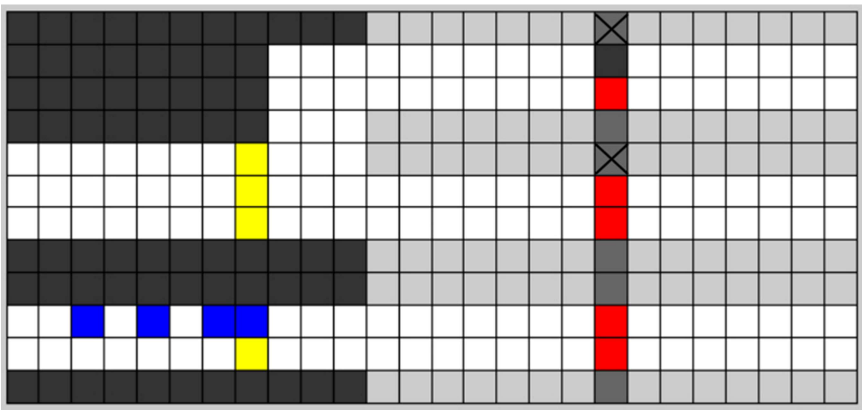

Fig.8 Diagram of the beginning situation

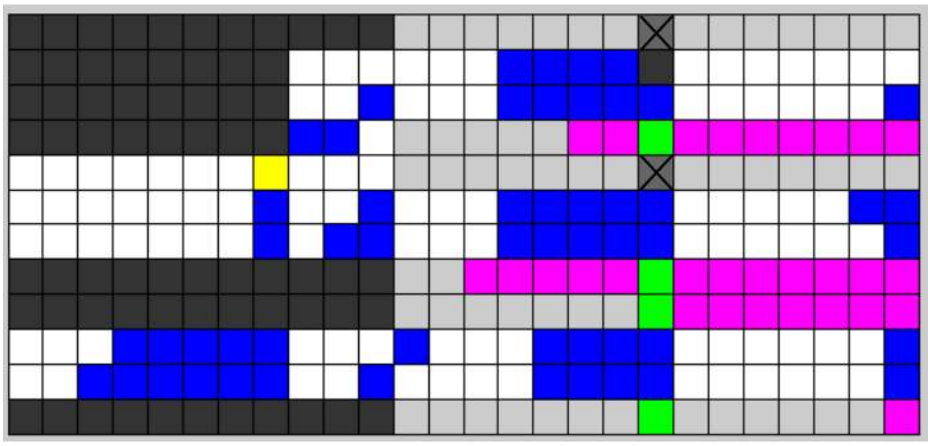

Fig.9 Diagram of the end situation

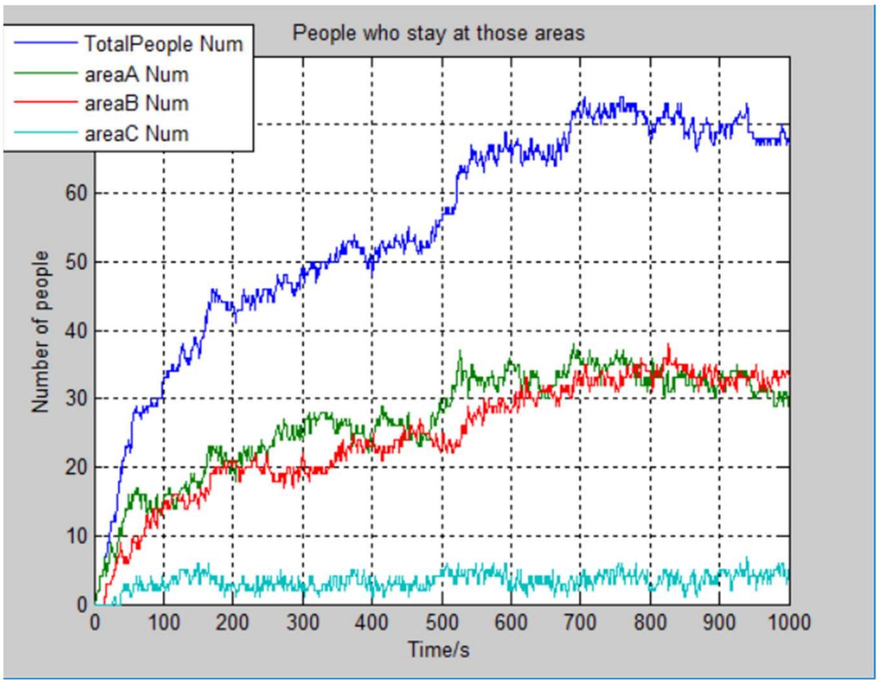

Chart.5 Real-time number of passengers in every area after step two

4. Wide applicability of the model

4.1 Task three: Considering cultural differences and sensitivity analysis

In our model, cultural differences mainly influence the parameter 'T'. Take American and Chinese people for an example, American people are more willing to keep a specific length of distance with others which will make them spend more time on waiting in queues, but Chinese people pay more attention on their speed of moving forward, and they will spend less time on waiting[5].
Thus, we add a parameter θ, which is associated with cultural differences, in this model and use it to adjust the value of T. Our sensitivity analysis is also based on θ.

