Problem D: Time to leave the *Louvre*

林师言201830570194，彭鹏，孙秦

SCUT，China

github:<https://github.com/XinShiQaQ/MathModelPrivate>

The increasing number of terror attacks in France requires a review of the emergency evacuation plans at many popular destinations. Your ICM team is helping to design evacuation plans at the Louvre in Paris, France. In general, the goal of evacuation is to have all occupants leave the building as quickly and safely as possible. Upon notification of a required evacuation, individuals egress to and through an optimal exit in order to empty the building as quickly as possible.

The Louvre is one of the world’s largest and most visited art museum, receiving more than 8.1 million visitors in 2017. The number of guests in the museum varies throughout the day and year, which provides challenges in planning for regular movement within the museum. The diversity of visitors -- speaking a variety of languages, groups traveling together, and disabled visitors -- makes evacuation in an emergency even more challenging.

The 380,000 exhibits located on these five floors cover approximately 72,735 square meters, with building wings as long as 480 meters or 5 city blocks. The pyramid entrance is the main and most used public entrance to the museum. However, there are also three other entrances usually reserved for groups and individuals with museum memberships: the Passage Richelieu entrance, the Carrousel du Louvre entrance, and the Portes Des Lions entrance. The Louvre has an online application, “Affluences” (https://www.affluences.com/louvre.php), that provides real time updates on the estimated waiting time at each of these entrances to help facilitate entry to the museum. Your team might consider how technology, to include apps such as Affluences, or others could be used to facilitate your evacuation plan.

Only emergency personnel and museum officials know the actual number of total available exit points (service doors, employee entrances, VIP entrances, emergency exits, and old secret entrances built by the monarchy, etc.). While public awareness of these exit points could provide additional strength to an evacuation plan, their use would simultaneously cause security concerns due to the lower or limited security postures at these exits compared with level of security at the four main entrances. Thus, when creating your model, your team should consider carefully when and how any additional exits might be utilized.

Your supervisor wants your ICM team to develop an emergency evacuation model that allows the museum leaders to explore a range of options to evacuate visitors from the museum, while also allowing emergency personnel to enter the building as quickly as possible. It is important to identify potential bottlenecks that may limit movement towards the exits. The museum emergency planners are especially interested in an adaptable model that can be designed to address a broad set of considerations and various types of potential threats. Each threat has the potential to alter or remove segments of possible routes to safety that may be essential in a single optimized route. Once developed, validate your model(s) and discuss how the Louvre would implement it.

Based on the results of your work, propose policy and procedural recommendations for emergency management of the Louvre. Include any applicable crowd management and control procedures that your team believes are necessary for the safety of the visitors. Additionally, discuss how you could adapt and implement your model(s) for other large, crowded structures.

1. **Introduction**

In this paper, we will introduce a method to analyze the information of the whole space of house and find out the emergency evacuation model.

Breadth First Search is one of the simplest graph search algorithms and the prototype of many important graph algorithms. It is a blind search method that systematically expands and examines all nodes in a graph for results. In other words, it doesn't consider the possible location of the result, and searches the entire graph thoroughly until it finds the result. We use Bread first search to analyze the room from the exit to the entire room space in order to get the path length from everywhere to exits.

Cellular Automata is a kind of mesh dynamic model with discrete time, space and state and local spatial interaction and temporal causality. In our work, we use Cellular Automata to simulate the crowd evacuation, including the shortest paths of different crowd to exits.

All of our model generations and code are based on Python.

[keywords]: Evacuation model Breadth First Search Cellular Automata

1. **Basic Assumption**

In this section, we discuss several assumptions based on the situation in the room.

**Assumption 1**. The room we consider of only consists of empty space, wall and exits. And the information of the crowds in the room is randomly generated and the map of the room is designed on our own because our evacuation model and algorithm are aimed to analyze every model of indoor room, not only the *Louvre.*

‘0’ represent wall, and Boolean value exit True represent a exit.

**Assumption 2**. The crowd in the room consists of young, adult, old and disable. Different crowd have different moving speeds.

