Framework for Agent-Based Model of Escape Dynamics from a Commercial Aircraft

Grace McLaughlin,¹ Sirui Yao,²

¹Department of Bioinformatics and Mechanical Engineering, Virginia Tech
²Department of Computer Science, Virginia Tech

Abstract

Introduction

Creating accurate models of pedestrian movement is of great interest due to the challenges and ethical concerns related to conducting human experiments. Models can be used to design areas that will be subject to high crowd density in order to maximize safety in emergency situations. One way to do this is with a social force model, where pedestrians are viewed as particles and their behavior is determined by long range forces that arise from other individuals in the simulation [10, 6]. In addition, theoretical work has been done to apply fluid dynamic models to large crowds [11, 12], and video analysis has uncovered wave-like patterns among people in panicked situations [9]. Continuum models have their place, but they fail to capture emergent behavior that arises from decision making by the individual and interaction with others. Agent-based modeling is an approach that allows for emergent phenomenon to be observed in the simulation [5]. Pre-determined characteristics such as sex, height, weight, and personalities can be assigned to the agents. An agent-based method provides a flexible framework to model individuals and observe collective behavior that emerges from their choices and interactions. Challenges include large computation time and difficulty determining parameters of the model, as human decision making is a complex process. Cellular automata (CA), a method where agents occupy cells and move and interact based on a series of preset rules, has been employed [6, 15, 4, 18, 7] to model pedestrian dynamics. In CA, each agents motion is calculated in parallel during each time step, making it very computationally efficient [8]. Implementing the rules is generally straightforward but can lead to complex pattern formation that can accurately mimic human behavior [3]. CA can be based on deterministic rules that lead to the same out-
come with each run, or alternatively a probabilistic approach can be taken. Burstedde et al. [6] combined the social force model idea with CA, and introduced the idea of a floor field. They created a second, underlying grid that, based both on location of pedestrians in the model and features of the room they were confined to, determined the probabilities of each agent moving in a given direction (see Figure 1). Feliciani and Nishinari [7] expanded upon this idea and created a sub-mesh (Figure 2) that allowed for a higher pedestrian density and more realistic movement. One challenge of any discrete pedestrian model on a fixed grid is that the intricacies of human movement may not be captured. Feliciani and Nishinaris solution represents one way of addressing this issue.

Figure 1: Burstedde et al. (2001), cellular automata model of pedestrian movement based on underlying floor field, illustration of possible directions an agent could move (left) and underlying probability values used to determine movement (right)

Aircraft disembarkment is a natural place to employ an agent based model, as a planes layout is well represented by a grid and has relatively low numbers of passengers to model leading to fast computation speeds. Furthermore, dynamics of panicked escape from an aircraft is a worthwhile problem to investigate due to limited data available to the public [1]. Modeling panic is often considered to be challenging due to the notion that humans behave erratically under high stress, but in reality individuals generally behave in a rational fashion based on the information that they have available in the moment [16]. Internal factors as well can alter someones behavior during an emergency, such as tendency towards altruistic behavior, presence of family members, and personal response to stress. An agent-based approach allows one to consider these factors and leads to increased realism of the model, given that one can determine the appropriate parameters. Although tracking algorithms for crowds are improving [13], video data from panic situations remains scarce. Kirchner et al. [14] applied Burstedde et al.s [6] floor field model to explore the role of competition and exit size on escape from an aircraft. They however did not model a realistic plane, but rather a simplified scenario consisting of one exit door. Qiang et al. [17] used CA to investigate factors that effect airplane embarking times to
try to find an optimum boarding strategy. More recently, Zou et al. [19] designed a CA model involving a realistic airplane layout and used it to test how different exit door configurations affect disembarkment time. Giitsidis and Sirakoulis [8] also used CA, allowing either one or two passengers to occupy a grid space, to see the role that number of exits and percent of plane initially filled (Figure 3) and presence of obstacles played on exit time.

Here we expand upon Giitsidis and Sirakoulis [8] model to add complexity that we believe better captures human behavior. We base our model on real airplane layouts and include more exits than Giitsidis and Sirakoulis. We include the size of passengers as a fundamental part of our model in order to determine the speed at which they can move through a crowded airplane cabin. Passenger size also determines how many can fit into a single grid space allowing for higher crowd densities. Passengers also have their own intrinsic speeds, which introduces the concept of weak and elderly people into the model. Finally we incorporate obstacles in the aisle to model fallen and dropped luggage that may slow passengers down. Due to a lack of access to data from real airplane emergencies, we present a framework here for an agent-based model of aircraft escape dynamics.

