For office use only	Team Control Number	For office use only
T1 ________________	**1921429**	F1 ________________
T2 ________________		F2 ________________
T3 ________________	Problem Chosen	F3 ________________
T4 ________________	**D**	F4 ________________

2019

MCM/ICM

Summary Sheet

The Evacuation Plan for the Louvre

Summary

Visitors evacuation is crucial to the safety of public places. How to evacuate the visitors quickly and safely is the key to making an evacuation plan. Due to the complex structures of buildings and diversity of visitors, taking factors of many aspects into account is significant. When a threat occurs in a public place, an efficient and flexible plan prevents visitors from getting lost or stuck, avoiding congestion. By referring to cellular automata, PSO, genetic algorithm and multi-agent theory, the complex structure of the Louvre is taken as an example. Discuss how the numbers of exits and the existence of evacuation instructions will influence the evacuation. Then, do some further discussion on the effects of emergency personnel, 3D structure and bottlenecks. The main contents are as follows:

Firstly, we construct the evacuation model. Considering the differences in individual behaviors, visitors are divided into 3 categories (young and middle-aged people, non-young-middle-aged people and group visitors). The Agent theory and the cellular automata theory are used to construct the models of people and to grid the environment. One person is regarded as an independent agent who follows specific rules.

Secondly, we make an evacuation plan. The evacuation plan is represented by evacuation instructions. Apply genetic algorithm to get the best evacuation instructions that minimizes the overall evacuation time. Under these instructions, visitors can escape to the exit quickly and safely.

Lastly, we focus on programming simulation analysis. Assume there are four kinds of evacuation environments (single exit without guidance, single exit with guidance, two exits with guidance, and four exits with guidance) and simulate evacuation behavior of the visitors. Develop an evacuation plan and calculate the evacuation time. Thus, we know that the optimized evacuation plan can greatly reduce the overall evacuation time. In addition, take group visitors, cut-off routes, 3D structure evacuation, the entry of emergency personnel, bottlenecks into consideration and give suggestions.

Keywords: Agent, Cellular Automata, PSO, Genetic algorithm, Simulation

Contents

I. Introduction 2
1.1 Background 2
1.2 Our Tasks 2
1.2.1 Task 1 2
1.2.2 Task 2 2
1.2.3 Task 3 2
II. Analysis of the Problem 2
III. Basic assumptions 3
IV. Symbols 3
V. Model 3
5.1 The Basic Agent Theory 3
5.1.1 Two-Dimensional Modeling of Environmental Space 3
5.1.2 Individual Agent Attribute Model[2] 4
5.1.3 Agent's Decision 4
5.1.4 The Action of Agent 5
5.2 The Initial Attributions 6
5.2.1 The Initialization of the Evacuation Environment 6
5.2.2 The Initialization of People's Attribution 6
5.3 Simulation Results Analysis 7
5.3.1 Evacuation Process Analysis 7
5.3.2 Further Study of the Above Model. 10
5.4 Other Factors 13
5.4.1 The Factor of Threat 13
5.4.1 The Entry of Emergency Personnel 15
5.5 Analysis of Multi-floor Evacuation 15
5.6 Bottlenecks 16
5.6.1 Congestions at Interior Crossings 16
5.6.2 The Waiting Time at the Exits 17
VI. Sensitivity Analysis 19
VII. Evaluation of Model 19
7.1 Strength 19
7.2 Weakness 19
VIII. Conclusion 19
IX. References 20
X. Appendix 20

I. Introduction

1.1 Background

France is suffering an increasing number of terror attacks in recent years, which happened or will probably happen in many scenic spot, such as The Louvre. So how to evacuate visitors from the museum when attacked becomes a tough problem.

The Louvre is the world's largest art museum and a historic monument in Paris, France. In 2017, the Louvre was one of the world's most visited art museum, receiving 8.1 million visitors, including many kinds of people. They come from all over the world, speaking different languages. They travel alone or with a group. Even some of them are disabled.

There are four entrances of The Louvre. The main entrance is the Pyramid entrance, which is the most used public entrance of the museum. The others are the Passage Richelieu entrance, the Carrousel du Louvre entrance, and the Portes Des Lions entrance, which are reserved for groups and memberships of the museum. There are also other emergency exits which are only known by emergency personnel and museum officials in the museum. However, they are usually smaller and narrower than the four entrances, so safety is not guaranteed in these exits when widely used.

1.2 Our Tasks

1.2.1 Task 1

Design evacuation plans at the Louvre.

1.2.2 Task 2

Optimize the model with different situations, considering the diversity of visitors, the safety of the known and unknown available exits, the entering of the emergency personnel and the potential bottlenecks.

1.2.3 Task 3

Discuss how to implement and adapt the model for other large, crowded structures.

II. Analysis of the Problem

It is required to design a reasonable evacuation plan. It is too difficult for us to consider the actual structure of the Louvre directly, so we simplify the problem at first. We concentrate on the simplified single-floor plane sketch and analyze the evacuation situation. Then the factors which influence the evacuation such as guidance and threat points are gradually added to the simulation analysis. Next, we solve the evacuation problem of multi-floor, and finally we can get a feasible evacuation plan based on our model.

In addition, adjusting input parameters and environmental factors can check whether the model is suitable for other circumstances.

III. Basic assumptions

- No more tourists enter the museum once the museum is attacked.
- Attacks only occur in one place at the same time.
- Every normal adults have the same moving speed.
- The disabled, the old and children have the same moving speed.
- The positions of people are randomly distributed.
- The area covered by each tourist is more than 0.5m*0.5m and the number of tourists is less than the maximum capacity of the museum.
- Tourists move straightly per unit time.
- Tourists will listen to the guide of personnel.
- The simplified structures inside the museum are as follows:
- The personnel can open the emergency exits at any time if needed.

IV. Symbols

Symbols	Significance
A_i	Ages of evacuees
(x_i, y_i)	Individual agent location and coordinate
v_i	Speed of individual agents
$P_{i,j}^t$	The value of cell (x_i, y_j) at time *t*
$\boldsymbol{t}$	time
d_i	Instructions for $district_i$
$\boldsymbol{c_i}$	The congestion level

V. Model

5.1 The Basic Agent Theory

5.1.1 Two-Dimensional Modeling of Environmental Space

In this paper, two-dimensional modeling of environmental space[1] is mainly used. Model of cellular automata describes the state of the system itself and predicts the next time state. The cellular automata theory is used to grid the environment of The Louvre. As the model built on Cellular Automata Theory belongs to discrete model, so the objects perform discrete at any time. The model divides the continuous evacuation plane into a series of two-dimensional cells with definite boundary by using a certain size of grid. Each cell has only three states, either occupied by an agent, occupied by obstacles, or empty.

In the model, triangles, quadrangles and hexagons can be chosen as the two-dimensional cells. But in actual simulation, most of the models will choose square, which is not only convenient to display but also easy to observe. The museum's environment is complex. To appropriately simplify the calculation, square cells are used in this paper. The horizontal projection area of an individual determines the size of the grid. According to some collected data, the average size standard of the world's population is about 0.5m*0.5m.

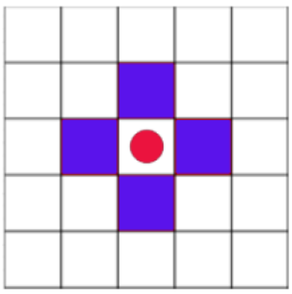
5.1.2 Individual Agent Attribute Model[2]

In the model, evacuees are regarded as agents. It is important to abstractly summarize individual behaviors into the model because evacuees have strong subjectivity and are affected by objective environment. Two factors are mainly considered in this model. First, the physiological characteristics are divided into two groups. Young and middle-aged people are one group; the elderly, children and disabled people are the other group (called non-young-and-middle-aged people). Second, the psychological characteristics are divided into the normal group and the panic group.
The attributes of each evacuee are abstracted as follows:

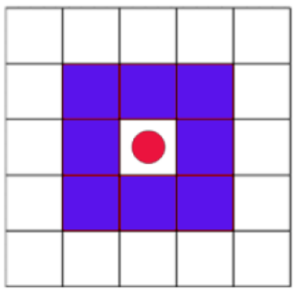
$$Agent\_Eva = [x_i, y_i, A_i, v_i]$$

5.1.3 Agent's Decision

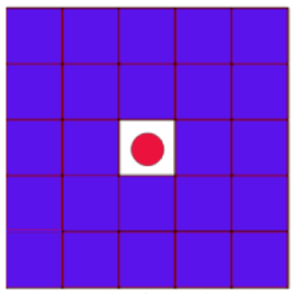
Build the decision-making module of each agent in the evacuation model by using the concept of neighborhood in cellular automata. According to cellular automata theory, the neighborhood rules must be determined before input in a cell. The common neighborhood models include Von Neumann neighborhood, Moore neighborhood and Extended Moore neighborhood. Three neighborhood grids are shown in the figure. In the picture, the red center circle represents the central cell and the blue squares represent its neighbors.



