Team Control Number

1919953

Problem Chosen

D

For office use only
T1 _____
T2 _____
T3 _____
T4 _____

For office use only
F1 _____
F2 _____
F3 _____
F4 _____

2019
Interdisciplinary Contest in Modeling (ICM)
Summary Sheet

Optimization of the Evacuation Plan for the Louvre

Summary

On the night of Nov.13, 2015, thousands of Paris residents, reveling tourists, and fans who were enjoying a soccer match between France and Germany, got struck by horror of an unprecedented terror attack. Ever since then, terrorists have been constantly impacting France with barbaric manner. Located in Paris, the Louvre, a magnificent palace previously, attracts millions of visitors every year. An ill-planned evacuation route would be devastating to both the visitors and the museum.

We are tasked to develop an emergency evacuation model to help the museum leaders explore options to evacuate visitors. Additionally, we should allow emergency personnel to enter the building quickly.

We develop a basic model of undirected graph (UGE) to simulate the process of evacuation. We creatively use Volumetric flow rate equation as an analogy to the movement of evacuees, ulteriorly develop Fluid Flowing Evacuees (FFE) model to calculate the edge weight in UGE. In consideration of group visitors and modernized technologies, we combine this model with Cellular Automation (CA) model to specify this process. The combination helps us find certain bottlenecks in an emergent situation.

In terms of the optimization of evacuation plan, we find an excellent but not optimal solution by Bellman-Ford algorithm. We are exhilarated to discover that the algorithm fits the property of Single-source Shortest Path problem well. Thus, we develop a complete model of optimization that is quite interpretable as well as efficient with little loss of efficiency in the result. We use Stimulate Annealing algorithm in the hope of obtaining optimal solution, but soon to find it far too time-consuming to calculate and of no help in the optimization.

Afterwards, we specify the model by introducing entrance of personnel and considering "unsafe" extra exits. The two variances are both modifying parameters of models we develop. As for the former, we divide the problem into two parts: optimization of total evacuation time and that of entering time. We use BFS with labeling method to solve it. The latter requires us to rectify the evaluation criteria. Hence we propose the Cost Evaluating Model (CEM) to solve it properly. However, Due to the lack of practical research, this model is incomplete.

Finally, we analyze the sensitivity and efficiency of our models, during which they show high stability, extensive applicability and efficiency.

Contents

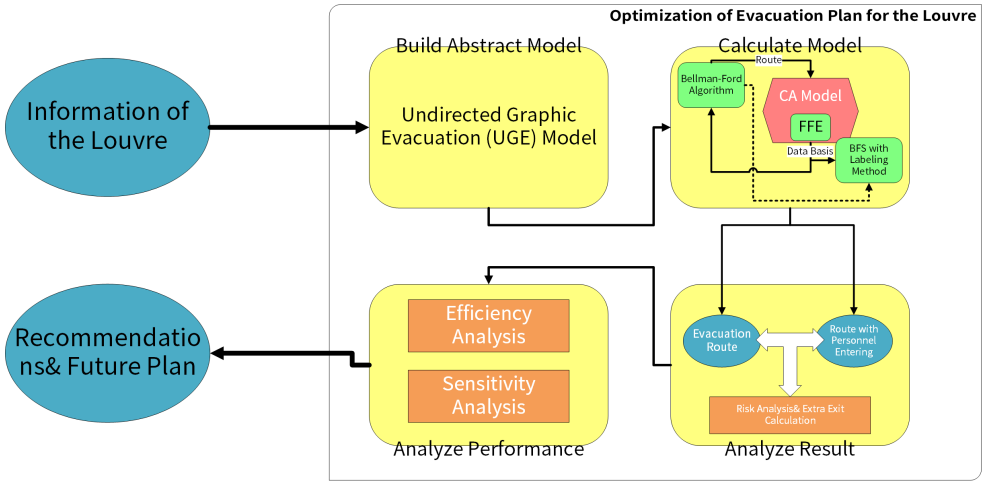


Figure 1: Structure of the Paper

1 Introduction

1.1 Problem Background

France, a country pursuing freedom, equality, and fraternity, has been constantly suffering from terror attacks in recent years[6], which evoked great indignation as well as consternation all over the world. Aside from terrorism itself, the situation of security in Europe, especially in France, raises awareness of effective evacuation, by which could save a great number of people when an emergency occurs. Thus, an optimization of evacuation plan becomes imperative and necessary.

Figure 2: Pyramide du Louvre, Paris, France[8]

As one of the world's largest and popular art museums, the Louvre receives millions of visitors from different countries every year[5]. Attracted by the masterpieces in it and its magnificent architectures, they pack each other in shoulder to shoulder to admire these artworks. It would be a disaster if these visitors evacuate blindly when an emergency happens.

1.2 Literature Review

Due to the rapid pace of urbanization caused by industrialization and commercialization, high rise buildings have been constructed in order to accommodate the large migration from rural areas. Since then, in the attempt to ensure safety in the time of an emergency (e.g.fire accidents) in these buildings, researchers have been conducting experiments[3, 4], simulation models, and algorithms to study the evacuation dynamics of complicated buildings under emergency. Liu et al. studied and established a model combining net work flow control and heuristic algorithm in 2016 [2]. Sheeba and Jayaparvathy developed a performance model of emergency evacuation on accidental fire occurrence, using generalized stochastic petri nets(SPN)[7].

1.3 Problem Analysis

The problem as a whole seems complicated to solve, so we divide it into five smaller parts.

1. **Developing an emergency evacuation model.**

   This model should describe visitors' action during emergency in the Louvre, taking necessary and crucial parameters into consideration.

2. **Optimizing the evacuation plan.**

   Considering the efficiency of evacuation primarily, we decide to seek for an optimal plan to guarantee it in the first place when in an emergency.