**Methods**

Our simulation framework involves four steps: loading the aircraft layout, placing agents, placing objects and targets, and evacuation, below we discuss the steps in each subsection.
Aircraft Layout

We consider aircraft layouts as grids and therefore load them as matrices where different numbers represent different positions, such as walls, seats, and aisles. As an example, figure 5 is the layout matrix of Boeing 737 shown in Figure 4, we ignored the differences between cabins and treated all seats the same, we distinguish the main aisle from the aisle in front of seats, since the latter is narrower, which makes passengers move slower. Areas such as restrooms are regarded as walls that cannot be entered during evacuation. The numbers are summarized in table 1.

Placing Agents and Objects

We stochastically place agents on the aircrafts. In our framework, agents are of different intrinsic speed and body size. With these parameters, we can address many realistic scenarios where, for example, agents with smaller body size can move faster in narrower spaces but those with bigger body size cannot move as fast as their intrinsic speed if they encountered a crowd, or
passengers who are big and slow may block the aisles and slow down the evacuation. Intrinsic speed and body size co-determine agents’ moving speed in each cell, the more crowded a cell is, the more it will slow down the agents passing through.

Correspondingly, we use an occupancy matrix to represent the crowdedness in each grid cell. The occupancy value stands for the percentage of space being occupied, ranging from 0, for a clear cell, to 1, for a fully occupied one. A agent can only enter a cell if his or her body size is smaller than the space currently left in that cell. We also compute the slack, which is the space left in the cell after he or she entered, if the slack is smaller than a certain threshold, the smaller the slack is, the more the agent is slowed down.

The value in the occupancy matrix is initiated according to the layout of aircrafts. Cells correspond to seats and walls are set as fully occupied; the space in front of seats are partially

<table>
<thead>
<tr>
<th>Position</th>
<th>Layout matrix Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wall</td>
<td>102</td>
</tr>
<tr>
<td>Exit</td>
<td>72</td>
</tr>
<tr>
<td>Aisle</td>
<td>0</td>
</tr>
<tr>
<td>Seat</td>
<td>51</td>
</tr>
<tr>
<td>Space in front of seats</td>
<td>255</td>
</tr>
</tbody>
</table>

Table 1: Layout matrix values of different positions on aircraft
occupied since they are narrower than the main aisle; we assume all passengers are in their seats when evacuations start, so the cells on the main aisle are set to be clear.

Each agent is able to estimate the travel time from their current location to any exit, which is the sum of the estimated speed in each cell along the path to that exit based on the current crowdedness. Considering the fact that the travel time may not be exactly estimated, we stochastically assign varying level of error to the judgment of each agent.

We also consider objects that may appear on the aircraft and affect evacuation time. Most passengers bring carry-on luggage on board and store them in the overhead bin, but these objects may fall on the ground during turbulence, blocking or slowing down evacuation until they are removed. Therefore, during simulation, especially at the beginning, we randomly place objects of varying size on the main aisle, and remove them after a certain amount of time when passengers approach them.

**Placing Targets**

Targets are the positions that passengers run to during evacuation. Exits are targets for all passengers, there are also scenarios where family members may move towards each other, or a passenger may need to first retrieve a very important luggage before leaving. We model targets as centers with different importance values. Note that the importance values of all exits are the same, and in most cases, they are the highest among all targets.

The attraction of each target to a passenger is computed based on the importance of the target decaying as the shortest travel time increases. Passengers would choose the target with the highest attraction to move to. In the case of urgent evacuation where we only consider exits as targets, passengers will choose the exit with the shortest travel time. As mentioned in the previous subsection, we assigned different levels of error in human judgment to agents to address the scenarios where passengers may wrongly estimate the travel time, which leads them moving away from an optimal direction.

**Evacuation**

We simulate the movement of passengers over time. At each time step, due to different moving speed, only the eligible passengers are move to the next position, then for all passengers on board, the next direction and the corresponding arriving time is computed. The evacuation algorithm is shown in algorithm 1, 2, and 3.
Algorithm 1 Simulate Evacuation of Aircrafts

