Agent Based Modelling Final Project

Rush or Crush?

Impact of Obstacle Geometry on Escape Efficiency Considering Social Groups

Group 6
Yingying Deng, Lingfeng Li, Meiqi Sun
Mingjie Li, Weikang Chen

Why ABM?

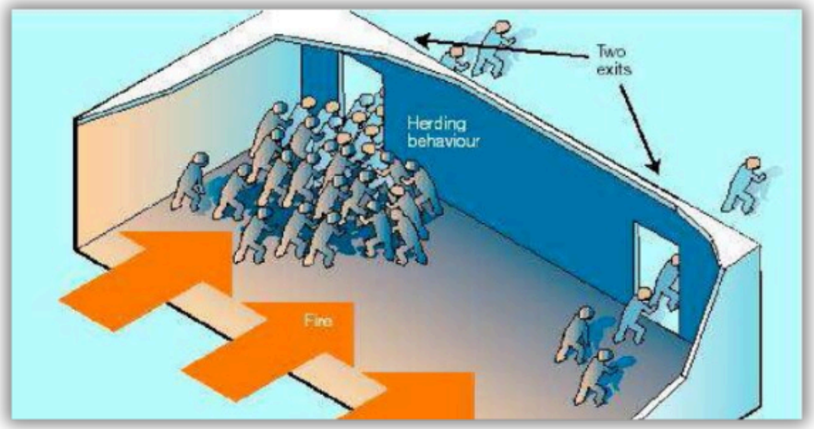
Herding Behavior

Self-organization Evacuation

Stampede pushing and shoving

Adaptation and Learning

Panicked Behavior



INTRODUCTION

[1]

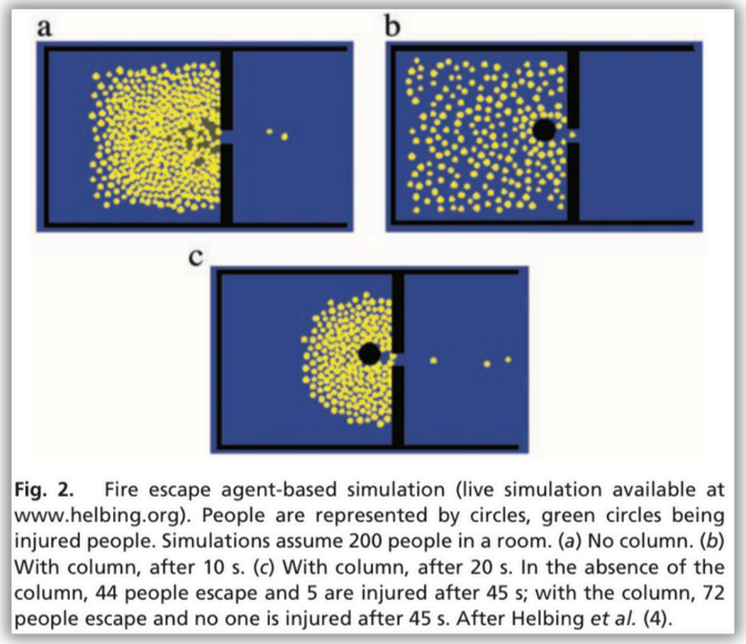
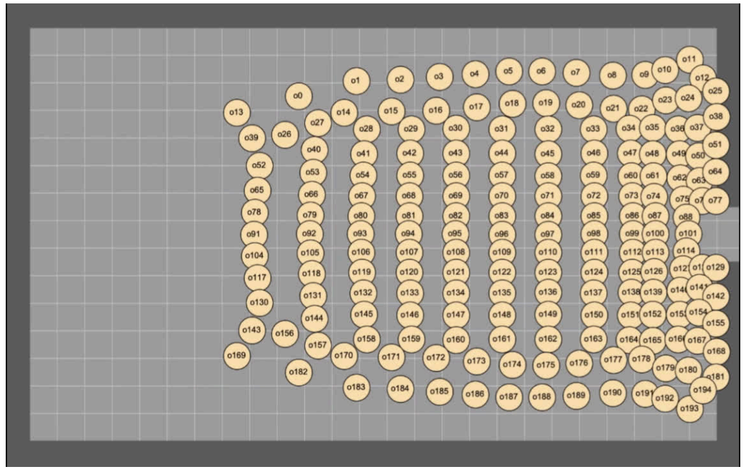
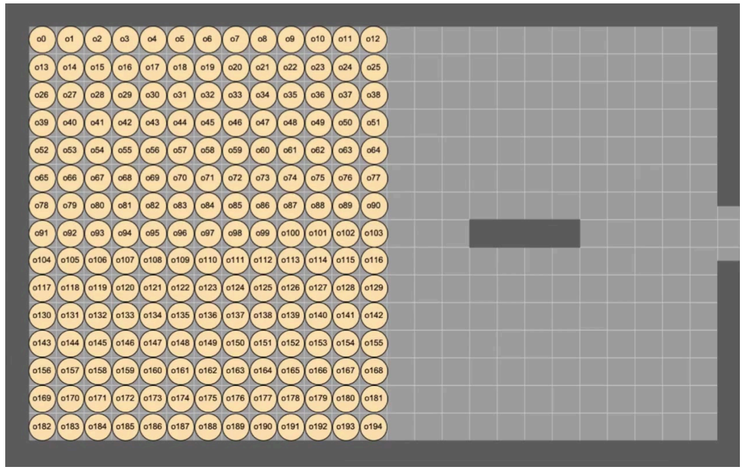


Fig. 2. Fire escape agent-based simulation (live simulation available at www.helbing.org). People are represented by circles, green circles being injured people. Simulations assume 200 people in a room. (*a*) No column. (*b*) With column, after 10 s. (*c*) With column, after 20 s. In the absence of the column, 44 people escape and 5 are injured after 45 s; with the column, 72 people escape and no one is injured after 45 s. After Helbing *et al.* (4).

Assuming a confined space with only one exit, **placing a suitable obstacle could improve the efficiency of the escape.** 😲





BUT individuals act independently during evacuation**!**

[1] Bonabeau, E. (2002). Agent-based modeling: Methods and techniques for simulating human systems. Proceedings of the national academy of sciences, 99(suppl_3), 7280-7287.

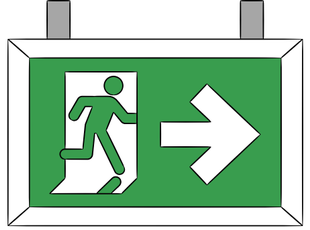
INTRODUCTION

Reaserch Problem:

Can the obstacles with appropriate geometry and layout improve escape efficiency considering population group splits?

Assumption:

1. The space is confined with only one exit.
2. People are heterogeneous (mass, shoulder length, velocity).
3. Obstacles will not be damaged, moved, or disappear.
4. Once outside, individuals will not re-enter the room.

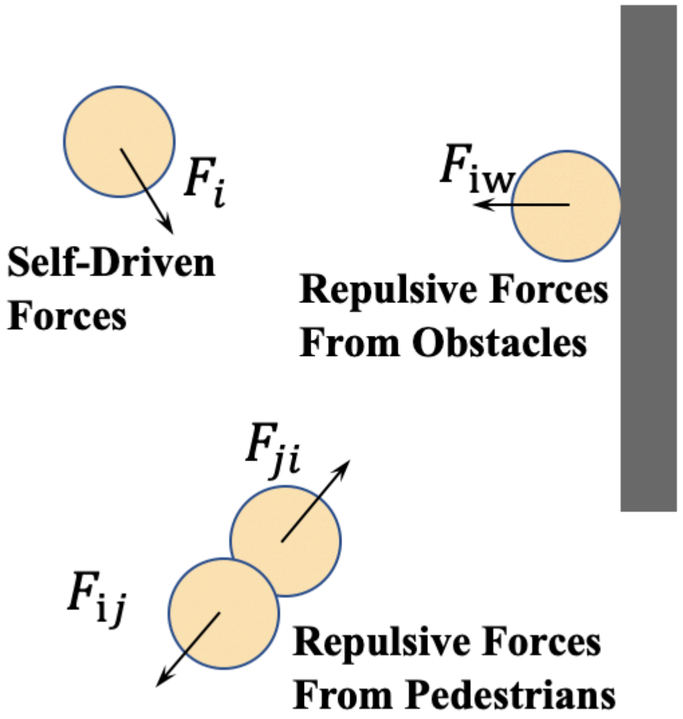


MODEL SETUP

- **The Social Force Model**

  It assumes the accerleration of each pedestrain i follows the equation:

$$m_i a_i = F_i + \sum_{j(\neq i)} F_{ij} + \sum_{w} F_{iw} + \overset{\text{random term}}{\epsilon_i(t)} + F_i^{vis} + \sum_{k(k\neq i)} F_{ik}^{att}$$



1. Self-Driven Force

$$F_i = m_i \frac{v_i^0(t)e_i^0(t) - v_i(t)}{\tau}$$

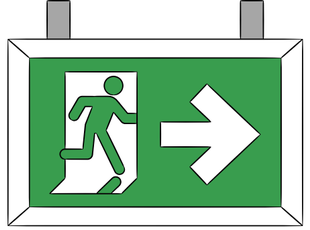
m_i : mass of pedestrian i.

v_i^0 : desired velocity of pedestrian i.