3. **Rectifying the plan by introducing entrance of personnel.**

   After the efficiency of retreat is solved, we try to complete this plan by turning several exits into entrance, then optimize it under a new condition.

4. **Specifying the model by considering "unsafe" extra exits.**

   Other available exit points(service doors, employee entrances, and old secret entrances built by the monarchy, etc.)play a more complex role in this optimization. We strive to keep the balance between speed and security with quantified measure.

5. **Drafting a proposal of policy and procedural recommendation.**

   Based on the result that we have achieved, we will propose the plan for emergency management of the Louvre.

2 Preparation of the Models

2.1 Assumptions

In order to solve this problem properly, we base our model on assumptions below:

1. **Only routes crowded with visitors cost time to pass.**

   Compared with the routes that are crowded with visitors, void ones cost so little time that we can ignore them.

2. **Visitors always choose the best route throughout evacuation.**

   Because of widely-used smartphones and convenient Internet, we assume that visitors keep up with the latest information during emergency evacuation. More flexible and electronic indicators can be equipped in the Louvre to guarantee this.

3. **Evacuation of dense crowd in a narrow passage is similar to liquid's flowing in a tube.**

4. **Visitors in the same gallery evacuates together.**

   Concerning the chaos when emergency occurs, it is both infeasible and unnecessary to direct panic visitors in a same gallery to evacuate separately. Some visitors are traveling in groups, which means they can be unwilling to separate from each other. In addition, due to the dense arrangement of galleries, this assumption can tremendously simplify the problem with little loss of efficiency.

5. **The total number of people on three steps in a staircase equals to the number of people standing in two rows on flat ground.**

2.2 Nomenclature

The primary notations used in this paper are listed in **Table 1**.

Table 1: Nomenclature

Symbol	Definition
v	Certain vertex
(u, v)	Certain edge between vertex u and vertex v
Cap_v	Capacity of vertex v
$P_{v,t}$	Current number of people in vertex v at moment t
$Q_{(u,v)}$	Maximum number of evacuees passing edge per second(u, v)
Q	Flow rate of certain fluid
$\boldsymbol{v}$	Flow velocity of the fluid
$\boldsymbol{A}$	Cross-sectional vector area
Q_e	Quantity of evacuees passing per unit time
$\boldsymbol{v_e}$	Velocity of the movement of evacuees
v_s	Speed of the movement of evacuees, which is a scalar quantity
$\boldsymbol{A_e}$	Vector of evacuees' passages
A_s	Quantity of evacuees per unit length
θ	Angle between the unit normal $\boldsymbol{n}$ and $\boldsymbol{v_e}$
$v_{corridor}$	Speed of evacuees in corridors
v_{stair}	Speed of evacuees on the stairs
$flow_{<u,v>,t}$	Number of people evacuates from u to v at moment t
$flow_{<u,v>}$	Sum of people evacuates from u to v

3 Model Construction

3.1 Undirected Graphic Evacuation Model

To describe the process of evacuation more concisely, we develop the Undirected Graphic Evacuation (UGE) model by abstracting the structure of the Louvre into an undirected graph $\boldsymbol{G}(\boldsymbol{V}, \boldsymbol{E})$. In this graph, a vertex stands for one room or several connected, relatively independent rooms; an edge stands for a corridor, an aisle, a

stair, or **an extra exit open to evacuees**. In particular, **vertex *0* stands for the safe area**. For the Louvre, the total number of vertexes is 85 and the total number of edges is 131. All the detailed data are in the appendix.

Cap_v depends on the size of the room and whether the room is open, while $P_{v,t}$ can be calculated according to the positioning information feedback from visitors' cellphones. Particularly, Cap_0 is $+\infty$.

For undirected edges (u, v), we set one parameter: $Q_{(u,v)}$, standing for the maximum number of evacuees passing this edge per second (despite whether they are evacuating from u to v or from v to u). We assume **the current number of people in an edge is always 0**. If one is standing in an edge $(\boldsymbol{u}, \boldsymbol{v})$, we regard him as in the nearest room, namely v or u.

A single example is shown in **Figure 3**.

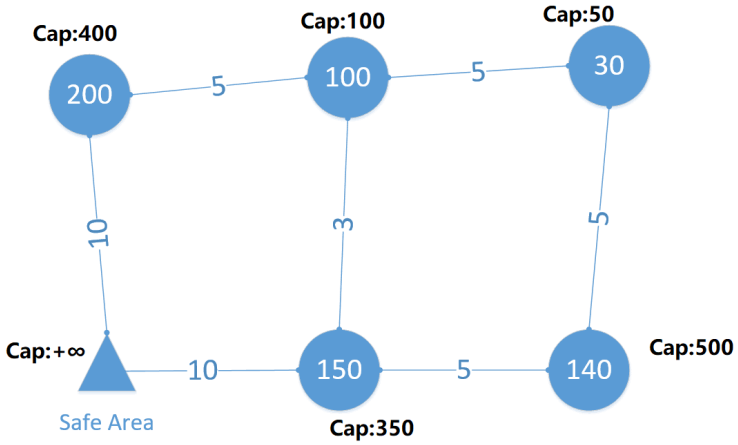


Figure 3: An example of the undirected graph $G(V, E)$

3.2 Evacuation Model

3.2.1 Overview

We develop a basic model of emergency evacuation in large buildings like the Louvre, simulating behaviors of evacuees in this situation. First of all, we mainly study the evacuees' behavior in edges, and set up an Fluid Flowing Evacuees(FFE) model to simulate the evacuation in edges. Afterwards, we use the Cellular Automaton(CA) model to simulate the whole evacuation in the Louvre **with a certain evacuation scheme**.