Von Neumann neighborhood



Moore neighborhood



Extended Moore neighborhood

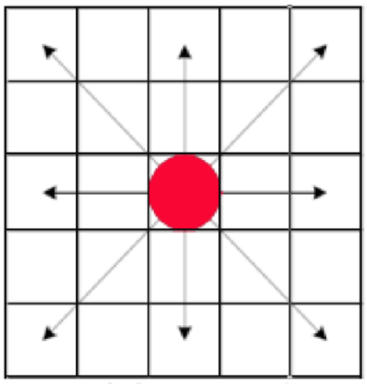
The values of cells are determined by the following formula at any time:

$$P_{i,j}^{t} = f(P_{i,j}^{t-1}, P_{i+1,j}^{t-1}, P_{i-1,j-1}^{t-1}, P_{i-1,j+1}^{t-1}, P_{i,j-1}^{t-1}, P_{i,j+1}^{t-1}, P_{i+1,j+1}^{t-1}, P_{i+1,j-1}^{t-1})$$

For young and middle-aged people, their actions are quick and flexible, and Moore neighborhood is used to analyze them. Assume that the young and middle-aged people move 2m per second. The direction of their movement during evacuation is shown in the figure

$P_{i-1,j}$		$P_{i,j+1}$		$P_{i+1,j}$
$P_{i-1,j}$		$P_{i,j}$		$P_{i+1,j}$
$P_{i-1,j-}$		$P_{i,j-1}$		$P_{i+1,j-}$

(a)The coordinate of young and middle-aged people's movement.

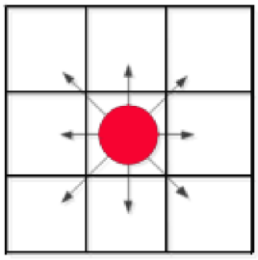


(b) The way that young and middle-aged people move.

For the old, children and the disabled, they lack flexibility because of their inconvenience in movement. Therefore, they move slowly. Assume that they move 1m per second. The direction is the same as that of young and middle-aged people.

$P_{i-1,j+1}$	$P_{i,j+1}$	$P_{i+1,j+1}$
$P_{i-1,j}$	$P_{i,j}$	$P_{i+1,j}$
$P_{i-1,j-1}$	$P_{i,j-1}$	$P_{i+1,j-1}$

(a)The coordinate of the old, children and the disabled people's movement.



(b)The directions that the old, children and the disabled people move.

5.1.4 The Action of Agent

In this model, moving targets for each individual are figured out according to their own attributes by using algorithms. The fitness values of part global optimized values, and their own ability of decision-making ultimately decide which direction to move.

(1) Check if there are obstacles in the next possible eight cells of each agent, whether they are occupied or they are out of the edge of the region, and exclude unavailable cells. If there are no available cells around, the agent just stays where he is.
(2) Calculate the distance from the cell to a random point in a given number of safety zones (using the distance formula between two points) among the remaining optional cells, and choose the nearest cell.
(3) Agent moves to the corresponding cell.
(4) Check if the agent arrives in the safety zone, if not, repeat the process until the agent reaches the safety zone.

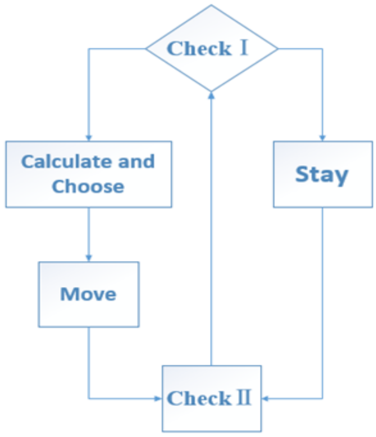


Fig5.1 The process of agent's action

5.2 The Initial Attributions

5.2.1 The Initialization of the Evacuation Environment

According to the related documents and the information mentioned in the problem, the simplified plane sketch of the Louvre is as follows:

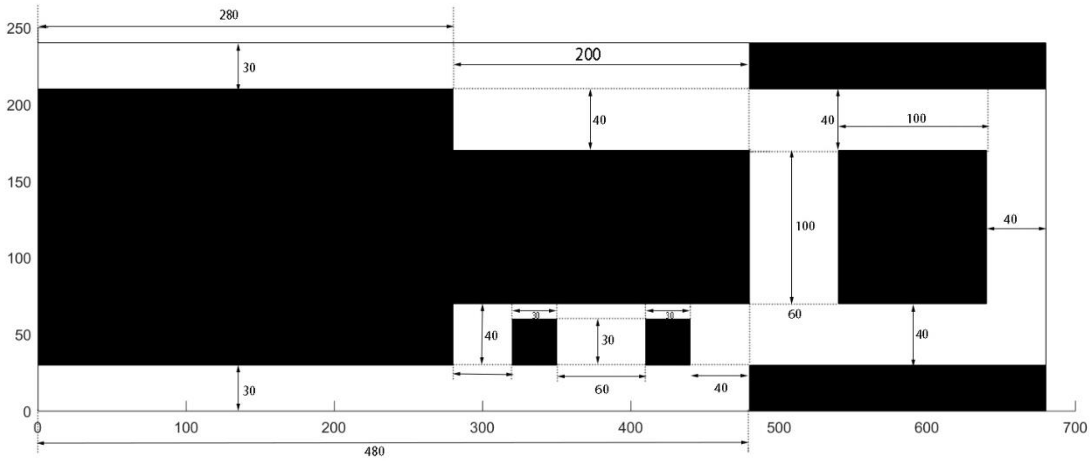


Fig5.2.1 The initialization of the evacuation environment

Black represents obstacles, white represents available areas and green represents exits.

5.2.2 The Initialization of People's Attribution

According to the information in the problem, 8.1 million visitors were received in 2017, with an average of 20,000 visitors per open day. Suppose that at a point in time, the number of people on each floor is about 1300(adjustable, refer to codes in appendix 1 for details). Divide all people into two groups, normal young and middle-aged people, which account for 75% of the total, and others, which account for 25% of the total. The area covered by each person is about 0.5m*0.5m.

In the museum evacuation, evacuees are randomly situated. The initial situation is as follows:

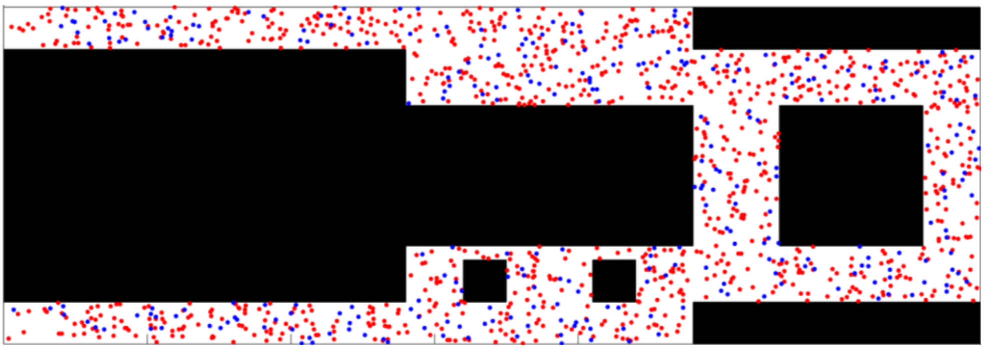


Fig5.2.1 Initial situation

Red dots represent normal young and middle-aged people, blue dots represent other people (including the old, children, disabled people, etc.)

5.3 Simulation Results Analysis

5.3.1 Evacuation Process Analysis

(1) Assume there is only one exit (its location is [240,250]×[5,10]) for a certain floor, and people evacuate with no guidance.

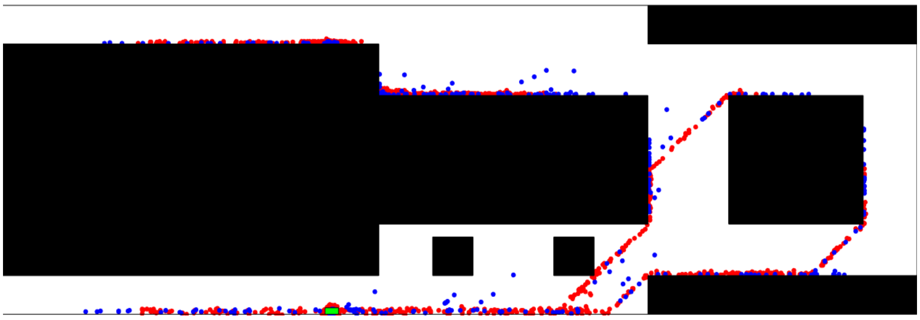


Fig5.3.1 Evacuation situation(t=50s)

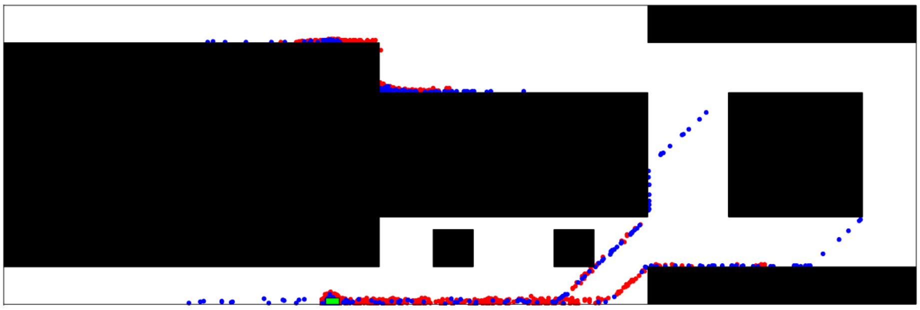


Fig5.3.2 Evacuation situation(t=120s)

From the above pictures, we can see that the young and middle-aged people move quickly. When the evacuation time is 120 seconds, they are the firsts to arrive at the exit during the evacuation process. Others move slowly and arrive late at the exit. However, people who are far away from the exit or are difficult to find the exit tend to gather, which may be caused by panic. It shows that the existence of obstacles has certain interference and influence on evacuation.

Through simulation calculation, when the number of evacuees is 1300, 85% of the people arrive at the exit in 205 seconds, and 15% of the people gathered because they are not able to find the exit due to panic or other reasons. In the next part, guidance will be added in emergencies to explore the influence of artificial guidance on evacuation.