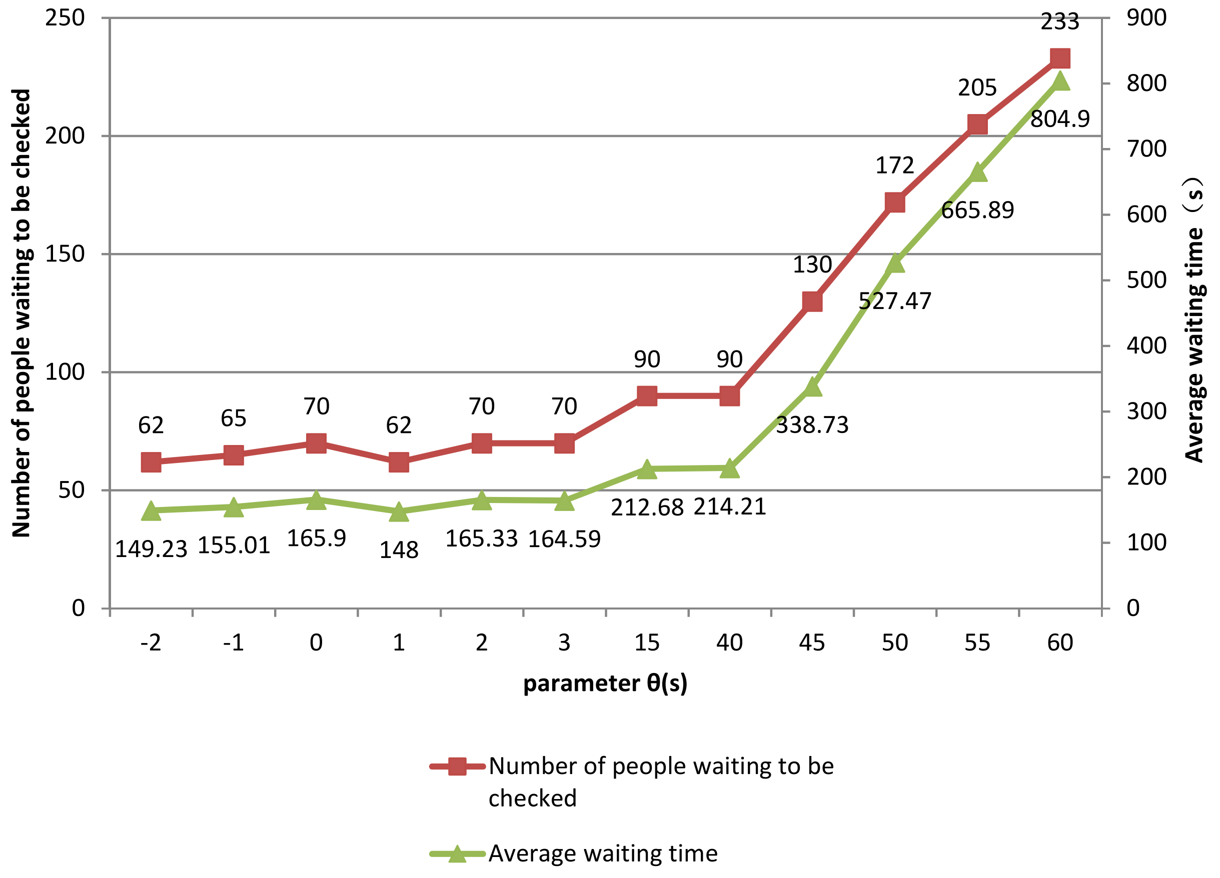

Chart.6

According to the chart, it is shown that when the value of θ is changed between 0 and 40, the average waiting time doesn't change greatly, but if it is changed bigger than 40, jam happens and the average waiting time becomes bigger with the growth of θ.
However, it is in low possibility that θ is bigger than 40 seconds, so it is reasonable to say that our solution of this problem is stable.

4.2 Task four: Recommendations

By input possible values of key parameter which may be the best answer, we get a good solution in the limitation of money

1. Increase the numbers of pre-check lanes.
2. Encourage more people choose pre-check,
3. Increase the number of Millimeter wave body scanner and close one belt to save money.