$e_i^0(t)$: desired direction of pedestrian i, determined by **A*** algorithm based on the current position.

$v_i(t)$: current velocity of pedestrian i.

τ : characteristic time.



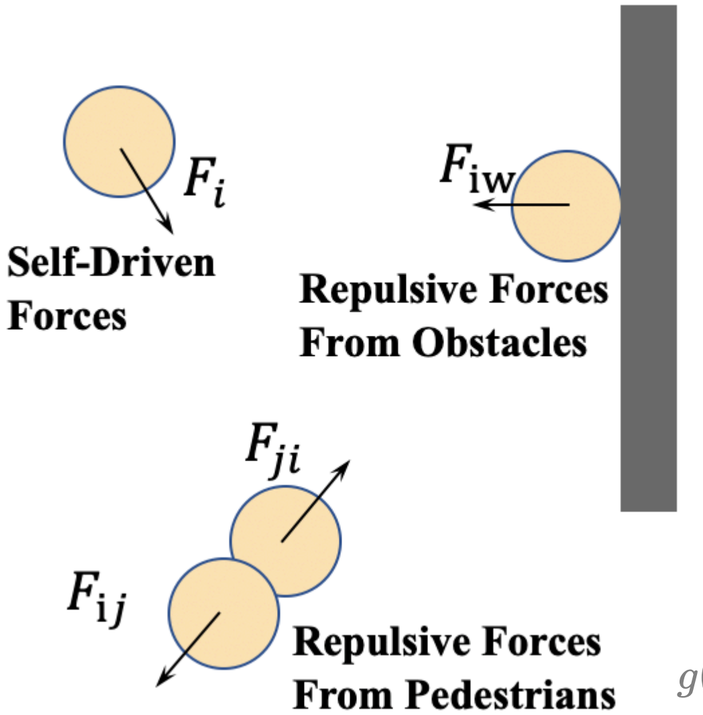
MODEL SETUP

- **The Social Force Model**

It assumes the accerleration of each pedestrain i follows the equation:

$$m_i a_i = F_i + \sum_{j(\neq i)} F_{ij} + \sum_{w} F_{iw} + \epsilon_i(t) + F_i^{vis} + \sum_{k(k\neq i)} F_{ik}^{att}$$

random term



2. Agents-Agents Repulsive Force

$$F_{ij} = Ae^{(r_{ij}-d_{ij})/B}\vec{n}_{ij} + kg(r_{ij}-d_{ij})\vec{n}_{ij} + \kappa g(r_{ij}-d_{ij})\Delta v_{ji}^t \vec{t}_{ij}$$

A :strength of repulsive force.

B :scale controls distance influence.

r_{ij} : $r_i + r_j$, where r denotes the radii, i.e. half of the shoulder length.

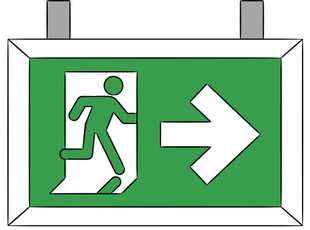
d_{ij} : distance between two pedestrians.

$g(r_{ij} - d_{ij})$: $max\{0, r_{ij} - d_{ij}\}$, i.e. effective when pedestrians touch each other.

$\vec{n}_{ij}$: normalized vector pointing from pedestrian j to i.

$\vec{t}_{ij}$:corresponding normalized tangential vector, i.e., $\vec{t}_{ij} = (-n_{ij}^2, n_{ij}^1)$

Δv_{ji}^t :$(v_j - v_i) \cdot t_{ij}$, the tangential velocity difference between pedestrian j and i.

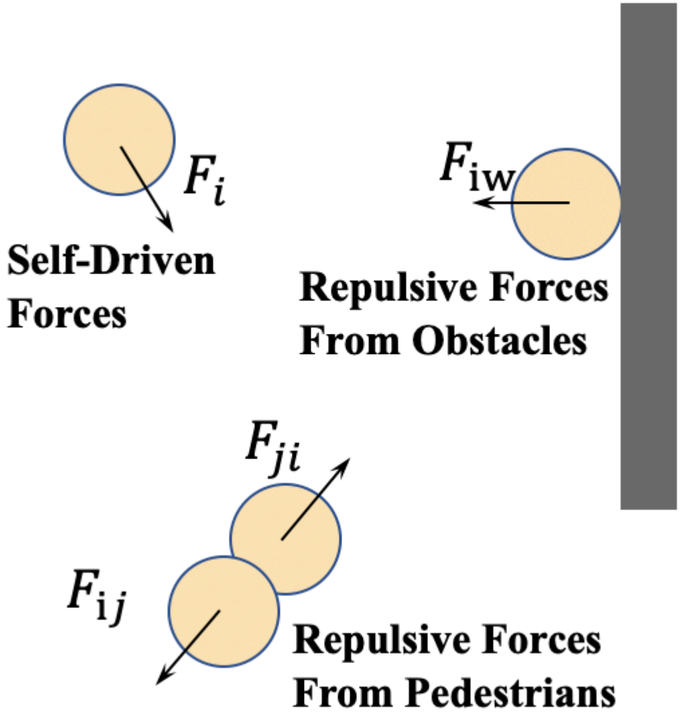


MODEL SETUP

- **The Social Force Model**

It assumes the accerleration of each pedestrain i follows the equation:

$$m_i a_i = F_i + \sum_{j(\neq i)} F_{ij} + \sum_{w} F_{iw} + \overset{\text{random term}}{\epsilon_i(t)} + F_i^{vis} + \sum_{k(k\neq i)} F_{ik}^{att}$$



3. Agents-Obstacles Repulsive Force

$$F_{iw} = Ae^{(r_i - d_{iw})/B}\vec{n}_{iw} + kg(r_i - d_{iw})\vec{n}_{iw} + \kappa g(r_i - d_{iw})(-\vec{v}_i\vec{t}_{iw})\vec{t}_{iw}$$

A :strength of repulsive force.

B :scale controls distance influence.

d_{iw} : distance to the obstacle.

$g(r_i - d_{iw})$: $max\{0, r_{ij} - d_{iw}\}$, i.e., effective when the pedestrian touches the obstacle.

$\vec{n}_{iw}$: normalized vector perpendicular to the obstacle.

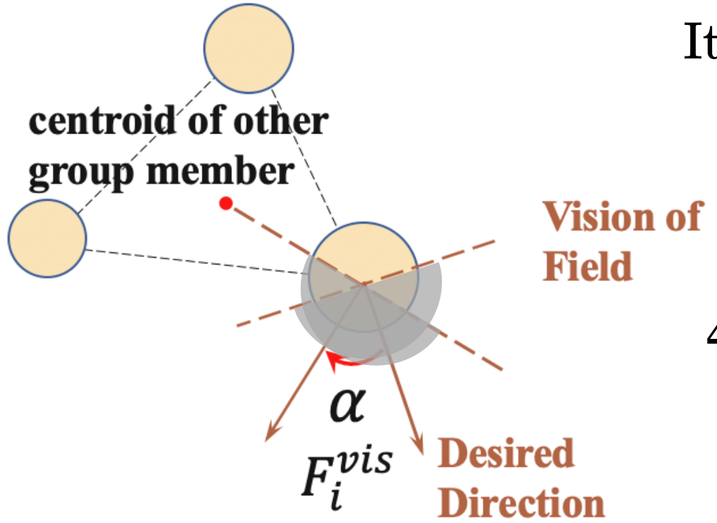
$\vec{t}_{iw}$:corresponding normalized tangential vector, i.e., $\vec{t}_{ij} = (-n_{ij}^2, n_{ij}^1)$

MODEL SETUP

- **The Social Force Model**

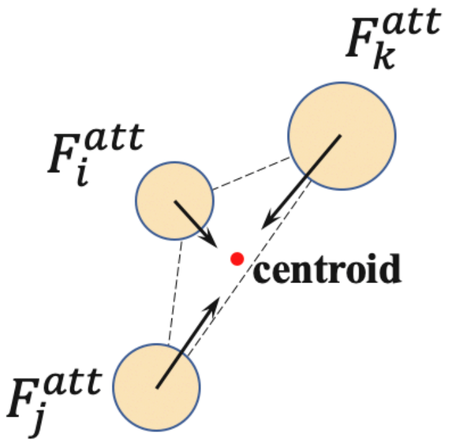
It assumes the accerleration of each pedestrain i follows the equation:

$$m_i a_i = F_i + \sum_{j(\neq i)} F_{ij} + \sum_{w} F_{iw} + \underset{\text{term}}{\overset{\text{random}}{\epsilon_i(t)}} + F_i^{vis} + \sum_{k(k\neq i)} F_{ik}^{att}$$



4. Vision Attraction Force

$$F_i^{vis} = -\beta_1 \alpha_i \vec{v}_i$$



α_i :minimum head rotation angle that pedestrian i adjusts to maintain visual contact with their partners.

