Agent-Based Modeling of Emergency Evacuations Considering Human Panic Behavior

Article *in* IEEE Transactions on Computational Social Systems · March 2018

DOI: 10.1109/TCSS.2017.2783332

2 authors:

Ashutosh Trivedi
International Institute of Information Technology Bangalore

Shrisha Rao
International Institute of Information Technology Bangalore

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Agent-Based Modeling of Emergency Evacuations Considering Human Panic Behavior

Ashutosh Trivedi and Shrisha Rao, *Senior Member, IEEE*

***Abstract*— During mass evacuations, many psychological and physical factors are responsible for stampedes and other life threatening situations. Quantitative and qualitative analyses of these factors are of high importance while devising optimal strategies for evacuations. In this paper, we present an agent-based model that considers psychological and physical factors that cause panic in such situations. We have also simulated some simple evacuation scenarios and presented a method to identify possible bottlenecks and shortcomings in the environments during emergency evacuations. Our method also helps in evaluation and analysis of different evacuation strategies. To enable this analysis, we have used a rule-based roadmap approach, where critical nodes in the environment are identified by the evacuation planner and each node has a special rule according to the strategy of the planner. We evaluate different strategies on parameters, such as evacuation time and physical discomfort caused to the agents.**

***Index Terms*— Agent-based modeling, egress, emergency evacuation, panic, simulation.**

I. Introduction

EMERGENCY scenarios, such as fires, earthquakes, terrorist attacks, and so on, may require evacuations of relatively large numbers of people in short times. However, the onset of panic in a crowd of people under such circumstances may trigger a stampede with serious, even fatal consequences—people crushed or trampled [1], [2]. Panic makes people behave irrationally at an individual level, and also creates undesirable collective behavior like stampedes, causing serious threats to the lives and well-being of all concerned—it is known from painful collective experience that panic can actually hinder timely evacuations and can even cause mass fatalities. The unpredictable and nonadaptive behaviors of crowds in panic situations constitutes a real danger to the lives of people [3]–[5]. Such dynamics of crowds apply when large numbers of people are gathered for a common purpose, such as at concerts, movie screenings, sporting events, rallies, and other mass events. Ease of transportation and communication in modern times has fostered the increased occurrence of such crowd events, which has in turn increased the probability of hazards due to stampedes, which is a most calamitous form of collective crowd behavior. People in crowds also show herding behavior and follow other people mindlessly, which may fuel stampedes [6]–[8].

Human crowd behavior, which is generally in order, is liable to be completely different and highly uncoordinated during an emergency evacuation, due to panic and fear; therefore, there is a need of a mechanism to manage crowds in such situations [9]–[11]. The current best practice in any mass gathering is to have enough emergency personnel to assist the crowd during emergency evacuation. In some places which are not so large, such as lectures halls, theater halls, supermarkets, and so on, this practice is mostly overlooked. Organizers mostly rely on precautions, such as signage, cues, and evacuation notifications, which, however, run counter to both individual and mass psychology during emergencies [12]–[14].

Mathematical models play major roles in explaining the dynamics of evacuations, and help engineers, architects, and scientists to make better decisions for evacuation planning. There are two broad types of models in use. One type consists of the *macroscopic* models, which give mathematical equations describing the escape dynamics [8], [15]. Helbing's *social force* model [6] is a notable member of this category. The other type consists of *individual-based* models, such as cellular-automata-like approaches to molecular dynamics and agent-based modeling [16]–[19]. Agent-based modeling has been very effective in asking "what-if" questions about crowd behaviors [20]–[22]. In the case of crowd safety and security, it becomes even more significant due to the lack of real-life experiments and the data needed by other modeling techniques [23]. It is easier to simulate cognitive behaviors—such as the occurrence and causes of panic, the heterogeneity of agents, communications between agents, and so on—in an agent-based model (ABM). For example, a decision an agent makes during an evacuation may affect its physical as well as psychological parameters. Changes to its psychological parameters may in turn affect future decisions made by the agent. Such issues can be accounted for in our model, which is individual-based, but not in any macroscopic model. Helbing *et al.* [8] discuss the characteristic feature of escape panic, but there are no cognitive or psychological parameters described in their model, which trigger the undesirable behavior, such as jamming and overcrowding. Quantifying these aspects and predicting their effect on how people choose exit doors is of high importance [24]–[27].

In this paper, we present an approach for the analysis of strategies for safe evacuations using an ABM, to improve the ability to design better evacuation strategies as has been long

Manuscript received December 4, 2017; accepted December 7, 2017. *(Corresponding author: Shrisha Rao.)*

A. Trivedi is with Spext, Bangalore 560 095, India (e-mail: ashu@spext.co).

S. Rao is with the International Institute of Information Technology Bangalore, Bengaluru 560100, India (e-mail: shrao@ieee.org).

The source code and documentation for our work can be accessed at https://github.com/codeAshu/ABMS-Panic-Evacuation.

Digital Object Identifier 10.1109/TCSS.2017.2783332

sought [3], [11], [28]. Using this approach, it is possible to identify bottlenecks in the floor plans of built areas. It is also possible to compare and evaluate different designs and locations of exit doors with respect to properties, such as position, width, and number of the doors (if there are doors more than one). Our evaluations consider evacuation times and the levels of physical discomfort of the human agents in the environment, and enable us to model and quantify the effect of panic generated among humans during life-critical evacuation situations. To incorporate cognitive aspects, our model permits us to identify the cause and effects of panic. Panic in our model is associated with certain factors, such as distance from the exit door, velocity of neighbors heading toward exit, count of nearby agents who have high degrees of physical discomfort, and lag in velocity (if an agent is moving slowly due to injury, then the panic level is taken to increase). The velocity of an agent is of course a vector; if the direction of an agent is not aligned with that of fellow agents, or if the speed of the agent is lower than that of nearby agents, then the panic of that agent increases. Because of the social nature of human physical discomfort, any occurrence of injury or severe discomfort among nearby agents also affects the panic level of an agent.

There may be many more factors affecting the panic level of a human being in a crisis situation [29], [30], but as an initial step, in our model, we have identified these factors and accounted for them in our mathematical model of panic.

For crowd path planning, we make use of the *boids* behavior model of Reynolds [31]. To fit it to human crowds, we amplify certain behaviors based on emotional parameters, such as panic, and physical parameters, such as location and velocity. We also use a relevant aspect of the *social force* model [8] for the calculation of physical pressure among agents in a crowd.

For complex environments with many obstacles, our approach is to evaluate assistance strategies. We may describe *assistance* in this regard as prior (standing) instructions or real-time instructions given to agents in a crowd during emergency evacuations. It may come from outside of the crowd environment through communications, signage, and announcements.

We build a roadmap of the environment, which is computed during the preprocessing stage. A roadmap is a graph of the environment which has nodes and paths [32], where every node contains certain rules, which an agent has to follow if it reaches that node. All these rules combined together form a strategy of evacuation; for example, at a narrow passage, the rule can be *follow the leader* or *form a queue*. We can implement different strategies and select the best of them, finding the best permutation of the rules based on some performance parameters. Another physical parameter in our model is the visibility of the exit door to the agents. *Visibility* here refers to the knowledge of the exit door and its immediate vicinity by the agents. In many evacuation situations, it is a critical factor.

We use MASON, a Java-based multiagent simulation tool [33], for simulation of some evacuation scenarios. In our simulations, we have considered different permutations of exit doors and their visibilities. Even the simpler cases prove to be informative enough to warrant a qualitative analysis of the positions of the exit doors. For instance, we have simulated the evacuation of a lecture hall with 160 agents (attendees). At the time of evacuation, one door that is most used for entering the hall is found to be crowded, but even with limited assistance with signage, crowd evacuation can be managed.

In summary, the following are the novel contributions of this paper:

1) We propose an ABM of human evacuation under panic. This model conforms to published studies in psychology which have analyzed the behavior patterns seen in panicked evacuees.
2) We present a simulation study of these psychological factors on the agents during evacuation.
3) We use the AMB and further simulate some evacuation scenarios to indicate how to identify bottlenecks in environments that may need to be evacuated.
4) We create a system to test different evacuation strategies using a rule-based approach on these agents; this is a way to analyze what-if scenarios on emergency evacuations.

The rest of the paper is structured as follows. Section II describes the context briefly with reference to the relevant prior literature. Section III gives our system model. Section IV discusses results of the simulations, and Section V presents the conclusion.

II. Related Work

The simulation and modeling of animal flocking behavior has been done extensively from years by researchers of computer graphics and robotics. More recently, mass human behavior and crowd behavior under panic have also been studied.