5. Model assessment

5.1 Questions about our solution's feasibility and its answers

a. Questions about Non-integer parameters:
   As cellular automation can only refresh values of cellular at the end of every unit of time which is an integer variable. So, when the time parameter is Non-integer, it will cause inaccuracy. To fix such kind of problems, we plan to improve the frequency of refreshing with 10^n (n is a integer that bigger or equal to zero), and change decimal fraction into integer.
b. Procedure that controls the waiting-time which follows a probability distribution:
   In every loop, in terms of a cellular that are waiting for the finish of a specific event, the value of one cell which belongs to a Global-count vector and counts the waiting-time of this cellular will be changed into a probability value. Then, the program will generate a random number, and if the random number is smaller than the value, we can say that the event is over and this cellular can move to the other cells.
c. Algorithm used to calculate the number of people who stay in this model:
   The number of people staying in this system is the answer of number of people arriving to the system minus people leaving this system..

5.2 Strengths

a. It can generate the values of parameters following specific probability distribution.
b. It can simulate the details of passengers' behavior in the Airport Security Checkpoint.

5.3 Weakness

Because of the limitation of time, the space matrix only has two dimensions, and we cannot give every cellular a specific ID-Number. Thus, the scalability of this model is limited. The redundancy of logical controlling code also cause the difficulty in updating and maintaining the code. What else, cellular in this model can only find the local optima.

5.4 Future improvement

We are planning to create a cellular automation based on matrix with 3 dimensions. Then, we can give every cellular a special ID-number which can help us control the behavior of the cellular more easily. What else, we will learn more about AI algorithms and make our model simulate the real life more precisely.

References

[1] Liao Mingjun, Sun Jian and Wang kaiying, Simulation of Queuing Behavior Based on Cellular Automata Model, Journal of Transportation Systems Engineering and Information Technology. 3 (2009) 140-145.

[2] Wang Shiyuan. Analysis on Real-Time Communication Technology of the Ethernet Based on the Queuing Theory[D]. Dalian: Dalian University of Technology, 2010.

[3] Zhao Yuehua. The Application Research of Queueing in Call Center[D]. Zhenjiang: Jiangsu University, 2008.

[4] Alibaba 24 1 2017
<https://www.alibaba.com/product-detail/xray-security-inspection-equipment-scanner-luggage_60490335925.html?s=p>

[5] Zhang Yu, The difference between Chinese and American on the consciousness of space, Jiangsu learned journal of School of Education. 3(1998) 140-142.

Appendix

An example of Space-matrix creation

```
%==========================================================================
%precheck part
for i=1:preIDcheck_Num
    space(idcheck_Point,i+1) = 3;
end
%draw precheck-part's forbidden area and blank area between IDcheck part
%and bag storage part
    space(1:midspot-var-1-1,1) = -900;
    space(idcheck_Point+1:midspot-var-1-1,2:precheck_width-1)=-890;
    space(1:midspot-var-1-1,preIDcheck_Num+2:max(precheck_width,precheck_Num*4))=-900;
    if(precheck_width>precheck_Num*4)
    space(1:spacelength,max(precheck_width,precheck_Num*4:precheck_width))=-900;
    end
    space(idcheck_Point+1:midspot-var-1-1,1+1:precheck_width-1)=-890;%blank


for i=0:precheck_Num-1%checkpoint part
    space(midspot,[1+i*4,1+i*4+3]) = -888;
end
```

An example of Space-matrix display

```
PLAZA(PLAZA==901)=0.2;%forbidden part
PLAZA(PLAZA==889)=0.4;%bag check part
PLAZA(PLAZA==887)=0.8;%bag storage part
PLAZA(PLAZA==891)=1;%blank-part between IDcheck and entrance of bag storage part
%===============================================================
%check point part color-strategy
for i = midspot
    for t = 0:total_Num-1
        for j=1:width
            if plaza(i,j) == 0;
                PLAZA(i,j,1) =0;
                PLAZA(i,j,2) =1;
                PLAZA(i,j,3) =0;
            else

                if plaza(i,j) == 3
                PLAZA(i,j,1) =0;
                PLAZA(i,j,2) =0;
                PLAZA(i,j,3) =1;
                else
                    if (plaza(i,j) == 2)
                    PLAZA(i,j,1) =1;
                    PLAZA(i,j,2) =0;
                    PLAZA(i,j,3) =0;

                    end
                end
            end
        end
    end
end
%===============================================================
```