(2) Assume there is only one exit for a certain floor, and people evacuate with guidance. Here's an explanation for artificial guidance.

First, the sequence of evacuation instructions refers to the set of instructions of the escape direction, which can drive people in a certain area to move in a certain direction. Eight evacuation directions are defined in this paper. They are up, down, left, right, left up, left down, right up and right down. These instructions are represented by 1, 2, 3, 4, 5, 6, 7, 8 respectively in the code and by green arrows in different directions in the following figures. Due to the psychological roles and behavioral inertia of people, the available space is divided into N blocks, labeled successively, and the evacuation instructions of N blocks are stored in a 1*N matrix.

$$Instructions = [d_1, d_2, \dots, d_n]$$
$$d_i = 1,2, \dots 8$$

Considering the museum evacuation environment and characteristics of the algorithm, N equals 53. The specific partition is as follows:

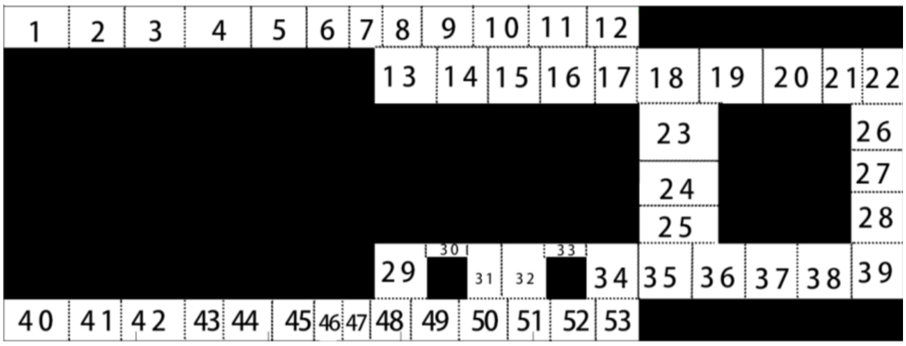


Fig5.3.3 Specific partition

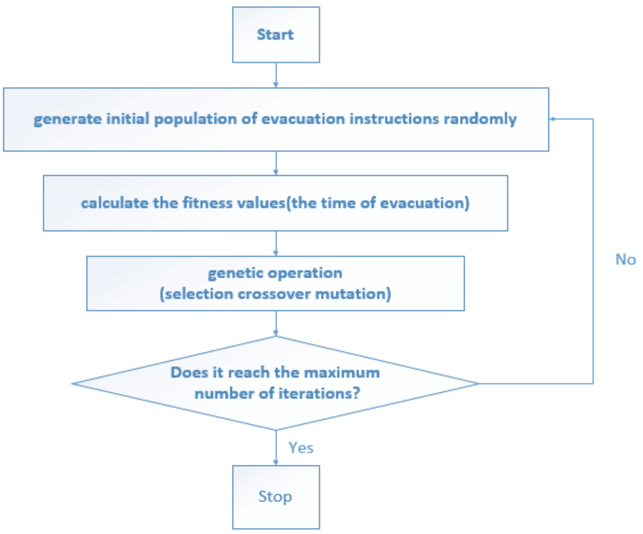


Fig5.3.4 The process of genetic algorithm[3]

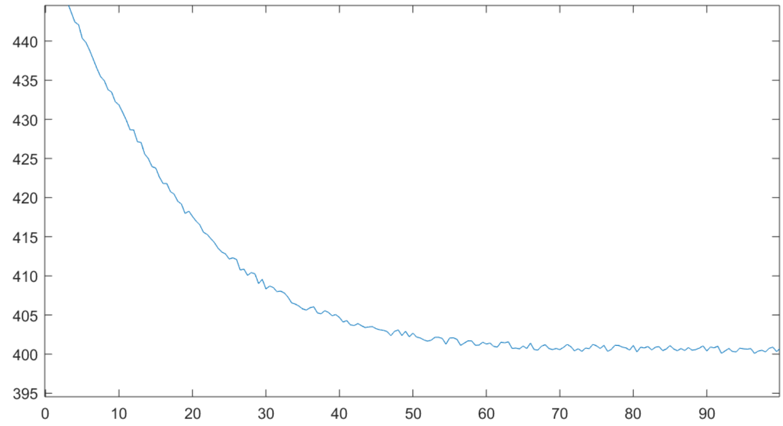


Fig5.3.5 The convergence of evacuation time

Optimize the sequence of evacuation instructions by genetic algorithm. The total time of evacuation is the fitness value. The best instructions are as follows:

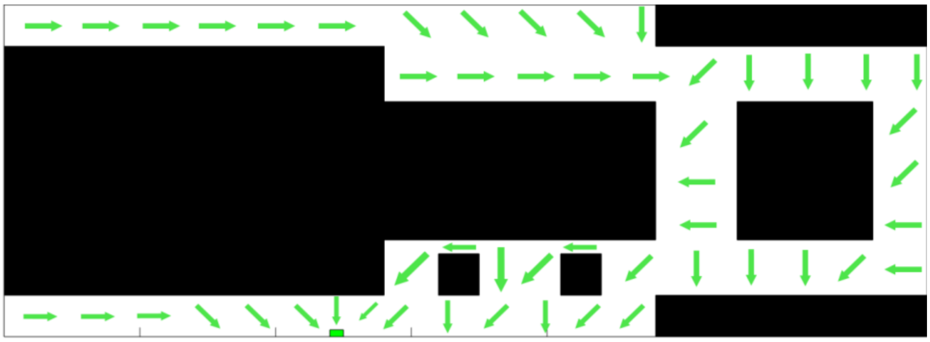


Fig5.3.6 Instructions 1

Improve situation mentioned in last subsection by adding guidance to gathering areas where people cannot escape quickly and safely. The improved simulation results are as follows:

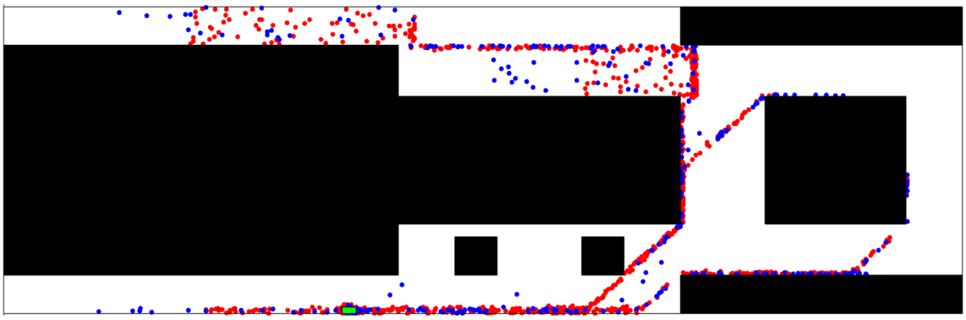


Fig5.3.7 Evacuation situation(t=50s)

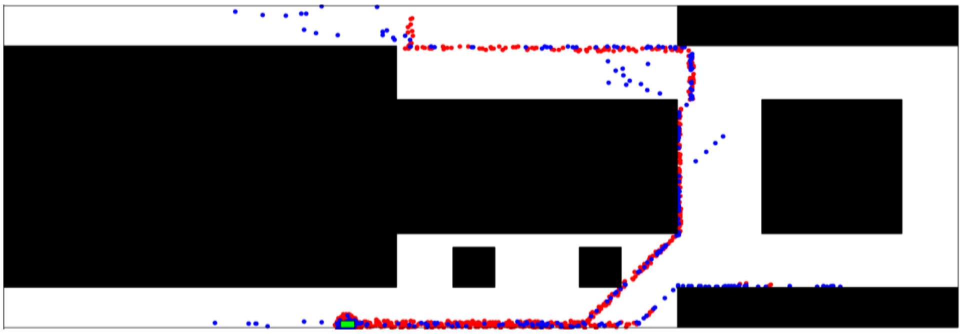


Fig5.3.8 Evacuation situation(t=120s)

As can be seen from the above figures, the phenomenon of people gathering in the previous part does not appear again with guidance. The young and middle-aged people move quickly when the evacuation time is 100 seconds. They are the firsts to arrive at the exit during the evacuation process. Others move slowly and arrive late at the exit. And people who are far away from the exit or are difficult to find the exit can also escape to exit after a while.

Through simulation calculation, when the number of evacuees is 1300, 85% of the people arrive at the exit in 170 seconds, and the total time for all people to escape to the exit is 320 seconds. Therefore, guidance is very important for people who are far away from the exit and are difficult to fine the exit in evacuation. So it is suggested that in case of emergency, people who are far away from the exit and are difficult to find the exit should be given guidance in time to escape, such as flashing green arrow lights, broadcasting in various languages, or using large screen inside the museum to display the guidance information.

5.3.2 Further Study of the Above Model.

(1) Assume there is at least one emergency exit (its location is [270,280]×[235,240]) that can open in the area where panic may occur (Staff are in charge of these entrances and exits). Open this exit in case of an emergency. So that the exit and the original exit can evacuate people at the same time.

Use the algorithm in the previous part to get the optimal guidance instruction.