A. Previous Crowd Evacuation Work

The theory of *boids* is one of the most influential in the behavioral study of flocks, by Reynolds [31], [34]. It is a model of polarized, noncolliding aggregate motion, such as that of flocks of birds and herds of beasts, which demonstrate emergent properties, i.e., where global behavior arises from simple local rules applied independently on each agent, such as rules governing separation, cohesion, and alignment. Studies can also be empirical; for example, Benthorn and Frantzich [17] recruited volunteers for an emergency evacuation and presented an experimental study.

An important later study by Frank and Dorso [15] showed that in the presence of obstacles, a shortcut people might take in order to get to the exit will probably do no better (*clever is not always better*). Ji and Gao [16] simulate the evacuation process in a dancing hall. They introduce a leader–follower model, where each group is assigned a *leader* and each leader is attached to one exit door; they also examine the coordination between leaders. The A* algorithm [35] is used for path finding in the dancing hall. Yang *et al.* [36] present a mathematically optimal leader–follower model.

To control a flock or group, there have been studies in robotics, and some have been described as *shepherding* behavior. Schultz *et al.* [37] introduce a genetic algorithm for a hybrid cognitive reactive system, with rule-based learning to control the movement of a reactive agent. Vaughan *et al.* [38] construct and simulate a robot that guides a flock of geese in a circular environment. Recently, Boukas *et al.* [39] present a model for a robot that assists in crowd evacuation.

Funge *et al.* [40] use shepherding behavior for territory safeguarding, which can be applied in mine sweeping and surveillance (e.g., preventing birds from flying over runways at airports and other restricted aerospace). Although all of the previous techniques provided simplistic planning and navigation capabilities, Bayazit *et al.* [32] show how global information from a roadmap of an environment allows more sophisticated flocking behaviors. They apply this technique to homing, goal searching, traversing narrow passages, and shepherding behaviors of flocks. Another work on shepherding behaviors by Lien *et al.* [41] extends this approach to shepherding by improving locomotion and verity of other behaviors, such as coverage, patrolling, and collection of a heard. They also describe the methods for selection of the steering point, how to approach a flock to collect them in groups and assign shepherd agents to groups.

B. Current Panic Models

The social force model of Helbing *et al.* [6] is based on psychological and social forces like nervousness and panic. In this model, panic acts as a force on individuals to alter their velocity. The model also accounts for the forces applied by fellow human beings while running and pushing. Taking velocity as a parameter, Helbing *et al.* [6] arrive at simulation results about evacuation timings which tend to validate the adage *faster is slower*, when it comes to evacuation in the event of panic. This paper is also supported by various sociopsychological studies and empirical observations [9].

On the studies of rescue robots, Murphy [42] and Bethel and Murphy [43] study how volunteers reacted in panic during urban disasters. They suggest using voice communications with human agents to ease anxiety and reduce panic. Robinette and Howard [44] try to apply the same concept in robots to guide the crowd in emergency evacuation. Pelechano *et al.* [18] consider communication as a key factor during the evacuation and show how a random leader from among the crowd can be helpful during emergencies if it has effective communication. Viswanathan *et al.* [45], [46] have studied some of the motion planning methods known today and have given a quantitative comparison, which proved very useful to us to choose social force model for our study.

A recent study on understanding panic by Shiwakoti *et al.* [47] proceeds by examining empirical data collected from panicking Argentine ants to study crowd panics at turnings and intersections. Sarshar *et al.* [48] demonstrate how panic can dynamically vary from passenger to passenger with different physical (or mental) conditions. Vermuyten *et al.* [49] present a survey of optimization models for pedestrian evacuation and design problems. They show that social force and ABMs are powerful tools for model complex and heterogeneous systems. Ma *et al.* [50] use the extended social force model to show the effect of pedestrian evacuation under limited visibility. Li *et al.* [51] present different simulation scenarios on passenger evacuation from the platform of urban metro stations.

III. Agent Attributes and Behavior Model

Our model is similar to classical many-particle system models, with each agent being similar to a particle with a size and a direction of motion, moving in a continuous 2-D Euclidean space. Agents can be added or removed (the model allows for simulations with different numbers of agents).

Each agent has certain *attributes* and *behaviors*; also there are *refinement factors* associated with the behaviors. An attribute x_i of an agent i (as given below) specifies the physical and psychological properties of the agent. Behaviors z are path-planning rules each agent follows. The *refinement factors* m_i^z alter the intensity of a behavior z based on their values. There are also some distances d_i^z associated with agent i, which define the scope of a certain behavior z. These distance are the radial distance from an agent, which define the boundary of the group we consider for any behavior. We can obtain the values of all these for an agent in real time, and can also record the values in logs for later analyses.

The environments modeled are the following:

1) the human agent environment;
2) the obstacle environment; and
3) the variable visibility environment.

A human agent environment has all the agents (with their attributes, behaviors, and refinement factors), which are to collectively simulate a crowd of human pedestrians.

The obstacle environment abstracts relevant properties of obstacles, which are its position and size. (Obstacles are static and do not move.)

The variable visibility environment is associated with exit doors. Every door has certain visibility range. If an agent is within the visibility range of a door, it is assumed that the agent has the information about the position of the exit door. Else, an agent is unaware of the exit door.

A. Modified Boids Model

Our model for the movement of the agents is based on *boids* rules [34]. The theory of boids posits that it is possible to simulate flock movement by following three simple rules.

1) *Cohesion:* Every individual agent tries to move toward the center of mass of the group.
2) *Separation:* An agent moves away from a fellow agent if the distance between them is less than a certain *avoid distance*.
3) *Alignment:* Every agent tries to align itself with the direction of the group.

To incorporate this model in human path planning, we vary the weights of these according to the psychological and physical parameters of the agent, using refinement factors. In our simulations, all agents follow these rules at the micro level, but their movements at macro level are governed by their goal velocities, which are calculated based on their relative positions with respect to the obstacle (if any), and the exit doors in the environment.

TABLE I
ATTRIBUTES OF AGENTS

Attribute of the agent i	Symbol
radius	r_i
weight	w_i
velocity	$\vec{v_i}$
position	$\vec{p_i}$
panic	γ_i
ease distance	l_i
avoid distance	d_i^a
centroid distance	d_i^c
alignment distance	d_i^l
goal factor	m_i^g
cohesion factor	m_i^c
separation factor	m_i^s
alignment factor	m_i^l
obstacle factor	m_i^o

Table I presents the attributes and the symbols we use to represent them. At any time t, an agent i is specified by the 5-tuple of primary attributes $\langle r_i, w_i, v_i, p_i, \gamma_i, l_i \rangle$. A brief description of these is as follows.

1) r_i: Since our simulation is particle like, it is the radius of an agent i (which represents half the shoulder width of a human [6]).
2) w_i is the weight of agent i. This attribute is used in calculating the physical force experienced by the agent.
3) $\vec{v_i}$ is the instantaneous velocity of an agent i at time t. It is a vector with magnitude and direction.
4) $\vec{p_i}$ is the position vector of agent i.
5) γ_i is the panic level of the agent, $0 \le \gamma_i \le 1$, where a value of 0 means no panic and 1 the highest.
6) l_i is the ease distance for agent i, which is the maximum acceptable distance to an exit door. If an agent does not see an exit door within this distance, its panic is liable to increase.

The radial distances associated with an agent i, $\langle d_i^c, d_i^a, d_i^l, d_i^e \rangle \in \mathbb{R}$, which are used for defining the scope of different behaviors of agent i. A brief description of these is as follows:

1) d_i^c: is the radial distance from the agent i, which define the boundary of the group we consider for cohesion behavior.
2) d_i^a: This is the avoid distance of agent i, i.e., the minimum distance by which i separates itself from an agent $j \neq i$.
3) d_i^l: is the radial distance from the agent i, which defines the boundary of the group we consider for alignment behavior.

There are some behavior refinement factors $\langle m_i^g, m_i^c, m_i^s, m_i^l, m_i^o \rangle \in \mathbb{R}$ (where $\mathbb{R}$ denotes the set of real numbers), used for altering the behaviors of agent at different situations. The values of these factors depend upon physical factors, such as position and velocity and sociopsychological factor like panic.

1) m_i^g is a factor associated with the behavior of *getting to the goal*. Every agent first decides its goal, which is the closest exit door. If agent i cannot see any door, its goal velocity component is zero, in such cases, this component does not contribute toward the motion of the agent. This factor for goal velocity always has higher value to make goal velocity dominate the motion at macro level, which is around 45%–50% of the total velocity.
2) m_i^c is a factor associated with *herding* behavior, which dominates when an agent's knowledge of the environment is limited. In that case, agents (like people) try to follow others. Even in very high panic situations people show herding behavior [4], and agents in our model do likewise.
3) m_i^s is a factor associated with *separation* behavior. On increasing this factor, an agent is liable to separate itself from a herd to a safe distance. Near exit doors where every agent is trying to get out, there is increased risk of injury or physical discomfort to agents. In such situations, we can separate agents from one another by increasing this factor.
4) m_i^l is a factor associated with the *alignment* behavior. When we require disciplined motion of agents such as queuing, increasing this factor increases the alignment of the group of agents. By combination of separation m_i^s and alignment m_i^l, we get the required queuing motion. This is beneficial when agents have to move through narrow passages while they are being assisted during an emergency evacuation.
5) m_i^o is associated with *obstacle avoidance* behavior. When there are many obstacles in an environment, increasing the value of this factor increases the behavior of avoiding obstacles swiftly while moving toward the goal.

