	Team Control Number	
For office use only	**1912457**	For office use only
T1 ________________		F1 ________________
T2 ________________		F2 ________________
T3 ________________	Problem Chosen	F3 ________________
T4 ________________	**D**	F4 ________________

2019
MCM/ICM
Summary Sheet

Abstract

The objective of our Internal Control Management (ICM) team was to investigate the emergency evacuation plan for the Louvre in Paris. We are given several challenges according the plan design requests.

Our model is divided into two major parts, abstraction part and simulation part. For the abstraction part, we propose a novel model called Recurrent Min-Cost Flow (RMCF) model based on min-cost max-flow model. The original max-flow problem is not time-related, all the flow completes immediately which could not be used for evacuation. So, our RMCF model extends the node of graph from spatial domain to temporal domain, and we can thus minimize the time cost and maximize the capacity.

Using the agent-based NetLogo model to study how passengers move through paths and rooms, we can generate time cost data based on careful parameter setting and simulation to the RMCF model to calculate the optimal evacuation route, which can also conduct the movements of simulation, forming a cycle.

To run simulations and make detailed analysis, we propose a square map with 4 paths and four rooms to represent the Louvre. To simulate human, we apply flocking model with collision avoidance to approximate the behavior of people in real life. Our model studied how several threats impact passenger flow including rocks, stairs, disabled visitors, and additional gates. We found that, with the same max speed reduction, if the population is large, stairs model is 10% slower than the disabled model with same configuration; it shows that the stairs are a huge bottleneck with more visitors. Another large bottleneck is the crowding of people, which cause rapid deceleration.

With our two-part model, we could handle larger structure with higher abstraction and more crowded place with more precise simulation.So our model is adaptive and thus allowed us to closely analyze the evacuation plan and shed light on how to evacuate so many people in such a complex structure.

1 Introduction

1.1 Background

According to the data of the Global Terrorist Database, from 2015 to 2017, there have been 115 terrorist attacks in France. The total casualty is 450 and the total number of the injured is 1,403, of which 25 occurred in Paris, accounting for 21.7% of the total. Most seriously, on 13 November 2015, there were five terrorist attacks in Paris, which caused 133 fatalities and 311 injuries. Former French President Francois Hollande immediately declared a national state of emergency and President Barack Obama said on the next day that the attack was a terrorist attack on humanity. Thus, the prevention of terrorist attacks has become the focus of national attention. And the crowded public places are the focused areas of terrorist attacks, such as stadiums, museums, squares, concert halls, etc., are often more vulnerable to attacks. Therefore, making reasonable and efficient evacuation plan for public places is very important to prevent the terrorist attacks.

The Louvre is one of the most famous attractions in Paris. The number of visitors officially claimed to the Louvre Museum reached 10.2 million in 2018, a 25% increase compared to 2017, setting a new record. The Louvre has five floors, including 380000 exhibits, covering an area of 72,735 square meters. Due to the complex internal structure of the Louvre, evacuation plan is affected by various types of emergencies. How to quickly use the exit to evacuate tourists, and to ensure the safety of the exit, as well as the normal entry of emergency personnel, is the key to our adaptive plan.

1.2 Restatement of the Problem

1. We have to give a model to conduct the behavior of people in evacuation process, considering both the mental and physical factors. We also need to take the diversity of people into considerations – speaking different languages, groups traveling together and disabled visitors.

2. Our model should offer a range of options to the museum leaders confronted with various of situations and potential threats, which means it should be highly-adaptive and capable of responding to changes swiftly.

3. Additional exits could be placed in the museum with careful analysis and validation simulations.

4. We need to figure out the bottlenecks of our models and give optimization to maximize the throughput of our plan.

2 General Assumptions

1. Each visitor visits alone. We assume that visitors are not in pairs, as families, or in groups, each makes decisions during evacuation that are not dependent on others.
2. There is no collision between visitors and visitors. There is a limit distance between the visitor and the visitor during evacuation. When the limit distance is reached, the visitor will adjust the speed and direction of the movement.
3. Visitors can only see a certain range of situations during evacuation, including obstacles and congestion level, and will not make adjustments in advance based on the overall situation.
4. Visitors appear randomly in the Louvre, including the floors, exhibition halls, etc., but the density of visitors in different pavilions is proportional to the popularity of the exhibits.
5. The evacuation model is obtained after processing the plan book given in the Louvre official guide. Different colors indicate different meanings (as shown in Figure 1).
6. Leaving the building is a successive process, and the regular exit locations are marked (as shown in Figure 2).
7. Evacuated visitors have no influence on non-evacuated visitors. After the evacuated visitors were evacuated, they were properly scattered by the staff outside the Louvre, and there were no congestion or impact on the non-evacuated people.
8. The visitors followed the guidance of the staff. When the visitor enters the scope of the staff's influence during evacuation, the adjustment will be made according to the instructions of the staff.

3 Overall Framework

The overall framework of our model is given in Fig.1. Our model can be divided into two parts, abstraction part and simulation part. The complex structures of the Louvre can be abstracted with graph model, where vertices and edges are used to represent the structure; by abstraction, we could apply several algorithms on it. However, merely abstracting the topographic model into a topological model may lose the microcosmic information, which the Louvre is actually filled with such as the walls, the obstacles or even the stairs. That's the reason why we not only use the highly abstracted graph model, but also use NetLogo to simulate the behaviors of visitors in the museum. The simulations with NetLogo could produce time cost or capacity approximations, which would be fed into the proposed Recurrent Min-cost Max Flow model (RMCF). The officials of the museum could use RMCF to calculate the optimal evacuation plan, which could be used as conductors in NetLogo model. With this cycling process, we could utilize the multi-level abstraction of the museum and propose a plan microcosmically and macrocosmically.

4 Recurrent Min-Cost Flow Model

In the routing process of the evacuation path, we use an algorithm called Recurrent Minimum-cost Flow (RMCF). This algorithm is a variant based on the Minimum-cost flow problem (MCF) [1,3,6,9].