3.2.2 Fluid Flowing Evacuees model

We find that the evacuation of dense crowd in a narrow passage is very similar to liquid's flowing in a tube. Therefore, we develop a Fluid Flowing Evacuees(FFE) model to stimulate evacuation in the edges.

As for liquid's flowing in tubes, the relation between Q, $\boldsymbol{v}$, and $\boldsymbol{A}$ called *Volumetric flow rate equation*, is

$$Q = \iint_{\boldsymbol{A}} \boldsymbol{v} \cdot d\boldsymbol{A}. \tag{1}$$

(1) is a surface integral, includes curved surfaces. Particularly, for flat, plane cross-sections, the equation becomes a dot product type:

$$Q = \boldsymbol{v} \cdot \boldsymbol{A}. \tag{2}$$

Similarly, we can calculate approximate "flow rate" of evacuees according to equations above. In the Fluid Flowing Evacuees model, we choose (2) to stimulate evacuation in edges, out of the consideration that topography of edges in the Louvre is changeless and flat. We redefine $\boldsymbol{A}$ as $\boldsymbol{A_e}$ in representation of the **average number** of evacuees in a **cross section**. Q_e and $\boldsymbol{v_e}$ are defined similarly to Q and $\boldsymbol{v}$, which are shown in **Figure 4**.

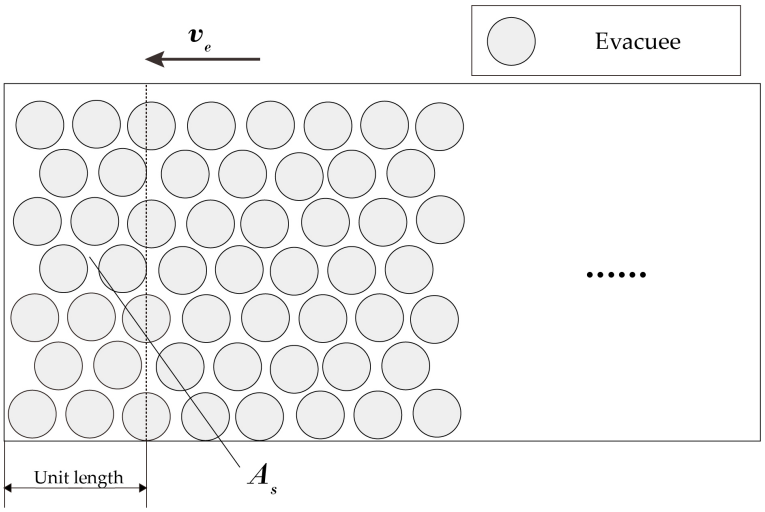


Figure 4: Schematic diagram of FFE model

Then we get

$$Q_e = \boldsymbol{v_e} \cdot \boldsymbol{A_e} \tag{3}$$

where the vector area is a combination of the quantity of the area through which evacuees pass, A_s, and a unit vector normal to the area , $\boldsymbol{n}$. The relation is

$$\boldsymbol{A_e} = A_s \boldsymbol{n}. \tag{4}$$

Based on our assumption, the velocity of evacuees is always parallel to the unit normal, which means $\theta = 0$. So we get the amount of Q_e :

$$\begin{aligned} Q_e &= v_s A_s \cos\theta \\ &= v_s A_s. \end{aligned} \tag{5}$$

We assume that the speed of evacuees on the stairs is equal to normal walking, while the speed of evacuees in the corridors is equal to normal jogging. So we can get estimation for $v_{corridor}$ and v_{stair}. Additionally, based on our measurements on the scale map, a corridor and a staircase measure both approximately 3.2m in width. Generally,

we suppose one person occupies an area of $0.4m \times 0.6m$ in the crowd on flat ground, so we can calculate $A_{corridor}$. According to our measurements of the stairs in public places, the total length of three steps approximately equals to 1m. Then we can calculate A_{stair}.

Using the data above we can calculate $Q_{corridor}$ and Q_{stair}. The results are listed in **Table 2**.

Table 2: Results of v_s, A_s, and Q_e

	Corridor	Stair
$v_s/m \cdot s^{-1}$	2.7	1.4
$A_s/person \cdot m^{-1}$	12.5	10
$Q_e/person \cdot s^{-1}$	33	14

3.2.3 Cellular Automation Model

We develop a special Cellular Automation (CA) model $\mathbb{A} = (L, d, S, N, f)$ to simulate the whole evacuation process **under a certain evacuation scheme**. In this model, a cell corresponds to a vertex.

L: Cellular Space Although the earth can be abstracted as a very large undirected graph according to the above rules (mentioned in **3.1**), we only consider the vertexes and edges corresponding to the Louvre, which means the cellular space can be illustrated as $\{v \mid v \in G(V, E)\}$

d: Dimension of the Cellular Space In this model, we let $d = 1$: the index of the vertex in $G(V, E)$

S: Domain of the States of a Cell We define the state of cell v as the current number of people in the room corresponding to vertex v, i.e. $P_{v,t}$. Therefore, the Domain can be illustrated as $\{P_{v,t} \in \mathbb{Z} \mid \ 0 \leq P_{v,t} \leq Cap_v\}$.

N: Neighborhood of a Cell N depends on the evacuation scheme. Firstly, we define $Path < u, v >$ as the **directed edge** in a evacuation route. Then the neighborhood of cell v can be illustrated as:

$$\{u \mid u \in Path < u, v >\} \cup \{u \mid u \in Path < v, u >\} \tag{6}$$

f:Rules for State Updates According to the state at moment t, the state at moment $t + 1$ is calculated under the following three rules:

1. If there exist a $Path < u, v >$, i.e. v **is u's parent in the tree**, then vertex v must be updated before vertex u is updated.