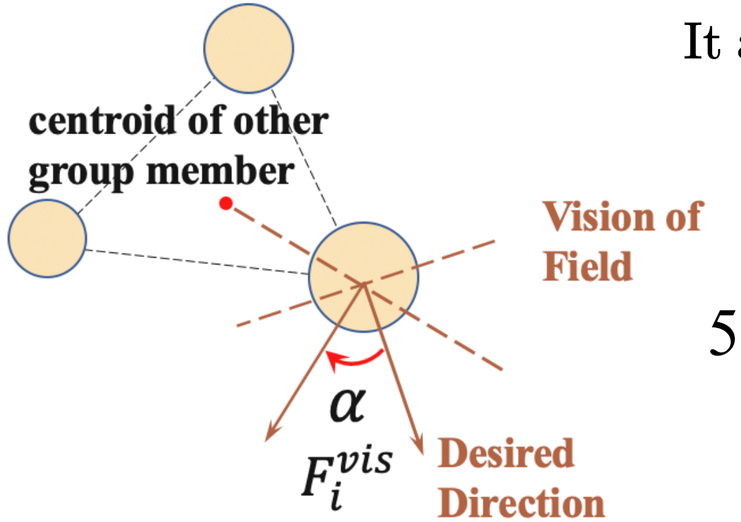
β_1 :parameter describing the strength of the social interactions between group members.

MODEL SETUP

- **The Social Force Model**

It assumes the accerleration of each pedestrain i follows the equation:

$$m_i a_i = F_i + \sum_{j(\neq i)} F_{ij} + \sum_{w} F_{iw} + \underset{\text{term}}{\overset{\text{random}}{\epsilon_i(t)}} + F_i^{vis} + \sum_{k(k\neq i)} F_{ik}^{att}$$



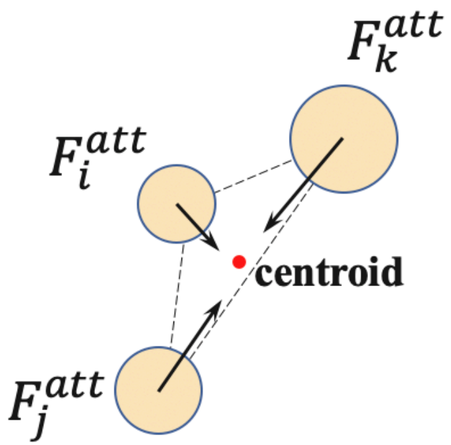
5. Group Attraction Force

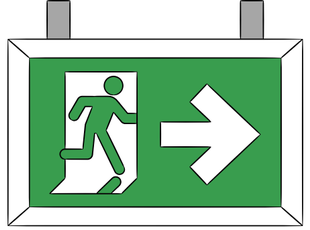
$$F_i^{att} = q_A \beta_2 \vec{u}_i$$

$\vec{u}_i$:unit vector pointing from pedestrian i to the centroid.

q_A :Set to 1 if the distance between pedestrian i and the group centroid exceeds a threshold, otherwise 0. The threshold is approximated as $\frac{N-1}{2}$ meters.

β_2 :parameter describing the strength of the attraction effects.





MODEL SETUP

- **Process Overview and scheduling**

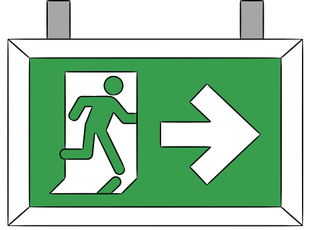
1. Randomly initialize mass, and radius (1/2 shoulder length) of each agent.
2. Agents are statically positioned on a grid.
3. Randomly form groups based on the ratio of each type of group.
4. At each timestep (0.005s):

   - Use the **A* algorithm** to **determine the desired direction** from the current position.

   - Calculate the net force, then **update the acceleration -> new velocity -> the distance moved**.

   - **Update** the agent's new **position**.
     (If an agent reaches the target position, drop it out.)

   - **Repeat** step 4 until all agents have been dropped out.



MODEL SETUP

- **Environment Parameter**

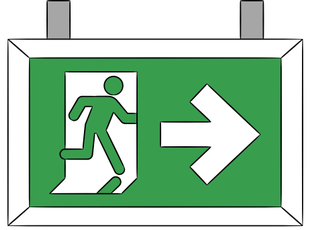
Parameter	Value	Units	Description
W_{room}	25	m	Room width
L_{room}	15	m	Room length
W_{door}	2	m	Exit width
$(x_{obs}^{start}, y_{obs}^{start})$	$([13, 23], [3, 7])$	(m, m)	Start node of the rectangle obstacle
$(x_{obs}^{end}, y_{obs}^{end})$	$([14, 24], [8, 12])$	(m, m)	End node of the rectangle obstacle

- **Agent Parameter**

Parameter	Value	Units	Description
N	195	-	Number of agents
m_i	$[30, 70]$	kg	Mass
r_i	[17.5, 20]	cm	Radius, i.e. half the shoulder width
v_i^0	0	m/s	Initial velocity
v_i^d	[2, 2.4]	m/s	Desired velocity
r_1	[0.6, 1]	-	Proportion of the total population in groups of size 1
r_g	[0, 0.4]	-	Proportion of the total population in groups of size g (g=2, 3)

- **Model Parameter**

Parameter	Value	Units	Description
A	2000	N	Strength of the repulsive force
B	0.08	m	Scale controls distance influence.
τ	0.5	s	characteristic time
β_1	1	-	Strength of the vision attraction force
β_2	3	-	Strength of the group attraction force



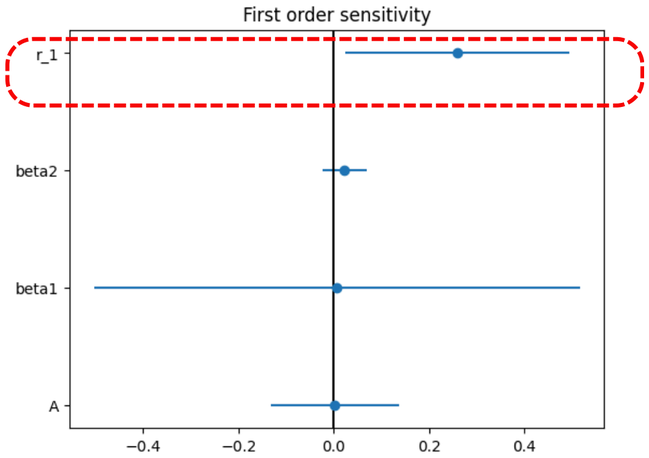
HYPOTHESIS

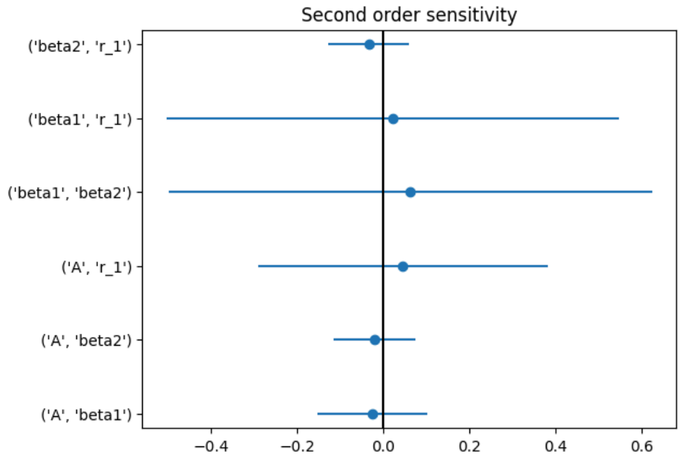
- Is **Individual Ratio (r_1)** important? **Global SA**
- Layout and Geometry of Obstacle
- Proportion of group size