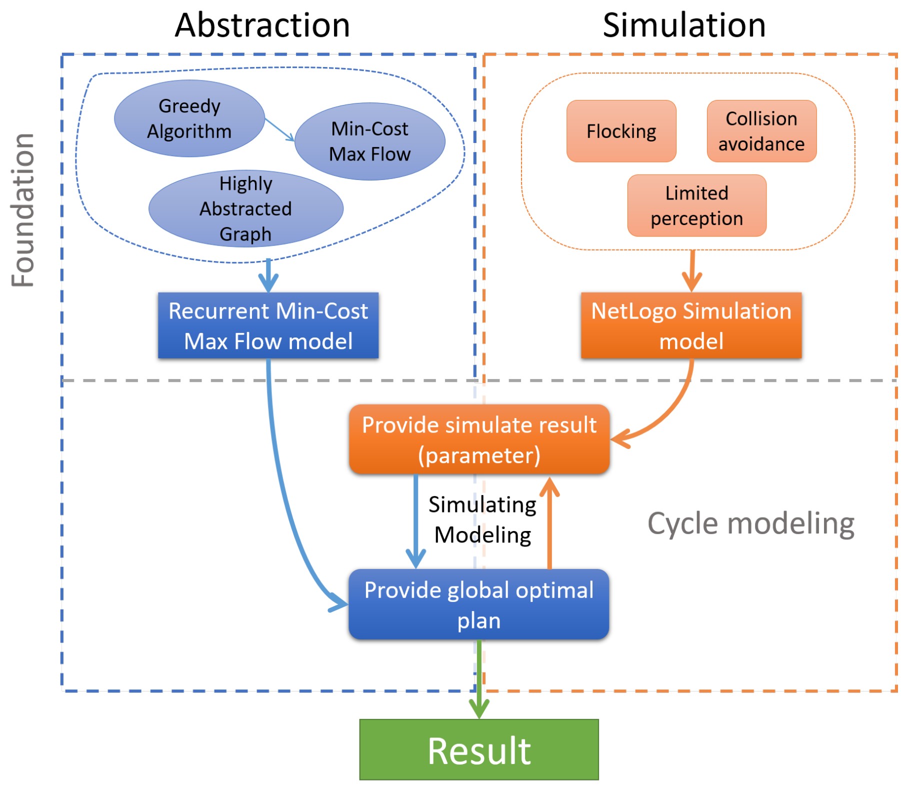


Figure 1: The overall structure of our cycle model

Minimum-cost flow problem is a kind of Flow network problem. In graph theory, a flow network (also known as a transportation network) is a directed graph where each edge has a capacity and each edge receives a flow. The amount of flow on an edge cannot exceed the capacity of the edge. A flow must satisfy the restriction that the amount of flow into a node equals the amount of flow out of it, unless it is a source, which has only outgoing flow, or sink, which has only incoming flow. The minimum-cost flow problem (MCFP) is an optimization and decision problem to find the cheapest possible way of sending a certain amount of flow through a flow network.

Definition of Original Flow Network A flow network is a directed graph $G = (V, E)$ with a source vertex $s \in V$ and a sink vertex $t \in V$, where each edge $(u, v) \in E$ has capacity $c(u, v) > 0$, flow $f(u, v) \geq 0$ and cost $a(u, v)$, with most minimum-cost flow algorithms supporting edges with negative costs. The cost of sending this flow along an edge (u, v) is $f(u, v) \cdot a(u, v)$. The problem requires an amount of flow d to be sent from source s to sink t. The definition of the problem is to minimize the total cost of the flow over all edges:

Solution of MCF

1. Find a "shortest distance" path from the source point to the sink point, and "distance" is measured by the sum of the unit costs of the edges on the path.
2. Then find the minimum value f of the capacity of the edge on this path, then the current maximum stream max_flow is expanded by f, while the current minimum cost min_cost is expanded by f×min_dist(s, t).
3. Decrease the capacity of each positive side on this path by f, and increase the capacity of each reverse side by f.
4. Repeat 1–3 until the path from the source point to the sink point cannot be found.

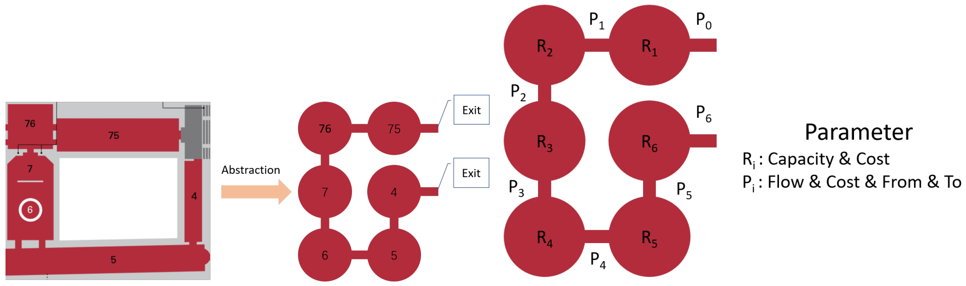


Figure 2: Step 1 for building RMCF

RMCF Our model is the modification of the MCF with time expansion and several modifications to fit the solution algorithm; that is to say, our model can be solved with the given solution above. First step of our model, we can take the Fig.2 as an example. The above picture shows the real terrain of the Louvre. We abstract the real topographic map: get the room and path elements. The room has two parameters, the room capacity: the maximum number of people that can be accommodated in this room; cost: indicates how long it takes to walk through this room. Path has two parameters, channel traffic: indicates how many people can walk this path in a unit of time; cost: how long it takes to walk through this path. After such abstraction, we can ignore the specific size and positional relationship of each room in the actual map. Just pay attention to the topological relationship between the room and the channel. However, the MCF algorithm introduced earlier cannot be used at this time. Because the nodes

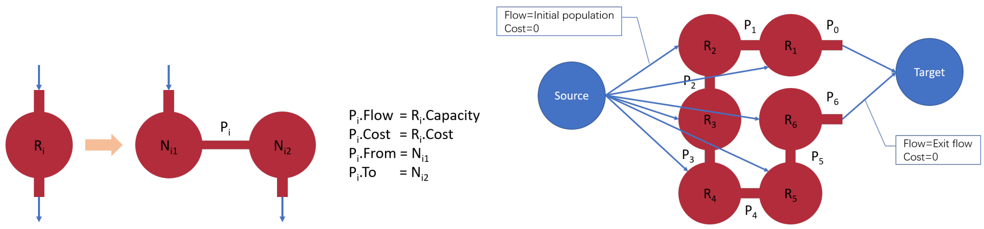


Figure 3: Step 2 for building RMCF

in the MCF algorithm do not capacity attribute. Therefore, the abstract diagram which is directly reflected the real room structure in the above need to be further processed. First of all, since the attributes of the nodes are different, the room nodes in the above figure are split into a structure consisting of two new nodes and one edge. The flow of the edge is equal to the capacity of the original room, and the cost of the edge is always the same as the cost of the original room. The path that was originally connected to this room is connected to two new nodes according to the outgoing edge and the incoming edge.

In addition, the MCF algorithm has only one starting point, and one end point. In the scene of evacuation, people are randomly distributed in all rooms according to a certain proportion, so there will be multiple starting points. So we need to create a new virtual starting point, he has the edge connected to all rooms, and the flow of these sides is the number of people in this room at the beginning of the evacuation, the cost is zero. This allows everyone to be assigned to the appropriate room at the very beginning of the algorithm. At the same time, the Louvre should have multiple evacuation exits, so we use a similar idea to create a new virtual destination that is connected to all available outlets. The flow for these edges is the corresponding exit flow, and the cost is zero. The meaning of this node is outside the Louvre. Since the MCF does not have the concept of time, it is assumed that a room with a capacity of 1 enters one person in the first second, the person goes out in the second second, and the other person enters the third second is not allowed. But obviously this is not realistic, and this can happen in the actual evacuation process. So we have to add the concept of time to MCF. The