‘person’ represents different crowd, ‘0’ represent old, ‘1’ is young and ‘3’ is disable. ‘v’ is the value to represent the speed of different crowd, ‘1’ is adult, ‘1.5’ represent old, ‘1.3’ is young and ‘2.5’ is disable.

**Assumption 3**. We assume that the crowd evacuate from the room by every exit, every exit is usable and not shut down.

1. **Definitions**

For each valid node in the map has some attributes.

|  |  |
| --- | --- |
| Msg | The minimum distance between the node and the exit node |
| Cap | The biggest number of people can stay in one node |
| Person | Four integer which represent number of different people in the node |

For easily represent, we use some method to represent some messages

|  |  |
| --- | --- |
| RemainVol | Cap - Σ Person |
| TotalPerson | Σ Person |

For different kind of people, we use

|  |  |
| --- | --- |
| Alpha | Acceptability of RemainVol in next node that the person will go to |

Meanwhile, we need some variable to represent initial settings

|  |  |
| --- | --- |
| Flows | The number of people leave at one exit node for each iteration |

1. **Models**
   1. **Data Preprocessing**

Before presenting our models, we would like to address the preprocessing work we have done to the data.

Step 1. We construct a random map of the room, which is the size of 6\*11. (in matrix)

Step 2. We add up the information about the room we prepare, including the capacity of an empty space, wall, people and exits.

Step 3. Combine the map and the information of room together, then build up a space model of the room, which is executed by Python program. There is the rule:

‘[x][y]’ represent the location information

‘.exit=True’ stand for a exit

‘.cap’ represent the crowd capacity

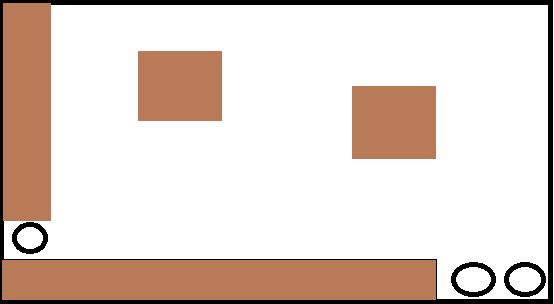
‘person=0’ represent old crowd, ‘alpha=1.5’ is the tolerance of the old for the number of people in queuing.

‘person=1’ represent young crowd, ‘alpha=1.3’ is the tolerance of the young for the number of people in queuing.

‘person=3’ represent disable crowd, ‘alpha=2’ is the tolerance of the disable for the number of people in queuing.

‘alpha=1’ represent the tolerance of the adult for the number of people in queuing.

The following is the design of the map of the room using to test our crowd evacuation models:



The whole black frame stands for the whole room, one circle stands for an exit and the brown blocks stand for walls.

After building up the information of the room, the crowd information is generated. The people will be distributed in the whole room randomly.

Building up room information and distributing people in the room will both be completed in our Python program.

1. **Breadth First Search**
   1. **Overview**

In this paper, in order to systematically expand and examine all nodes in the diagram, we use breadth first search. The breadth first search in our model is to analyze the whole map of the room and then find out the path length from every space to exit. This algorithm is aimed to make people crowd on everywhere in the room can find the shortest path length to reach the exit and evacuate. On this evacuation, people can not stand on the wall block and people need to devour to get the exit if there is wall on the evacuating route. So our Breadth First Search algorithm need to consider the effect of wall.

* 1. **Methodology**

Initially, for the exit node, the Msg will be signed as 0. Beginning from the exit node, BFS is used to sign all node’s Msg. Each time when visit one node (represent by A) from another node (represent by B), we have

1. Msg = B.Msg + 1

After one time of BFS method, all the nodes’ Msg is defined as the shortest path to the exit due to BFS’s principle

* 1. **Result and Analysis**

The following is the running result of our Breadth First Search algorithm:

0 5 6 7 8 9 8 7 6 5 5

0 4 5 0 0 8 7 6 5 4 4

0 3 4 0 0 7 6 0 0 3 3

0 2 3 4 5 6 5 0 0 2 2

0 1 2 3 4 5 4 3 2 1 1

0 0 0 0 0 0 0 0 0 0 0

The number ‘0’s in location (1,5), (10,6) and (11,6) stand for the three exits. And the other ‘0’s stand for the wall and people have to devour when the wall beside their evacuating route. The other number stand for the number of pace and path length people need to walk from every grid to the nearest exit. The path length is one of the factor affects the people choosing different path to evacuate.