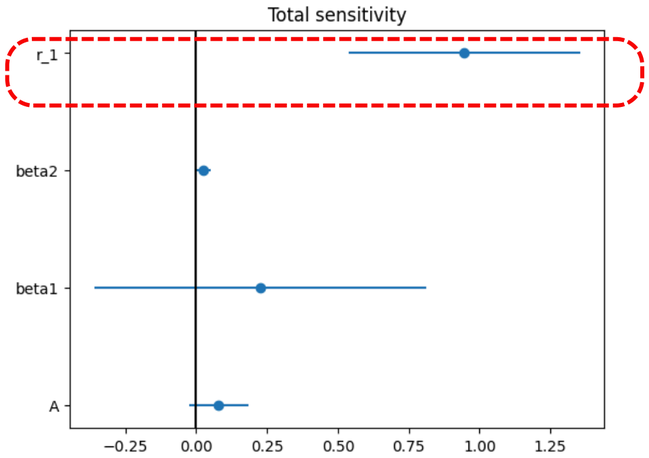
} **Experiment**

GLOBAL SA

variance explaination







r_1 is the most important parameter

Individual Ratio

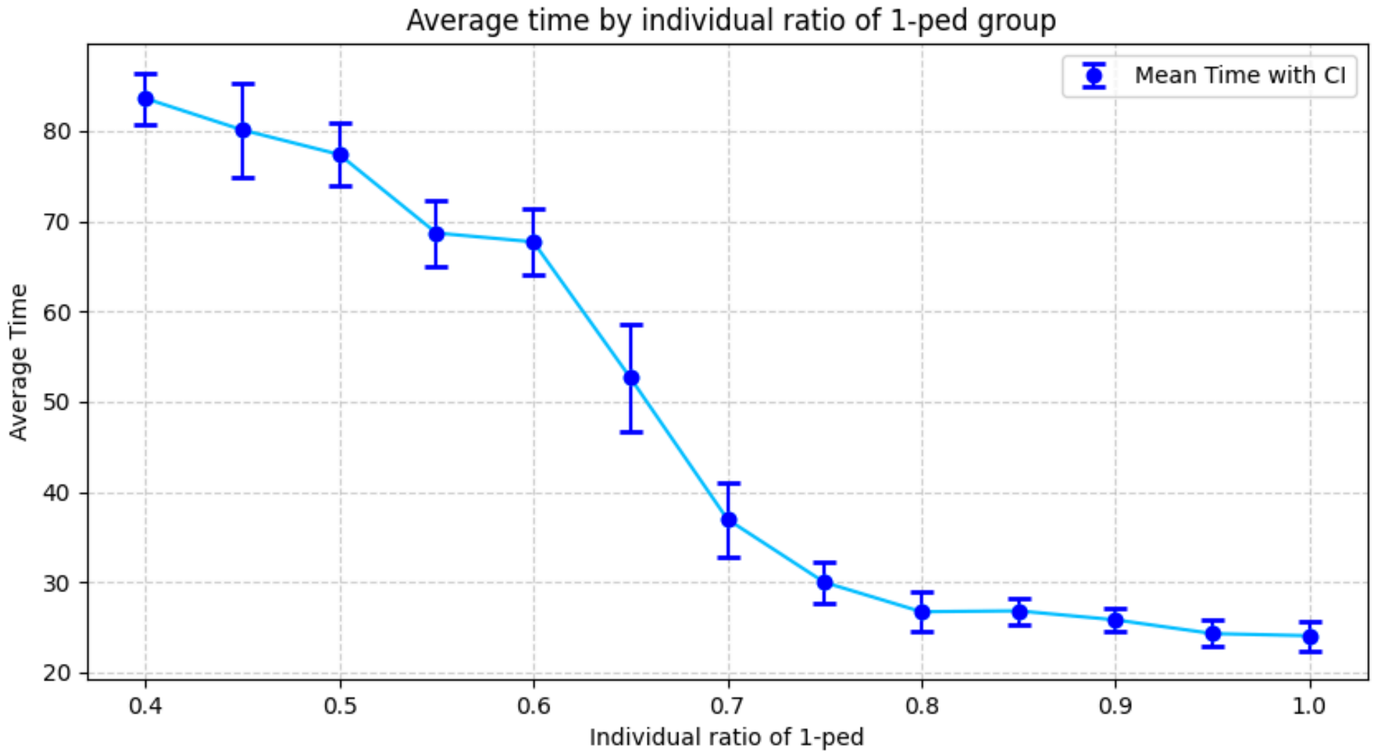
LOCAL SA

- **Individual Ratio (γ_1)**

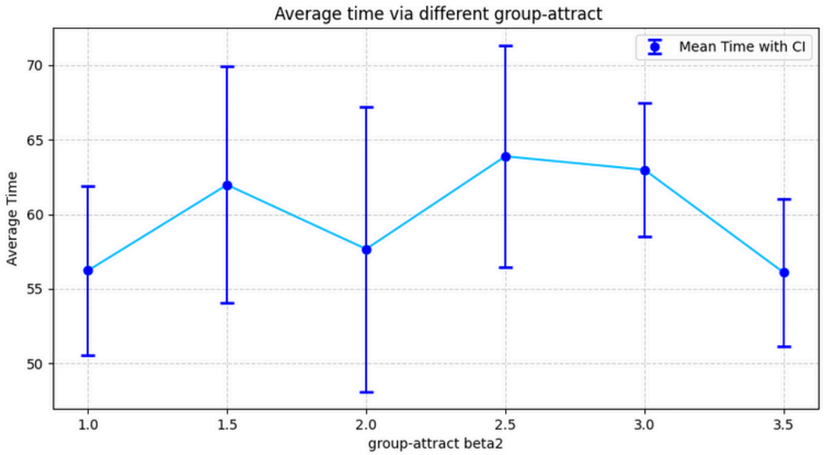
monotone decreasing

significantly reduces from 0.6 to 0.7

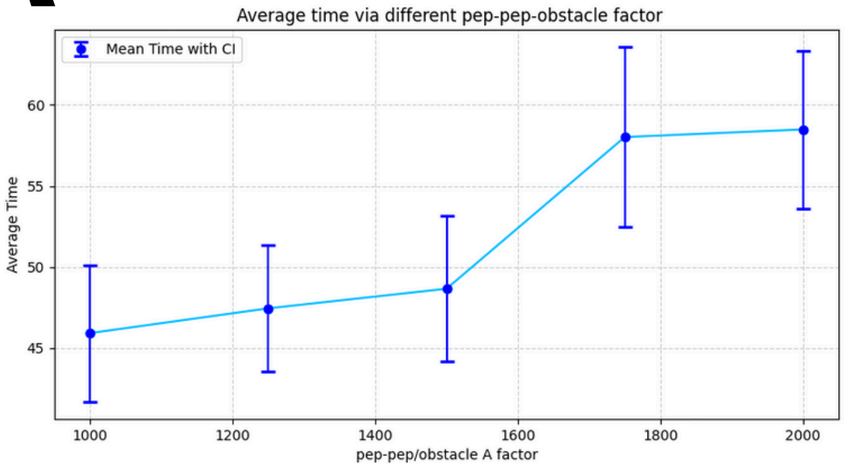
Selfishness is better for survival



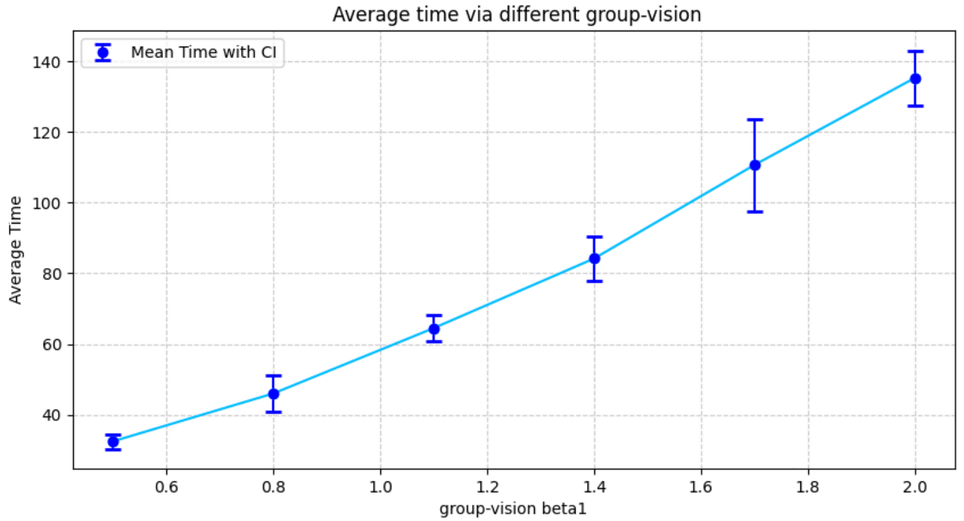
LOCAL SA



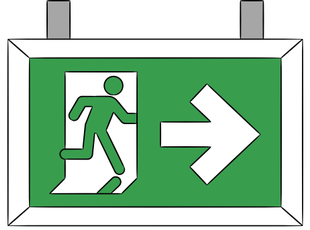
group attraction



force factor

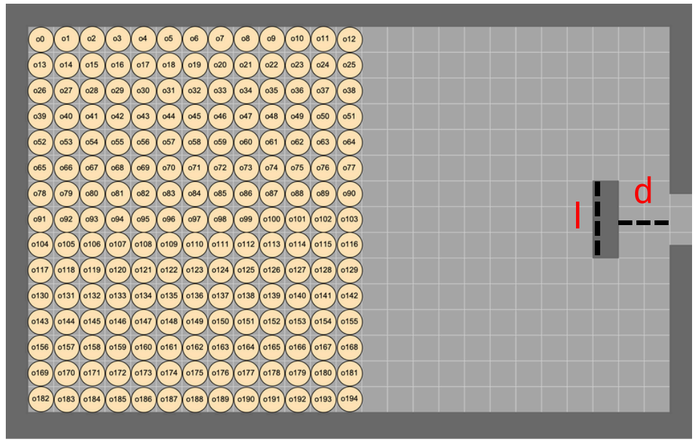


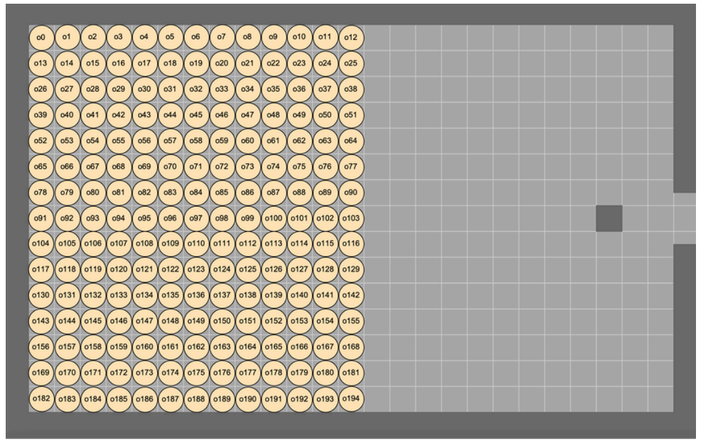
group vision

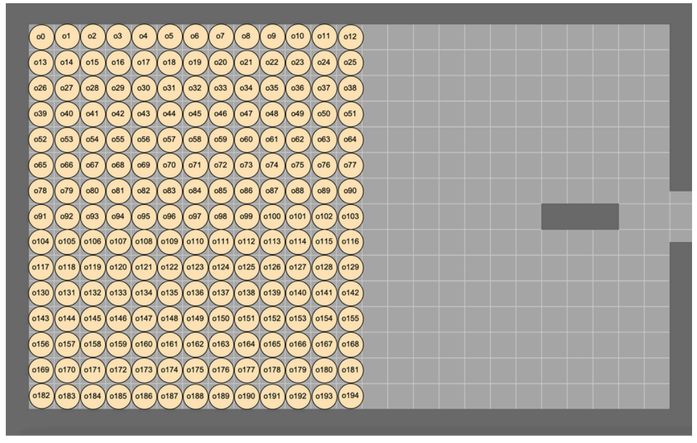