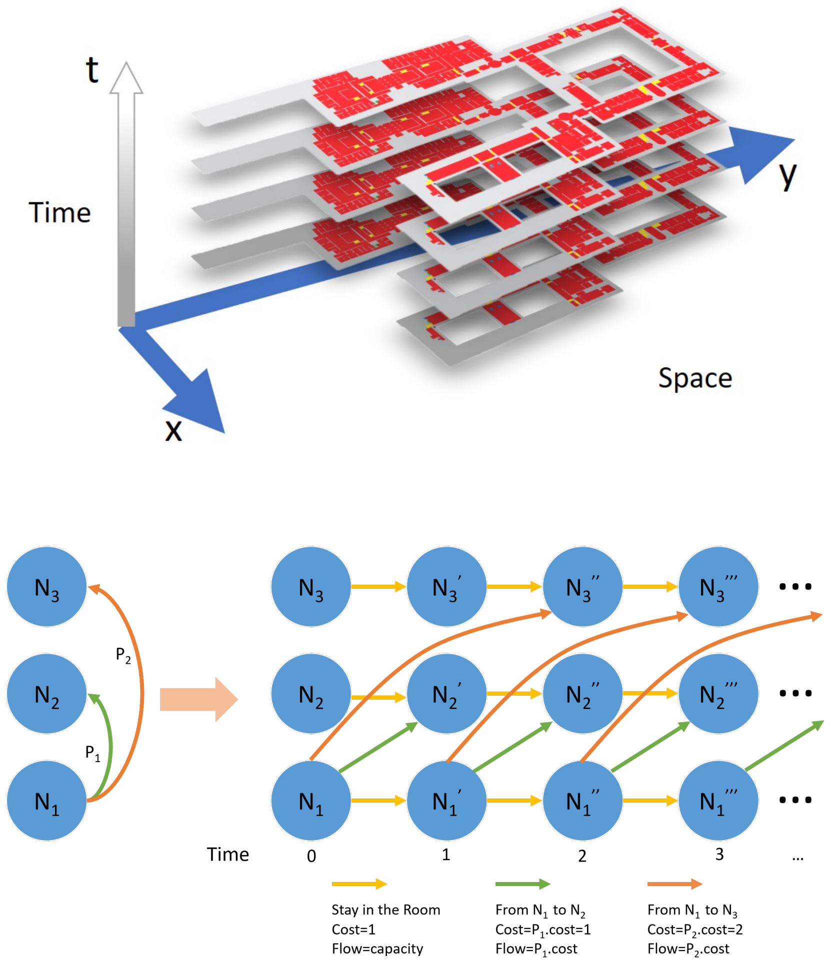


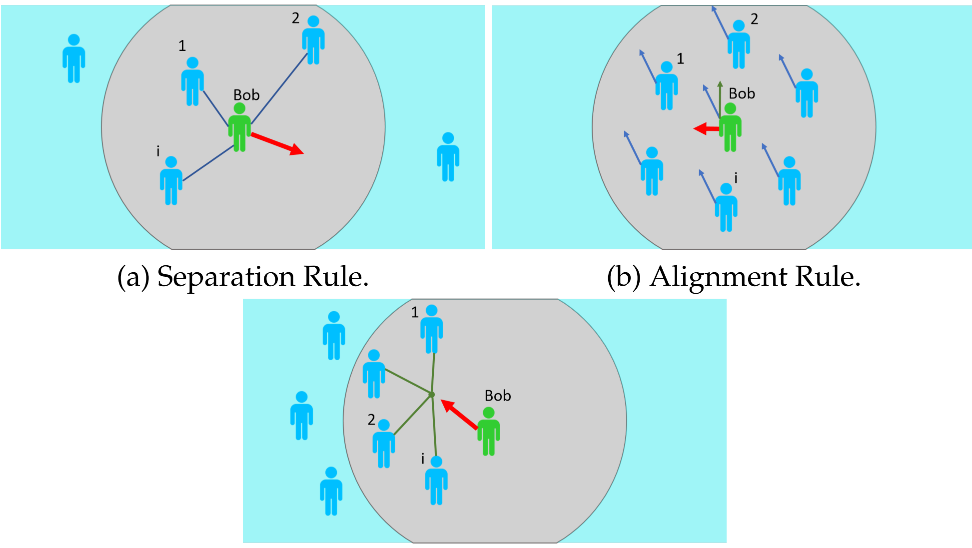
Figure 4: Step 3 for building RMCF

specific method is to expand the spatial map created above into time and become a space-time map.

The same spatial point can be directly connected to the corresponding two points in the two connected time. The flow of this edge corresponds to the capacity of the room corresponding to this point. The cost of this edge is 1. The meaning of this side is the number of people who stay still in this room. The edge existing in the original picture is connected to the destination node after the cost unit time according to the cost. In addition, the virtual starting point in the above is only connected to each room node in the first time unit. The virtual end point above is connected to the egress node at all times. Since this step involves connecting each node to itself in the next time, our model is called Recurrent Minimum-cost Flow. At this point, all the real maps to the abstract map have been modeled, and the optimal path can be solved directly according to the CMF algorithm.

5 NetLogo Model

To evaluate our assumptions and generate data, we use an agent-based NetLogo model. With NetLogo, we could easily visualize and simulate the behavior of visitors, officials and even terrorists in a time-accurate model. We could also use NetLogo to produce experimental data.



(a) Separation Rule.

(b) Alignment Rule.

(c) Cohesion Rule.

Figure 5: Basic Flocking Rules.

5.1 Visitors Modeling

The visitors of our model are represented by "turtles", icons in the grid capable of performing actions. Each visitor has several attributes including the ones that restore the doors it has visited to prevent re-searching. Visitors are represented by orange people. Human are group animals, so there exists some interactions among people such as following or grouping. We apply flocking model [4, 5, 13, 15] to describe such properties in the following section.

5.1.1 Flocking Model

Flocking is a cooperative behavior of social animals. It is mainly manifested by the swarming phenomenon caused by a large number of individuals moving together. The simple rules generated by this behavior only cover the individuals and do not concern any central coordination. Migratory birds as an example: a group of migratory birds, there must be some behavior, allowing them to coordinate their movements with clusters. These behaviors seem particularly unique, but all creatures have them to some extent.

A contradictory balance is kept: the desire to maintain close relationship with group of birds and also to avoid collisions. A lot of behaviors of the crowds are similar between those of human and birds. For example: grouping, crossing a sidewalk, escaping from danger, etc. Building a common crowd model is very difficult; on the one hand, human-beings are the most complex creatures of nature; on the other hand, every individual in the real world is an independent agent. Even if they have a common goal, each person has different ideas and different personalities. In this article we only introduce some simple crowd behavior simulations. Our proposed flocking model has three basic rules:

Separation The central person in Fig.7 measures the distance between the surrounding group and himself. If the distance is too short, then the individual goes away from the local grouping partner to avoid crowding. Just like a magnet, the distance between them can be maintained when the distance is appropriate.

Alignment The central individual in Fig.7 measures the clustering partners within a certain distance, seeks the average direction of the local partners, and the individual turns to their average direction.
Cohesion The central individual in Fig.7 measures the clustering partners within a certain distance, seeks the average position of the local partners, and individual turns to their average position.

5.1.2 Collision Avoidance Model

We model the process of acceleration and deceleration process of people in evacuation with this model. We assumed that people would avoid collision not only by separation rule in flocking model but also slow down their speed. Because people could not accurately know the speed of others, they would adjust their speed to be slightly slower. $speed(P) = speed(P_front) - deceleration$. And if there is nobody in front of the individual, he would accelerate till the speed reaches the limit.

5.2 Ablation Simulation

Our model can be divided into several parts including flocking model, collision avoidance model, and recurrent min-cost flow model. We would like to do ablation study to give effective analysis of our model. The capacity of the room and the flow of the exit will have a major impact on the evacuation, and the shape of the room itself will not have much impact. We used NetLogo simulation to design multiple maps, each of which contains only one room of the same size but different shapes and an exit with the same location and size. The same amount of people were randomly generated in the room, and the time of ten evacuation was calculated. It was found that the evacuation time had no obvious correlation with the shape of the room. And this conclusion is also the basis for the abstraction of the entire Louvre as an abstract map containing only points and edges.