**6. Cellular Automata**

**6.1 Overview**

After finishing analyzing the information of the map, we need to construct the evacuation model of the crowd. Cellular automata is a good way to simulate people evacuation in a space, it also is widely used in researching, such as formation and outbreak of economic crisis, The sociality of individual behavior, epidemic, etc. The design idea of cellular automata itself comes from the idea of biological self-reproduction, and diffusion simulation of biomes based on cellular automata model is also a hot application. Therefore, we design a Cellular automata model to build up the evacuation model of the room.

**6.2 Methodology**

For each iteration, starting from the exit node, we use BFS to travel through the map to calculate node transfer because the empty space for person to move is from the exit node radially broadcast to the whole map.

For each node that is not the exit node, we consider the four directions, left, right, up, down and not move, stored as called new node. For each new node and different person, we calculate a weight as w,

W = - max(NewNode.Msg, t) \* Alpha + (Node.Msg - NewNode.Msg)

Where t represent expected time. Considering several matrix formed by NewNode and a symmetric node to different exit, we calculate all people in the matrix (N) and divide by the valid size (S)

t = N / S

The maximum function represent a situation that the node is not at the crowd sector area which will be talking about later.

The left side of plus represent a influence of crowd while the right hand side represent shortest path influence.

For each person in the initial node we want to transfer will select the new node that has the biggest w till the new node is full, then they will choose the secondary new node.

The node which is the exit node will just process the people leave the area we talking about,

Exit.totalPerson = Exit.totalPerson - flows

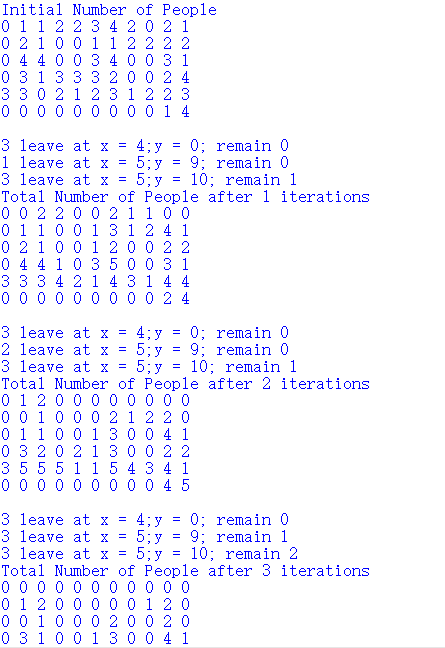
When ΣtotalPerson = 0, finish the processing.

**6.3 Result and Analysis**

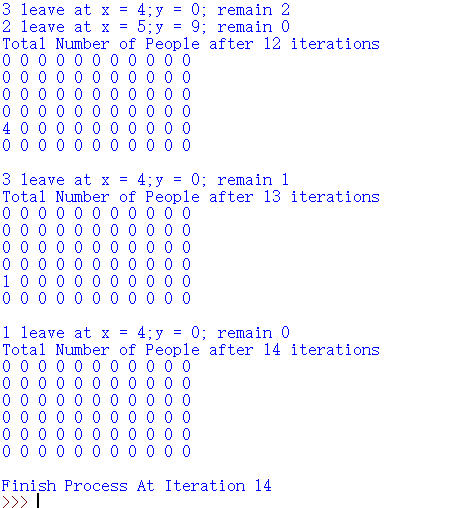
Based on Breadth first search and Cellular automata model, we then can construct a evacuation model by Python programming.

In the real world, usually there are many people in exhibit room, so we set up 140 people (crowd) in our space model. We construct a randomly distribution model and the people are distributed randomly at everywhere in the room. And we set up 3 exits at the edge of the room. There are two rectangular pillars near the center of the room and two walls at the left edge and down edge of the room. people have to detour when beside the wall.