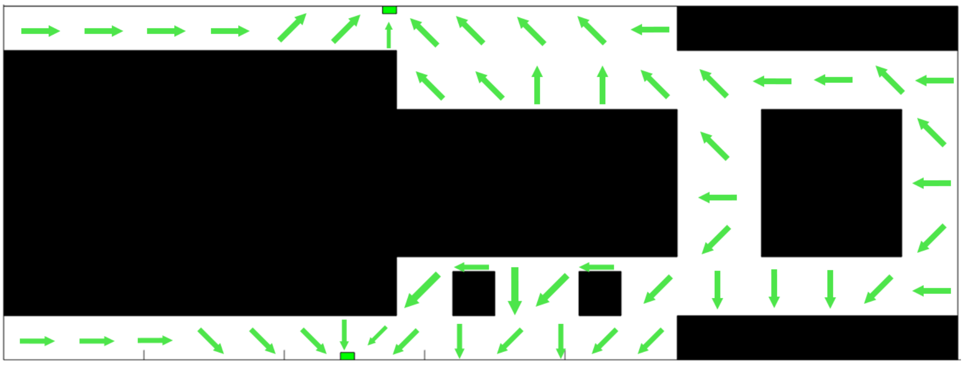


Fig5.3.9 Instructions 2

After adding this guidance, the simulation results are as follows:

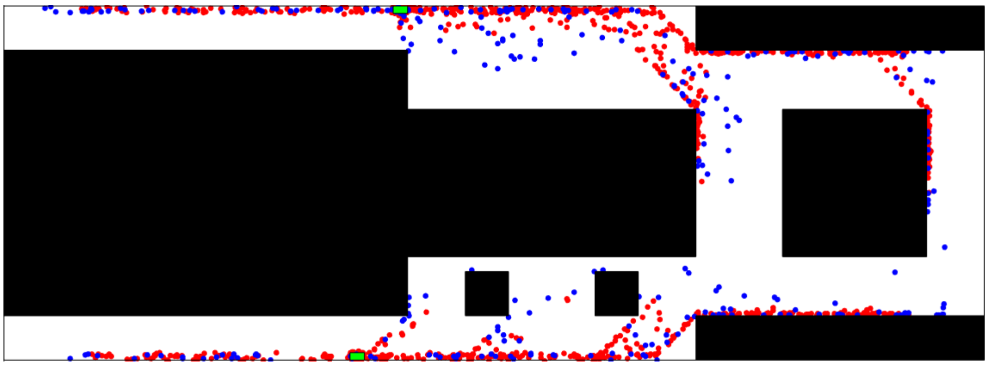


Fig5.3.10 Evacuation situation(t=30s)

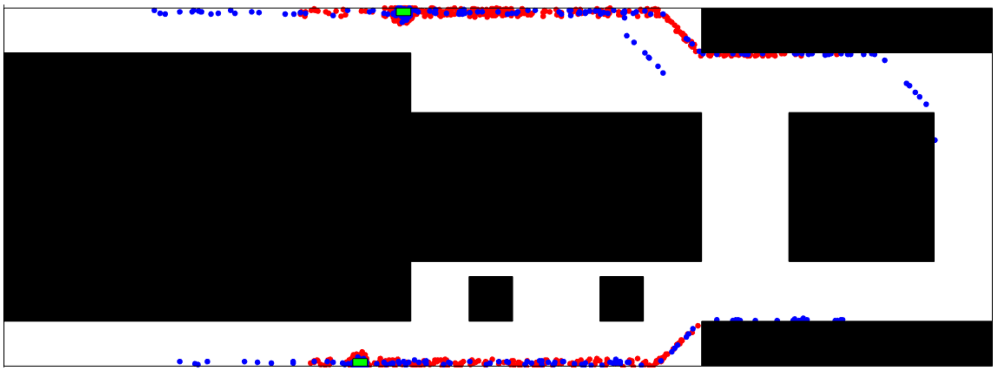


Fig5.3.11 Evacuation situation(t=100s)

From the above pictures, it can be seen that the phenomenon of people gathering does not occur, and the overall evacuation efficiency is greatly improved. Through simulation calculation, when the number of evacuees is 1300, 85% of the people arrive at the exit in 82 seconds, and the total time for all people to escape to exit is 151 seconds. It shows that opening one more exit is much better for evacuation. Therefore, it is suggested that in case of emergencies, the existing available exits should be opened to those areas which are far away from the original exits or where it is difficult to find the original exits.

(2) Specific analysis of the Louvre's -2nd floors with 4 existing exits (their locations are $[240,250]\times[0,5]$, $[270,280]\times[235,240]$, $[480,485]\times[100,110]$, $[470,480]\times[235,240]$ respectively) which is shown in the following figure.

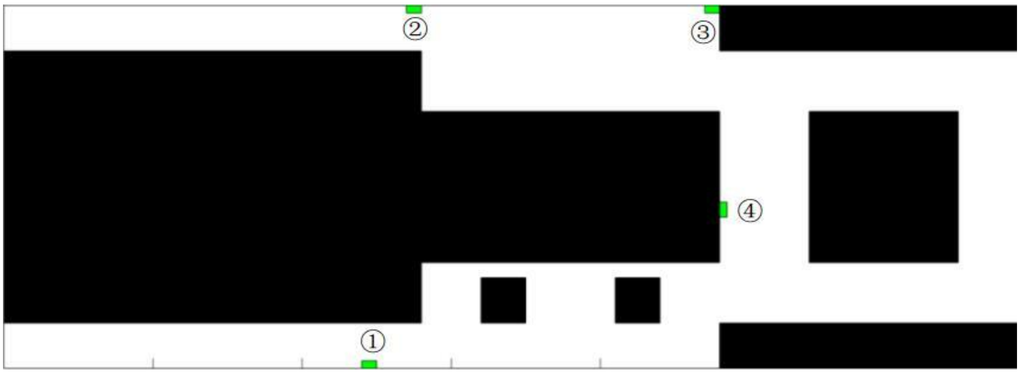


Fig5.3.12 4 existing exits

Similarly, get the optimal guidance instruction

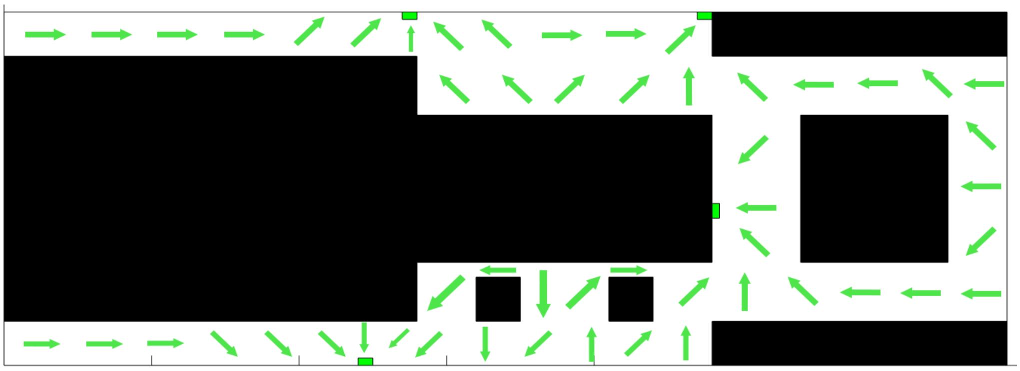


Fig5.3.13 Instructions 3

After the simulation, the results are as follows:

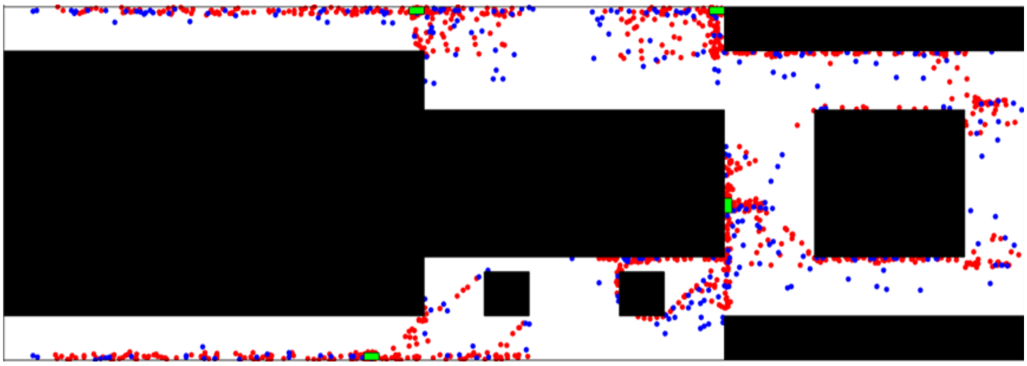


Fig5.3.14 Evacuation situation(t=15s)

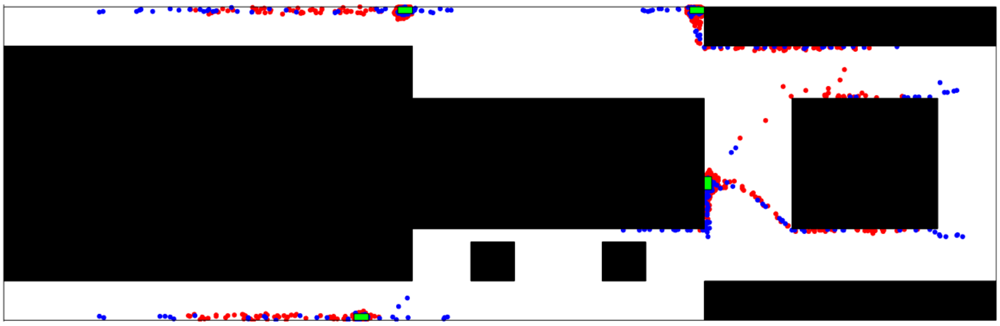


Fig5.3.15 Evacuation situation(t=60s)

As can be seen from the pictures above, the escape efficiency is improved. When the number of evacuees is 1300, 85% of the people arrive at the exits in 23 seconds, and all people arrive at the exits in 97 seconds. Therefore, guidance is very important for evacuation in areas where it is difficult to determine the optimal route. It also ensures that the evacuation time is as small as possible and avoids people choosing the non-optimal route. So it is suggested that in case of emergency, people should be given guidance in time to escape, such as flashing green arrow lights, broadcasting in various languages, or using the large screen to display guidance information.