An example of basic movement

```
                if space(i+1,j)==0||space(i+1,j)==3
                    space(i+1,j)=space(i+1,j)+1;
                    space(i,j)=space(i,j)-1;
```

An example of finishing an event costing time that follows probability distribution

```
%======================================================
%idcheck part
for i = idcheck_Point
    for j = 1:width
        if space(i,j) == 3+1
            a=Gs_dis(checkpause_vector(j));
            b=rand(1);
            if((a>=b)&& (space(i+1,j)==-890));
                space(i+1,j)=space(i+1,j)+1;
                space(i,j)=space(i,j)-1;
                checkpause_vector(j)=1;
                 Plot_vector(2)=Plot_vector(2)+1;
            end
         checkpause_vector(j)=checkpause_vector(j)+1;
        end
    end
end
%======================================================
```

An example of finishing an event costing certain time

```
            if space(i-1,j) == -887
                if((xraypause_vector(j)>=xraypause_Time)&& (space(i,j)==-886));
                    space(i,j)=space(i,j)+1;
                    space(i-1,j)=space(i-1,j)-1;
                    for k=i:spacelength-1
                        if (space(k+1,j)~=-886)
                            break
                         else
                          space(k+1,j)=-885 ;
                          space(k,j)=-886;
                        end
                    end
```

An example of finding the lanes containing fewer people.

```
                    EW=direct_selection(space,i,j);
                    end
                    if(EW>0)
                        if space(i,j+EW)==-890
                           space(i,j+EW)=space(i,j+EW)+1;
                           space(i,j)=space(i,j)-1;
                           flag111=1
                           break
                        end
                    end
```

```
function EW= direct_selection( space,y,x )
%====================================
%calculate the number of people in every lane
global precheck_Num
global regular_Num
global precheck_width
global regular_width
global midspot
global var
global doublebelt_vector
global Ifpre_Two
global Ifreg_Two

total_Num=precheck_Num+regular_Num;
queue_vector=zeros(total_Num,2);%column 1 stores the number of people in everylane, column 2 stores the left-most position of every lane

minx=0;

for i=1:precheck_Num
    for j=midspot-var-1:midspot
        for k=1+1+(i-1)*4:1+(i-1)*4+2
            queue_vector(i,2)=1+(i-1)*4+1;
            if(space(j,k)==1||space(j,k)==3)
                queue_vector(i,1)=queue_vector(i,1)+1;
            end
        end
    end
    if((doublebelt_vector(queue_vector(i,2))+doublebelt_vector(queue_vector(i,2)+1)==2)&&(Ifpre_Two==1))
        queue_vector(i,1)=queue_vector(i,1)/2;
    end
end
```

```
if(x>0&&x<precheck_width)
    flag=0;
    min=queue_vector(1,1);
    minx=queue_vector(1,2);
    for i=1:precheck_Num
        if( queue_vector(i,1)<min)
            minx=queue_vector(i,2);
            if(queue_vector(i,1)==min)
                flag=flag+1;
            end
        end
    end
    if(flag==precheck_Num-1)
        minx=0;
    end

    if(minx>x)
        if(space(y,x+1)==-890)
            EW=1;
        else
            EW=0;
        end
    else
        if(minx<x)
            EW=-1;
        else
            if(minx==x||minx==0)
                EW=0;
            end
        end
    end
end
```

Virtual storage of bags (In some cases, the space of the matrix are not big enough to store the bags of the passengers)

```
        if ((space(i+1,j-1)==-886)|| (space(i+1,j+1)==-886)||space(i+1,j-1)==-885||space(i+1,j+1)==-885)
            if (space(i+1,j-1)==-886&&Savebagflag_vector(j)==0)
               space(i+1,j-1)=space(i,j-1)+1;
               Savebagflag_vector(j)=1;
            else
                if (space(i+1,j-1)==-885&&Savebagflag_vector(j)==0)
                    savebag_vector(j-1)=savebag_vector(j-1)+1;
                    space(i+1,j-1)=space(i,j-1)+1;
                    Savebagflag_vector(j)=1;
                end
            end
            if (space(i+1,j+1)==-886&&Savebagflag_vector(j)==0)
                 space(i+1,j+1)=space(i+1,j+1)+1;
                 Savebagflag_vector(j)=1;
                else
                  if (space(i+1,j+1)==-885&&Savebagflag_vector(j)==0)
                   savebag_vector(j+1)=savebag_vector(j+1)+1;
                   Savebagflag_vector(j)=1;
                  end
             end

            if(space(i+1,j)==0)
               space(i+1,j)=space(i+1,j)+1;
                space(i,j)=space(i,j)-1;
                Savebagflag_vector(j)=0;
            end
        end
```