A complete architecture of the system model is presented in Fig. 1. An agent i first decides its goal and calculates its goal velocity, which is then multiplied by m_i^g to decide its weighted goal velocity component. After that, velocity components due to boids behavior rules (cohesion, alignment, and separation) are added. These constitute 15%–20% of the total velocity. Finally, the velocity component for obstacle avoidance is added, which constitute around 25%–30% of the total velocity. All these components are variable, and the corresponding refinement factors too continually change during simulation.

After the velocity calculation, we calculate the panic level γ_i of agent i. How we calculate panic and on what parameters it depends, is explained in Section III-B. After the calculation of panic, we update the velocity of the agent at time $t+1$ by exponential smoothing [52], where the smoothing factor is γ_i which represents the value of panic. The granularity of the time intervals (taken to be unity) is small enough that the total velocity of an agent is constant during an interval.

We update the position and the velocity of the agent for time $t+1$ based on the values at time t, and calculate the physical discomfort of the agent.

In Algorithm 1, from lines 1 to 4, we initialize the vector components for each behavior.

In lines 5–9, we calculate the center of the mass $\vec{w}$ of the group for an agent i for the cohesion behavior. First,

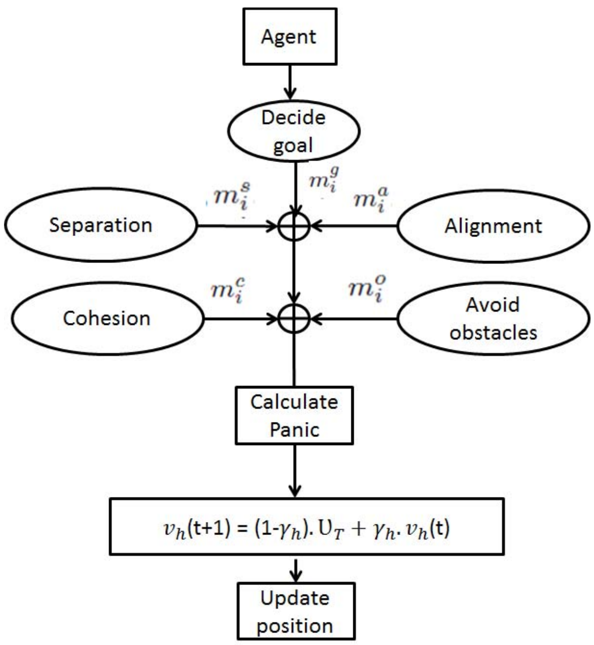


Fig. 1. System model.

Algorithm 1 Motion Algorithm

1: **procedure** MOVE (agent i)
2: *Vector* $\vec{w}, \vec{x}, \vec{y}, \vec{z} \leftarrow 0$
3: *Vector* $\vec{v} \leftarrow 0$
4:
▷ *toward center of the mass of the group*
5: **for** ***each*** *human j in* d_i^c **do**
6: **if** $j \neq i$ **then**
7: $\vec{w} = \vec{w} + \vec{p}_j$
8: $\vec{w} = \vec{w}/(N_c - 1)$ ▷ mean position
9:
▷ *try to keep a small distance*
10: **for** ***each*** *human j in* d_i^a **do**
11: **if** $j \neq i$ **then**
12: $\vec{x} = \vec{x} - (\vec{p}_j - \vec{p}_i)$
13:
▷ *try to match velocity*
14: **for** ***each*** *human j in* d_i^l **do**
15: **if** $j \neq i$ **then**
16: $\vec{y} = \vec{y} + \vec{v}_j$
17: $\vec{y} = \vec{y}/(N_l - 1)$ ▷ mean velocity
18:
▷ *avoid obstacles*
19: **for** ***each*** *Obstacle O in* d_i^a **do**
20: $\vec{z} = \vec{z} - (\vec{p}_o - \vec{p}_j)$
21:
▷ *add all the velocity components*
22: $\vec{v} = (\vec{w} - \vec{p}_i) \cdot m_i^c + \vec{x} \cdot m_i^s + (\vec{y} - \vec{v}_i) \cdot m_i^l + \vec{z} \cdot m_i^o$
23: **return** v

we calculate the sum of the positions of the agents, which lie within distance d_i^c, and after that we divide it by N_c, the count of agents in the group.

Next, in lines 10–13, we calculate the vector component for the separation behavior $\vec{x}$. We calculate the vector away from all agents j which lie in the avoid distance d_i^a, where $j \neq i$. We add all these components to get the resultant repulsion vector.

In lines 14–18, we calculate the vector component for alignment behavior $\vec{y}$. We calculate the sum of the velocities of other agents, which are within distance d_i^l of agent i. After that, we divide it by N_l to calculate the mean velocity of the group, where N_l is the count of agents in the group.

In lines 19–21, we calculate the vector away from all the obstacles $\vec{z}$, which are within the avoid distance.

In line 22, we calculate the vector toward the mean position of the group and multiply it with the cohesion refinement factor m_i^c. Similarly, we multiply $\vec{x}$ with the separation factor m_i^s. After these, we calculate the vector toward the mean velocity of the group and scale it with the alignment factor m_i^l. We also adjust $\vec{z}$ with factor m_i^o. Finally, we add all these components and get the resultant vector.

In the worst case, any group can have all the agents, so the complexity of this algorithm is also $\mathcal{O}(n)$.

B. Model of Panic

Panic has an effect on intelligent judgment [2], [4], [22], [53]. If an agent is highly panicked, it tends not to make rational decisions according to the normal path planning rules.

We present a model that accounts for some of the factors which trigger panic, and mimic the behavior seen in panicked humans. Based on related studies [2], [4], [53] about panic in human crowds, the factors we consider are as follows.

1) The distance from the exit door, or the visibility of the door to the agent.
2) The velocity of neighbors moving toward exits. We define velocity as a vector. So, if the direction toward which a agent is moving is not aligned with fellow agents, or if the speed is slower than others', then it is very likely that the panic of that agent will increase.
3) The count of nearby agents who have high degrees of physical discomfort. This factor has a significant effect on panic—because of the social nature of physical discomfort of humans, physical discomfort of nearby agents tends to increase the panic of an agent.
4) The lag in velocity. If an agent is moving slowly in relation to its group (which may be due to injury), then its panic level is liable to increase.

We define the individual panic component at time t due to these factors as $\delta_k(t)$, $k \subset [1, 2, 3, 4]$ representing one of the above described factors. These $\delta_k(t)$ are all normalized to between 0 and 1. We define $\zeta_i(t)$ as the sum of all the $\delta_k(t)$ (this also scaled to lie between 0 and 1) for an agent i at time t. If $\gamma_i(t-1)$ is the panic level of the agent at time $t-1$, the total panic at time t is the average of $\zeta_i(t)$ and $\gamma_i(t-1)$. In this manner, the panic level of agent i varies with time depending on the changing values of the panic components.

For an agent i with position $\vec{p}_i$, the position of any door d is defined as (l_d, r_d), where l_d and r_d are the coordinates of the left and right sides of the door; we calculate vector $\vec{m_d}$ as the midpoint of the door. The Euclidean distance of an agent from nearest m_d is as given in (1)

$$D_i = (\vec{m_d} - \vec{p_i}). \tag{1}$$

If the agent is not in within its visibility range of any door, then we take D_i as the maximum value L, which is the length of the environment. If $D_i \geq l_i$, then panic is liable to increase. If r_i is the radius of the agent, we define the ease distance as

$$l_i = c \cdot r_i \tag{2}$$

where c is a proportionality constant. We calculate δ_1 due to distance as

$$\delta_1 = \frac{1}{L} \cdot (\|D_i\| - \|l_i\|) \tag{3}$$

To calculate the panic factor δ_2 due to neighboring velocity, we first calculate the mean velocity $\tilde{v_i}$ of the group of agents within alignment behavior boundary d_i^l. If N_l is the count of agents within d_i^l and $\vec{v}_{\max}$ is the maximum allowed velocity, we calculate the component of panic due to this factor in (5)

$$\ddot{v_i} = \frac{1}{N_l} \sum_{\left(j \in \text{agent within } d_i^l\right)} \vec{v_j} \tag{4}$$

$$\delta_2 = \frac{1}{\|\vec{v}_{\max}\|} \cdot (\|\ddot{v_i}\| - \|\vec{v_i}\|). \tag{5}$$