2. For $Path < u, v >$, there are evacuees moving from u to v at moment t**if and only if**:

$$\begin{cases} P_{v,t} < Cap_v \\ P_{u,t} > 0 \end{cases} \tag{7}$$

The number of evacuees moving through this edge at moment t satisfies:

$$flow_{<u,v>,t} \leq Q_{(u,v)}. \tag{8}$$

3. For a vertex v, if there is more than one path pointing to v, then we assume that evacuees move out of these edges **in turn** (Until $P_{v,t} = Cap_v$).

4. For a vertex v at moment t, if all the vertexes in its subtree is empty:

$$\sum_{u \in Subtree_v} P_{u,t} = 0 \tag{9}$$

then this vertex will be empty all the time. We define $Last_v$ as the earliest moment when this condition is satisfied in vertex v.

The pseudo-code for implementing this cellular automaton model is shown in **Program 1**.

Program 1: `CA Model`

```
def Update(v,t)
    if equation(9) is satisfied then:
        Last[v]:=t
        return
    end if
    while P[v,t]<Cap[v] do:
        if every u in Path<u,v> does not satisfies (P[v,t]>0 &&
        Flow<u,v,t> < Q(u,v)) then:
            break
        end if
        for each Path<u,v> satisfies P[v,t]>0 && Flow<u,v,t> < Q(u,v) do:
            P[u,t]:=P[u,t]+1
            P[v,t]:=P[v,t]-1
            Flow<u,v,t>:=Flow<u,v,t> + 1
        end for
    end while
    for each Path<u,v> do:
        Update(u,t)
    end for
end
def CA()
    get the initial state
    t:=0
    while equation(9) is satisfied for at least one vertex except vertex 0 do:
        state t+1 := state t
        t:=t+1
        Update(0,t)
    end while
end
```

A working example is shown in **Figure 5** and **Figure 6**. Note that red edges are the evacuation route.

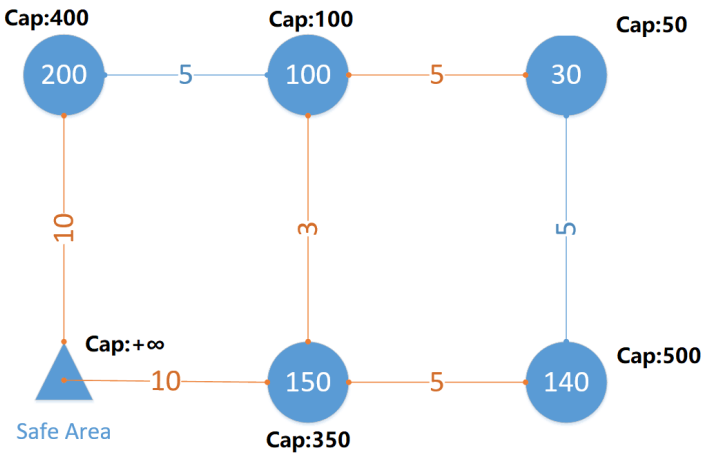


Figure 5: State t

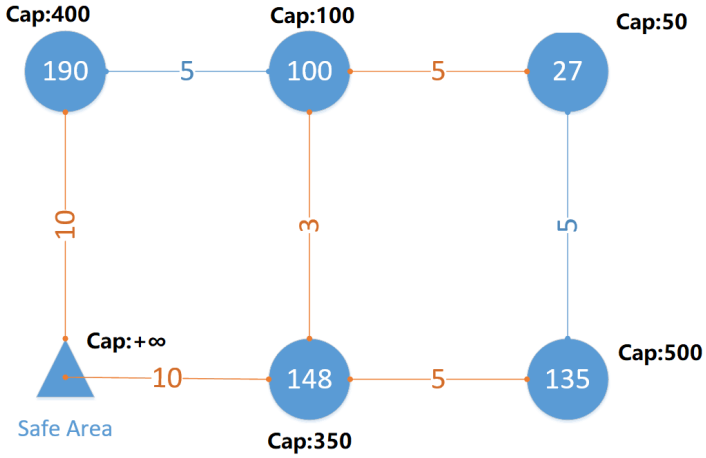


Figure 6: State $t+1$

The Role of CA Model From CA model, we can get important data: $Last_v$. We can figure out the **total time taken for the whole evacuation**:

$$TotalTime = \max_{v \in G(V,E)} Last_v \tag{10}$$

We can also identify the **bottlenecks** in a evacuation route – if vertex v satisfies: for every $Path < u, v >$, the earliest moment when vertex u is "completely finished" is always much earlier than the moment in vertex v:

$$Last_u \ll Last_v \tag{11}$$

The example is shown in **Figure 7**

3.3 Bellman-Ford Algorithm for Evacuation Routes

Based on Evacuation model, we can calculate the total evacuation time *Totaltime* in routes. In order to optimize evacuation routes by calculation, we define TP_v for a vertex v as:

$$TP_v = \sum_{u \in route_v} P_{u,0} \tag{12}$$

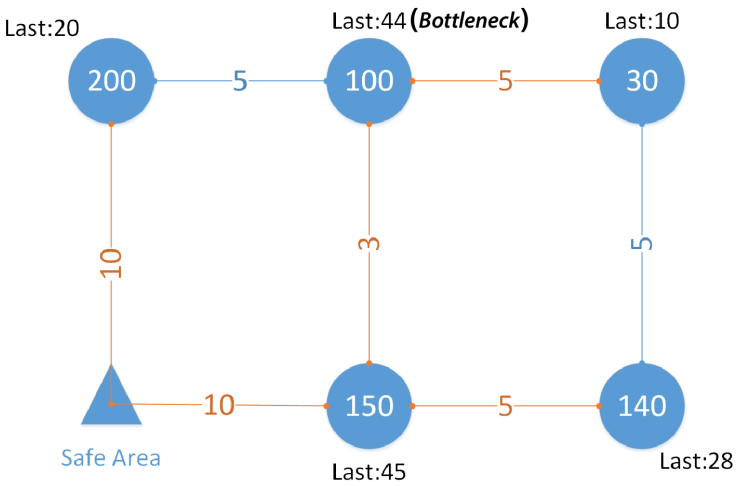


Figure 7: $Last_v$ and the bottleneck

where $route_v$ refers to the evacuation route from vertex v to vertex 0. TP_v is used to estimate the total evacuation time spent by people in vertex v. The reason is that TP represents the total number of people "in front of them" in the evacuation route, and the latter is obviously positively correlated with "waiting time in queue". Now we have two relations: the total evacuation time equals to the time spent by the last evacuee to leave the Louvre, and the time spent by him is positively correlated with TP_v:

$$\begin{cases} TotalTime = \max\limits_{v \in G(V,E)} Time_v \\ Time_v \propto TP_v \end{cases} \tag{13}$$

Therefore, our target to minimize $TotalTime$:

$$\hat{routes} = \arg\min_{routes} \max_{v \in G(V,E)} Time_v \tag{14}$$

can be transformed into (although not equivalently):

$$\hat{routes} = \arg\min_{routes} \max_{v \in G(V,E)} TP_v \tag{15}$$

which is **a typical Single-source Shortest Path Problem**, where all the routes form a tree whose root is vertex 0. We use the Bellman-Ford Algorithm to solve it. **The evacuation routes are exactly the shortest paths**. Notably, according to the property of the Single-Source Shortest Path, the paths eventually form a tree, whose root is **also vertex *0***, which means our evacuation route happens to fit that previous requirement.

An example is shown in **Figure 8**.

3.4 BFS with Labeling Method for personnel's routes

3.4.1 Overview

The reason we calculate the personnel's routes at last is that personnel cannot enter the scene **against the flow of evacuees**, which is extremely perilous. Hence before

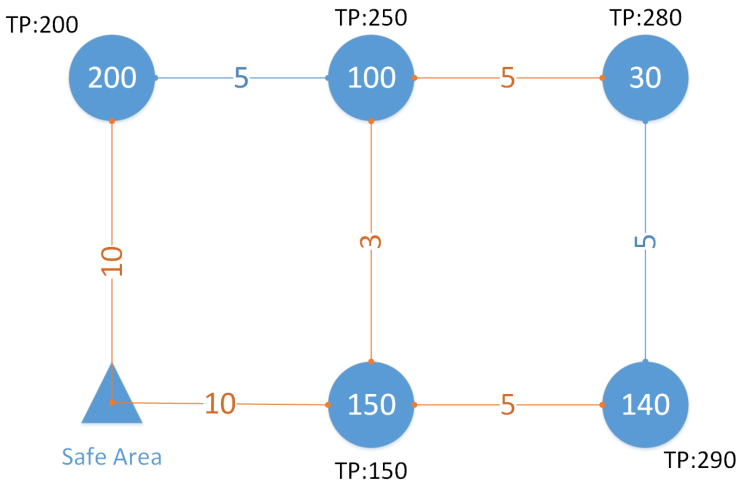


Figure 8: TP_v and the Evacuation Routes

we calculate the personnel's routes, we have to get $Last_v$, which is calculated by CA model.

According to assumption 1 (only "waiting" costs time), we can define the total time spent by the personnel $Enter_v$, whose terminal is vertex v as:

$$Enter_v = \max_{u \in EnterRoute_v} Last_u \tag{16}$$

while the target is

$$\hat{EnterRoute_v} = \arg\min_{EnterRoute_v} Enter_v \tag{17}$$

Note that **equation** (17) is different from **equation** (14) because of the assumption that the number of personnel is small, so there will not be any "crowding". Consequently, different terminals can be calculated separately.

3.4.2 Validation

The **equation** (17) can be solved by BFS with labeling method. Before implementing this algorithm, we should **open all the "extra exit" for the personnel**, because the "potential safety hazard" is just for those visitors.

It is quite convoluted to describe BFS with labeling method algorithm with mathematical language, so we have to use pseudo-code to interpret the process of algorithm implementation.The pseudo-code for implementing this algorithm is shown in **Program 2**.

Program 2: `BFS with Labeling Method`

```
def BFS ()
    for each Enter[v] do:
        Enter[v]:=MAX_INT
    end for
    List.push(0)
    while List is not empty do:
        u=List.front()
        List.pop()
```

```
        for each v in edge(u,v) and Extra-edge(u,v) do:
            if max(Enter[u],Last[v])<Enter[v] then:
                Enter[v] := max(Enter[u],Last[v])
                Update EnterRoute from vertex 0 to vertex v
                if v is not in List then:
                    List.push(v)
                end if
            end if
        end for
    end while
end
```

An example is shown in **Figure 9** .

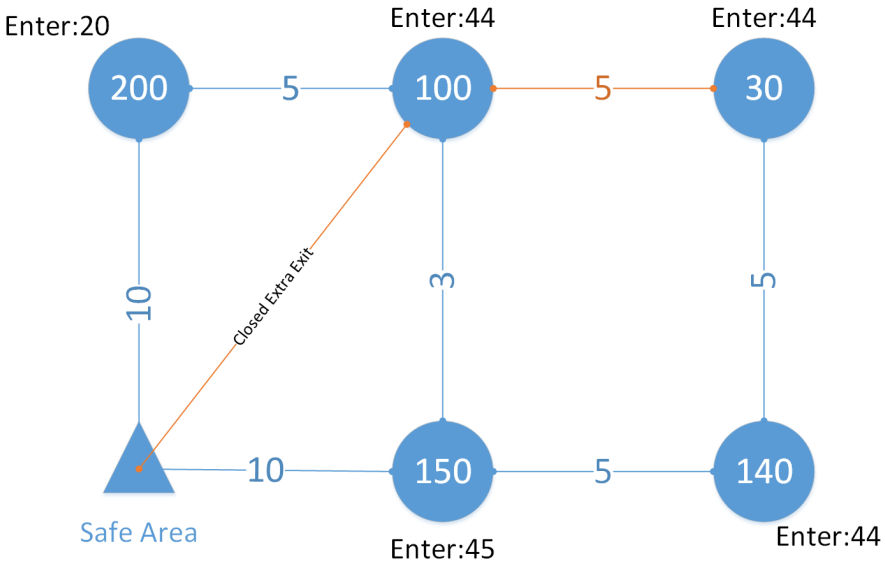


Figure 9: $Enter_v$ and the Personnel Routes (accident happens at the vertex in top right corner)