(3) Additionally, take the group visitors into consideration. Assume that there are 50 groups on each floor, which are composed of nine visitors on average, and each group move together. Here comes a problem. Will these groups interfere with the individuals' evacuation? In this model, group visitors are regarded as special individuals of large size, who move in the same way following the action rules mentioned in the Chapter 5.1.4.

Assume their speed is 1.5m/s. The initial situation is as follows:

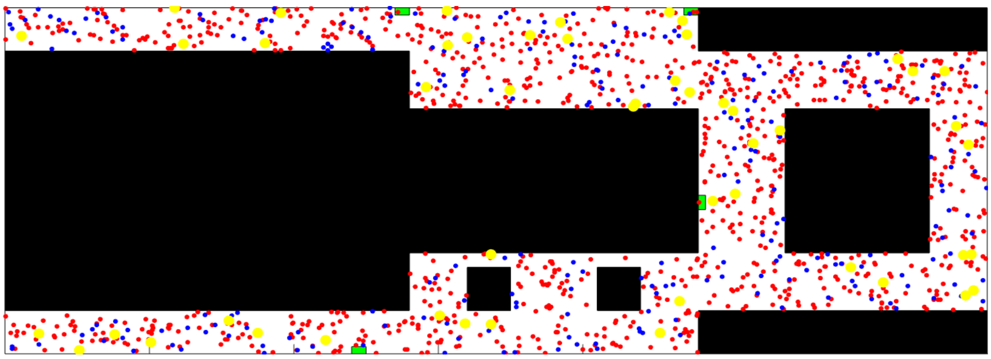


Fig5.3.16 Initial situation

Yellow dots represent group visitors, red dots represent normal young and middle-aged people, blue dots represent other people (including the old, children, disabled people, etc.) and green rectangles represent the exits.

Still use the algorithm in the previous part to get the optimal guidance instruction. The result is as follow:

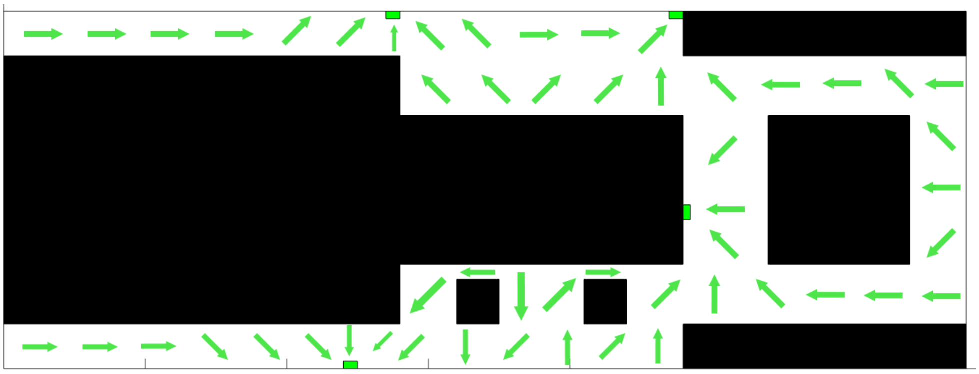


Fig5.3.17 Instructions 3

It is a little different of the model with no group visitors mentioned above. It shows that the existence of group visitors will influence the evacuation strategy.
The simulation results are as follows:

Fig5.3.18 Evacuation situation(t=30s)

When the number of evacuees is 1300, 85% of the people arrive at the exits in 43 seconds, and all people arrive at the exits in 135 seconds. So the evacuation time will be extended because of the existence of group visitors. Therefore, staff can flexibly make some effective plans according to the number of group visitors at that time.

5.4 Other Factors

5.4.1 The Factor of Threat

Assume that a threat has altered or removed the segments of possible routes to exits, and could be regarded as new obstacle. For example, if there is a threat at [580,585]×[170,210] (It is just an example. The coeds in the appendix can be changed in other situations.) The light blue pattern in the picture refers to the threat site, which cut off the route completely. Now the initial situation is as follows:

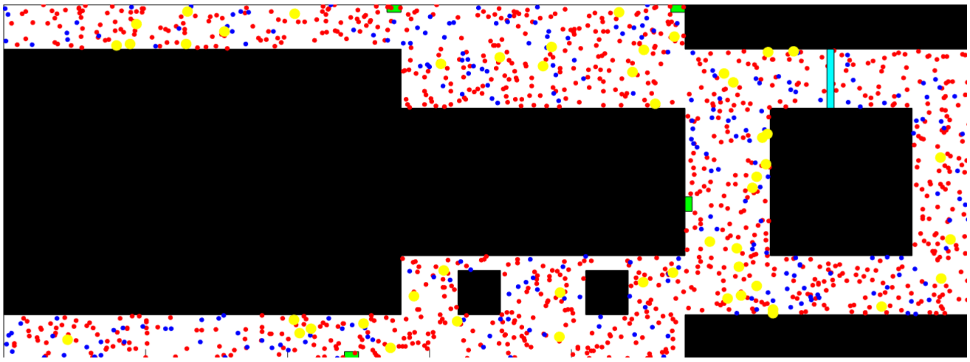


Fig5.4.1 Initial situation

Similarly, calculate the optimal guidance instructions by genetic algorithm in this case. The result is as shown in the figure below.

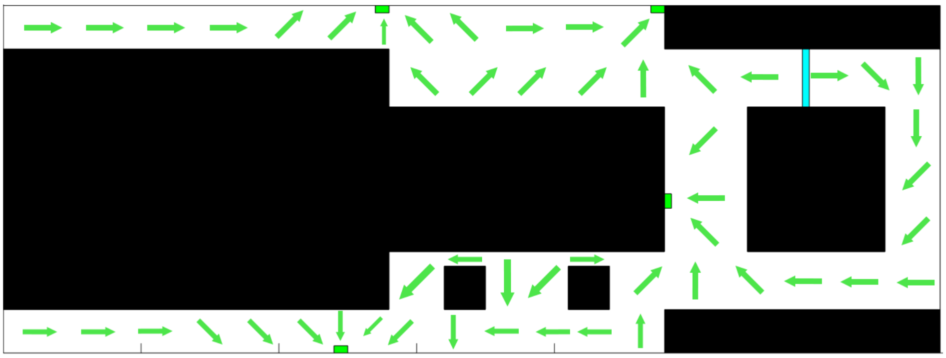


Fig5.4.2 Instructions 4

Analyze the instructions at this time and compare it with the instructions when no threat occurs. It shows that the original optimal escape route on the right side of the threat point is cut off and the instructions for this area are updated. At the same time, in order to alleviate the evacuation pressure of the 4th exit, areas which are directed to the 4th exit by instructions were optimized by algorithm and changed to another exit.

The simulation evacuation is as follows:

Fig5.4.1 Evacuation situation(t=30s)

It can be seen that the algorithm can update the guidance instructions when some routes are cut off and flexibly change the evacuation instructions to achieve the shortest overall evacuation time, taking the flow of people at each exit into account.

5.4.1 The Entry of Emergency Personnel

There are two key factors that affect the time when emergency personnel arrive at a threat point. One is the distance d between the entrance of emergency personnel and the threat point; the other is the congestion level c of the entrance.

$$T = g(d, c)$$

Assume the rules of emergency personnel' movement are the same as those of tourists in previous models, but the speed is 4m/s.

Assuming that there is only one rescue team (consisting of 10 people) at the same time. Calculate the time when all emergency personnel from different entrances arrive at the threat point, which is represented by t_1, t_2, t_3, t_4.

Position	Time
①	t_1=103s
②	t_2=83s
③	t_3=42s
④	t_4=30s

Then compare the congestion level (the number of people per unit area), which is represented by c_1,c_2,c_3,c_4.

Position	Congestion
①	c_1=8/m^2
②	c_2=5.5/m^2
③	c_3=4.7/m^2
④	c_4=7/m^2

Get the most suitable rescue entrance by Analytic Hierarchy Process. Regard "selection of entrance" as the goal, the "time for all emergency personnel to reach the threat point from the entrance" and "congestion level" as the criterion, and the "position (1), (2), (3) and (4) as the alternatives. The specific analysis process is relatively simple and will not be described in detail here.

Get the best rescue entrance ((3) entrance) and the route (the opposite direction to the instructions mentioned in the picture Fig5.4.2) by AHP. The rescue simulation results are as follows:

If there are additional special entrances that can allow emergency personnel into the museum, staff can also use AHP to get the best entrance.

5.5 Analysis of Multi-floor Evacuation

According to the overall structure of the Louvre, it can be simplified as follows:

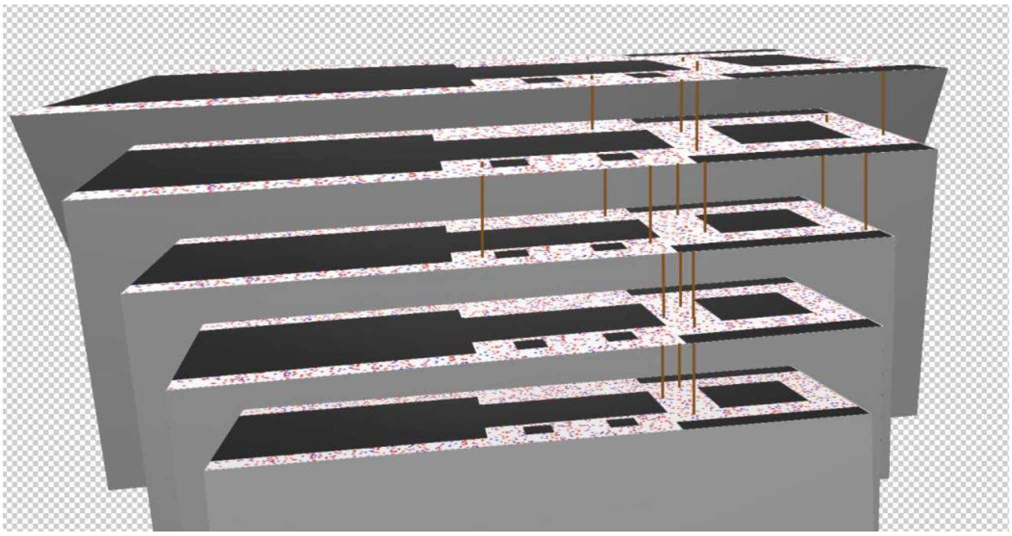


Fig5.5.1 3D structure

The brown tubes represent the simplified stairs.