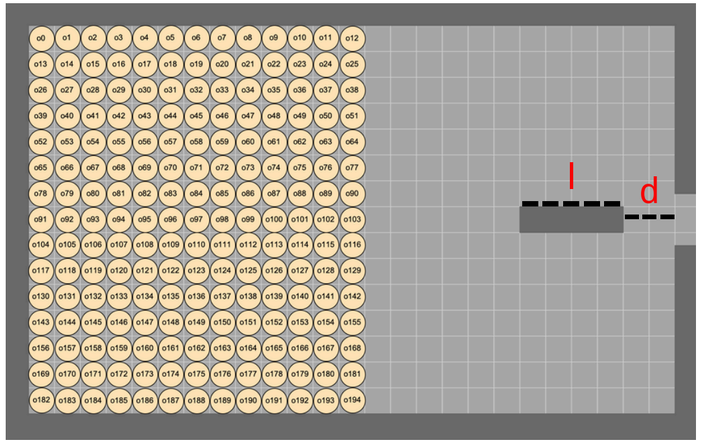
EXPERIMENT-1

- **Determine the optimal layout and geometry of an obstacle without considering group splitting.**





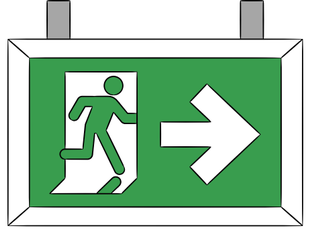




d: distance from the door

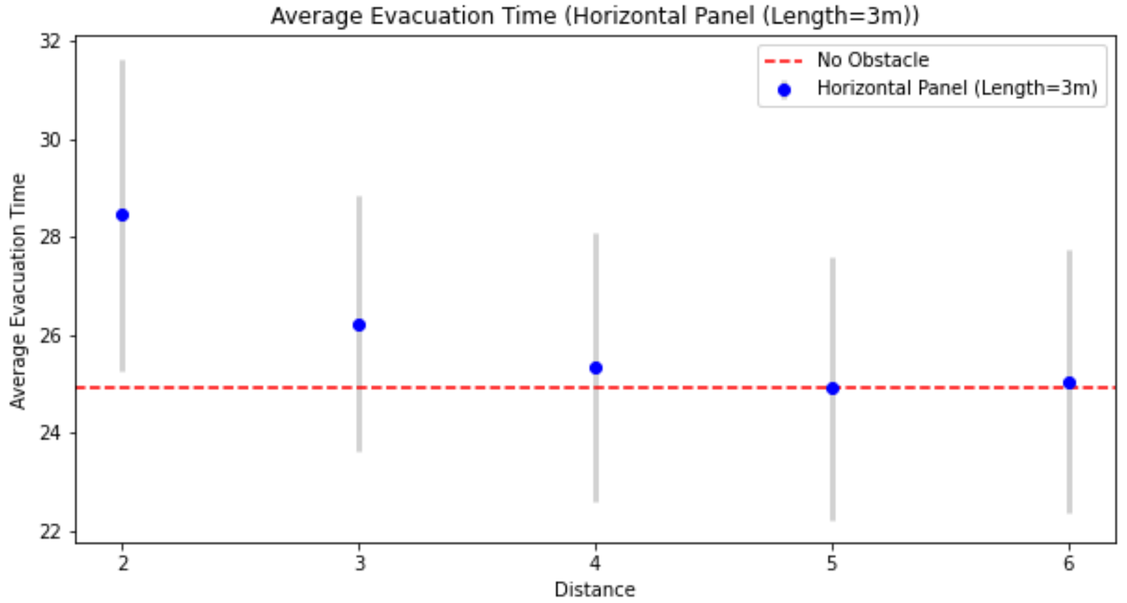
l: the length of the obstacle

- Examine obstacles with different length and distance from the door
- Test four different placements



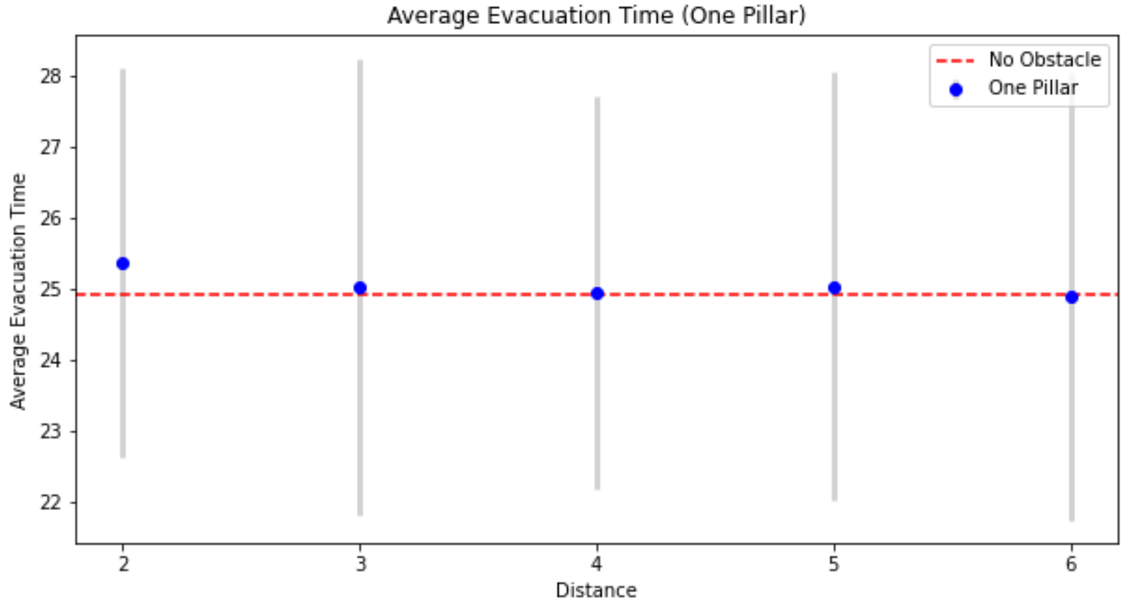
EXPERIMENT-1

Horizontal Panel

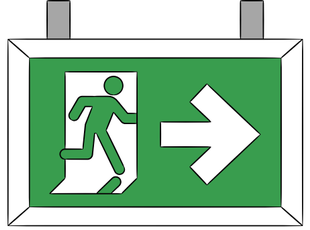


Red line represents the escape time without obstacles.

One Pillar

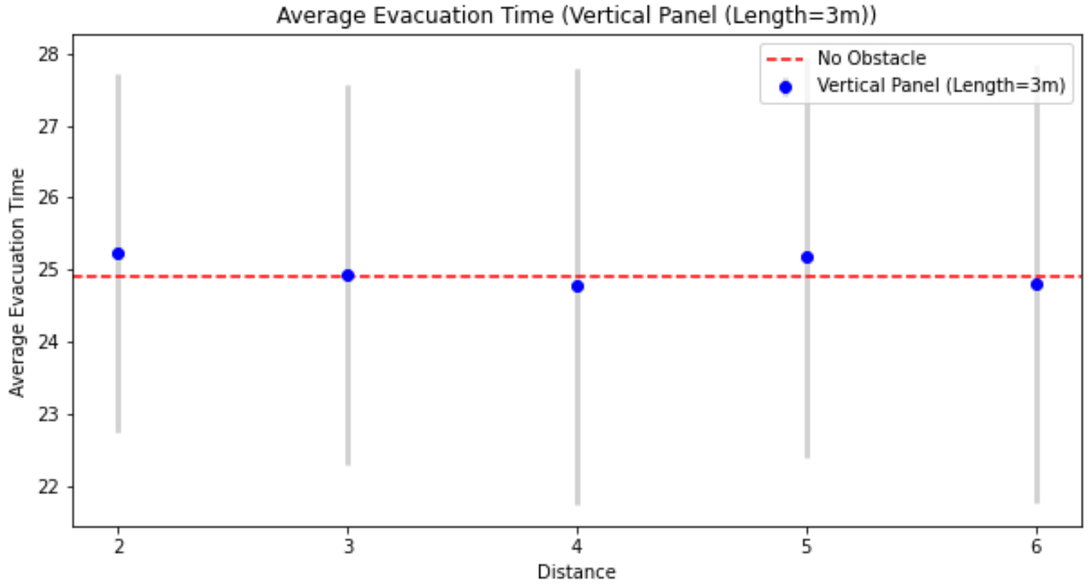


The obstacle fails to improve escape time.