3.5 Cost Evaluating Model for Opening Extra Exit

It is widely acknowledged that opening an extra exit can provide additional strength to an evacuation plan but causes potential safety hazard. Therefore, in order to comprehensively assess the risk of open extra exits, we develop a Cost Evaluating Model:

$$Cost = C_{eva} + C_p + C_{exit} \tag{18}$$

where

$$C_{eva} = \sum_{v \in G(V,E), t \leq TotalTime} E_{eva}(v, t, P_{v,t}) \tag{19}$$

$$C_p = \sum_{v \in terminals} E_p(v, Enter_v) \tag{20}$$

$$C_{exit} = \sum_{(u,v) \in extra\ exit} E_{exit}(flow_{u,v}) \tag{21}$$

Equation (19) calculates the expected number of casualties in evacuation process. These causes include but are not limited to: stampedes, carbon monoxide poisoning,

etc. The value of the function is related to many factors such as accident type and location.

Equation (20) calculates the expected number of casualties mainly related to the arrival time of the personnel.

Equation (21) calculates the expected number of casualties due to the safety hazard. Once the accident type and location are known, the above three functions are basically determined. Then we can use enumeration method to determine each extra exit should be open or closed.

Equation (18) calculates the total expected number of casualties. Once the accident type and location are known, the above four functions are basically determined. Then we can use enumeration method to determine each extra exit should be open or closed, that is:

$$\hat{decision} = \arg\min_{decision} Cost \tag{22}$$

3.6 Optimization Plan for the Louvre

3.6.1 Overview

Specifically focus on the Louvre, we firstly abstract the map into an undirected graph, then use the models above to calculate an optimized emergency evacuation plan for the Louvre.

3.6.2 Abstraction of Map into Graph

We redistribute the areas in each floor of the Louvre, then number them afresh. An example of this abstraction is shown in **Figure 10**.

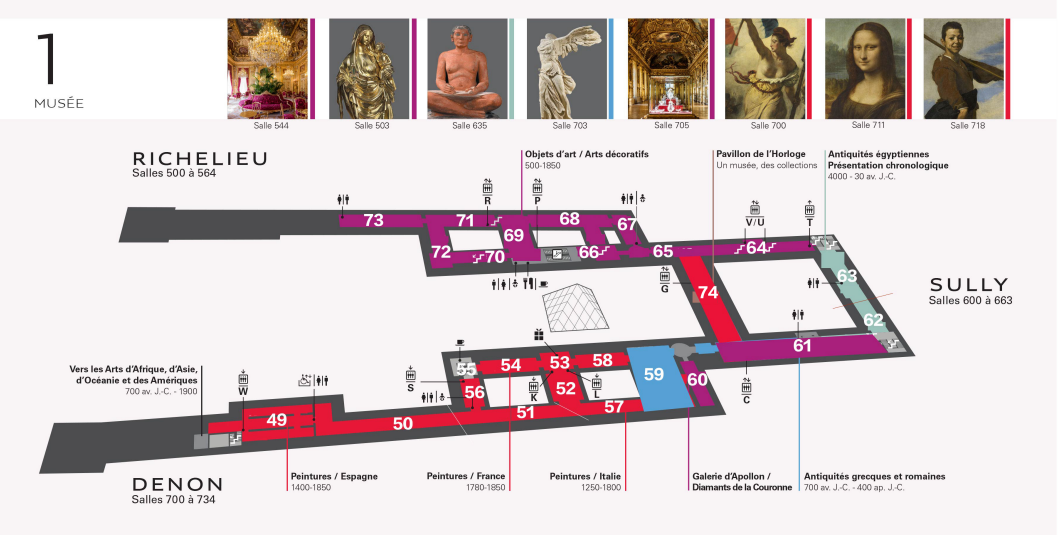
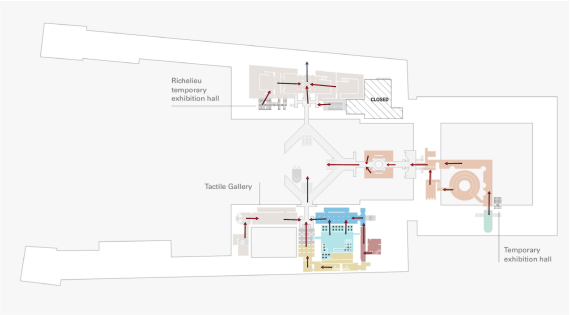


Figure 10: Renumbered Floor 1 of the Louvre

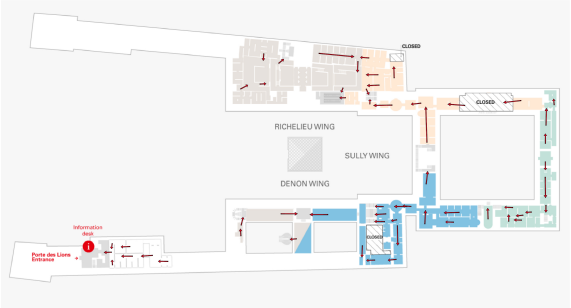
Afterwards, we determine Cap_v of each v , $P_{v,t}$, and $Q_{(u,v)}$ by measuring the size of every gallery, corridor and room. Assuming that 70% of the space in every gallery and room is able to contain people, synthesizing the basic room that a person need to move freely, which is approximately $0.75m^2$, Cap_v can be calculated. Based on $Q_corridor$ and Q_stair that are calculated in 3.2.2, $Q_{(u,v)}$ is determined. $P_{v,t}$ is estimated by popularity of artworks with a randomized fluctuation. Thus, basic parameters of the models are determined completely.

