

Hybrid agent-based methodology for testing response protocols

Enrique J. López-Ortiz^a, Fernando Sancho-Caparrini^a, Miguel Á. Martínez-del-Amor^a,
Luis M. Soria-Morillo^b, Juan A. Álvarez-García^{b,*}

^a Dept. Computer Science and Artificial Intelligence, Universidad de Sevilla, Avda Reina Mercedes S/N, 41012 Seville, Spain

^b Dept. Languages and Computer Systems, Universidad de Sevilla, Avda Reina Mercedes S/N, 41012 Seville, Spain

A B S T R A C T

In recent years we have seen multiple incidents with a large number of people injured and killed by one or more armed attackers. Since this type of violence is difficult to predict, detecting threats as early as possible allows to generate early warnings and reduce response time. In this context, any tool to check and compare different action protocols can be a further step in the direction of saving lives. Our proposal combines features from continuous and discrete models to obtain the best of both worlds in order to simulate large and crowded spaces where complex behavior individuals interact. With this proposal we aim to provide a tool for testing different security protocols under several emergency scenarios, where spaces, hazards, and population can be customized. Finally, we use a proof of concept implementation of this model to test specific security protocols under emergency situations for real spaces. Specifically, we test how providing some users of a university college with an app that informs about the type and characteristics of the ongoing hazard, affects in the safety performance.

Keywords:

Agent-based modeling
Simulation
Emergency protocols
Behavior modeling

1. Introduction

Unfortunately, in recent years we have seen multiple serious incidents with a large number of people injured and killed by one or more armed attackers. Although in many cases they are due to psychotic outbreaks, terrorism has been one of the causes that most concerns citizens, with attacks on shopping malls², newspaper offices³, nightclubs⁴, or streets⁵, among others. In all these cases, as in other risk situations, a key aspect of minimizing damage is reaction time, which is directly related to the time taken to detect the threat, and response plan. Although this type of violence is difficult to predict, it would be desirable to detect threats as early as possible and generate early appropriate warnings that reduce response time and, therefore, the number of victims, since it is known that the greatest number of victims occur during the first few minutes of these events. For this reason, any tool helping to check the efficiency of the different response protocols can be a further step in the direction of saving lives.

In mass settings such as shopping malls, open-air festivals, concerts or sports events, evacuation policies are proposed based on widely studied crowd dynamics [1]. Even so, events usually occur where forecasts and evacuation policies fail, and can only be studied in retrospect to generate future recommendations [2]. In the literature we can find evacuation management systems [3], mechanisms to control the number of people evacuated [4], models for understanding the influence of terrorist attacks on the population behavior [5], mathematical algorithms that generate an optimal evacuation route [6], or analysis of evacuees' behavior obtained from mobile sensors through machine learning methods such as SVM [7] or Artificial Neural Networks [8]. Other systems focus on supplying the exit route through the appropriate use of information provided by emergency management systems [9]. In some cases, they integrated into the users' own smartphone [10], allowing the information to be spread immediately around the affected persons.

Under conditions of very little real data to apply data analysis techniques, computational approaches for this purpose require procedures to simulate both the environment in which the hazard occurs and the behavior of the individuals in it. Although the problem of simulating an environment under an emergency situation has been previously addressed by different researchers using a variety of techniques, the use of multi-agent systems is by far the most common approach.

When using agent-based models to study this problem, there are two main aspects that will determine how the simulation will be developed. The first one is about the environment modeling,

* Corresponding author.

E-mail addresses: enrloport@alum.us.es (E.J. López-Ortiz), fsancho@us.es (F. Sancho-Caparrini), mdelamor@us.es (M.Á. Martínez-del-Amor), lsoria@us.es (L.M. Soria-Morillo), jaalvarez@us.es (J.A. Álvarez-García).

which usually falls under one of two options, discrete or continuous representation. Both offer pros and cons, and choosing one of them will depend on our goals and requirements. The second one is about modeling the decision making process of the human-like agents, where the use of Belief-Desire-Intention (BDI) paradigm [11] is at a clear advantage over other options.

Our proposal is inspired by continuous (Social Forces, Helbing and Molnár [12]) and discrete (Cellular Automata, Bandini et al. [13]) modeling to simulate large and crowded spaces through statistical agent-based models. With this proposal we aim to provide a tool for testing different security protocols under varying conditions. For this, the tool must keep track of the several aspects of the simulation, specifically the casualties that have occurred, the agents who have suffered an accident due to the pressure of other agents and those who have managed to reach a safe room or exit the hazard space. The goal is to provide a virtual environment where to test different security protocols under several emergency scenarios, where spaces, hazards, population, etc. can be customized with less effort.