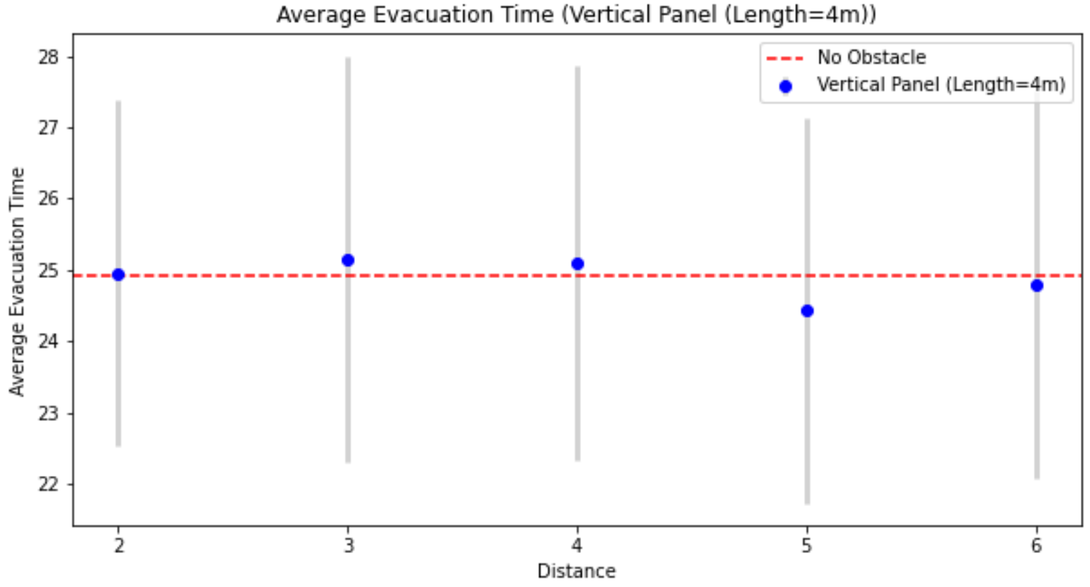
EXPERIMENT-1

Length of Vertical Panel (3m)

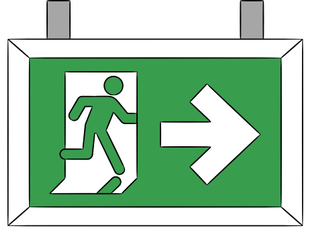


These obstacles can reduce escape times under certain conditions.

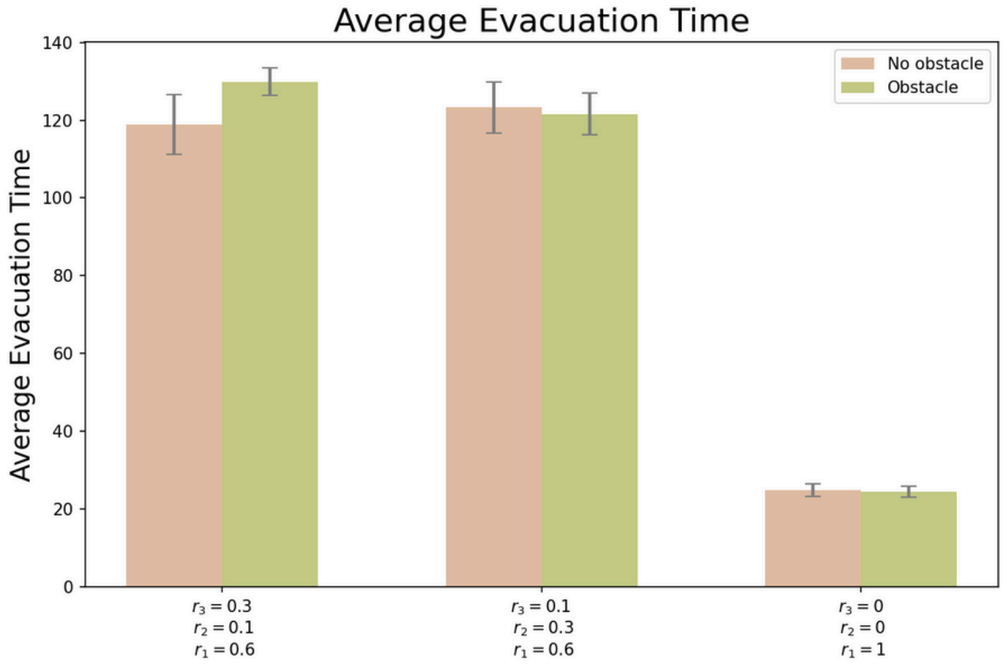
Length of Vertical Panel (4m)

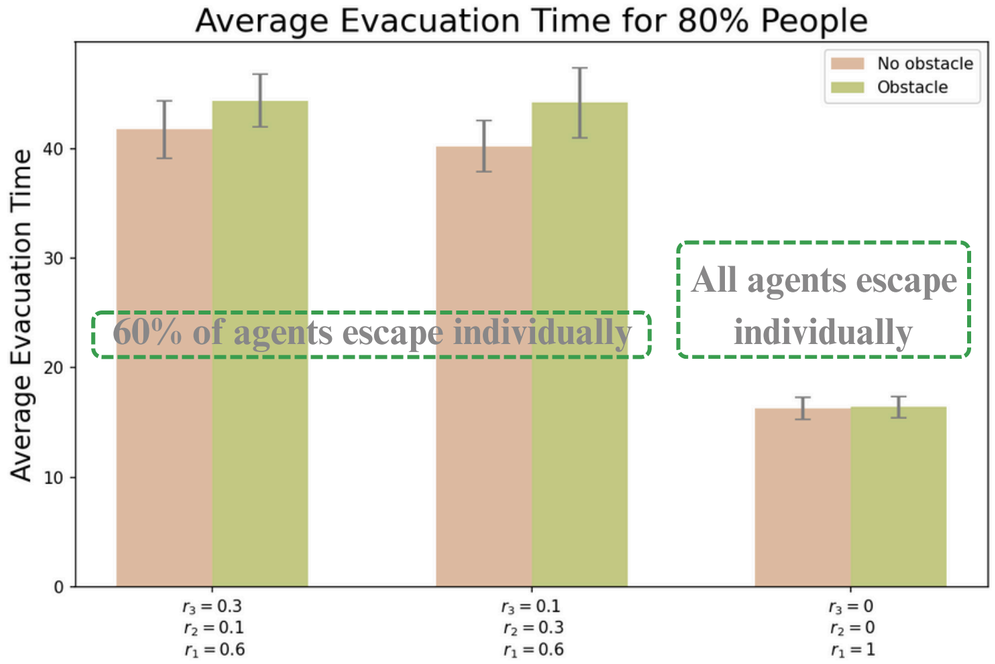


A 4-meter obstacle placed 5 meters from the door is the most effective.



EXPERIMENT-2



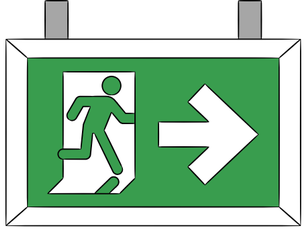


Obstacle can extend escape times, indicating negative impact.

R1: the proportion of agents escape **individually**

R2: the proportion of agents forming **groups of two**

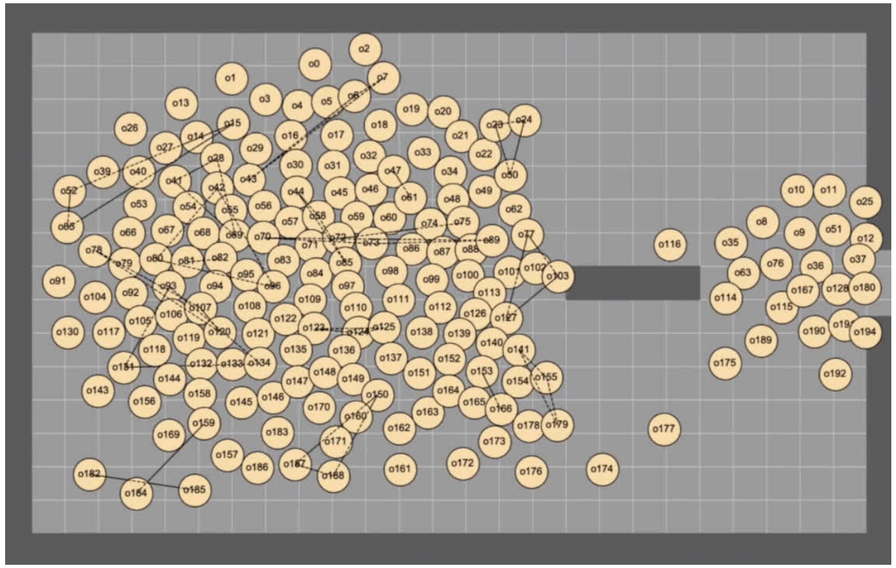
R3: the proportion of agents forming **groups of three**