Running a simulation algorithm on a complete Louvre map requires a huge amount of computation. Moreover, the shape of the room itself does not have much influence on the evacuation speed. So the actual map of the Louvre can be simplified into a collection of rooms and channels in a certain positional relationship. In this case, the actual situation in the Louvre Museum can be broken down into a series of repetitive small structures. For example, a plurality of rooms in an area and a passage connecting the rooms to each other; a room connected to a staircase that causes deceleration; a multi-branch junction. Then we can carry out the simulation experiment of the actual evacuation on the abstracted small module map with universal effects, without directly carrying out the simulation experiment on the complete large map. If you need to verify the effectiveness of the complete global evacuation route, you need to use a higher-configuration computer to perform simulations with tens of thousands of agents at the same time. Considering the reasons of its own computing equipment, this paper adopts NetLogo simulation for local details and typical building structure. For the planning of the overall evacuation route of the Louvre, the RMCF highly abstract modeling calculation method is adopted.

5.2.1 Simulation of Collision Avoidance Model

The parameters set include the number of people, the acceleration of the person, and the deceleration of the person. The observation point is randomly set to observe the speed change of a certain person to display the overall movement as shown in Fig. 6

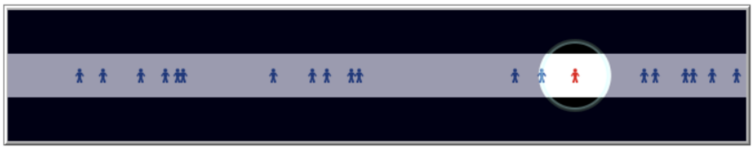


Figure 6: The motion simulation of people

5.2.2 Simulation of Flocking Model with Collision Avoidance

When we call **setup**, people would be created randomly with the number of **population**. When the model is set to **go**, people would do the flocking rules including separation, alignment and cohesion. **Vision** decides how far an individual could see when flocking.

Here we would like to analyze the influence of different parameters to our model. Our world is 50×50 patches. We fix the **acceleration** to be 0.0030, and **deceleration** to vary. And we set all the maximum turning degree with flocking operations to be 10 to see how the **minimum-separation** distance affect our model. **Vision** is 15. We use the **ticks** it takes to evacuate 100 people as our metric. We set the exit gate to be three patches, and if people enter the gate they would **die** in NetLogo. The **maximum-speed** of people is 1 patch per tick and the **minimum-speed** would vary. The initial speeds of people are randomly assigned from 0.1 to 1.

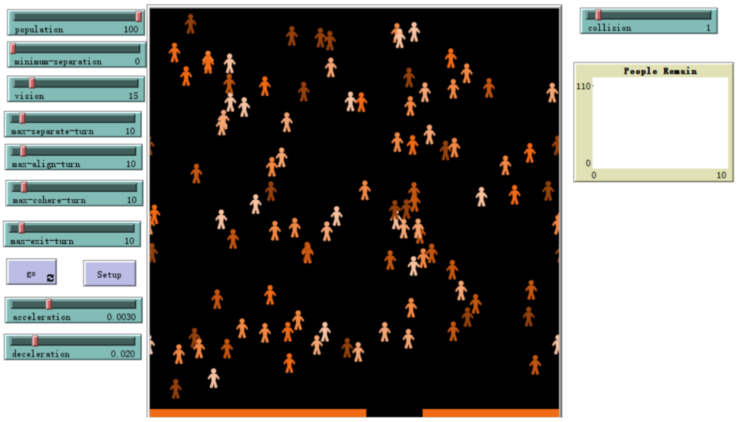


Figure 7: The screenshot shows the flocking simulations

6 Experiments and Evaluations

6.1 Experiments of RMCF Model

The following test experiments of the RMCF algorithm are performed: Take a small local map as an example for testing. The actual map is as Fig.8. Use NetLogo to simulate the corresponding flow and capacity data of each room and intersection, input it into the RMCF mapping program, and get an abstract map that can use MCF algorithm, the screen shot is given in Fig.10.

The text on the left is the original map data entered, and the text on the right is regenerated according to the rules described above. Convert the six rooms in the original map, seven paths, and expand 20 time units into 242 nodes and 368 edges. The expanded graph is input into the program for calculating the optimal evacuation path, and the result is calculated, and converted

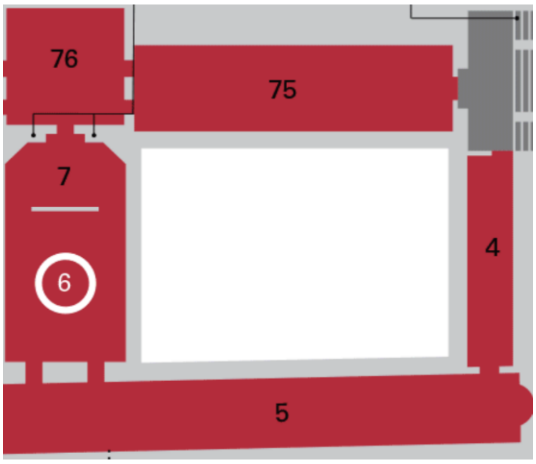
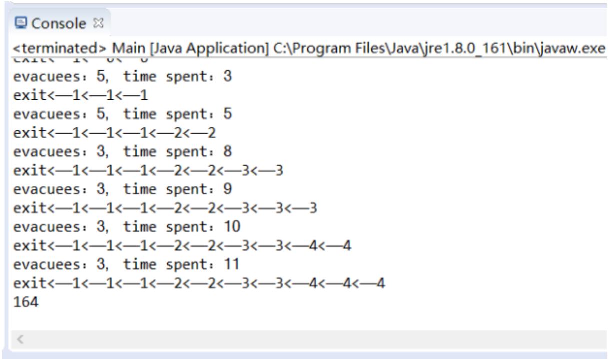


Figure 8: The topographic representation of a part of Louvre



```
Console
<terminated> Main [Java Application] C:\Program Files\Java\jre1.8.0_161\bin\javaw.exe
evacuees: 5, time spent: 3
exit<—1<—1<—1
evacuees: 5, time spent: 5
exit<—1<—1<—1<—2<—2
evacuees: 3, time spent: 8
exit<—1<—1<—1<—2<—2<—3<—3
evacuees: 3, time spent: 9
exit<—1<—1<—1<—2<—2<—3<—3<—3
evacuees: 3, time spent: 10
exit<—1<—1<—1<—2<—2<—3<—3<—4<—4
evacuees: 3, time spent: 11
exit<—1<—1<—1<—2<—2<—3<—3<—4<—4<—4
164
```

Figure 9: The screen shot of results by RMCF Model

back to the corresponding original room number to obtain the result, as shown in Fig.9. The evacuation routes for the last two lines in the figure are as follows:
evacuees:3, time spent10
$exit \leftarrow 1 \leftarrow 1 \leftarrow 1 \leftarrow 2 \leftarrow 2 \leftarrow 3 \leftarrow 3 \leftarrow 4 \leftarrow 4$
evacuees3, time spent11
$exit \leftarrow 1 \leftarrow 1 \leftarrow 1 \leftarrow 2 \leftarrow 2 \leftarrow 3 \leftarrow 3 \leftarrow 4 \leftarrow 4 \leftarrow 4$
It is indicated by the path from the fourth room to the exit after passing through the third room No. 3No. 2 and No. 1. Among the lines with "time spent" of 11 seconds, there is more than one line from "fourth room" to "fourth room" than the line with "time spent" of ten seconds. This means that due to the maximum flow limit of the route, only three people can start to go to Room 3, and the other three need to wait for a time unit in Room 4 and then evacuate from Room 4.