1: procedure EVACUATE(numPassengers, passengers, targets)
2: \[ t \leftarrow 0 \]
3: \[ \text{while } numPassengers \neq 0 \text{ do} \]
4: \[ \text{for } p \in \text{passengers do} \]
5: \[ \text{if } p\text{.moveTime} = t \text{ then} \]
6: \[ \text{move}(p) \]
7: \[ \text{if } p \text{ reaches an exit then} \]
8: \[ \text{Remove } p \text{ from } \text{passengers} \]
9: \[ \text{numPassengers} \leftarrow \text{numPassengers} - 1 \]
10: \[ \text{end if} \]
11: \[ \text{end if} \]
12: \[ \text{end for} \]
13: \[ \text{for } p \in \text{passengers do} \]
14: \[ \text{computeNextMove}(p, \text{targets}) \]
15: \[ \text{end for} \]
16: \[ t \leftarrow t + \text{timeStep} \]
17: \[ \text{end while} \]
18: \[ \text{return } t \]
19: \[ \text{end procedure} \]

Algorithm 2 Compute Next Move Cell and Next Move Time

1: procedure COMPUTENEXTMOVE(p, targets)
2: \[ \text{Initialize } \text{nextCell} \]
3: \[ \text{highestAttr} \leftarrow 0 \]
4: \[ \text{for } \text{target} \in \text{targets do} \]
5: \[ \text{travelTime, cell} \leftarrow \text{graphSearch}(p, \text{target}) \]
6: \[ \text{Compute } \text{attr} \text{ using travelTime} \]
7: \[ \text{if } \text{attr} > \text{HighestAttr} \text{ then} \]
8: \[ \text{nextCell} \leftarrow \text{cell} \]
9: \[ \text{end if} \]
10: \[ \text{end for} \]
11: \[ p\text{.nextCell} \leftarrow \text{nextCell} \]
12: \[ \text{update } p\text{.moveTime} \]
13: \[ \text{end procedure} \]
Algorithm 3 Move Passenger

1: procedure MOVE(p)
2:   if p.nextCell still has space then
3:       p.loc ← p.nextCell
4:       Update occupancy values
5:   else
6:       p.nextMoveTime ← p.nextMoveTime + timeStep
7:   end if
8: end procedure

Results and Discussion

The two layouts considered in Figures 4 and 5 are based on Uniteds Boeing 737 and 777 [2]. In both, aisle cells represent 0.43x0.43 m and the spaces front of seats represent 0.38x0.38 m. The Boeing 737 is a smaller plane consisting of two rows of seats. The Boeing 777 is a larger with three rows of seats, and it contains more exits than a 737.

Figure 4 shows a plot of the time in seconds to evacuate the plane plotted against the initial proportion full. This figure only considers the Boeing 737. Note that for the plot and to generate averages, data points along the x-axis were binned. 10 runs are included in each bin. The average speed was set to 1.3m/s [6], and passengers were randomly placed throughout the plane for each run. The variation observed here is due to a variety of things, including initial placement of passengers, their randomized sizes and speeds, and stochasticity in their decision making leading to errors. Supplementary videos 1-4 provide examples. Figure 7 was created by removing randomness from the simulation. All passengers were fixed to the same size, their speeds were set to all be 1.3m/s, and passengers initial locations within the plane were constant. We varied the initial proportion of the plane that was filled and compared it to the time it took for everyone to escape. The same Boeing 737 layout from Figure 4 was used (blue data points) and was compared to that of a United Boeing 777 (black data points). Note that the proportion plotted is in reference to the smaller plane, so each x-value represents the same initial number of passengers in both planes. Removing these aspects of the simulation led to faster escape times. This is largely due to the presence of larger individuals, people with slower speeds, and the errors in their decision making. The deterministic version of the simulation shows that just based on layout alone the escape time from a 777 does not drastically differ from a 737. Although the 777 is larger and has more ground to cover during escape, there are more exits available which accounts for the equivalence to the 737 escape times.
Figure 6: Time in seconds to evacuate plane plotted against initial proportion full. Blue dots are data points and black dots are the averages. This is all for a Boeing 737 layout which contains two rows of seats.

Figure 7: Time in seconds to evacuate plane plotted against initial proportion full for a deterministic simulation. Black points show results for a larger Boeing 777 layout, which has 3 rows of seats. Blue points are for a Boeing 737, as in Figure 6.
**Future Work**

In our simulation, though we have considered other targets than just exits, such as important luggage or family members, that may cause passengers to move towards unpredictable directions, we have not formally investigated how much these factors would affect the evacuation time. This requires further study of human movements. For example, to model parent-child movements, depending on their location on the aircraft and the child’s age, they may move toward each other, or the parent may move towards the child, or they may simultaneously move toward an exit that is optimal for both of them.

We also need to optimize how to incorporate all factors that may affect passengers’ judgment. For example, how to balance physical distances and shortest travel time in choosing the optimal direction to move toward.

**Group member contributions**

Sirui- Methods section, simulation code
Grace - Introduction section, results section, animations

**References**