EXPERIMENT-2

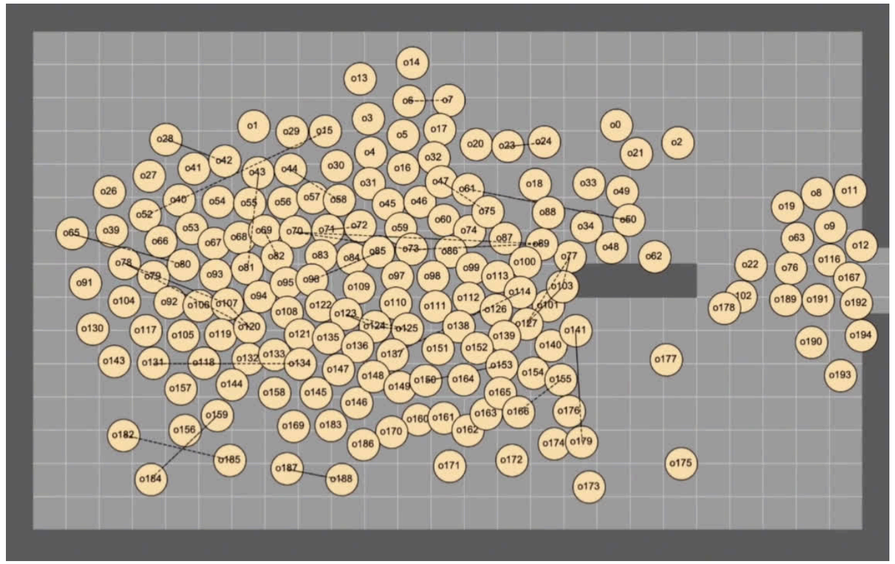
$R_3 = 30\%$

$R_2 = 10\%$

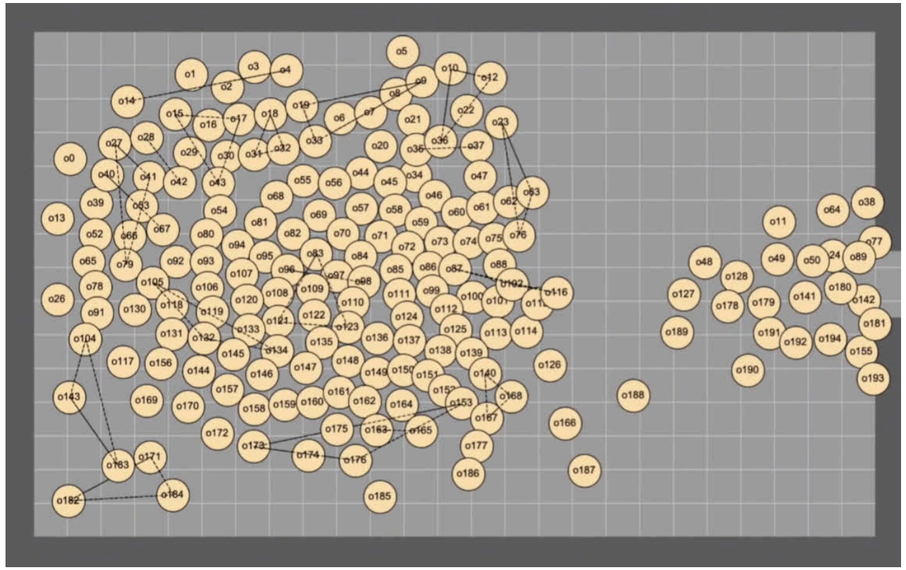


$R_3 = 10\%$

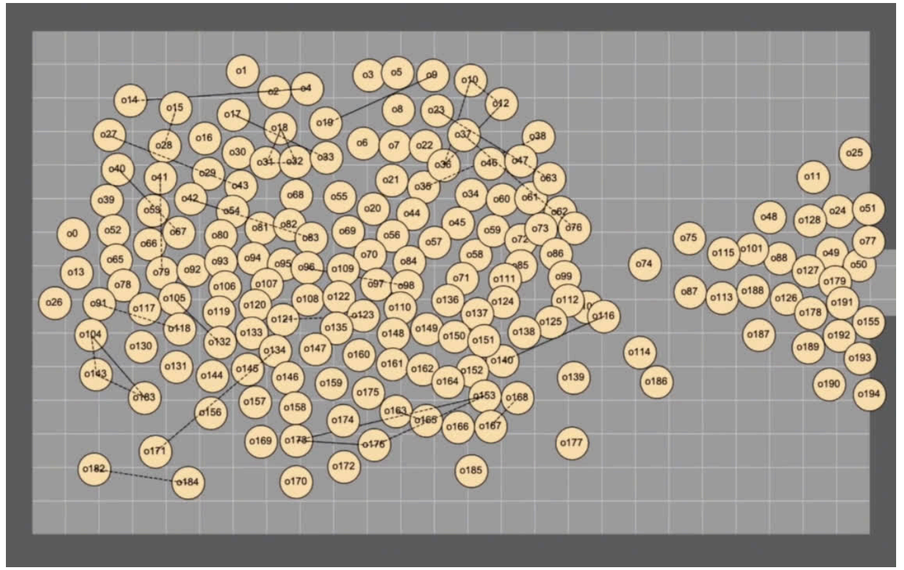
$R_2 = 30\%$

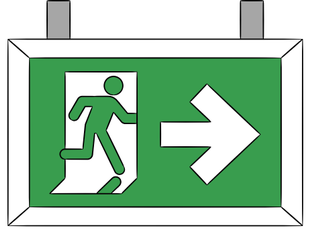


Agents move slowly not only due to obstacles but also because they are blocked by other groups.

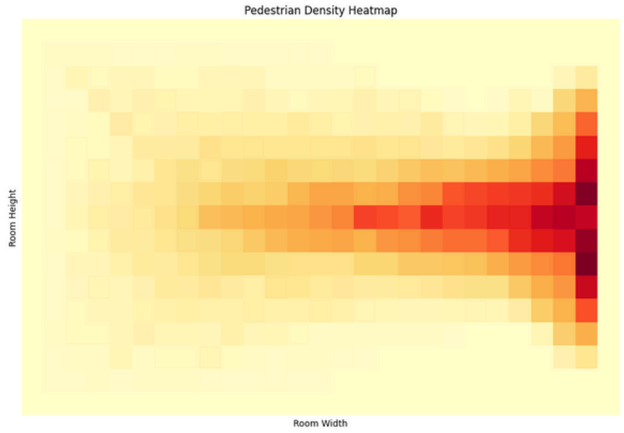
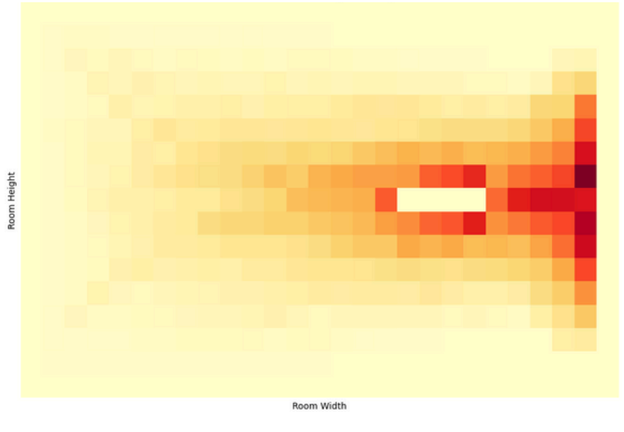
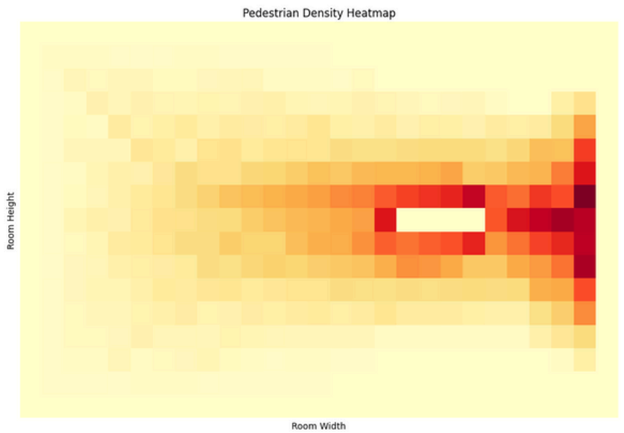
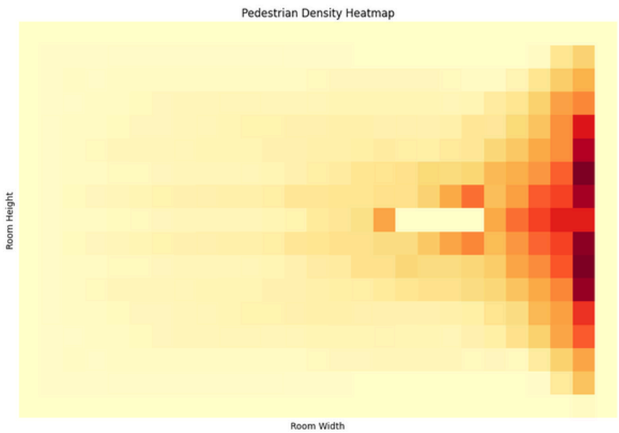


Many agents cluster in the middle, which hinders their escape.