To address the factor of physical discomfort, we take the count of nearby agents whose discomfort is above a certain threshold. How to calculate physical forces among agents is explained in Section III-C. If ω agents within distance d_i^c of agent i have levels of physical discomfort above the threshold T, and there are total n agents in the environment, δ_3 is the panic component due to this is the count ω

$$\delta_3 = \frac{1}{n} \cdot \omega. \tag{6}$$

The lag also contributes toward building the panic level. If $\vec{v}_{\min}$ is the usual agent velocity (when there is no panic), then the factor due to this is

$$\delta_4 = \frac{1}{\|\vec{v}_{\max}\|} \cdot (\|\vec{v}_{\min}\| - \|\vec{v_i}\|). \tag{7}$$

Finally, we sum all the component and normalize the value of γ_i with η. We give equal weights to each factor and take the value of η as 4, and we calculate the panic at time t as the average of $\gamma_i(t-1)$ and $\zeta_i(t)$ in (9)

$$\zeta_i(t) = \frac{1}{\eta} \sum_{k=1}^{4} \delta_k \tag{8}$$

$$\gamma_i(t) = \frac{\gamma_i(t-1) + \zeta_i(t)}{2}. \tag{9}$$

Quantifying panic in this manner is a good tool to have in evacuation dynamics, as it provides a real-time sense of any disorderly decisions made by agents during evacuation. The effect of panic on motion planning is explained in Section III-E.

C. Model of Physical Discomfort

For quantitative analysis of physical force among agents, we use (10), taken from Helbing *et al.* [6]. According to the original model, three types of forces act on a given agent i with mass m_i and instantaneous velocity $v_i(t)$. The first force f_i is a restorative force which steers the agent toward a desired velocity v_0. The second f_{ij} is a repulsive force between agents i and j. The third f_{iW} is between an agent and the wall. In the original model, the repulsive force is given as

$$f_{ij} = \{Ae^{(R_{ij}-d_{ij})/B} + k\eta(R_{ij} - d_{ij})\}m_{ij} + \kappa\eta(R_{ij} - d_{ij})\Delta v_{ji}^t t_{ij} \tag{10}$$

where

$$\eta(x) = \begin{cases} x, & \text{if } x \geq 0; \\ 0, & \text{if } x < 0. \end{cases} \tag{11}$$

Here $R_{ij} = r_i + r_j$ is the sum of the radii of the two agents, $d_{ij} = |r_i - r_j|$ is the physical distance between the two agents, and $m_{ij} = ((r_i - r_j)/d_{ij})$ is the unit vector pointing from agent j to agent i. In addition, $A = 2000$ N and $B = 0.08$ m are the repulsion coefficient and the fall-off length of interacting agents respectively. The second term of the right hand side (RHS) of (10) contributes to counteracting body compression, and the third term to the sliding friction force. In addition, $k = 12\,000$ kg/s^2 and $\kappa = 24\,000$ kg/ms are, respectively, the body force constant and the sliding friction force constant used.

We require a measure of the physical pressure acting upon an agent i due to other agent j. Because of the physical nature of the force, it occurs when agents physically touch other, i.e., where $R_{ij} - d_{ij}$ is positive. We observe from (10) that the second and third terms of the RHS (which are counteracting body compression, and the sliding friction force, respectively) model the measure of physical pressure, and come into play only when agent i and j touch each other, i.e., $R_{ij} - d_{ij}$ is positive. The first term of the RHS of (10) is nonzero even when agents i and j do not touch each other, but due to the very low value of B, it is not significant.

We use (10) as a measure of the physical force acting among agents. According to the original model of Helbing *et al.* [6], any agent above a desired velocity is injured and becomes an unmoving obstacle for others, if the sum of magnitude of radial force acting on it divided by its circumference exceeds a pressure of 1600 N/m.

D. Simulation Algorithm

In Algorithm 2, for each agent i, we first initialize the vector components in line 3.

In line 4, we get the goal velocity $\vec{\vartheta}_g$ by calling the GOAL method for an agent i, which we also scale with the factor m_i^g. The GOAL method is strategic to the floor plan, the visibility of the exit doors, and so on. For example, in the basic case [see Fig. 5(a)], when there is only one exit door and its visibility is universal among all agents, the goal vector points toward the position of the exit door. Section III-E discusses the GOAL method when agents do not have complete information about the environment, and when there are obstacles in the environment.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Algorithm 2 Simulation Algorithm

1: **procedure** EVACUATION( )
2: **for *each*** *agent* i **do**
3: *vector* $\vec{\vartheta}_i, \vec{\vartheta}_g, \vec{\vartheta}_m \leftarrow 0$
4: $\vec{\vartheta}_g \leftarrow$ GOAL $(i) \cdot m_i^g$
5: $\vec{\vartheta}_m \leftarrow$ MOVE(i)
6: $\vec{\vartheta}_i \leftarrow \vec{\vartheta}_g + \vec{\vartheta}_m$
7: **if** $\gamma_i \geq 0.5$ **then**
8: $\vec{\vartheta}_i \leftarrow 0$
9: $m_i^s, m_i^l, m_i^o \leftarrow 0$
10: $\vec{\vartheta}_i \leftarrow$MOVE$(i)$
11: $\gamma_i \leftarrow \gamma_{\text{mean}}$
12: $\vec{v_i}(t) \leftarrow (1-\gamma_i) \cdot \vec{\vartheta}_i + \gamma_i \cdot \vec{v_i}(t)$
13: $\vec{p_i}(t+1) \leftarrow \vec{p_i}(t) + \vec{v_i}(t)$
14: $\vec{v_i}(t+1) \leftarrow \vec{v_i}(t)$
15: $F_i = \sum_{(j \in \text{agent within } d_i^c)} f_{ij}$

In line 5, we compute all other components of the velocity by procedure MOVE (which is as explained in Section III-A). In line 6, we add these velocity components together.

We check the panic level of the agent in line 7 and if it is greater than or equal to a threshold (0.5 in our simulations), lines 8–11 execute. Panicky individuals tend to show herding behavior, so we call the MOVE method with only cohesion (herding) m_i^c factor, with all other factors set to zero, to simulate the herding behavior seen under panic.

After that, in line 11, we update the panic of the agent as the mean panic of the agents around it, $\gamma_i \leftarrow \gamma_{\text{mean}}$. Due to social nature of human judgments, it is known that high panic in agent i eventually comes down to mean panic level of its local group.

In line 12, our equation for assigning the human velocity at time $t+1$ takes the lack of intelligent judgment under high panic into account by updating the velocity at time $t+1$ by exponential smoothing, with panic γ_i as the smoothing factor. When γ_i has high value, the weighted sum of the velocity $\vec{\vartheta}_i$ (calculated by intelligent path planning) is low in comparison to the existing velocity $\vec{v_i}(t)$. Whereas at very low panic situation, $\vec{\vartheta}_i$ have much higher weighted sum than previous velocity $v_i(t)$

$$\vec{v_i}(t) \leftarrow (1-\gamma_i) \cdot \vec{\vartheta}_i + \gamma_i \cdot \vec{v_i}(t) \tag{12}$$

After calculating the resultant velocity at $\vec{v_i}(t)$, we update the position of the agent in environment at time $t+1$. We take τ as the time fraction to update the velocity and position. Finally, we update the position as

$$\vec{p_i}(t+1) \leftarrow \vec{p_i}(t) + \vec{v_i}(t) \cdot \tau. \tag{13}$$

In simulations, we run the algorithm every τ time interval taken to be unity and small enough that the velocity of an agent does not change within the interval. We also update the velocity at $t+1$ with $\vec{v_i}(t)$.

In line 15, we calculate physical discomfort F_i on the agent i by other agents, which are in centroid distance d_i^c, by (10) explained in Section III-C.

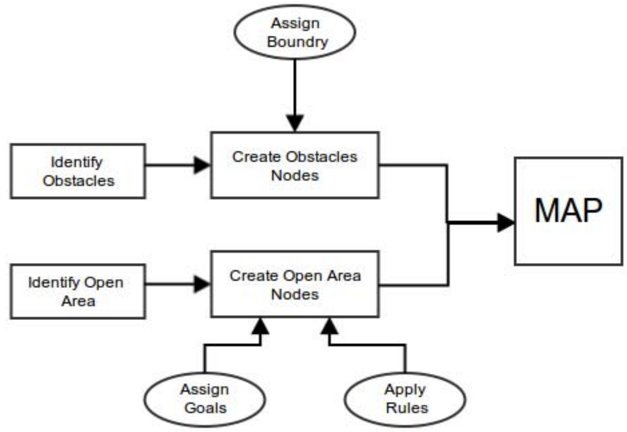


Fig. 2. Flowchart for building the map.