The final figure is the sum of the time of evacuation for all, 164 unit hours, and the average evacuation time for each person is 6 unit time.

For sudden changes in the map, a small amount of changes will be made to the input and the program will be re-run. For example, if an accident occurs in the passage from room 2 to room 1, it is impossible to continue. Then just enter the file and change the available flow for this route to 0. as follows:
Original input: from:2 to:1 flow:6 cost:1
Modified input: from:2 to:1 flow:0 cost:1
Then after re-running the program, the obtained path will automatically avoid the above channel and change to:

evacuees3, time spent11
$exit \leftarrow 5 \leftarrow 5 \leftarrow 5 \leftarrow 5 \leftarrow 5 \leftarrow 4 \leftarrow 4$
evacuees3, time spent12
$exit \leftarrow 5 \leftarrow 5 \leftarrow 5 \leftarrow 5 \leftarrow 5 \leftarrow 4 \leftarrow 4 \leftarrow 4$
Although the number of rooms that have passed has been reduced, the time it takes to escape from each other is longer because of the longer time spent on each room on this route. This is also the case when the route can be evacuated on both sides, the program prefers to travel from $4 \rightarrow 3 \rightarrow 2 \rightarrow 1$. This also proves the correctness of the program from the side.

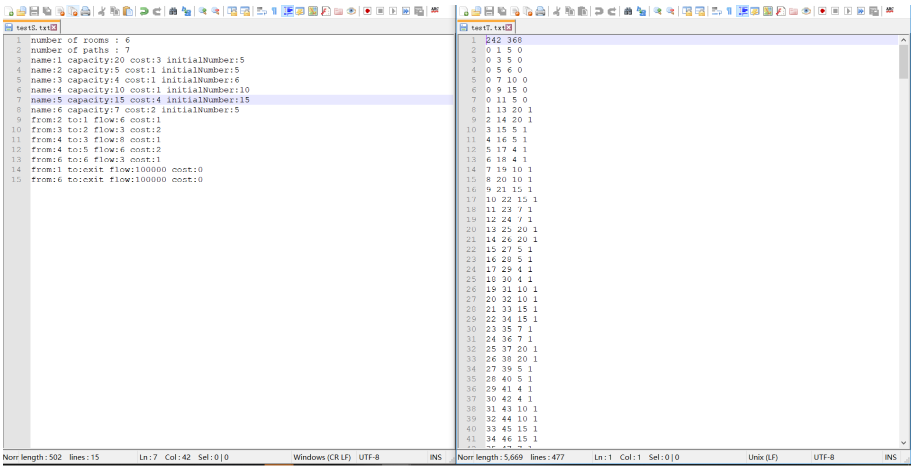


Figure 10: The screen shot of nodes expanded results produced by our RMCF program

6.2 Evaluations of RMCF

After the above test, it can be seen that the RMCF algorithm can give the shortest evacuation result of the total evacuation time of all personnel according to the number of people in the current venue and the terrain structure of the museum. The complete Louvre map has a large amount of data and there is no detailed map of the Louvre. Therefore, this article does not complete the evacuation route planning of the whole Louvre museum. However, the path planning algorithm proposed in this paper is universal and can be used for any map.

This article currently only provides a core algorithm for calculating the evacuation path. If there is sufficient time and money, a complete evacuation procedure can be designed based on our RMCF algorithm.

Any museum can first initialize the program with its own architectural drawings, so that the program completes the creation of the abstract map. The capacity of each room in the hall and the traffic at key intersections can be measured by the staff in the hall, or can be simulated by NetLogo described above and the results can be obtained. These are preparations that can be completed before the terrorist attacks, without affecting the museum's official response speed in the event of an emergency.

At the same time, the monitoring equipment of each exhibition hall in the museum can be used to obtain the images in the exhibition hall, and the computer visual technology is used to automatically count the current time in each area. In the event of a terrorist attack, you can use the real-time information on the number of people in the library, quickly run the evacuation route planning process, get results, and guide accordingly. For some areas that cannot enter, or main exits cannot be used, the corresponding changes can be set through the corresponding interface of the program. The program itself can then modify the abstract map based on the input and run a new evacuation route.

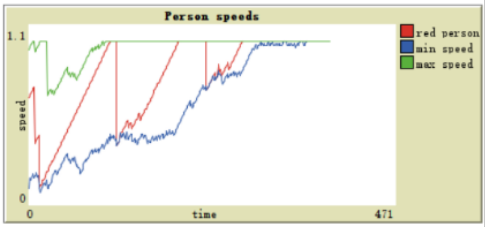
For the case where a rescuer needs to enter the hall, the situation is similar to the above. After the rescuer's entry route is determined, the route can be calculated in the program. The program reduces the traffic of the route by itself, so that the route gets a certain amount of capacity, and rescuers can pass. After the rescuer completes the entry, the original map settings can be restored. For an example of a program application scenario:

1. In the event of a terrorist attack, terrorists placed bombs at the main entrance, causing the main entrance to be impassable. The staff knows the actual situation, use the program, select the main door, make it unreachable, and turn the capacity of the path to zero. Then calculate the evacuation path according to the current situation, and notify the staff in the hall to go to the post and guide accordingly.

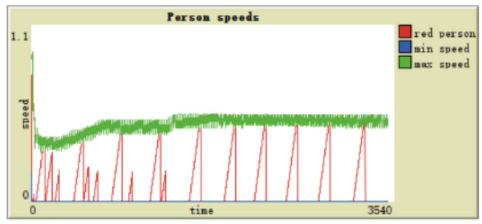
2. A fire broke out in a certain area of the museum, which made the entire area inaccessible. After the staff understands the situation, close all elevator-like passages, close the areas where the fire broke out, and close the fastest path from the main entrance to the fire area as a fast-track for firefighters; then calculate the evacuation path according to the current situation, and notify the staff in the hall to go to the positions and guide accordingly.

3. For disabled visitors in wheelchairs, these visitors cannot use the ordinary stairs to go up and down the stairs. For the evacuation route of these visitors, the staff can make all the ordinary stairs unreachable, leaving only barrier-free access. Then run the algorithm according to this situation and get the evacuation route for the disabled. And inform the staff in charge of the guidance in the museum to conduct targeted guidance for different personnel.

6.3 Accelerations vs. Decelarations



(a) When acceleration is big.