EXPERIMENT-2

	$R_3 = 30\%$ $R_2 = 10\%$	$R_3 = 10\%$ $R_2 = 30\%$	$R_1 = 100\%$ $R_2 = R_3 = 0$
No Obstacle			
Obstacle			

Agents pile up in the middle because the group blocks them.
This negates the dispersing effect of the obstacle, and agents have to find new route.

All agents crowd along the right line, but obstacles help spread them out.

CONCLUSION

Single person escaping:

The **optimal obstacles** indeed **reduce** the escape time.

Group escaping: **(Key Finding!)**

Obstacles may have a **negative** impact.

Individual Ratio:

The escape time **significantly reduces** as the proportion of singly escaping agents increases from 60% to 70%.

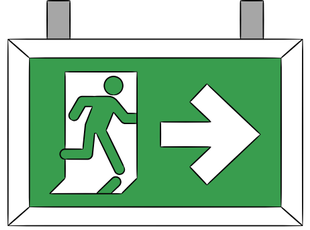
FUTURE WORK

- Might **cylindrical obstacles** works better due to the smooth surface?
- For individually escaping, are there any potential **emergent behaviors**? - Large-scale selfishness may bring benefits, even to those who need help.

Agent Based Modelling Final Project

Thanks for your attention!

Group 6
Yingying Deng, Lingfeng Li, Meiqi Sun
Mingjie Li, Weikang Chen



MODEL SETUP

- **Astar Algorithm**

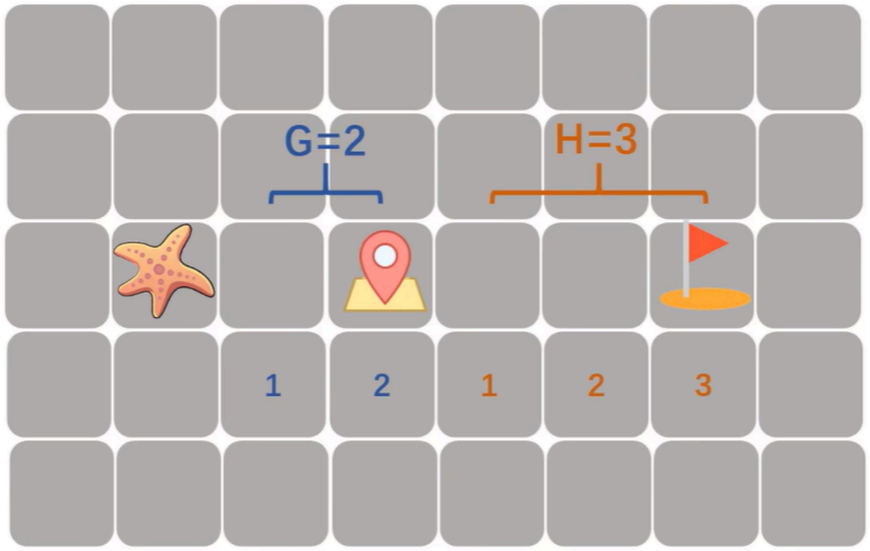
Creat a data structure Node:
struct Node{g; h; f; *father*} // the g, h, f value and pre-node of the node.
Function Next_Loc(*start node*, *barrier_list*):
  Define *target node*
  Initialize *open list*, *close list*, and *barrier list* as empty lists
  If *start node* is in *barrier list*
    return None
  Add *start node* **to** *open list*

  While *open list* is not empty:
    Select *currrent node* **from** *open list* the minimum f value
    Remove *currrent node* **from** *open list* and add it to *close list*
    For each *direction* **from** *currrent node*:
      temp node = *currrent node* + *direction*
      Continue if *temp node* is out of bounds, in *barrier list*, or in *close list*
      If *temp node* not in *open list*:
        Calculate and **set** g, h, f values for *temp node*
        Add *temp node* **to** *open list*
        Set *father* of *temp node* as *currrent node*
      Else if g value at *temp node* can be improved:
        Update g, $father$, and f values for *temp node*
      If *temp node* is *target node*
        break the loop

  Trace path **back** from *target node* to *start node*:
    While *father* of *temp node* is not *start node*:
      Move to *father* of *temp node*

  Return the direction from *start node* to *temp node*

ASTAR



F=H+G

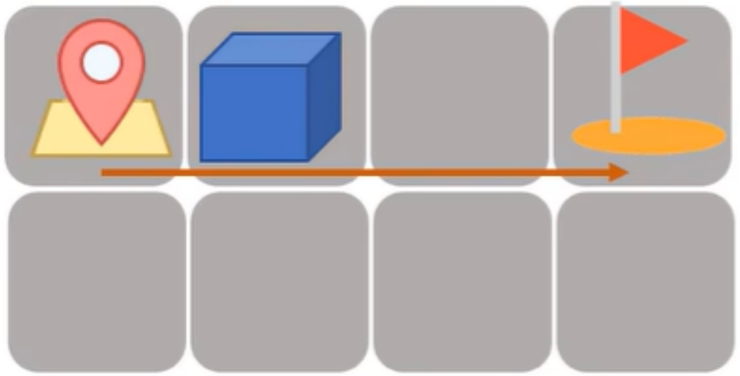
F: Total cost from the start to goal

G: Cost from start

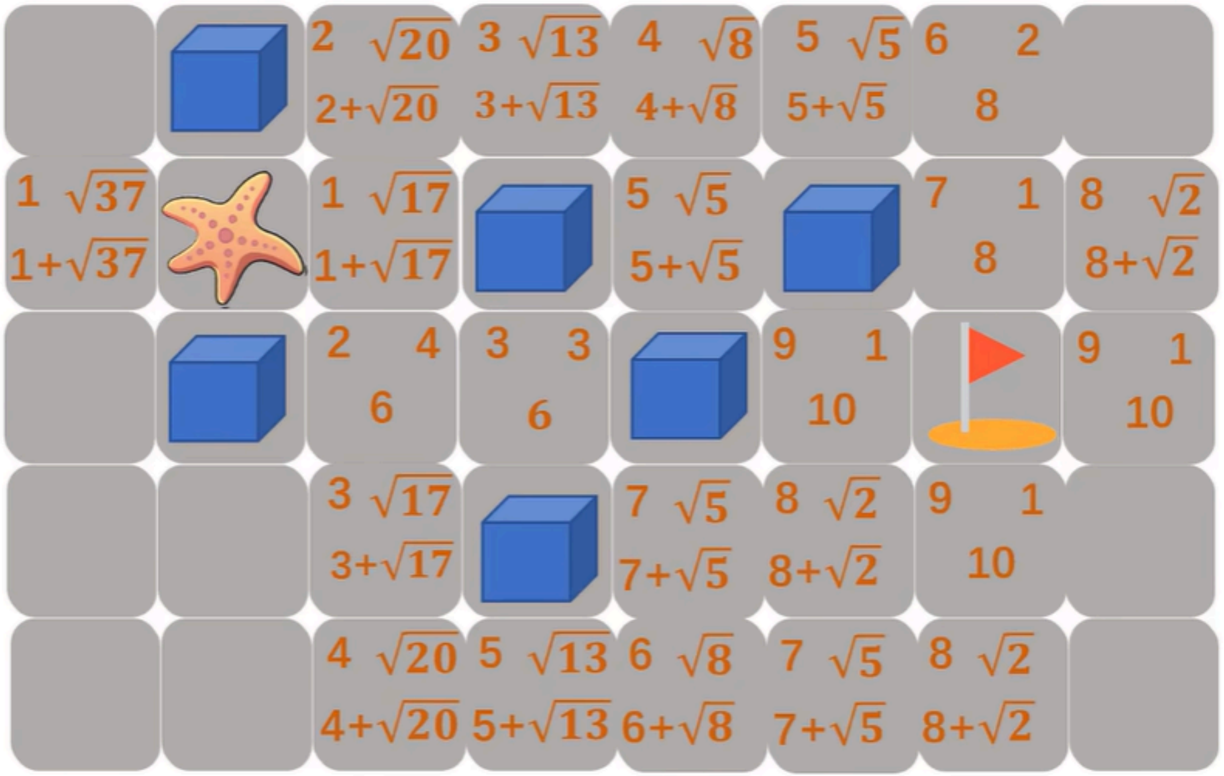
H: Heuristic estimate of cost to goal

$$H = \sqrt{(x2 - x1)^2 + (y2 - y1)^2}$$

might unestimate H



ASTAR



F=H+G

F: Total cost from the start to goal

G: Cost from start

H: Heuristic estimate of cost to goal

$$H = \sqrt{(x2 - x1)^2 + (y2 - y1)^2}$$

ASTAR

F=H+G

F: Total cost from the start to goal

G: Cost from start

H: Heuristic estimate of cost to goal

Backward Searching

Iterate to find smaller F