To consider the time complexity of Algorithm (2), the procedure MOVE can in the worst case be of order $\mathcal{O}(n)$. Because of the **for** loop which runs once for each agent, the total complexity of this algorithm is of order $\mathcal{O}(n^2)$.

E. Model of Assistance

When a crowd is trained for evacuation by prior instructions, or guided by some mode of external communication, such as announcements and exit signs, there arises a need to precisely validate and evaluate the effects and results of these techniques, though any assistance is helpful in emergency evacuations. In our simulations, shown in Figs. 5(b)–7(a), when agents do not know about the exit doors, the evacuation time increases. The results are shown in (Fig. 8). In addition, introduction of an obstacle in some environments renders earlier successful evacuation strategies ineffective. Some rational agents can help by searching for hidden (unknown) exits, but it is best if a crowd is guided by a tested strategy.

In practical settings, it is very unlikely that pedestrians would remember the map of the whole environment, and it is always better to guide a crowd through exit signs. The model of assistance is primarily the GOAL (see Algorithm 3) procedure, which computes the *goal velocity* $\vec{\vartheta}_g$. It can be executed only given complete knowledge of the environment. A rule-based roadmap approach enables us to distribute the knowledge of the environment.

In Fig. 2, we describe the procedure to build the map, which we represent with symbol $\triangle$. A map is a directed graph which consist of nodes and edges. Every node in the map has the following characteristics: 1) node boundary; 2) destination; and 3) rule. *Node boundary* is the area in the environment, which is represented by the node. *Destination* is the next node and there is an edge from the node to destination. *Rule* defines the behavior agents follow in the path while moving from node to destination.

While building the roadmap, we first identify the critical areas in the environment, and then identify possible nodes and the node boundary. Second, we assign the destination of each node. In this process of building the map, we also implant our strategies as rules on the nodes of the map. These rules

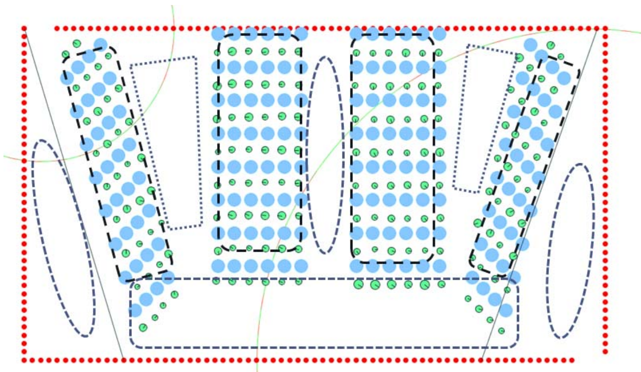


Fig. 3. Identifying nodes in the environment.

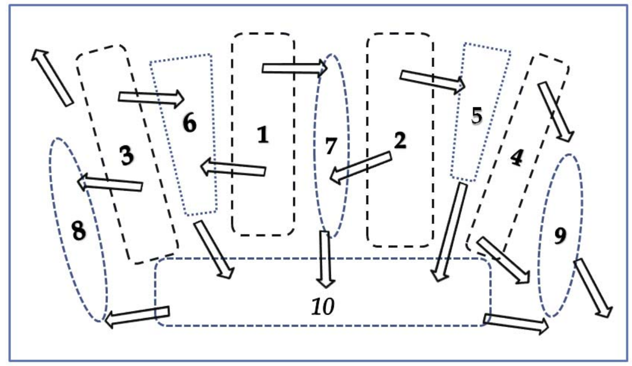


Fig. 4. Assigning destination for nodes.

are values of refinement factors to simulate certain motion behavior. Every agent at that node follows these predefined rules.

Algorithm 3 Goal Finding Algorithm

1: **procedure** GOAL(agent i)
2: *vector* $\vec{g}$ $\leftarrow$*Initialize to* 0
3: **if** $\triangle = NULL$ **then**
4: $\triangle$ $\leftarrow$*Build Map*
5: *Set Factors* ($\triangle$)
6: $\vec{g} \leftarrow$ *Next Node* ($\triangle$, $\vec{p}_i$)
7: **return** $\vec{g}$

Algorithm 3 presents the steps to apply a rule-based roadmap approach. In line 2, we first initialize the vector. After that, in line 3, we check if the map of the environment $\triangle$ is built already. If the map is not built, we first build the map $\triangle$ by calling method *Build Map* explained in Fig. 2.

After that, in line 5, we assign the rules that agent i has to follow, by setting refinement factors of different behaviors. In method *Set Factors*, based on the position p_i of the agent in relation to the node boundary of the node in the environment, we identify the node in which the agent lies. We assign the rule implanted at that node in the map $\triangle$ to agent i.

Finally, in lines 6 and 7, we assign a goal velocity pointing toward the destination node.

Build Map and *Set Factors* are strategy- and environment-dependent. It is up to the user (the designer of the area evacuation plan) to identify critical nodes and evacuations rules as per their strategy. We are creating a general framework to test many strategies; it is not specific to any environment and strategy, which is the primary aim of this paper—to enable the process of validating different evacuation strategies.

We show this approach by a simulation of a lecture hall emergency evacuation (see Fig. 3); based on the floor plan of a real lecture hall known to the authors, there are 160 agents (lecture attendees) and 160 seats, which seat the agents but also become obstacles during evacuation. There are two exit doors, one at the lower right corner and one at the upper left corner. Agents are more familiar with the lower door, which is regularly used, so its visibility is very high. The upper door leads to an uncommon area and is not used frequently, hence its visibility is low.

TABLE II
PROPERTIES OF AGENT IN OUR SIMULATION

Attribute	Symbol	Value
radius	r_i	0.25m – 0.4m
weight	w_i	40 Kg – 80 Kg
velocity	$\vec{v_i}$	0.35 m/s–1.95 m/s
panic	γ_i	0–1
ease distance	l_i	$10 * r_i$
avoid distance	d_i^a	$2 * r_i + 2$
centroid distance	d_i^c	$8 * r_i$
alignment distance	d_i^l	$4 * r_i$
goal factor	m_i^g	6.5
cohesion factor	m_i^c	1.5
separation factor	m_i^s	2.5
alignment factor	m_i^l	1.5
obstacle factor	m_i^o	3.5
threshold force	T	750 N/m

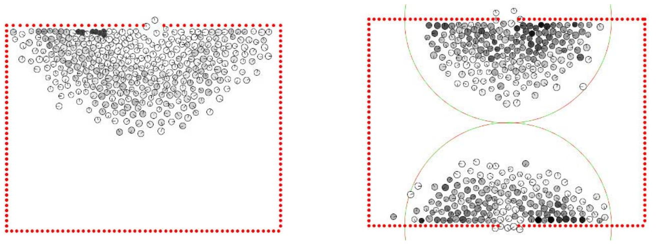
For a proposed evacuation of this lecture hall, we have identified the nodes which are presented in Fig. 3. The nodes with their node area and next destination as presented in Fig. 4. The arrows represent the connectivity between nodes.

Rules implanted at the nodes abstract the overall evacuation strategy. Initially, agents are at nodes 1–4. During evacuation, an agent at one of those nodes is guided toward the nearest of nodes 5–9. Considering node 5 whose area is quite narrow, we increase m_i^l for that node. Node 10 is wider, so mild herding behavior (increased m_i^c) is better to get more agents to the exit easily. In our simulation, we have some randomness, e.g., an agent at node 1 can either go to node 6 or node 7 (also because of a blocked path, and so on).

Path planning for each agent is classified into two categories: Macro level, rule-based road map approach which gives the goal direction and micro level path planning by native agent behavior rules, such as cohesion, separation, alignment and obstacle avoidance with their varied refining factors decided by the rule applied at that node.

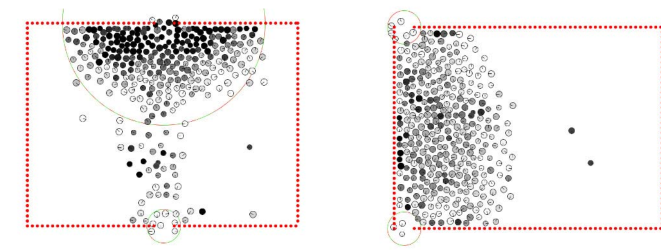
IV. RESULTS

We have conducted some simulation experiments based on our algorithms and describe the results here. We make sure to do all simulations using identical parameter settings for better comparison and evaluation. It is of course not possible to claim that these (or any other) simulation settings precisely



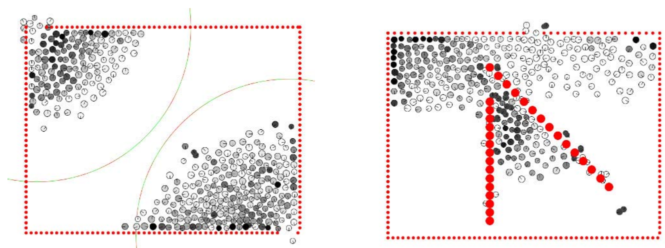
(a) (b)

Fig. 5. Experiment with one and two exit doors. (a) Case I—one exit door in the middle. (b) Case II—two exit door in the middle.