We present a proof-of-concept implementation of this tool to test some specific security protocols under emergency situations for real spaces. Specifically, we test how providing (some) individuals of a space with an app that informs about the type and features of the hazard in progress, affects the safety performance. The developed model was implemented in Netlogo (version 6.x) and published at OpenABM in [14] under license GPL3. It can also be downloaded from the open repository <https://github.com/fsancho/Response-Emergency-Protocol>.

The paper is structured as follows. Section 2 is devoted to introduce some motivation and related works, where continuous and discrete environment modeling is presented and their pros and cons are emphasized, some notes about human behavior modeling is also introduced at the end of the section. Our proposal is presented in Section 3, detailing the environment modeling approach and how the BDI paradigm has been applied to model decision making processes in human-like agents. In Section 4, we present the experimental setup and the results obtained by the simulations to test the influence of different world configurations and security protocols to the safety performance. Finally, Section 5 shows some conclusions and future work that can be derived from our work.

2. Motivation and related work

Over the past few decades, many researchers have addressed the evacuation of buildings and crowded places [15–20]. Models implemented in these works, although having their own characteristics, have in common some techniques for modeling agent behavior and their interaction. An agent-based model can be generally subdivided into two main parts: one for defining the environment where the agents operate, and another for defining the decisions made by the agents.

Taking into account the nature of space where the agents act and the time where they evolve, the world can be essentially modeled in two different ways: either discretely or continuously.

Generally speaking, in discrete models, both space and time are considered discrete. The space is usually divided in cells that can contain a single agent (or an obstacle) in each instant of time. In this context, a unit of time is usually set to be the time an average agent needs in order to move between adjacent cells. Since these models are inspired by multi-state cellular automata, they are usually called Discrete CA models.

In order to reflect more realistic scenarios, it is common to extend agents' behavior to avoid collisions while choosing specific routes (minimal distance, for example) while moving through the environment. Because of this, when simulating crowded/big/

complex spaces these models require a lot of computing power and complex states/rules, so they are usually designed to study just specific aspects of human or crowd behavior, or in very specific spaces (classrooms, open spaces, theaters, etc.). For example, Bandini et al. [13] uses a discrete CA model to study how groups of agents behave under different simple scenarios defined through three key elements: environment, pedestrians and rules.

Group membership is a key concept in the model, and pedestrians belonging to the same group have similar objectives. The obtained results were very similar to those from real situations, but only limited to these very simple situations.

Closer to our goal, in [21] the authors design a discrete CA model to evaluate building plans in evacuation scenarios. The effectiveness of the different designs were measured by the average distance traveled by the agents and the total evacuation time, and their speed of movement can be affected by physical and mental condition (panic state). Moreover, some experiments are carried out through a NetLogo model [22] to evaluate single room features (like the number of exits or the presence of information about escape route).

As [23] points out when describing the key role of accidents between agents, the way a discrete model divides the space is a central problem. On the one hand, Dadová and Ferko [24] proposes a more relaxed space division to provide freedom of movement to agents, where cells size is 4 to 5 times larger than the necessary for an agent. On the other hand, Pluchino et al. [25] studies local dynamics of individual decision making and behavior considering geometry, randomness, social preferences, and local and collective behavior of other individuals.

Although this type of discrete models is the most widely used when simulating evacuation events, there are other discrete techniques to address this problem, like *gas models* [26], *potential field models* [27], *electrostatic induction models* [28], or *potential cost models* [29]. It is noteworthy that all of them require higher computational cost, since in order to calculate the movement of the agent, multiple environmental factors must be taken into account, as well as the directions chosen by the neighboring agents.

In addition, these models enable parallel computing easily, such as those based on GPU computing [30]. However, there are a number of issues that need to be addressed: how to express forces in a confined space with high-density crowds?, or how to describe the non-local interactions or non-adaptive actions (falls, injuries, disability or surprise)? One of the goals of our approach is to apply features learned from continuous models to discrete models in order to fill these gaps.

On the contrary, in continuous models space and time are not made up of discrete units: space is usually an open subset of $\mathbb{R}^2/\mathbb{R}^3$, and time is a real interval (although it is sometimes discrete as an approximation for effective calculation).