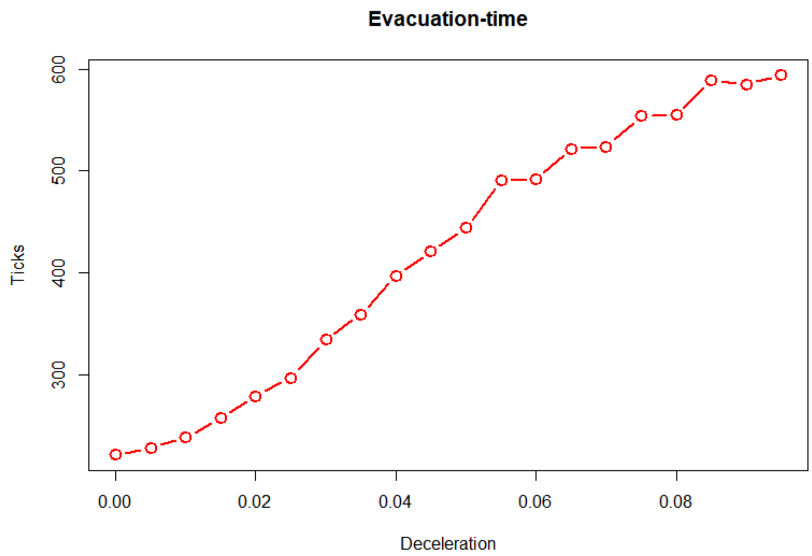


(b) When deceleration is big.

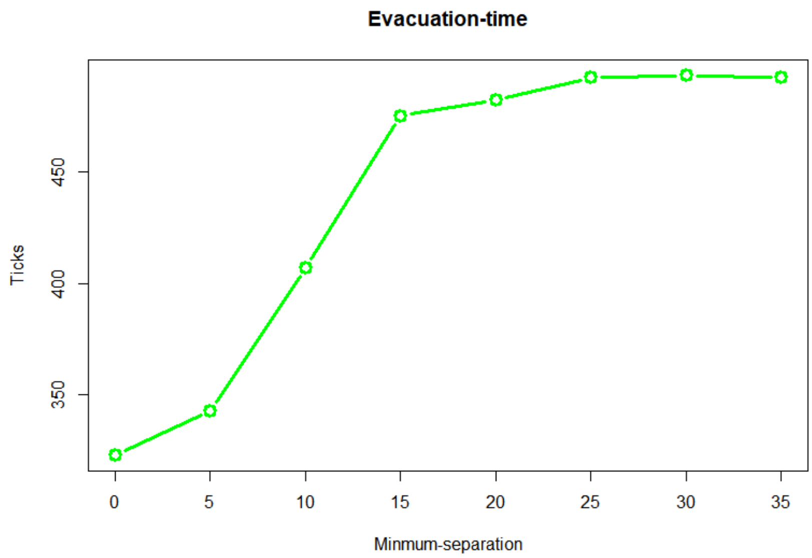
Figure 11: Speed changes according to acceleration and deceleration

Considering high acceleration, when the distance between the latter person and the preceding person is bigger than the limited distance, the latter person will immediately reach the speed of the preceding person, so after a period of time, since the latter person can always follow the previous person immediately, each person advances at the same acceleration with a fixed distance. The speed change is shown in Fig.11.

When it comes to high deceleration, if the distance between the latter person and the former person is less than the limited distance, the latter person will immediately stop, and the influence will be gradually transmitted to the whole crowd. Since the latter person can always stop immediately, the congestion situation It will always appear periodically and eventually stabilize.



(a) Varying deceleration with other parameters fixed



(b) Varying min-separation with other parameters fixed.

Figure 12: Distance and Deceleration Study

6.4 Varying min-separations and decelerations

We think deceleration is a very important mental factor, since people tend to decelerate more under depression and anxiety. The separation rule in our flocking model could avoid collision so that people are less likely to decelerate or collide. However, a big **min-separation** parameter would cause people to over-separate and they could hardly enter the exit. The minimum speed in Fig.12 is set to 0.1, so people would forward at a minimum speed of 0.1, which is the reason why the ticks of evacuation process increase when **min-separation** gets bigger. And the ticks stabilize because vision is limited.

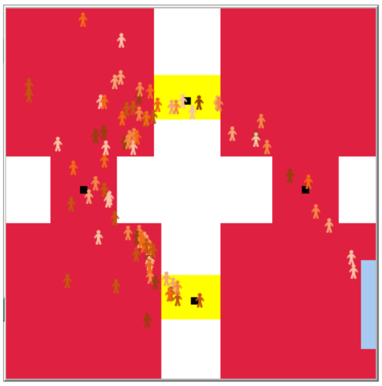


Figure 13: Square Map Abstraction

From Fig.14 we could clearly see the difference, with a pretty low minimum speed, the cost of collision is pretty high so when **min-separation** is close to zero, the evacuation could be really slow; and if min-separation is too big, people need to separate from others frequently disturbing them to go to the destination. From Fig.14 the best setting of min-separation is

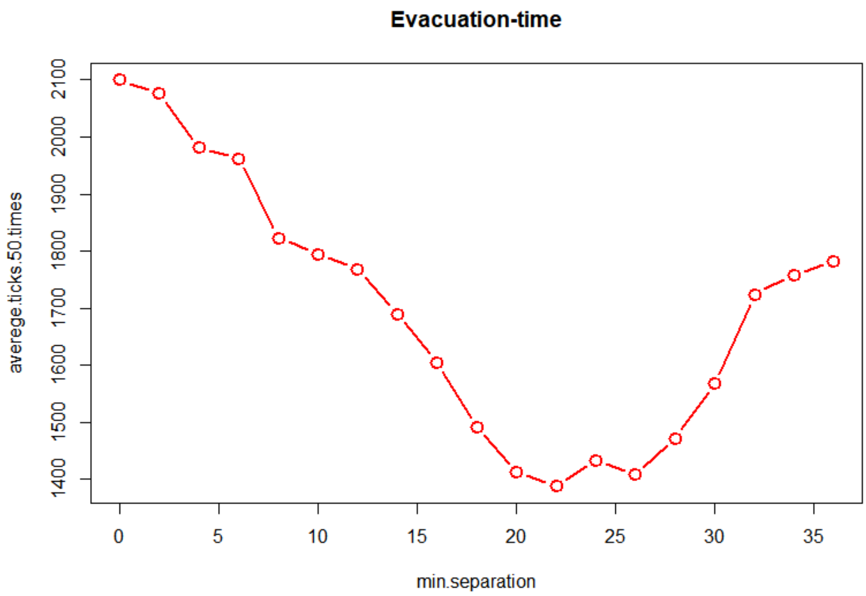


Figure 14: Varying min-separation with 0.01 min-speed

around 22, which we think is the natural group choice for efficiency in reality, and we would also adopt it for further simulations. To clearly simulate the reality, we would set min-speed at 0.01 in further simulations.