(a) (b)

Fig. 6. Experiment with varying visibility. (a) Case III—varying visibility. (b) Case IV—low-visibility exit doors at the corners.



(a) (b)

Fig. 7. Experiment with obstacle. (a) Case V—two exit doors at the corners. (b) Case VI—obstacle in environment.

describe all real scenarios, but wherever possible, we have calibrated the model close enough to real parameters as known from prior sources. The physical properties of an agent are taken from previous studies [7], [45], [46], whereas other factors have been calibrated individually to produce more realistic maneuvering of the agents for obstacle avoidance, collision avoidance, and so on. All the simulations serve the purpose well to achieve our aim of presenting a qualitative analysis of evacuation planning, marking out the bottlenecks, critical paths, and complete evacuation strategy evaluation and comparison.

Initial values of each property of agents in simulation are presented in Table II. We take a normal human with 0.5–0.8 m shoulder length as the physical diameter of the agent, and an average weight of 65 kg. The observed free velocities for leaving a room correspond to $\vec{v_i} \approx 0.6\, m/s$ under relaxed, $\vec{v_i} \approx 1\ m/s$ under normal, and $\vec{v_i} \approx 1.5 m/s$ under nervous conditions. Force calculation according to (10) in Section III-C, the force bound of 1600 N/m to declare an agent unable to move [7]. All these properties are scaled accordingly from meters to pixels for the simulation in MASON.

These kind of simulations can be very effective in pointing out the shortcomings of the floor plan of any building, and also indicate the locations and types of signage or other evacuation assistance needed to avoid obstacles and prevent bottlenecks.

A. Experiments Without Obstacles

In this section, we present results of our experiments on evacuation without any obstacles in the environment. We consider different arrangements of exit doors in a box environment with 400 agents. The dimension of the box is $26m \times 24m$ and the width of the door is 1 m, which is wide enough for an agent but difficult for two or more agents to pass through simultaneously. Large circles represent the visibility of a door. Absence of a circle near the door means the door has universal visibility, i.e., every agent has the knowledge of that exit door.

The panic level of each agent is shown in grayscale, where white being a least panicked and black being a most panicked agent. Black static agents are injured agents who cannot move after receiving a physical pressure above threshold 750 N; they further become obstacles for other evacuating agents. Figs. 5(a) and (b), 6(a) and (b), and 7(a) show these cases. In Cases II–V, we consider different permutations of two exit doors. We measured the performance of the different cases, which is presented in Figs. 8 and 9.

1) *Case I:* This is a basic case in which there is just one exit door for 400 agents. It turns out that just one door is not enough for these many agents. An arc-shaped crowd builds up near the exit door, and the time of evacuation was 4500 clock ticks, which is very high. The level of physical discomfort was very high among all the agents and for 16–20 agents crosses the injury threshold, and they become immobile (seriously injured).
2) *Case II:* There are two doors in the middle of two opposite walls with equal visibility. Approximately half of the crowd went to the first door and rest to the second door at the bottom. It is the most favorable, both in terms of reduced evacuation time and discomfort.
3) *Case III:* There are two doors in the middle of two opposite walls but one door has low visibility of 3 m. This situation arises when signage is poor or absent, and most people are unaware of an alternative exit door like a fire exit.
4) *Case IV:* There are two doors, both on the left side instead of in the middle of the wall, with low visibility. In this case, the knowledge of the environment among the agents is very low, which is of course not conducive to desirable outcomes.
5) *Case V:* There are two exit doors which are diagonally opposite with high visibility. It is different from Case II in door positioning. Though the crowd gets divided evenly between the doors, the observed evacuation time is higher than in Case II.

A comparative analysis of these cases is presented in Figs. 8 and 9.

The following may be noted: between Cases II and III, there is no much difference in evacuation times. In Case II, the visibility of the doors is very high, so agents are divided

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

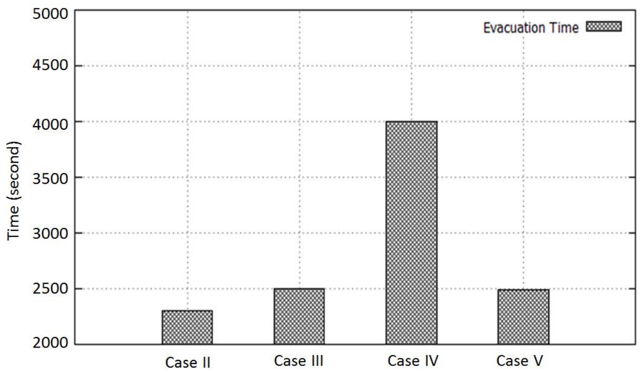


Fig. 8. Comparison of evacuation time of the cases with two exit doors.

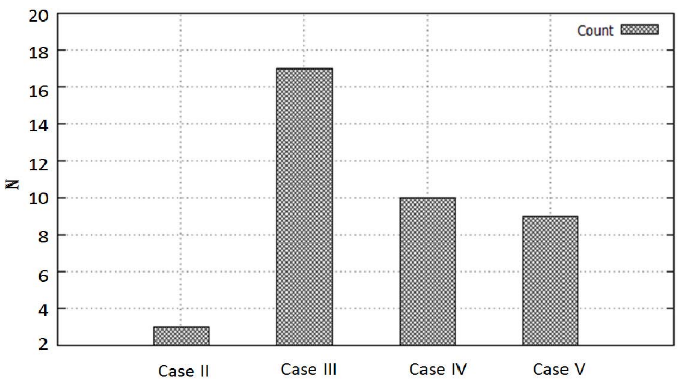


Fig. 9. Comparison of average count of injured agents during evacuation for all the cases with two exit doors.

evenly at both the doors, while in Case III one door (the lower one) has low visibility, so the agents do not get divided evenly, which increases the physical discomfort level of the agents at the upper door. The count of agents who experience pressure above the threshold (resulting in injury) is really high in Case III.

Case IV has a low evacuation rate due to the low visibility of both doors, but the physical discomfort count is less in comparison to Case III because the crowd gets divided evenly between the doors due to herding behavior.

There are good things to be said about both Cases II and V, but Case II proves to be the best.

B. Evacuation Given Obstacles

Case VI: This is similar to Case I, but with obstacles added. Unlike the previous cases where we had not considered the presence of obstacles in the environment, this shows how the presence of obstacles creates bottlenecks for evacuations and can completely change the locations, which are considered best for an emergency door or normal exit door.

So, there is a need to identify these locations and bottlenecks which can cause a lot of trouble at the time of emergency evacuation. We have a simple bottleneck obstacle in front of the exit door.

C. Lecture Hall Evacuation

In this scenario (see Fig. 10), based on the floor plan of a real-lecture hall known to the authors, there are 160 agents

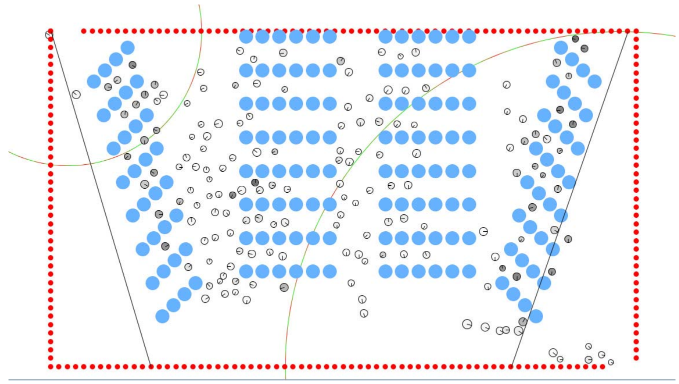


Fig. 10. Lecture hall evacuation.

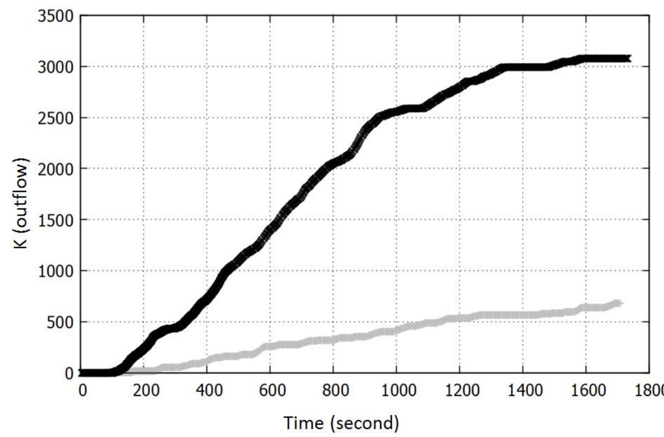


Fig. 11. Flow of lower door and upper door in the lecture hall case.