Among them, *Social Force Models* are widely spread for crowd simulation [12], where it is assumed that each agent is subject to a set of forces acting on her, not only a force is trying to move her in a desired direction. We can consider forces to avoid obstacles or walls, and also compression forces and sliding friction forces, of great importance during emergency evacuations. In [31] the author uses these forces to compute evacuation times from simple areas (rooms). As we will see later, in our model we can use a similar approach to obtain some statistical features of single regions in our spaces (average flows, density, etc.). In other works also motivational or psychological forces are considered [32], some examples are: self-driving force (self-induced and responsible for the acceleration of the agent towards a desired position), repulsive psychological force (the desire of agents to maintain a certain distance from other agents and obstacles), and viscosity force (avoids agents from reaching unrealistic speeds).

From a computational point of view, continuous models in general (and those based on social forces, in particular) have higher requirements than discrete models, and may be inappropriate when looking for some near real time testing. Therefore, since our goal is to create a tool for experimenting and validating evacuation protocols, and the degree of uncertainty that derives from any simulation of human behavior (and, even more, under emergency situations) is very high, our contribution tries to balance the best of both modeling approaches by means of a methodology that extracts some parameters from social forces approaches describing the characteristics of the spaces under certain conditions (like in [33] or [34]), and later executes a discrete model that uses those characteristics to evaluate the dynamics of the event in those conditions.

We also want to emphasize *Model Validation* as a main problem to handle (in discrete and continuous models): qualitative validation aims to reproduce classic collective effects and self-organizing phenomena that can be observed in real events; while quantitative validation compares simulation results with data from real events. The problem is that there is a lack of real data about these kind of events, and only very simple phenomena are known or only simple evacuation exercises are available, see [35]. Consequently, the validation of our model is based on the goodness of the behavior expected from the point of view of security experts.

The way social relationships are structured in human groups is a very important factor in modeling group behavior under emergency scenarios. Following [36], three interconnected levels of interaction can be considered: the individual, the interactions between individuals, and the group.

From the point of view of the individual, decision-making under emergency can be summarized in three factors: instinct [37], experience, and bounded rationality [38].

From the perspective of interactions between individuals, individual behavior is often shaped by the social structures where she is immersed. The most important factors in this sense are: social identity [38], personal space [39], and social proof [40].

Finally, by analyzing the group as an entity, we can identify some factors contributing to the crowd behavior: density [41], environmental restrictions [42] and emotions and perceived tensions [43].

The goal when modeling human behavior through multi-agent systems is to restrict rules and interactions to the level of individuals, expecting the emergence of group rules from the joint action of all the individual rules. Nevertheless, the fact is that individual rules must be helped with some explicit group rules. In our approach, trying to provide a more computationally efficient model, some of these group rules are designed through aggregated values (density occupancy of areas, capacity flow of doors and corridors, presence of leaders to follow, etc.).

It is not easy to establish the appropriate parameters for social forces and, in particular, how the sense of hazard in an emergency affects the individuals. In [44], some relevant concepts for our case of study were introduced to better adjust this relation: in a panic situation, agents behave in a more instinctive way and need more time to carry out a certain action plan; in an emergency situation, the location of the hazard should be the most influential factor in the behavior of an agent, even more than obstacles or other surrounding individuals; and some individuals move faster in an emergency situation. Under these considerations, some experimental tests were conducted by simulating real situations with a few existing recordings, showing better validation than more classical social force models.

The most common tools to model individual behavior follow the BDI paradigm [11], which is reliant on folk-psychological view of reasoning (from an observable point of view, and not the actual

mechanism of the brain). Initially proposed as a theoretical model of thought, it was soon used for computational modeling purposes (by the author himself). In BDI paradigm, decision making is divided into three phases [45]:

1. *Belief*: The information that the agent receives from the world, including herself and inference rules that lead to new beliefs. It is not knowledge, but the uncertain belief of the agent for his environment (often modeled through Fuzzy Logic).
2. *Desires*: Objectives or situations that the agent wishes to achieve (often modeled through Bayesian Networks).
3. *Intentions*: Represent what the agent has chosen to do (desires that the agent has begun to carry out).