3.6.3 Emergency evacuation plan for the Louvre

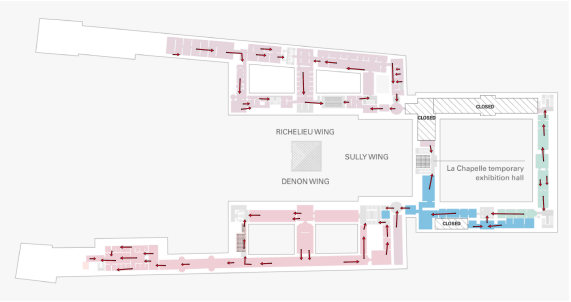
After abstraction, we use models to optimize the plan in the Louvre. The result is shown in **Figure 11**. Note that Floor -2 is an entrance, so we omit its plan.



(a) Floor -1.



(b) Floor 0.



(c) Floor 1.

(d) Floor 2.

Figure 11: Emergency Evacuation Plan for the Louvre

4 Model Analysis: Sensitivity and Efficiency

Our models contain several parameters, most of which are determined by knowledge in the article and standardized measures. In the following section, we produce sensitivity analysis to show whether our model is sensitive to different values of parameters in the mean time to analyze the efficiency by using the same map.

According to Aharon Ben-Tal et.al[1], we note that if we want to find an optimal solution, a graph with only 27 nodes already costs 4 hours to run. Therefore, it is **impossible to find the optimal solution to Louvre** because of the large scale. However, we can prove that our model works efficiently.

4.1 Robustness of parameter TP_v

We prove the strong correlation between TP_v and $Time_v$. For comparison, we define another parameter $Dist_v$ as:

$$Dist_v = \sum_{(x,y)\in route} Len_{(x,y)} \tag{23}$$

Where $Len_{(x,y)}$ is the **actual distance** between two adjacent vertexes x and y.

We randomly pick V (V=20, 50, 80) vertexes and corresponding edges in $G(V, E)$ (contains 85 vertexes and 131 edges), so we get a new graph $G'(V, E)$. Then by Bellman-Ford Algorithm we can get the certain evacuation routes. Note that we simply replace the TP_v with $Dist_v$ and then we can get the routes corresponds to $Dist_v$ by the same Bellman-Ford algorithm. Then we use our CA model to calculate the $Time_v$ corresponding to the two routes respectively. Now for vertex v, we get $Time_v$ and TP_v ($Dist_v$ for "Dist routes"). We gather these two parameters of all vertex $v \in G'(V, E)$ and then we get two sets $\{Time_v\}$ and $\{TP_v\}$ ($\{Time_v\}$ and $\{TP_v\}$ for "Dist routes") .

Then we calculate the correlation coefficient between TP_v and $Time_v$ ($Dist_v$ and $Time_v$ for "Dist Route").The result is shown in **Figure 12**.

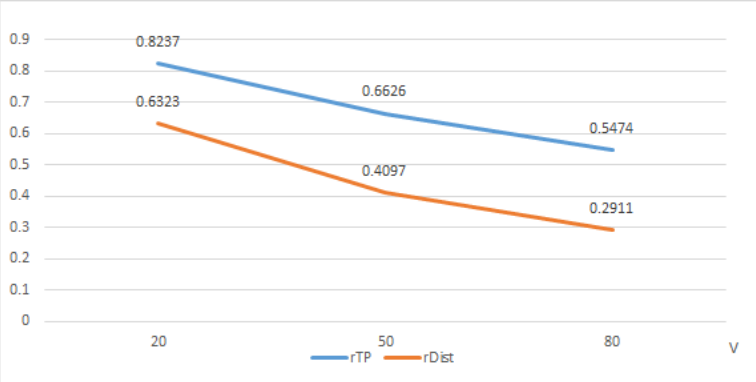


Figure 12: Correlation Coefficient

We can see that r_TP is **always greater than 0.5**, which suggests a strong correlation between TP_v and $Time_v$.

4.2 Coefficient of vertex weight and capacity

We define coefficient k as the ratio of vertex weight and capacity, k describes the extent of crowdedness of galleries.

$$k = \frac{\sum_{u\in route_v} P_{v,0}}{\sum_{u\in route_v} Cap_{v,0}} \tag{24}$$

As shown in **Figure 13**, as k increases, $Totaltime$ stays stable at first, but surges when k approaches 1.