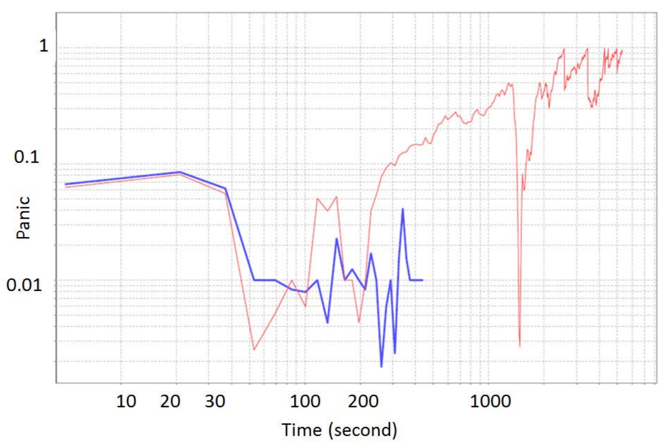


Fig. 12. Average panic of an agent near lower door and upper door.

(lecture attendees) and 160 seats, which seat the agents but also become obstacles during evacuation. There are two exit doors, one at the lower right corner and one at the upper left corner. Agents are more familiar with the lower door, which is regularly used, so its visibility is very high. The upper door leads to an uncommon area and is not used frequently, hence its visibility is low.

The framework of this simulation is explained in Section III-E. Figs. 3 and 4 explain the nodes we have identified. In this case, our rules were simple. All agents at nodes 2 and 4 move to their left or right randomly. Once the

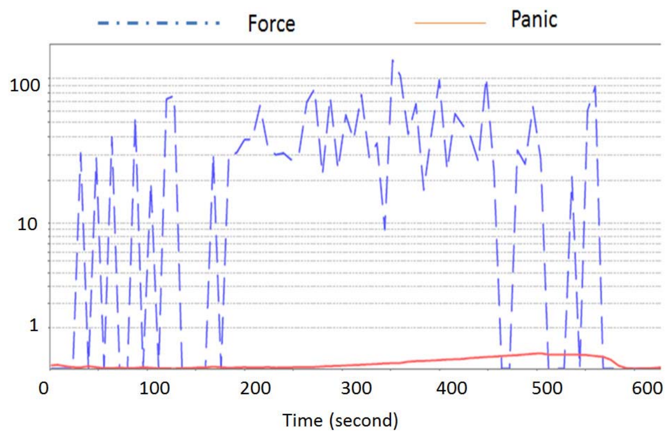


Fig. 13. Distribution of panic and physical force received by the agent. When agent get a sustained physical force its panic starts to increase.

agents reach nodes 5 and 7, they follow queuing behavior by adjusting their multipliers. At node 9 and 10, agents follow herding behavior to go to the nearest exit door.

Fig. 11 presents a graph which is proportional to the count of the agents coming out from both the doors. As we have noted that lower door has high visibility, so the count of agents coming out from it is higher than from the low-visibility upper door. Fig. 12 presents the comparison of average panic levels of the agents near the lower door and the upper door, which clearly shows that the upper door area has higher levels of panic during evacuation, so this result suggests that. This result is encouraging for our panic model. Fig. 13 also presents an interesting result which shows that if physical pressure is sustained at high value for some time the panic level of the agent starts increasing rapidly.

V. Conclusion

Our method of crowd path planning during emergency evacuation is seen to be successful at giving qualitative and some useful quantitative analyses. Qualitative analysis is also useful to understand the complex psychological and physical behavior of human, and the task of giving any quantitative analysis may or may not give any useful results and may not be trusted every time. We have successfully evaluated some very simple floor plans and how positions of obstacles can significantly change the environment in terms of emergency evacuation. Our results for the model of panic are encouraging and suggest the validity of the factors we have chosen to associate with panic on emergency evacuations.

We can suggest with confidence that positioning doors in the middle of a wall has an advantage in evacuation scenarios, compared with similar doors at the corners. From the lecture hall evacuation, we can conclude that the visibility of the doors play a huge role in ease of evacuation and sustained physical pressure increase the panic and increase the evacuation time even in simple scenarios. We have also given a general framework for testing various evacuation strategies, which is very difficult to analyze through mock drills.

Possible future work that could extend our model would be to incorporate more sophisticated human behaviors (e.g., clustering by families and groups of friends). Special scenarios requiring evacuations, such as fires, explosions, active shooters, and so on, can also be explored, as can evacuations in special settings, such as aircraft, ships, and large buildings and urban complexes.

References

[1] D. Elliott and D. Smith, "Football stadia disasters in the United Kingdom: Learning from tragedy?" *Org. Environ.*, vol. 7, no. 3, pp. 205–229, 1993.

[2] N. R. Johnson, "Panic at 'The Who concert stampede': An empirical assessment," *Social Problems*, vol. 34, no. 4, pp. 362–373, 1987.

[3] J. M. Chertkoff and R. H. Kushigian, *Don't Panic: The Psychology of Emergency Egress and Ingress*. Westport, CT, USA: Praeger, 1999.

[4] E. L. Quarantelli, "The nature and conditions of panic," *Amer. J. Sociol.*, vol. 60, no. 3, pp. 267–275, 1954.

[5] A. Mintz, "Non-adaptive group behavior," *J. Abnormal Social Psychol.*, vol. 46, no. 2, p. 150, 1951.

[6] D. Helbing, I. Farkas, and T. Vicsek, "Simulating dynamical features of escape panic," *Nature*, vol. 407, no. 6803, pp. 487–490, 2000.

[7] D. Helbing, I. J. Farkas, P. Molnar, and T. Vicsek, "Simulation of pedestrian crowds in normal and evacuation situations," *Pedestrian Evacuation Dyn.*, vol. 21, no. 2, pp. 21–58, 2002.

[8] D. Helbing, A. Johansson, and H. Z. Al-Abideen, "Dynamics of crowd disasters: An empirical study," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 75, no. 4, p. 046109, 2007.

[9] *Panic: A Quantitative Analysis*. Accessed: Jun. 14, 2014. [Online]. Available: http://angel.elte.hu/~panic/

[10] A. Langley, H. Mintzberg, P. Pitcher, E. Posada, and J. Saint-Macary, "Opening up decision making: The view from the black stool," *Org. Sci.*, vol. 6, no. 3, pp. 260–279, 1995.

[11] T. T. Pires, "An approach for modeling human cognitive behavior in evacuation models," *Fire Saf. J.*, vol. 40, no. 2, pp. 177–189, 2005.

[12] E. Kwon and S. Pitt, "Evaluation of emergency evacuation strategies for downtown event traffic using a dynamic network model," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 1922, no. 1, pp. 149–155, 2005.

[13] J. D. Sime, "Movement toward the familiar: Person and place affiliation in a fire entrapment setting," *Environ. Behav.*, vol. 17, no. 6, pp. 697–724, 1985.

[14] J. M. Watts, "Rescuing truth from familiarity," *Fire Technol.*, vol. 36, no. 1, pp. 1–2, 2000.

[15] G. Frank and C. Dorso, "Room evacuation in the presence of an obstacle," *Phys. A, Stat. Mech. Appl.*, vol. 390, no. 11, pp. 2135–2145, 2011.

[16] Q. Ji and C. Gao, "Simulating crowd evacuation with a leader-follower model," *Int. J. Comput. Sci. Eng. Syst.*, vol. 1, no. 4, pp. 249–252, 2007.

[17] L. Benthorn and H. Frantzich, "Fire alarm in a public building: How do people evaluate information and choose evacuation exit?" Dept. Fire Saf. Eng. Syst. Saf., Lund Univ., Lund, Sweden, Tech. Rep. LUTVDG/TVBB–3082–SE, 1996.

[18] N. Pelechano, K. O'Brien, B. Silverman, and N. Badler, "Crowd simulation incorporating agent psychological models, roles and communication," in *Proc. 1st Int. Workshop Crowd Simulation*, 2005, pp. 21–30.

[19] L. Zhang, J. Wang, Q. Shi, and T. Zang, "A new dynamic cellular automaton model for pedestrian evacuation process in stadium," in *Proc. 32nd Chin. Control Conf. (CCC)*, Xi'an, China, Jul. 2013, pp. 1181–1186.