An example of a BDI model for buildings evacuation using a CA discrete model can be found in [46], where the goal is to analyze the effect of information about exits in evacuation time. Agents have 3 basic behavioral modes: follow the signs avoiding the dangers until reaching an exit; dismiss the signs and going directly to a known exit; or explore the environment when forced to abandon its previous behavior. Results showed that the evacuation time was reduced under available information.

The link between BDI and Fuzzy Logic can be seen in works like [47], where a model of traffic congestion uses fuzzy rules to represent the relationship between events and how attractive these events are to the agent. Or [48], where Genetic Algorithms are used to evaluate the fitness of agents that combine BDI and Fuzzy Logic for decision-making in specific environments. Our proposal makes use of some similar tools to improve the use of individual rules.

Li et al. [18] presents a decision making system with the aim of enabling agents to evacuate dangerous scenarios in the best possible way. When an attack occurs, a series of signals are produced that help individuals to evaluate hazards and to directly influence the decisions they make. The information perceived by an agent may not be conclusive and therefore she may not have an accurate idea of the hazard event. Through Bayesian Belief Networks and by using data from the Global Terrorism Database (GTD), the model allows to differentiate between seven types of attacks and build the associated probability network, from where the agent makes an estimation of probability for different types of events and decides whether to evacuate or not in each of the possible situations. The results were very similar to those observed in videos extracted from real attacks in some specific events. Our approach uses a similar decision making system for agents to evaluate the type of hazard they perceive.

Other works can also be found using BDI to model human groups under hazard events: Park et al. [49] uses a system based on swarm intelligence for the optimization of survivors in rescue tasks against terrorist actions, and Goyal [50] introduces collective attitudes to model the behavior of a team in a fire hazard.

3. Proposed model

Since our goal is to simulate the global behavior of individuals under a changing information scenario, we will not add functionalities detailing the way agents interact with each other and with the environment. This does not mean that the model does not consider them, but that it will not take care of its formal derivation (for example, calculating the specific location of each agent within a room, as continuous models of social forces would do). For this purpose, the model will make use of aggregated values obtained from the simulation of auxiliary social forces models or from plan buildings/experts, for example, to compute the flow of agents that can move between two connected rooms/areas, the

maximum speed that can be reached in a passage of particular conditions, or the capacity of sound transmission through a wall [16] and [20].

In any case, our aim is not to provide a model capable of calculating the effective values involved in an evacuation process under emergency situations, but one that can efficiently use these values to simulate the behavior of a heterogeneous group of individuals under changing conditions. Its applicability to a specific scenario will require an intensive analysis of its characteristics.

As we do not need to mimic the exact local behavior of agents, we choose to represent the environment through a graph (discrete model) where nodes represent specific areas (rooms, passages, corridors, room areas, etc.), and the edges provides information about connection between areas (flow capacity, sound transmission, visibility, etc.). *Agents* in our model represent the individuals (peaceful and attackers) involved in the simulation of the event. They have a belief about the world and an internal state according to these believes.

Broadly speaking, the interactions that can occur between the elements of the model are (note that this is a cycle of interactions that repeat forever):

1. Agent actions produce signals in the node they occupy.
2. Signals are transmitted through the graph taking into account the dynamical underlying structure.
3. Agents receive information/signals from the graph and build their believes.
4. Agents make decisions and modify internal state and desire using their believes.
5. Agents perform actions (as a result of the intentions to reach their desires).
6. Agents change properties of the graph.

In each simulation step all these interactions take place, and cause changes in the graph, both at the level of nodes (e.g., the room occupation) and edges (e.g. modifying the connectivity by blocking a room in order to increase their security level), and at the level of agents (e.g. their position or state). Moreover, following the BDI model, agents can take decisions that will affect the flow of the simulation according to: the information that they perceive in the form of signals from the nodes and the intrinsic information that these nodes contain in their properties; their own internal state; and the information and interaction that other agents around them provide (see Fig. 1).

In order to address the inherent complexity of these scenarios, the model offers a wide variety of properties in nodes, edges and agents, and also a set of functions that allow the properties of these elements to be modified (from the intensity an acoustic signal is perceived in a node, to the goals of an agent in the process of pursuit or escape during a violent attack).

In the following subsections we will give a slightly more detailed view of some issues related to the design of the parts we consider more interesting in the model, and that have been implemented as proof of concept through NetLogo in order to be able to perform the experiments presented in the next section.