Though the multi-floor problem seems very complicated, in fact it is only a combination of the simple models discussed earlier. Only at the stair entrance will there be entry and exit problems.

According to the data, there is one exit on the 0th floor and four exits on the 2nd floor. Build three-dimensional model. The staircase entrance on each floor can be regarded as the safety zone in the model above. The visitors who reach other floors through the staircase can be regarded as the non-evacuated people on this floor and follow the evacuation instructions as well as other visitors on this floor. Assume that there are 1,300 tourists on each floor, totaling 6,500. It takes an average of 15 seconds for each of them to pass through each staircase. Each staircase has a maximum carrying capacity of 80 people. The entrance of the staircase and nearby regions are regarded as safety zones with an area of about 5m*10 m. The genetic algorithm mentioned above can still be used to calculate the optimal evacuation instructions.

5.6 Bottlenecks

5.6.1 Congestions at Interior Crossings

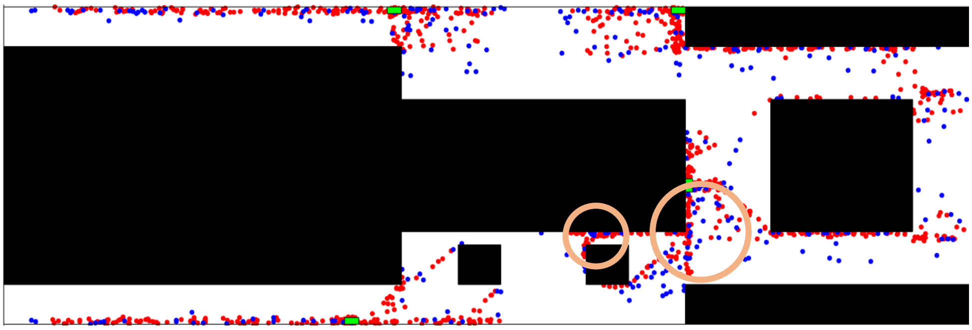


Fig5.6.1 Potential bottlenecks

From the analysis of the simulation results, it can be seen that queuing and congestion may probably occur in the circled areas because many routes cover these areas and there are many optimal escape routes pass through these places.

If the shape of the black part (the structure of the site) can be adjusted, staff can change the layout of the museum in the crowded areas shown above, increasing the width of these areas to facilitate evacuation. If it is not adjustable, it is suggested to broadcast voice messages in crowded areas to remind tourists to stay calm, follow evacuation instructions, and pass through these areas safely and cautiously, so as to avoid reducing evacuation speed due to the panic caused by crowding.

5.6.2 The Waiting Time at the Exits

In 2018, the total number of visitors is 10.3 million, averaging 20-30,000 people per day. Assumed that the number of people on each floor at some point is 1300, and the total number of people in the museum is 6500.

According to Affluence, the waiting time of several main exits is counted every 10 minutes. The tables are as follows:

Time	Pyramide	Pyramide e-bullet	Carrousel	Time	Pyramide	Pyramide e-bullet	Carrousel
9:10	10	10	5	11:30	5	5	5
9:20	15	15	5	11:40	5	5	5
9:30	15	5	10	11:50	5	5	5
9:40	5	5	5	12:00	5	5	5
9:50	5	5	5	12:10	5	5	5
10:00	5	5	10	12:20	5	5	10
10:10	5	5	10	12:30	5	5	5
10:20	15	5	closed	12:40	5	5	5
10:30	15	5	closed	12:50	5	5	5
10:40	15	5	closed	13:00	5	5	5
10:50	20	5	closed	13:10	5	5	5
11:00	10	5	closed	13:20	5	5	5
11:10	5	5	5	13:30	5	5	5
11:20	5	5	5	13:40	5	5	5

By analyzing the table, it can be assumed that all the people obey the order, the evacuation rate v1 of the normal exit is 240 people per minute when the queue passes through the exit in turn, and the evacuation rate v2 of the emergency exit is 160 people per minute.

In practical analysis, the influence of congestion level on evacuation rate should also be considered, which is represented by the density of people. According to some documents, there is an empirical formula between the density of people and the speed of people's movement.

$$v = 1.1\rho^{-0.7954}$$

v is the speed of people's movement
ρ is the density of people

Calculate the time when people arrive at the nearby areas of the four exits with guidance and get the form of the situations with congestion and without congestion.

(1) Without congestion:

Number of people	Coordinate of people	Number of arrival safety zone	Time of reaching the area within 50m^2 around the safety zone/s	Time of reaching the safety zone/s	Difference value of the times/s
1	(276,23.5)	①	13	19	6
2	(432,37.5)	④	18	23	5
3	(637,72)	②	30	35	5
4	(16.5,24)	①	21	28	7
5	(303,176)	③	17	23	6
6	(650,71.5)	④	31	36	5
7	(298,28)	①	12	16	4
8	(320,220)	②	15	21	6
9	(550,60)	③	18	25	7
10	(670,36)	④	34	39	5

The average time of passing through the nearby areas of safety zone is about 6s.

(2) With congestion:

Number of people	Coordinate of people	Number of arrival safety zone	Time of reaching the area within 50m^2 around the safety zone/s	Time of reaching the safety zone/s	Difference value of the times/s
1	(462,36)	④	26	45	19
2	(450,37.5)	④	23	43	20
3	(237,18)	①	6	26	20
4	(8,26)	①	28	43	15
5	(341,186)	④	9	25	16
6	(310,223)	①	15	33	18
7	(471,203)	①	10	27	17
8	(379,20)	①	18	35	17
9	(650,37)	②	24	42	18
10	(3,236)	②	35	54	19

The average time of passing through the nearby areas of safety zone is about 19s. So the efficiency of situations without congestion is about 3 times higher than that of situations with congestion.

VI. Sensitivity Analysis

Adjust the parameters of the initial number of people in single-exit model N_1, multi-exit model N_2 and multi-exit model with threat point N_3. The changes of parameters ranges from -10% to 10%. The percentage of evacuation time change is shown in the table below.

	N_1（±10%）	N_2（±10%）	N_3（±10%）
Change	±4.2565%	±.9127%	±2.4681%

VII. Evaluation of Model

7.1 Strength

The model is based on evacuation theory, learns from cellular automated, PSO, genetic algorithm and multi-agent theory, and put forward the concept of evacuation instruction. The Louvre's complex structure is decomposed into many small modules to manage the known and unknown entrances and exits flexibly, and can be adjusted flexibly in actual situations (actual terrain, number of groups, threat level, threat point, etc.). The algorithm combines genetic algorithm and PSO algorithm and improve both of them, preventing local convergence, and solving the optimal evacuation instructions quickly. So it can evacuate people safely and quickly.

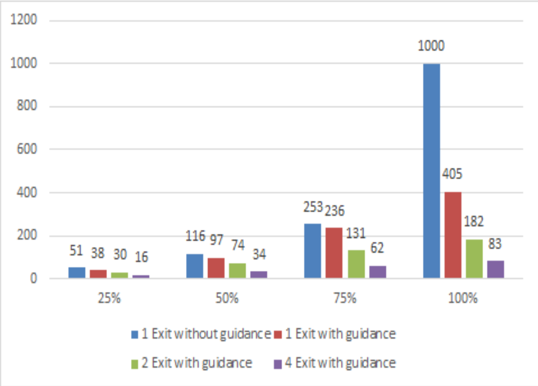
In addition, the model is flexible and universal. If it is applied to other buildings, the optimal evacuation route can be obtained by adjusting the relevant parameters (obstacles, available areas, site size, exit number location, and number of people in the site) according to the model provided in this paper.

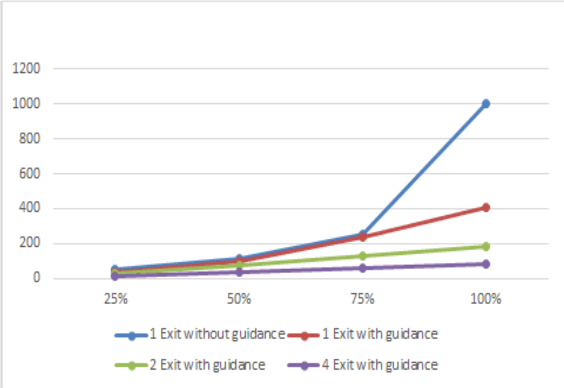
7.2 Weakness

(1) We did not take the fact that panic people do not follow the guidance and affect the normal evacuation of others into account.
(2) We did not take the fact that injuries are not able to move into account.
(3) We did not consider the situations that many threats happen at the same time.