6.5 Experiments on Square Map

6.5.1 Decision Tree of Human with multiple gates

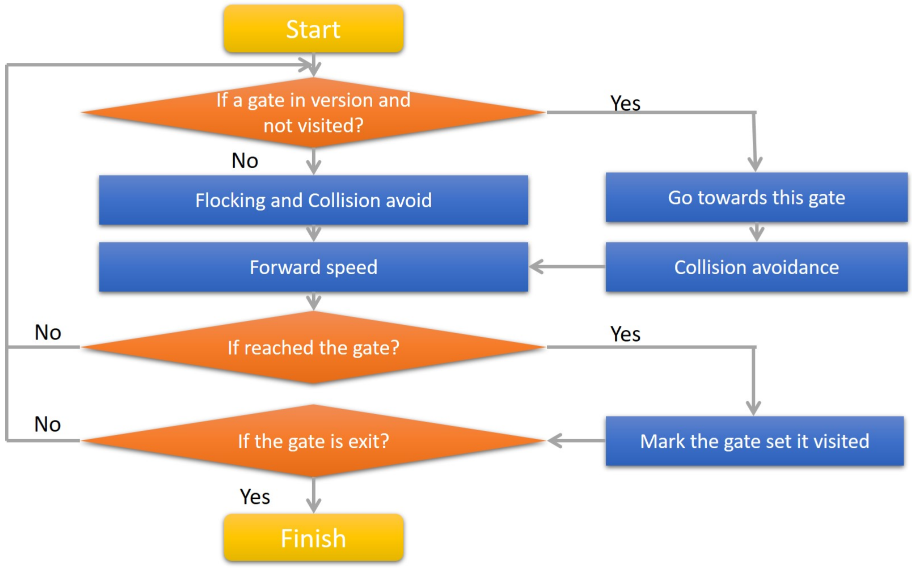


Figure 15: Decision Flow Chart of "turtles" in NetLogo

We have done the simulation of people of only one exit, and situations could be really complex with real images:

1. There would be multiple gates, "turtles" need to adaptively go to the nearest gate and ignore the ones they have already visited.
2. There would be obstacles, so our "turtles" need to automatically avoid collision.
3. There would be stairs where "turtles" need to slow down

To adapt our model to various situations, we propose a Square Map in Fig.13 to represent the real map approximately. The yellow area represents stairs, where "turtles" need to slow down, the narrow area represents the paths, the decision flow chart of "turtles" is given in Fig.15

From Fig.13 we could see "turtles" tend to get crowded in the corners near the gates. This satisfies our intuition of real visitors, because when evacuating people would all tend to go straight lines, causing frequent decelerations and collisions. Even with separation in flocking, since every one run towards gates, alignment would make them impossible to separate enough to avoid collisions. Here we need officials to control the movements of "turtles" to scatter scatter them or control the flow to avoid collisions, and the behavior of microcosmic conduction by officials could be given by our macrocosmic routing algorithm, RMCF. RMCF only takes the time cost and capacity as input, so the graph-based abstraction would not retain such microcosmic information.

To analyze how the officials should respond to various situations, we add four interferences to the model, and run the NetLogo model to get three evaluation metrics, **total evacuation time**, **average evacuation time**, and **standard deviation of evacuation time**, all give as **ticks**.

1. Randomly add several rocks on the map, each has a size of 2×2 patches.
2. Randomly generate "disabled visitors", which are represented by several "turtles" tied together to simulate wheelchairs and reduce their speed to 0.5.
3. Slow down on stairs.
4. Additional evacuation exits.

6.6 Generating Rocks

When we generate rocks with the probability on every patch, we could see from Table.4 that when the rock probabilities are the same, the effect of more people is bigger than more rocks with 0.10 and 0.15 in generation probability. People are good at dodging small obstacles, but if generation probability is 0.20 for every patch and rock probability is 0.1, the average time does not increase pretty much. Compared with Table.2, when generation probability is 0.10, adding some rocks seem to scatter people from crowding and collisions and reduce the evacuation time, while in 0.2 of generation probability the total time decreases but the mean and the standard deviation increases about 30 ticks.

6.7 Different Proportions of the disabled

We represent the disabled as "turtles" with low speed-limit of 0.3 and other "turtles" should turn-away if they meet them to avoid any possible collisions. From Table.5 we can see that, compared with Table.2 the three metrics all increase around 40%, and changes of disabled probability have a large impact on evacuation (33 ticks in average time of generation probability 0.10).

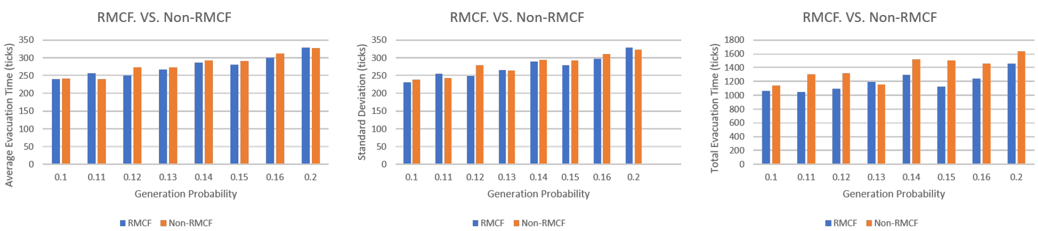


Figure 16: The simulations with and without RMCF supervision

6.8 Stairs that slow down people

When "turtles" are on the stairs (represented as yellow patches), their speed-limit would be reduced to 0.3 the same as the one in disabled limit. Comparing Table.5 and Table.3, we could see that with generation probability of 0.10, stairs model is faster, while things reverse in generation probability of 0.20. This is likely because with more people, it would occur more collisions and decelerations than dodging the disabled.

Generation Probability	Average Evacuation Time (ticks)	Standard Deviation (ticks)	Total Evacuation Time (ticks)
0.10	239.52	231.14	1062.14
0.13	267.61	265.07	1196.52
0.14	286.98	290.10	1297.14
0.15	280.02	279.12	1127.66
0.16	300.12	297.22	1242
0.20	328.89	329.23	1459.19

Table 1: Simulation results conducted by RMCF routing

6.9 Simulations conducted by RMCF routing

We did experiments with routing results from RMCF model, and we get the results as in Table.1. Compared with non-RMCF results in Table.2, we could witness more than 20% decrease in time usage. With generation probability lower than 0.16, the total evacuation time is all lower than 1300 ticks. The agent-based decision model reached 1500 ticks at 0.14 of generation probability.

7 Policy and Procedural Recommeandations

Based on our observations, we recognize several problems for evacuation. The stairs will slow down the people, and with high tension,people tend to decelerate more to avoid collisions. And people tend to walk straight lines, which makes them extremely crowded near the corner. To improve it, we give several suggestions:

1. Add more indications to make full use of paths and avoid collisions.
2. Build special stairs for the disabled.
3. Draw the evacuation route on the floor to separate the flow.

In order to help use the RMCF model, and get enough data,wo also suuggest the following:

1. Add passenger counters in every corner to provide capacity information.

The people evacuate the slowest are those who do not know where the door is or can not read the language. We suggest building the policy to help those who have cultural differences in emergency situations.