3.1. Environment

As noted in the previous paragraphs, nodes represent rooms/areas, but more than one node can be used for large rooms/areas. It is up to the modeler to choose a more or less detailed representation of the real space (of course, the global structure of the graph will be different as well as the properties in nodes and edges). Fig. 2 shows an example of how a small environment can be represented as a graph. This space has five small rooms of $3 \times 3 \text{ m}^2$ (Rooms 1 to 5), two large rooms of $6 \times 7 \text{ m}^2$ (Rooms 6 and 7) and a couple of corridors that connect these spaces. In

a standard CA representation, cells of $0.5 \times 0.5 \text{ m}^2$ will result in a graph with more than 750 nodes and around 2500 transitable edges, but in our model we can choose, for example, 14 nodes and 15 transitable edges for the same environment.

Using a discrete representation with a reduced number of nodes simplifies several tasks necessary for the simulation, like path-finding and visibility. But, to get closer to a realistic behavior when testing new spaces, we can use a continuous approach to model the local dynamics of agents in and between the nodes or through expert advisers (architects and security officers). These values will be used as physical parameters for nodes and edges.

Main parameters for nodes and edges are shown in Tables 1 and 2, respectively. They are categorized into three types: *physical*, they depend on the intrinsic features/design of the area (usually, they will not change over time); *state*, they reflect the current state of the area (they can change over time); and *signal*, if they are related to signal transmission/perception (they can also be *state* or *physical*).

In order to reduce computational costs, some parameters are stored in the graph, in this way agents can read/write information directly from the graph structure instead of duplicating computations that can be similar for all the agents in the same area. For example, nodes have several parameters to control and store the intensity of signals that can be perceived in them, visual or acoustic (*fire*, *fire-sound*, *attacker*, etc.). As we will see in the next subsection, agents can access these parameters to know/update the state of the world in every instant time, and combine with their intrinsic state to obtain a customized experience of the world. Also, aggregated parameters from continuous dynamics can be found here. For example, the movement of agents between nodes is managed by the edges through *flow* and *max-flow* parameters.

By modifying the parameters of the graph we can model very different scenarios, even with changing conditions over time (e.g. where visibility changes in certain areas due to the appearance of smoke, or some corridors or rooms are blocked). This ability to model and use very high dynamical scenarios is one of the goals and main guides for the kind of approach we decided, motivating to combine the good computational features of discrete models with realistic behavior modeling of social force models through statistical aggregation, something that can be hardly obtained by using one only approach.

3.2. Agents and decision making

The second core of the model is focused in agents modeling. Agents can move around the world and change their location. In order to simplify the model, agents' location will be reduced to know the graph node/edge where she is in every instant of time as a rough estimate of their position in the world.

In the events simulated in experiments, there are two main types of agents (more types can be easily added): *peaceful*, and *attacker*, with a clear meaning about their goals and actions in the world. Table 3 shows the main parameters of the agents and a description of their meaning and functionality in the model.

Almost all the functionality of the model is located in the action of the agents, that goes through three well differentiated phases that aligned with the BDI paradigm (see Fig. 1):

1. **Perception.** In this phase, agents update the information they obtain from the environment by means of perceptions. With these perceptions and her internal state of mind (fear, panic-level, sensibility, ...), the agent constructs her *beliefs* of the world. Perception is handled under the following assumptions in the model: (a) events occur in the world in

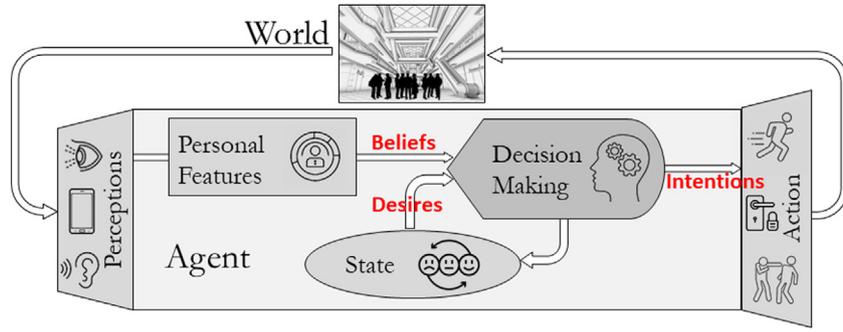


Fig. 1. Life cycle of an agent.