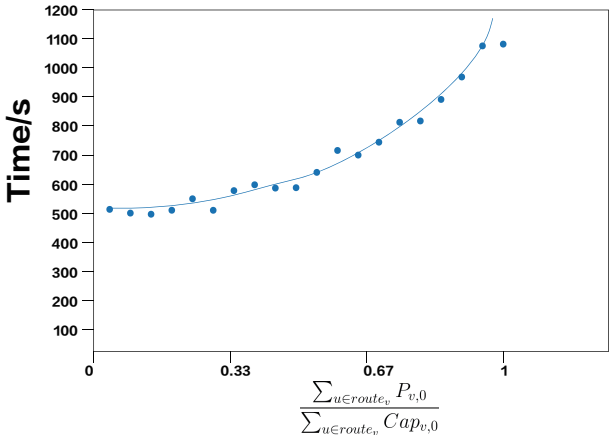


Figure 13: $Totaltime$ with different ks

4.3 Proximity between Our Model and Optimal Solution

We developed a simulated annealing(SA) algorithm which number of iterations is 5000 (its runtime is 15 minutes for our graph, we cannot increase the number of iterations anymore because of the time complexity). We use the routes calculated by Bellman-Ford algorithm (mentioned in 3.3) as initial routes in the SA program. Note that in this program we still use the CA model to calculate the evacuation time. Then, we set the $P_{v,0}$ in $G(V, E)$ as a random number. We define the result (evacuation time) of our model as 1, and calculate the ratio of $TotalTime$ in "SA routes" to $TotalTime$ in "TP routes". We do the same calculation for "Len routes" for comparison. We generated 13 graphs with random data, the result is shown in **Figure 14** We can easily see that "TP

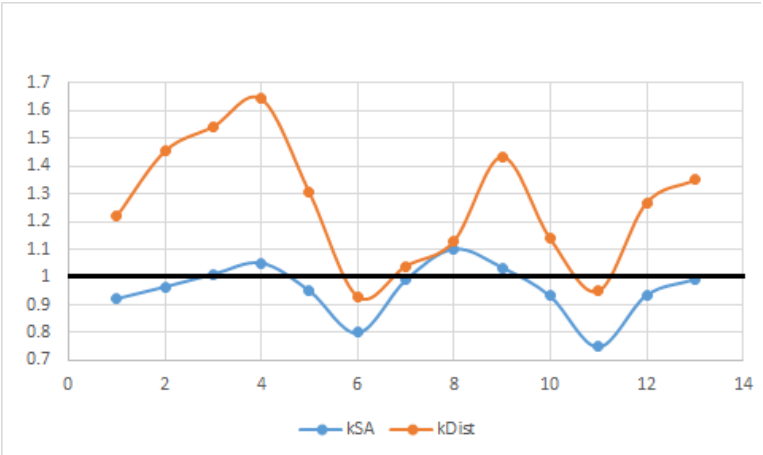


Figure 14: Ratios

routes" is much better than "Len routes", and is just a little bit worse than "SA routes". In other words, even algorithms like simulated annealing can hardly "improve" our model's result – because it is closed to the optimal solution.

Therefore, our model prove to be of high efficiency.

4.4 Strengths and Weaknesses

4.4.1 Strengths

- The model of evacuation, constructed by graph theory and analogy of liquid's flowing, is concise and elegant. Based on it, further calculations and optimizations are largely simplified and efficiently working.
- Our models are adaptable to different architectural structures. With little loss of efficiency on the results, BFS algorithm is quite swifter than stimulate annealing algorithm. Thus they are easily extended to more complicated structures.
- Our models can be widely applied. Because all the public places can be abstracted into an undirected graph, and different emergencies can be represented by modifying the structure of the graph, our models can work in different places and situations.
- Our works merely consist of intuitive models and graphic algorithm, making the models strongly interpretable.

4.4.2 Weaknesses

- Our models idealize the motion of visitors as flowing liquid, ignoring the complexity of human sentiments and small emergencies. They only work in a serene situation.
- In the CA model, we consider evacuees are all well-informed about directed instructions by managers, which may not be appropriate.
- Specifically about the Louvre, we consider escalators as stairs to evacuate, which may bring problems when in emergency.
- Due to the lack of statistics, there are many parameters in our model can't get exact number.

5 Conclusion

In this paper, we first abstracted the Louvre into a undirected graph and then designed a cellular automaton model to simulate the evacuation.

Afterwards, according to the working principle of cellular automate model, we design a Bellman-Ford algorithm and BFS algorithm to get a good solution. Finally, we analyze the high efficiency of our model and do the sensitivity analysis.

For future plans, in order to get concrete data of unknown parameters, large numbers of field researches are required.

References

[1] Aharon Ben-Tal, Byung Do Chung, Supreet Reddy Mandala, and Tao Yao. Robust optimization for emergency logistics planning: Risk mitigation in humanitarian relief supply chains. *Transportation Research Part B: Methodological*, 45(8):1177 – 1189, 2011. Supply chain disruption and risk management.

[2] Chang Liu, Zhan li Mao, and Zhi min Fu. Emergency evacuation model and algorithm in the building with several exits. *Procedia Engineering*, 135:12 – 18, 2016. 2015 International Conference on Performance-based Fire and Fire Protection Engineering (ICPFFPE 2015).

[3] Jake Pauls. International life safety and egress seminar, maryland, november, 1981: Summary of presentations and discussion. *Fire Safety Journal*, 5(3):213 – 221, 1983.

[4] R.D. Peacock, B.L. Hoskins, and E.D. Kuligowski. Overall and local movement speeds during fire drill evacuations in buildings up to 31 stories. *Safety Science*, 50(8):1655 – 1664, 2012. Evacuation and Pedestrian Dynamics.

[5] Louvre Press Release. 8.1 million visitors to the louvre in 2017. `http://presse.louvre.fr/8-1-million-visitors-to-the-louvre-in-2017`. 25 Jan. 2018.

[6] Telegraph Reporters. Terror attacks in france: From toulouse to the louvre. `https://www.telegraph.co.uk/news/0/terror-attacks-france-toulouse-louvre/`. 24 Jun. 2018.

[7] Angel A. Sheeba and R. Jayaparvathy. Performance modeling of an intelligent emergency evacuation system in buildings on accidental fire occurrence. *Safety Science*, 112:196 – 205, 2019.

[8] Best Wallpapers.net. Louvre city lights night pyramid. `https://cn.best-wallpaper.net/Paris-France-\Louvre-city-lights-night-pyramid-wallpapers.html`. 25 Jan. 2018.