When the evacuation starts, people on everywhere in the room move toward closest exits. The information of people number on the matrix refreshes every time the time updates.



On the 2-D matrix, the integer value represents the number information of people on every space. (note: the value just stands for the total number of people, the different crowd can stand on one space, for example, young and old in one space.) The iteration represents a period of time. ‘After n iterations’ means that the time consumption of the evacuation is n period of time. Every iteration, it shows that the integer values on the 2-D matrix are changing, which means that all the people crowd are evacuating and moving toward the closest exit. Different crowd moving in different speed. In this test model, the exits are at (1,5), (10,6) and (11,6).



From the program running result, the total time consumption of whole evacuation is 14 iteration times and all people reach the exits in the room. The result is that 140 number of people all get out of the room.

The one time iteration stand for a time period and the period can be defined on our own. For instance, if we define that one iteration represent 5 sec, then 14 iteration represent 70 sec and the total time consumption of whole evacuation is 70 sec.

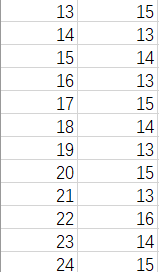
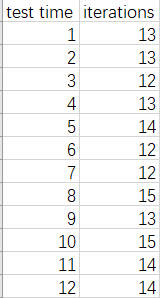
The size of every grid can also be defined by self. For instance, if we define the size of a grid is 10m\*10m, then the size of the whole room is 110m\*60m.

Considering the different number of people on every exit, we found that the people number of exit (5,1) is much more than that of the other two exits. In fact, too many people get across one exit in real world will cause jam and slow down the speed of people getting out the room. But in our consideration, if there are many people getting out of the room by one exit, the people will queue in a shape like fan and get out smoothly. Therefore, the probable accident causing by crowded exit will not be considered into our algorithm and the whole evacuation situation.

When most of people crowd near the exit node, it shows a circular arc with the center of exit node. We can find that in most case, people near the exit will keep waiting and the waiting length is the circular arc with radius equal to the distance between the person and the exit node. While the person far away from the exit node may try to go to another exit node that may have lower time spending.

1. **Result and Analysis**

After constructing our total method to analyze the room evacuation, we ran our program and collected some necessary data.



The average times of iteration is 13.75

As we mention before, we set up one time period of an iteration is 5 sec, then the average time consumption of the whole evacuation is 68.75 sec.

1. **Conclusion**

**8.1 Overview**

In this paper, we need to discuss and find the solution of designing evacuation plans at the *Louvre* in Paris, France. In general, the goal of evacuation is to have all occupants leave the building as quickly and safely as possible. We construct a room which represent a place need evacuation. Our goal is to design a general evacuation model designing algorithm. Not only to solve the evacuation model in *Louvre*, but also to apply our algorithm in every map and condition. That’s why we design our own room information to test our model.

Breadth First Search helps find the shortest path from every grid to the exits.

**8.2 Strength and weakness**

In our works, we mainly focus on solving the balance between crowd degree and the shortest distance. It is hard to choose the global best but easily to find the semi-better. During our work, we propose a variable weight to measure the goodness of each node that make the path choosing convenient. In our examinations, our methods can do better than normal BFS searching method for shortest path. In the same map with random number of people in each node, we normally get about 5% better and in some crowd map we can do better about 10%

Weakness is about the transfer equations, we still not find the best balance between the two influences. For different map with different degree of crowd, the correlation of them is totally different that we cannot find the best formula but the better in our examples. Meanwhile, we focus on the process of the balance but ignore some important ingredient such as randomly path break and exit node break. In real life of the background, we must put them in consideration. However, due to the limit of ability and time, we just ignore them.

**8.3 Conclusion**

In this work, we mainly solve the problem of balancing two influences of the escape problem, crowd degree and the shortest path. We try to find the semi-better solving for each node in the matrix by using weight that defined by this two variables. During this methods, when crowded situation, we can increase escape velocity for more than 10% and 5% in average.