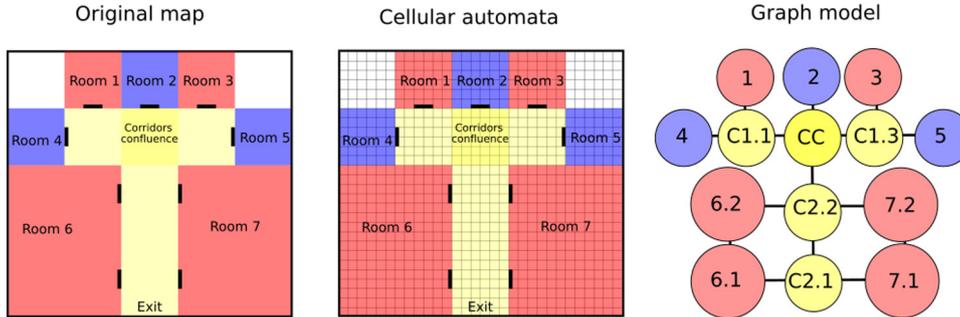


Fig. 2. Representation of a small building plan (left) as a cellular automata grid (center) and a graph (right), where a room/area can be represented by more than one node, according to its capacity and characteristics.

Table 1
List of main parameters for nodes of space graph.

Parameter	Type	Description
<i>capacity</i>	Physical	Maximum number of agents that can be in the node/area
<i>hidden-places</i>	Physical	Number of places in the node where an agent can successfully hide
<i>lockable?</i>	Physical	Can the node be locked?
<i>info?</i>	Physical	Are there signs in the node indicating an exit?
<i>residents</i>	State	Current number of agents in the node
<i>hidden-people</i>	State	Number of people hidden in the node
<i>nearest-danger</i>	State	Node with nearest danger
<i>leaders</i>	State	Number of leaders currently in the node
<i>fire/fire-sound</i>	Signal/State	Intensity of visual/acoustic signals derived from fires
<i>attacker/attacker-sound</i>	Signal/State	Intensity of visual/acoustic signals derived from attackers
<i>bomb/bomb-sound</i>	Signal/State	Intensity of visual/acoustic signals derived from explosions
<i>scream</i>	Signal/State	Intensity of acoustic signals derived from screams
<i>running-people</i>	Signal/State	Intensity of visual signals derived from people on the run
<i>corpses</i>	Signal/State	Intensity of visual signals derived from corpses in the floor

Table 2
List of main parameters for edges of space graph.

Parameter	Type	Description
<i>length</i>	Physical	Distance between areas connected by the edge
<i>lockable?</i>	Physical	Is the edge lockable?
<i>max-flow</i>	Physical	Maximum number of agents that can use simultaneously the edge
<i>locked?</i>	State	Is the edge locked?
<i>flow</i>	State	Number of agents currently using the edge
<i>sound</i>	Signal/Physical	Sound transmission intensity across the edge
<i>transitable</i>	Signal/Physical	Modifier on the speed of the agents that pass through it
<i>visibility</i>	Signal/Physical	Intensity of visual information transmitted by the edge

an objective way, and (b) agents have a subjective perception of the events (this individual subjectivity is managed through an agent-specific parameter: *sensitivity*). Specifically, in our implementation some of these stimuli are modeled through parameters of the nodes/edges in the graph. At the beginning of each iteration all agents must update the stimuli received since, as we will see, new

inputs may cause a change of their desires and actions that reflect their intentions. The perceptible characteristics of the nodes provoke an accumulation of stimuli in the agents that can eventually trigger changes in their beliefs about the world. In each iteration the agents read the “objective facts” of the scenario by querying the node where they are, and then, they update their accumulators. Table 3 shows

Table 3
List of parameters for agents.