[20] S. Picascia and N. Yorke-Smith, "Towards an agent-based simulation of housing in urban beirut," in *Agent Based Modelling of Urban Systems*. Cham, Switzerland: Springer, 2017, pp. 3–20. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-51957-9_1

[21] P. Perez, A. Banos, and C. Pettit, "Agent-based modelling for urban planning current limitations and future trends," in *Agent Based Modelling of Urban Systems*. Cham, Switzerland: Springer, 2017, pp. 60–69. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-51957-9_4

[22] S. Pluchino, C. Tribulato, A. Caverzan, A. M. Quillan, G. P. Cimellaro, and S. Mahin, "Agent-based model for pedestrians' evacuation after a blast integrated with a human behavior model," in *Proc. Struct. Congr.*, 2015, pp. 1506–1517.

[23] L. Crociani, G. Lämmel, and G. Vizzari, "Simulation-aided crowd management: A multi-scale model for an urban case study," in *Agent Based Modelling of Urban Systems*. Cham, Switzerland: Springer, 2017, pp. 151–171. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-51957-9_9

[24] A. Schadschneider, W. Klingsch, H. Klüpfel, T. Kretz, C. Rogsch, and A. Seyfried, "Evacuation dynamics: Empirical results, modeling and applications," in *Encyclopedia of Complexity and Systems Science*. Springer, 2009, pp. 3142–3176.

[25] D. R. Parisi and C. O. Dorso, "Microscopic dynamics of pedestrian evacuation," *Phys. A, Stat. Mech. Appl.*, vol. 354, pp. 606–618, Aug. 2005.

[26] L. Benthorn and H. Frantzich, "Managing evacuating people from facilities during a fire emergency," *Facilities*, vol. 17, nos. 9–10, pp. 325–330, 1999.

[27] S. T. Rassia and C. I. Siettos, "Escape dynamics in office buildings: Using molecular dynamics to quantify the impact of certain aspects of human behavior during emergency evacuation," *Environ. Model. Assessment*, vol. 15, no. 5, pp. 411–418, 2010.

[28] N. R. Johnson, "Panic and the breakdown of social order: Popular myth, social theory, empirical evidence," *Sociol. Focus*, vol. 20, no. 3, pp. 171–183, 1987.

[29] J. Drury, D. Novelli, and C. Stott, "Representing crowd behaviour in emergency planning guidance: 'Mass Panic'or collective resilience?" *Resilience*, vol. 1, no. 1, pp. 18–37, 2013, doi: 10.1080/21693293.2013.765740.

[30] R. F. Fahy, G. Proulx, and L. Aiman, "Panic or not in fire: Clarifying the misconception," *Fire Mater.*, vol. 36, nos. 5–6, pp. 328–338, 2012, doi: 10.1002/fam.1083.

[31] C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," *ACM SIGGRAPH Comput. Graph.*, vol. 21, no. 4, pp. 25–34, Jul. 1987.

[32] O. B. Bayazit, J.-M. Lien, and N. M. Amato, "Better group behaviors using rule-based roadmaps," in *Algorithmic Foundations of Robotics V*. Berlin, Germany: Springer, 2004, pp. 95–112.

[33] S. Luke, C. Cioffi-Revilla, L. Panait, K. Sullivan, and G. Balan, "MASON: A multiagent simulation environment," *Simulation*, vol. 81, no. 7, pp. 517–527, 2005.

[34] C. W. Reynolds, "Steering behaviors for autonomous characters," in *Proc. Game Developers Conf.*, 1999, pp. 763–782.

[35] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Trans. Syst. Sci. Cybern.*, vol. SSC-4, no. 2, pp. 100–107, Jul. 1968.

[36] Y. Yang, D. V. Dimarogonas, and X. Hu, "Optimal leader-follower control for crowd evacuation," in *Proc. 52nd IEEE Conf. Decision Control (CDC)*, Florence, Italy, Dec. 2013, pp. 2769–2774, doi: 10.1109/CDC.2013.6760302.

[37] A. C. Schultz, J. J. Grefenstette, and W. Adams, "Roboshepherd: Learning a complex behavior," in *Proc. Robot. Manuf., Recent Trends Res. Appl.*, vol. 6. 1996, pp. 763–768.

[38] R. Vaughan, N. Sumpter, J. Henderson, A. Frost, and S. Cameron, "Experiments in automatic flock control," *Robot. Auto. Syst.*, vol. 31, no. 1, pp. 109–117, 2000.

[39] E. Boukas, I. Kostavelis, A. Gasteratos, and G. C. Sirakoulis, "Robot guided crowd evacuation," *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 2, pp. 739–751, Apr. 2015, doi: 10.1109/TASE.2014.2323175.

[40] J. Funge, X. Tu, and D. Terzopoulos, "Cognitive modeling: Knowledge, reasoning and planning for intelligent characters," in *Proc. 26th Annu. Conf. Comput. Graph. Interact. Techn.*, 1999, pp. 29–38.

[41] J.-M. Lien, B. Bayazit, R. T. Sowell, S. Rodriguez, and N. M. Amato, "Shepherding behaviors," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, vol. 4. Apr. 2004, pp. 4159–4164.

[42] R. R. Murphy, "Human-robot interaction in rescue robotics," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 34, no. 2, pp. 138–153, May 2004.

[43] C. L. Bethel and R. R. Murphy, "Survey of non-facial/non-verbal affective expressions for appearance-constrained robots," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 38, no. 1, pp. 83–92, Jan. 2008.

[44] P. Robinette and A. M. Howard, "Incorporating a model of human panic behavior for robotic-based emergency evacuation," in *Proc. 20th IEEE Int. Symp. Robot Hum. Interact. Commun. (IEEE RO-MAN)*, Aug. 2011, pp. 47–52, doi: 10.1109/ROMAN.2011.6005265.

[45] V. Viswanathan, C. E. Lee, M. H. Lees, S. A. Cheong, and P. M. Sloot, "Quantitative comparison between crowd models for evacuation planning and evaluation," *Eur. Phys. J. B*, vol. 87, no. 2, pp. 1–11, 2014.

[46] V. Viswanathan and M. Lees, "An information processing based model of pre-evacuation behavior for agent based egress simulation," in *Pedestrian and Evacuation Dynamics 2012*. Cham, Switzerland: Springer, 2014, pp. 125–133.

[47] N. Shiwakoti, M. Sarvi, C. Dias, and M. Burd, "Understanding crowd panic at turning and intersection through model organisms," in *Pedestrian and Evacuation Dynamics 2012*. Springer, 2014, pp. 1175–1183.

[48] P. Sarshar, J. Radianti, and J. J. Gonzalez, "Modeling panic in ship fire evacuation using dynamic Bayesian network," in *Proc. 3rd Int. Conf. Innov. Comput. Technol. (INTECH)*, Aug. 2013, pp. 301–307.

[49] H. Vermuyten, J. Beliën, L. De Boeck, G. Reniers, and T. Wauters, "A review of optimisation models for pedestrian evacuation and design problems," *Saf. Sci.*, vol. 87, pp. 167–178, Aug. 2016.

[50] Y. Ma, E. W. M. Lee, and M. Shi, "Dual effects of guide-based guidance on pedestrian evacuation," *Phys. Lett. A*, vol. 381, no. 22, pp. 1837–1844, Jun. 2017.

[51] F. Li, S. Chen, X. Wang, and F. Feng, "Pedestrian evacuation modeling and simulation on metro platforms considering panic impacts," *Procedia-Social Behav. Sci.*, vol. 138, pp. 314–322, Jul. 2014.

[52] C. C. Holt, "Forecasting seasonals and trends by exponentially weighted moving averages," *Int. J. Forecasting*, vol. 20, no. 1, pp. 5–10, 2004.

[53] J. P. Keating, "The myth of panic," *Fire J.*, vol. 76, no. 3, pp. 57–61, 1982.

Ashutosh Trivedi received the M.Tech. degree in computer science from the International Institute of Information Technology Bangalore, Bengaluru, India, in 2015.

He is currently an Entrepreneur involved in building products with machine learning and artificial intelligence (AI). He is the Founder of Spext, an AI-based content conversion platform.

Shrisha Rao (M'08–SM'13) received the M.S. degree in logic and computation from Carnegie Mellon University, Pittsburgh, PA, USA, and the Ph.D. degree in computer science from The University of Iowa, Iowa City, IA, USA.

He is currently a Professor with the International Institute of Information Technology Bangalore, Bengaluru, India. His primary research interests are in artificial intelligence and other applications of distributed computing, including in bioinformatics and computational biology, algorithms and approaches for resource management in complex systems such as used in cloud computing. He also has interests in energy efficiency, sustainable computing ("Green IT"), renewable energy and microgrids, applied mathematics, and intelligent transportation systems.

Dr. Rao is a contributing member of the LITD 14 "Software and System Engineering" sectional committee (a national mirror committee of the ISO sub-committees IEC/JTC 1/SC 7 Software and System Engineering, and JTC 1/SC 38 Cloud Computing) of the Bureau of Indian Standards (BIS). He is also a member of the Intelligent Transport Systems Sectional Committee, TED 28, of the BIS.