VIII. Conclusion

In this paper, a flexible and adjustable model is constructed, and the factors influencing the evacuation plan are analyzed from many aspects. Also, the effect and possible problems of evacuation in four situations are simulated and analyzed.





At the same time, some evacuation suggestions are given to museum staff, which can be flexibly adjusted and applied to other more complex sites according to the actual situation.

IX. References

[1] Evacuation Simulation in Narrow Passage Under Fire Scenario Based on Cellular Automaton[J]. NATURAL SCIENCE, Vol. 46 No.8, Aug. 2018.
[2] Research on Safety Evacuation Route of New Ant Colony Optimization Based on Agent System[D]. Nanchang Hong Kong University, Nanchang, China, June, 2018
[3] Research on Optimization of Multi-exit Evacuation Path Problem using Genetic Algorithm

X. Appendix

10.1 Appendix 1

```
function [x,y,pop]=initialpop(m,n)
%a=1 for adults,a=2 for the olds,kids and the disabled
%m is the number of adluts vistors on this floor
%n is the number of the olds,kids and disabled vistors on this floor
%pop=[position_x,position_y,age,reaction,speed,direction]
%assume that Louvre is of 680m*240m size
X=randperm(680*2,1360);
Y=randperm(240*2,480);
%create adults' group
x=[];
y=[];
%% spaces that are not available
xv1=[0;280;280;0;0];
yv1=[30;30;210;210;30];
xv2=[280;480;480;280;280];
yv2=[70;70;170;170;70];
xv3=[540;640;640;540;540];
yv3=[70;70;170;170;70];
xv4=[480;680;680;480;480];
yv4=[210;210;240;240;210];
xv5=[480;680;680;480;480];
yv5=[0;0;30;30;0];
xv6=[320;350;350;320;320];
yv6=[30;30;60;60;30];
xv7=[410;440;440;410;410];
yv7=[30;30;60;60;30];
% plot(xv1,yv1)
% rectangle('Position',[0 30 280 180],'FaceColor', 'k')
% hold on
% rectangle('Position',[280 70 200 100],'FaceColor', 'k')
% rectangle('Position',[540 70 100 100],'FaceColor', 'k')
% rectangle('Position',[480 210 200 30],'FaceColor', 'k')
% rectangle('Position',[480 0 200 30],'FaceColor', 'k')
% rectangle('Position',[320 30 30 30],'FaceColor', 'k')
% rectangle('Position',[410 30 30 30],'FaceColor', 'k')
% rectangle('Position',[0 0 680 240])
%% set the points randomly at the beginning
i=0;
while i<m+n
    x0=X(ceil(1360*rand(1)))/2;
```

```
y0=Y(ceil(480*rand(1)))/2;
    in=inpolygon(x0,y0,xv1,yv1) ...
        || inpolygon(x0,y0,xv2,yv2)...
        || inpolygon(x0,y0,xv3,yv3)...
        || inpolygon(x0,y0,xv4,yv4)...
        || inpolygon(x0,y0,xv5,yv5)...
        || inpolygon(x0,y0,xv6,yv6)...
        || inpolygon(x0,y0,xv7,yv7);

    %1 for in,0 for out
    if ~in
        x=[x;x0];
        y=[y;y0];
        i=length(x);
    end
end
% for k=1:m
%     plot(x(k),y(k),'.g','Markersize',6)
%     hold on
% end
% for k=m:m+n
%     plot(x(k),y(k),'.b','Markersize',6)
%     hold on
% end
%r for adults,blue for the olds,kids and disabled vistors on this floor
%% set the innitial pop
pop=zeros(m+n,4);
for i=1:m
    pop(i,:)=[x(i),y(i),1,0];
    %pop=[position_x,position_y,age,reaction,speed,direction]
end
for i=m+1:m+n
    pop(i,:)=[x(i),y(i),0,0];
    %pop=[position_x,position_y,age,reaction,speed,direction]
end
```

10.2 Appendix 2

```
function flag=checkavailability(x0,y0,x,y)
flag=1;%means this point is available for others to move on
xv1=[0;280;280;0;0];
yv1=[30;30;210;210;30];
xv2=[280;480;480;280;280];
yv2=[70;70;170;170;70];
```

```
xv3=[540;640;640;540;540];
yv3=[70;70;170;170;70];
xv4=[480;680;680;480;480];
yv4=[210;210;240;240;210];
xv5=[480;680;680;480;480];
yv5=[0;0;30;30;0];
xv6=[320;350;350;320;320];
yv6=[30;30;60;60;30];
xv7=[410;440;440;410;410];
yv7=[30;30;60;60;30];
for k=1:length(x)
    %       if x0==x(k)&&y0==y(k)|| inpolygon(x0,y0,xv1,yv1) ...
    %             || inpolygon(x0,y0,xv2,yv2)...
    %             || inpolygon(x0,y0,xv3,yv3)...
    %             || inpolygon(x0,y0,xv4,yv4)...
    %             || inpolygon(x0,y0,xv5,yv5)...
    %             || inpolygon(x0,y0,xv6,yv6)...
    %             || inpolygon(x0,y0,xv7,yv7)...
    %             || x0>680 ||y0>240
    if x0==x(k)&&y0==y(k)
    flag=0;
    break
    end
end
if inpolygon(x0,y0,xv1,yv1)
    flag=0;
%       break
elseif inpolygon(x0,y0,xv2,yv2)
    flag=0;
%       break
elseif inpolygon(x0,y0,xv3,yv3)
    flag=0;
%       break
elseif inpolygon(x0,y0,xv4,yv4)
    flag=0;
%       break
elseif inpolygon(x0,y0,xv5,yv5)
    flag=0;
%       break
elseif inpolygon(x0,y0,xv6,yv6)
    flag=0;
%       break
elseif inpolygon(x0,y0,xv7,yv7)
    flag=0;
```

```
%       break
elseif x0>680 ||y0>240
      flag=0;
%       break
end
```

10.3 Appendix 3

```
function [xnew,ynew]=updatepos(xi,yi,x,y,l)
positionnext=[xi-l,yi+l;xi-l,yi;xi-l,yi-l;xi,yi+l;xi,yi-l;xi+l,yi+l;xi+l,yi;xi+l,yi-l;];
flagall=[];
for k=1:8
      flag=checkavailability(positionnext(k,1),positionnext(k,2),x,y);
      flagall=[flagall;flag];
end
% xnew=[];
% ynew=[];
%% randomly move
% avapos=find(flagall==1);
% numchos=randperm(length(avapos));
% xnew=positionnext(avapos(numchos(1)),1);
% ynew=positionnext(avapos(numchos(1)),2);
%% move towards the safe zone
avapos=find(flagall==1);
if xi>280&&xi<490&&yi>170&&yi<210
      if find(avapos==7)
      xnew=positionnext(7,1);
      ynew=positionnext(7,2);
      elseif find(avapos==8)
      xnew=positionnext(8,1);
      ynew=positionnext(8,2);
         elseif find(avapos==5)
      xnew=positionnext(5,1);
      ynew=positionnext(5,2);
%       elseif find(avapos==5)
%       xnew=positionnext(7,1);
%       ynew=positionnext(7,2);
      end
elseif xi>0&&xi<290&&yi>210&&yi<240
       if find(avapos==7)
      xnew=positionnext(7,1);
      ynew=positionnext(7,2);
      elseif find(avapos==8)
      xnew=positionnext(8,1);
```

```
      ynew=positionnext(8,2);
         elseif find(avapos==5)
      xnew=positionnext(5,1);
      ynew=positionnext(5,2);
       end
elseif length(avapos)==0
      xnew=xi;
      ynew=yi;
else
      safepoint=[240+floor(rand(1)*10),floor(5*rand(1))];
      distance=zeros(length(avapos),1);
      for i=1:length(avapos)
            distance(i)=norm(positionnext(avapos(i),:)-safepoint);
      end
      [a,b]=min(distance);
      xnew=positionnext(avapos(b),1);
      ynew=positionnext(avapos(b),2);
end
```

10.4 Appendix 4

```
function pop=checksafety(pop,xsafe,ysafe)

flag=inpolygon(pop(:,1),pop(:,2),xsafe,ysafe);
pop(flag,:)=[];
%set safe zone location
% xsafe=[240;250;250;240;240];
% ysafe=[0;0;5;5;0];
```