Parameter	Agent type	Param type	Description
<i>location</i>	All	State	Current node of location
<i>bad-area</i>	All	State	Nodes where the agent identifies any hazard
<i>route</i>	All	State	Planned route to reach a destination
<i>state</i>	All	State	State of the agent
<i>speed</i>	Attacker	Physical	Speed of attacker
<i>effectiveness</i>	Attacker	Physical	Attacker effectiveness
<i>detected?</i>	Attacker	Physical	Whether the agent has been detected as a threat or not
<i>max/base-speed</i>	Peaceful	Physical	Maximum/Base speed that can be reached by the agent
<i>sensibility</i>	Peaceful	Physical	Intensity with which the stimuli affect the agent
<i>fear</i>	Peaceful	State	Level of fear (accumulator)
<i>leadership</i>	Peaceful	Physical	Leadership level of an agent
<i>hidden?</i>	Peaceful	State	Whether the agent is hidden or not
<i>app?</i>	Peaceful	Physical	Whether the agent has installed the app or not
<i>attacker-heard/sighted</i>	Peaceful	Signal	Accumulators to measure perceived signals
<i>fire-heard/sighted</i>	Peaceful	Signal	
<i>bomb-heard/sighted</i>	Peaceful	Signal	
<i>scream-heard</i>	Peaceful	Signal	
<i>corpse-sighted</i>	Peaceful	Signal	
<i>running-people</i>	Peaceful	Signal	
<i>police/leader-sighted</i>	Peaceful	Signal	
<i>perceived-risk</i>	Peaceful	State	Risk to which the agent believes she is exposed
<i>panic-level</i>	Peaceful	State	Level of panic
<i>in-panic?</i>	Peaceful	State	Whether the agent has exceeded his capacity for risk stimuli or not

the different accumulators that the agents currently have for this purpose, for every of these accumulators there is one equivalent parameter in the nodes in Table 1. The grouping in the table is done by purpose, not by type. The model uses a small fuzzy logic engine to process the internal state of the agent and helps her to form an individual belief about what is happening in the world. Table 4 shows some samples of these fuzzy rules. In this way, all the agents inside one specific node will perceive the same environmental information in a different way, providing heterogeneity, flexibility and realism to the model. Also, in this way the model can be easily adapted to different conditions by non-expert users.

- 2. Decision Making.** In this phase, agents build a sequence of decisions mainly aimed at achieving their goal of escaping from hazard. As represented in the agents' life cycle (Fig. 1), each agent builds the next (*desire, intention*) pair from the (*belief, desire*) pair. Thus, it will determine the next behavior and action from her beliefs, as well as the next desire (hence working as an internal state of the agent). This decision/behavior making process is performed through a set of rules (see an extract of these rules in Table 5) that can provoke a change in her current desire (see Fig. 3 for the graph of available desires and their relationship through the decision making process) as well as an intention that translates into an action to be taken (processed in the next phase). The application of the rules applied in this process are prioritized by the urgency of each of the possible situations in which each agent may be. After receiving new stimuli, agents evaluate a *perceived-risk* which, if exceeded a threshold, causes the agent to leave the not alerted state (to which it will never return during the simulation). *Not alerted* and *Evacuate* are the two only desires that act as initial and terminal states with no predefined intentions in our model. When the agent is in *not alerted* state, she will continue with the default behavior in the world (keep working, walking, studying, etc.), and when she is in *evacuate* state, she will leave the BDI loop (also if the agent is killed/injured by the attacker). The application of these rules will determine the agent's desires, which will be transformed into intentions in the

Table 4
Samples of fuzzy rules to process Agents' Beliefs.

Premises	Consequents
<i>fear</i> is high and <i>sensibility</i> is very high	<i>panic-level</i> is very high
<i>fear</i> is low	<i>panic-level</i> is low
<i>fear</i> is high and <i>sensibility</i> is high	<i>panic-level</i> is high
<i>perceived-risk</i> is low and <i>distance-attacker</i> is far	<i>perceived-risk</i> is low
<i>perceived-risk</i> is high and <i>distance-attacker</i> is close	<i>perceived-risk</i> is high

next phase. For example, if an agent become aware of the presence of an attacker, she will probably get into the *avoid_violent* desire (since the condition of this rule is fulfilled) and will enter the next phase with this desire. However, the final behavior of the agent is affected by the particularities of the situation (for example, it is not the same situation when both attacker and agent are in the same room as when they are at opposite ends of a corridor). Consequently, although there may be several agents that reach the next phase with the same particular state, their final actions will be conditioned by their individual situation.

- 3. Performing Actions.** Finally, depending on the desire of the agent, different *intentions* will be produced and different actions can be taken. The model take into account the current state of the world (like the distance to the hazard) and the desire of the agent, and also a hierarchy of rules that prioritize their applicability based on the urgency of the situation. For example, under *follow-leader* desire, if the agent comes across an attacker, she will prioritize escaping from the attacker. The intentions of the agents are transformed in a set of actions in the world that can produce changes in the environment and in the future perceptions of the agents, closing the cycle of life of the agents.