8 Strengths and Weaknesses

8.1 Analysis of RMCF Model

The time complexity of the algorithm is $O(F \times E \times T \times log(V \times T))$ (F is the flow rate,E is the number of edges, V is the number of vertices, and T is the number of expansions). Where F is the number of routes to evacuate in this article, taking into account the number of evacuation

Generation Probability	Average Evacuation Time (ticks)	Standard Deviation (ticks)	Total Evacuation Time (ticks)
0.10	241.27	238.81	1138.57
0.13	272.16	263.58	1159.00
0.14	293.11	294.66	1518.90
0.15	290.86	293.03	1502.09
0.16	312.44	311.18	1454.80
0.20	326.45	322.21	1638.42

Table 2: Simulation results of different generation probability

Generation Probability	Average Evacuation Time (ticks)	Standard Deviation (ticks)	Total Evacuation Time (ticks)
0.10	296.39	281.33	1460.19
0.15	347.10	341.52	1829.90
0.20	405.86	389.67	2030.24

Table 3: Results with Stairs

Generation Probability	Rock Probability	Average Evacuation Time (ticks)	Standard Deviation (ticks)	Total Evacuation Time (ticks)
0.10	0.01	226.20	217.58	971.23
0.10	0.10	251.19	246.02	1053.00
0.15	0.01	296.58	296.20	1412.52
0.20	0.10	359.05	353.66	1490.14

Table 4: Randomly generate rocks

Generation Probability	Disabled Probability	Average Evacuation Time (ticks)	Standard Deviation (ticks)	Total Evacuation Time (ticks)
0.10	0.10	301.67	299.70	1570.43
0.10	0.30	334.74	315.89	1661.29
0.15	0.20	358.86	350.52	2013.90
0.20	0.10	393.87	377.14	1953.71

Table 5: Different proportions of the Disabled visitors

routes for different batches and the daily number of visitors to the Louvre, approximately 10^4. The number of rooms and the number of sides are all orders of magnitude below 10^3. The number of expansions T can be estimated and adjusted according to the size of the map and the number of people in the venue, and can generally be limited to 10^2. Since the modern computer operation speed is generally 10^9 or more per second, when calculating the evacuation route of the Louvre-level museum, the calculation result can be obtained in less than one second. So the time complexity of this algorithm is completely satisfactory. At the same time, if you want to further speed up the calculation, you can use the Fibonacci heap to optimize the program, so that the time complexity will be $O(E \times T \times log(log(V \times T)))$

Strengths

1. The algorithm has strong adaptability. He provides an algorithm that be given the corresponding map and the distribution of personnel in the hall to calculate the optimal evacuation plan. This approach is more adaptable than a separate set of specific evacuation options and can easily cope with unexpected situations and unforeseen circumstances.

2. The simulation of the real scene is very real. This algorithm abstracts the map into room and path. The selection of specific parameters is very targeted. For the room, because of its large space and people may stay for a long time, set its parameters to capacity. For path, because its space is small and people can pass in a short time, its parameter is not capacity, but the flow that can pass the number of people per unit time. In many other path planning algorithms, nodes are mostly without attributes, and the edge only has the attribute of traffic.

3. Many other algorithms used in path planning for path planning, such as the simple Dijkstra's Algorithm and the Bellman-Ford Routing Algorithm , can only satisfy the single-source shortest path with the smallest computational cost when calculating the path. But it can not consider global information. However, our algorithm is the shortest strategy for calculating the total evacuation time of all required persons in the case of ensuring a complete global evacuation.

Weaknesses

1. In order to calculate the evacuation route of a map, the algorithm first needs to divide the map into several rooms and paths, and input the corresponding capacity and flow information. This process is best for accurate measurements, but it is possible to use NetLogo simulation instead. The process of input the map is cumbersome and complicated, but the official capacity of the Louvre is relatively easy to do. Fortunately, this work only needs to be initialized once when the evacuation route is calculated for the first time. In the actual work in the future, if a small amount of map transformation occurs, such as blocking of a small area and the interception of the channel requires only a small amount. Modify the processing.

2. The algorithm seeks to calculate the shortest sum of the evacuation time of all the people required to ensure complete global evacuation. But the shortest sum of all evacuation time does not mean that the last person has the shortest time to evacuate. This means that for specially designed data, it is possible to have the shortest time sum, the rapid evacuation of most people, and the fact that very few people have been evacuated for too long; instead of calculating the shortest strategy for the slowest evacuation of evacuators . However, in the actual situation, the probability of occurrence of this situation is small, and the result value sought by the algorithm is also an acceptable better solution.

8.2 Analysis of NetLogo Model

8.2.1 Strengths

1. Using NetLogo for simulation is convenient and fast, which can save a lot of manpower and material resources.

2. Netlogo is highly authentic compared to abstract path planning algorithms. It can reflect the true reactions of human beings in the face of emergencies.

3. Different types of visitors can be programmed separately to design different parameters. Can meet a variety of situations, just need to set the corresponding operating rules

4. The programming language is simple and easy to learn, and it is convenient to make the model you need.

8.2.2 weaknesses

1. In the simulation of large-scale real scenes, because each agent independently decides its own strategy and at the same time it may affect other surrounding agents, a large number of agent simulations may put forward the computing power of the computer. Certain requirements. At the same time, the speed of the simulation will be slower.

2. Because the grammar itself is simple and easy to understand. Therefore, there are some ambiguities in grammatical expression, and there is no rigor of other high-level programming languages. At the same time, due to the small size of the program, there is no debug tool similar to the debug program.

References

[1] Doris Benda, Xiaoli Chu, Sumei Sun, Tony Q. S. Quek, Alastair Buckley: Modeling and Optimization of Renewable-Energy Sharing among Base Stations as a Minimum-Cost-Maximum-Flow Problem. VTC Spring 2018: 1-5.

[2] "How To Visit the Louvre Quickly and Efficiently" Youtube.

[3] Spiliopoulou A D, Papamichail I, Papageorgiou M. Toll plaza merging traffic control for throughput maximization[J]. Journal of Transportation Engineering, 2009, 136(1): 67-76.

[4] Wenwu Yu, Guanrong Chen, Ming Cao: Distributed leader-follower flocking control for multi-agent dynamical systems with time-varying velocities. Systems and Control Letters 59(9): 543-552 (2010).

[5] Reza Olfati-Saber: Flocking for multi-agent dynamic systems: algorithms and theory. IEEE Trans. Automat. Contr. 51(3): 401-420 (2006).

[6] "Min-Cost Flow Problem"https://en.wikipedia.org/wiki/Minimum-cost_flow_problem,Definition, Wikipedia.

[7] "Flow Network"https://en.wikipedia.org/wiki/Flow_network,Wikipedia.

[8] Elisa Valentina Moisi, Benedek Nagy, Vladimir-Ioan Cretu: Maximum flow minimum cost algorithm for reconstruction of images represented on the triangular grid. SACI 2013: 35-40

[9] Makhlouf Hadji, Djamal Zeghlache: Minimum Cost Maximum Flow Algorithm for Dynamic Resource Allocation in Clouds. IEEE CLOUD2012: 876-882

[10] T. H. Szymanski: Achieving minimum-routing-cost maximum-flows in infrastructure wireless mesh networks. WCNC 2012: 2031-2036

[11] Hao Xu, Luis Rodolfo García Carrillo: Distributed Optimal Flocking Design for Multi-agent Two-Player Zero-Sum Games with Unknown System Dynamics and Disturbance. ISVC (1) 2016: 603-614

[12] "The number of visitors" http://news.meishujia.cn/?act=app&appid=4096&mid=64870&p=view

[13] Center for Connected Learning and Computer-Based Modeling, http://ccl.northwestern.edu/netlogo/4.0.4/docs/NetLogo_manual_chinese.pdf,2008

[14] "Famous Attractions in France"https://www.parisdigest.com/france/famous-places-in-france.htm

[15] Distributed Tracking Algorithms for Multi-Agent Systems to Solve the Leader-Follower Flocking of Lagrange Networks and Dynamic Average Tracking Problem of Second-Order Systems. University of California, Riverside, USA 2016