10.5 Appendix 5

```
clear;clc;
%% set the initial pop
m=1000;%m is the number of adluts vistors on this floor
n=300;%n is the number of the olds,kids and disabled vistors on this floor
[x,y,pop]=initialpop(m,n);
%plot the unavailable spaces
rectangle('Position',[0 30 280 180],'FaceColor', 'k')
hold on
rectangle('Position',[280 70 200 100],'FaceColor', 'k')
rectangle('Position',[540 70 100 100],'FaceColor', 'k')
rectangle('Position',[480 210 200 30],'FaceColor', 'k')
rectangle('Position',[480 0 200 30],'FaceColor', 'k')
```

```
rectangle('Position',[320 30 30 30],'FaceColor', 'k')
rectangle('Position',[410 30 30 30],'FaceColor', 'k')
rectangle('Position',[0 0 680 240])
%% set the safe zone
xsafe1=[240;250;250;240;240];
ysafe1=[0;0;5;5;0];
rectangle('Position',[240 0 10 5],'FaceColor', 'g')
% xsafe2=[270;280;280;270;270];
% ysafe1=[235;235;240;240;235];
% rectangle('Position',[270 235 10 5],'FaceColor', 'g')
%%plot initial pop
for k=1:m
    plot(x(k),y(k),'.r','Markersize',10)
    hold on
end
for k=m+1:m+n
    plot(x(k),y(k),'.b','Markersize',10)
    hold on
end
hold off
%red for adults,blue for the olds,kids and disabled vistors on this floor
t=1;% time variable
%% when an emergency occurs
while t<10000
    for i=1:m
        [xnew,ynew]=updatepos(pop(i,1),pop(i,2),pop(:,1),pop(:,2),2);
        pop(i,1)=xnew;
        pop(i,2)=ynew;

    end
    for i=m+1:m+n
        [xnew,ynew]=updatepos(pop(i,1),pop(i,2),pop(:,1),pop(:,2),1);
        pop(i,1)=xnew;
        pop(i,2)=ynew;

    end

   % checksafety(pop,xsafe1,ysafe1)

    for k=1:m
        plot(pop(k,1),pop(k,2),'.r','Markersize',10)
        hold on
    end
    for k=m+1:m+n
```

```
        plot(pop(k,1),pop(k,2),'.b','Markersize',10)
        hold on
    end
    rectangle('Position',[0 30 280 180],'FaceColor', 'k')
    hold on
    rectangle('Position',[280 70 200 100],'FaceColor', 'k')
    rectangle('Position',[540 70 100 100],'FaceColor', 'k')
    rectangle('Position',[480 210 200 30],'FaceColor', 'k')
    rectangle('Position',[480 0 200 30],'FaceColor', 'k')
    rectangle('Position',[320 30 30 30],'FaceColor', 'k')
    rectangle('Position',[410 30 30 30],'FaceColor', 'k')
    rectangle('Position',[0 0 680 240])
    rectangle('Position',[240 0 10 5],'FaceColor', 'g')
    hold off
    t
    t=t+1;
    pause(0.01)
end
```

10.6 Appendix 6

```
function [xnew,ynew]=updateposmulti(xi,yi,x,y,l)
positionnext=[xi-l,yi+l;xi-l,yi;xi-l,yi-l;xi,yi+l;xi,yi-l;xi+l,yi+l;xi+l,yi;xi+l,yi-l;];
flagall=[];
for k=1:8
    flag=checkavailability(positionnext(k,1),positionnext(k,2),x,y);
    flagall=[flagall;flag];
end
% xnew=[];
% ynew=[];
%% randomly move
% avapos=find(flagall==1);
% numchos=randperm(length(avapos));
% xnew=positionnext(avapos(numchos(1)),1);
% ynew=positionnext(avapos(numchos(1)),2);
%% move towards the safe zone
avapos=find(flagall==1);
if xi>640&&xi<680&&yi>=65&&yi<=210
%       if find(avapos==4)
        xnew=positionnext(5,1);
        ynew=positionnext(5,2);
%       elseif find(avapos==1)
%           xnew=positionnext(1,1);
%           ynew=positionnext(1,2);
```

```
        %          elseif find(avapos==5)
        %       xnew=positionnext(5,1);
        %       ynew=positionnext(5,2);
        %       elseif find(avapos==5)
        %       xnew=positionnext(7,1);
        %       ynew=positionnext(7,2);
%       end
elseif xi>580&&xi<=640&&yi>=170&&yi<=210
%       if find(avapos==5)
        xnew=positionnext(7,1);
        ynew=positionnext(7,2);

%       elseif find(avapos==3)
%           xnew=positionnext(3,1);
%           ynew=positionnext(3,2);
        %       elseif find(avapos==5)
        %       xnew=positionnext(5,1);
        %       ynew=positionnext(5,2);
%       end
elseif length(avapos)==0
    xnew=xi;
    ynew=yi;
else
    safepoint1=[240+floor(rand(1)*10),floor(5*rand(1))];
    safepoint2=[270+floor(rand(1)*10),235+floor(5*rand(1))];
    safepoint3=[480+floor(rand(1)*5),100+floor(10*rand(1))];
    safepoint4=[470+floor(rand(1)*10),235+floor(5*rand(1))];
    distance1=zeros(length(avapos),1);
    distance2=zeros(length(avapos),1);
    distance3=zeros(length(avapos),1);
    distance4=zeros(length(avapos),1);
    for i=1:length(avapos)
        distance1(i)=norm(positionnext(avapos(i),:)-safepoint1);
    end
    for i=1:length(avapos)
        distance2(i)=norm(positionnext(avapos(i),:)-safepoint2);
    end
    for i=1:length(avapos)
        distance3(i)=norm(positionnext(avapos(i),:)-safepoint3);
    end
    for i=1:length(avapos)
        distance4(i)=norm(positionnext(avapos(i),:)-safepoint4);
    end
    %       [a,b]=min(distance1);
```

```
    %       [c,d]=min(distance2);
    com=[distance1;distance2;distance3;distance4];
    [mm,nn]=min(com);
    if nn<=length(distance1)
        xnew=positionnext(avapos(nn),1);
        ynew=positionnext(avapos(nn),2);
    elseif nn<=length(distance1)+length(distance2)
        xnew=positionnext(avapos(nn-length(distance1)),1);
        ynew=positionnext(avapos(nn-length(distance1)),2);
    elseif nn<=length(distance1)+length(distance2)+length(distance3)
        xnew=positionnext(avapos(nn-length(distance1)-length(distance2)),1);
        ynew=positionnext(avapos(nn-length(distance1)-length(distance2)),2);
    elseif nn<=length(distance1)+length(distance2)+length(distance3)+length(distance4)
        xnew=positionnext(avapos(nn-length(distance1)-length(distance2)-length(distance3)),1);
        ynew=positionnext(avapos(nn-length(distance1)-length(distance2)-length(distance3)),2);

    end
end
```

10.7 Appendix 7

```
clear;clc;
%% set the initial pop
m=1000;%m is the number of adluts vistors on this floor
n=300;%n is the number of the olds,kids and disabled vistors on this floor
g=50;%n is the number of the group vistors on this floor
[x,y,pop]=initialpop(m,n);
%plot the unavailable spaces
rectangle('Position',[0 30 280 180],'FaceColor', 'k')
hold on
rectangle('Position',[280 70 200 100],'FaceColor', 'k')
rectangle('Position',[540 70 100 100],'FaceColor', 'k')
rectangle('Position',[480 210 200 30],'FaceColor', 'k')
rectangle('Position',[480 0 200 30],'FaceColor', 'k')
rectangle('Position',[320 30 30 30],'FaceColor', 'k')
rectangle('Position',[410 30 30 30],'FaceColor', 'k')
rectangle('Position',[0 0 680 240])

%% set the safe zone
xsafe1=[240;250;250;240;240];
ysafe1=[0;0;5;5;0];
rectangle('Position',[240 0 10 5],'FaceColor', 'g')
xsafe2=[270;280;280;270;270];
ysafe2=[235;235;240;240;235];
```

```
rectangle('Position',[270 235 10 5],'FaceColor', 'g')
xsafe3=[480;485;485;480;480];
ysafe3=[100;100;110;110;100];
rectangle('Position',[480 100 5 10],'FaceColor', 'g')
xsafe4=[470;480;480;470;470];
ysafe4=[235;235;240;240;235];
rectangle('Position',[470 235 10 5],'FaceColor', 'g')
rectangle('Position',[580 170 5 40],'FaceColor', 'c')
%%plot initial pop
for k=1:m
    plot(x(k),y(k),'.r','Markersize',10)
    hold on
end
for k=m+1:m+n-g
    plot(x(k),y(k),'.b','Markersize',10)
    hold on
end
for k=m+n-g+1:m+n
    plot(x(k),y(k),'.y','Markersize',25)
    hold on
end
%% set the threatened zone
rectangle('Position',[580 170 5 40],'FaceColor', 'c')

hold off
%red for adults,blue for the olds,kids and disabled vistors on this floor
t=1;% time variable
%% when an emergency occurs
while t<10000
    for i=1:m
        [xnew,ynew]=updateposmulti(pop(i,1),pop(i,2),pop(:,1),pop(:,2),2);
        pop(i,1)=xnew;
        pop(i,2)=ynew;

    end
    for i=m+1:m+n
        [xnew,ynew]=updateposmulti(pop(i,1),pop(i,2),pop(:,1),pop(:,2),1);
        pop(i,1)=xnew;
        pop(i,2)=ynew;

    end
    %checksafety
%       pop=checksafety(pop,xsafe1,ysafe1);
%       pop=checksafety(pop,xsafe2,ysafe2);
```

```
%
    for k=1:m
        plot(pop(k,1),pop(k,2),'.r','Markersize',10)
        hold on
    end
    for k=m+1:m+n-g
        plot(pop(k,1),pop(k,2),'.b','Markersize',10)
        hold on
    end
      for k=m+n-g+1:m+n
        plot(pop(k,1),pop(k,2),'.y','Markersize',25)
        hold on
    end
    rectangle('Position',[0 30 280 180],'FaceColor', 'k')
    hold on
    rectangle('Position',[280 70 200 100],'FaceColor', 'k')
    rectangle('Position',[540 70 100 100],'FaceColor', 'k')
    rectangle('Position',[480 210 200 30],'FaceColor', 'k')
    rectangle('Position',[480 0 200 30],'FaceColor', 'k')
    rectangle('Position',[320 30 30 30],'FaceColor', 'k')
    rectangle('Position',[410 30 30 30],'FaceColor', 'k')
    rectangle('Position',[0 0 680 240])
    rectangle('Position',[240 0 10 5],'FaceColor', 'g')
    rectangle('Position',[270 235 10 5],'FaceColor', 'g')
    rectangle('Position',[480 100 5 10],'FaceColor', 'g')
    rectangle('Position',[470 235 10 5],'FaceColor', 'g')

    rectangle('Position',[580 170 5 40],'FaceColor', 'c')
    hold off
    t
    t=t+1;
    pause(0.01)
end
```