A set of additional reports and functions are needed to perform this decision making and several other actions, and also to update the information stored in agents and graph elements. These reports and functions have been named self-descriptive in order to facilitate the adaptation of the same model to different types of events and agents, and to ease the task of building concrete simulation scenarios in a simple and high-level way. Some of them are

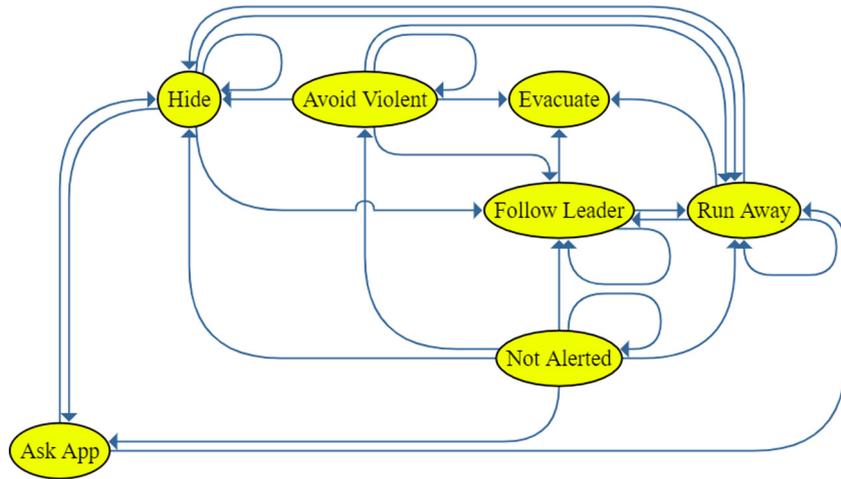


Fig. 3. Diagram of Peaceful Agents' Desires in the BDI model and their relationship through the decision making process. We do not detail here the different actions and conditions that can change a desire into a different one along the simulation.

Table 5

Some hierarchical rules for decision making, only from Beliefs to Desires. Rules from (*belief, desire*) pair to the next (*desire, intention*) pair are not here represented for simplification. Beliefs are calculated from the perceptions of the agent, here simplified in natural language.

Priority	Beliefs (the agent...)	Desire
1	Perceived a leader	Follow leader
2	Reached an exit	Evacuate
3	Reached a secure room or a place to hide	Hide
4	Believe that there is an attacker near	Avoid violent
5	Alerted with the app	Ask App
6	Try to escape with no location info	Run away
7	Not enough stimuli to alert	Not alerted

intimately related to the intentions that agents' decision making can produce to satisfy their desires. For peaceful agents we have: *avoid-crowd, go-to-exit, to-wait, to-stay, better-location?, avoid-violent, In-the-same-area?, best-visible, follow-leader, etc.* For attacker agents we have: *be-aggressive, find-target, violent-advance, attacker-next-location, walk, attack-target, attack, shoot, etc.*

Here we will only get into details for a couple of functions that will be of interest in our experimental stage: *Any-leader-near?* and *Is-the-app-sending-info?*. When the simulation starts and the agents are positioned in the map, maybe some of them do not have any global information about the world. This is controlled through the *leader* role of the agents. A leader can be seen as an authority figure (a teacher in a school, the director of an office, a security agent in a mall, etc.) or a person that, by her way of acting, seems to know the building very well and take decisions under hazard situations. Agents that do not know the building will follow a leader (if *any-leader-near?* is true). They will continue processing new stimuli but their intentions will mainly consist in following the leader [51]. Like any other agents, leaders can make wrong decisions under uncertainty (choose a route that takes them closer to an attacker, for example).

Finally, *app-pack?* allows to simulate the transmission of useful information about what is happening in the world by a supposed security application that some agents can have installed on their smartphones. This way, we can launch simulations and study the differences in scenarios where no information is given to the agents compared to those where it is. This app will provide information about routes to exits or rooms where the agents can achieve safety. We assume this app gathers real-time information about the attack from the infrastructure (e.g. sensors, cameras, location-based applications, etc.) such as in [52].

4. Experimentation and analysis

A map inspired by one of the pavilions of a high school has been designed to carry out the experiments. This pavilion has an approximate size of $20 \times 50 \text{ m}^2$ (see Fig. 4a), containing 6 classrooms (with sizes between $3 \times 6 \text{ m}^2$ and $6 \times 7 \text{ m}^2$), 26 offices (with sizes between $2 \times 2 \text{ m}^2$ and $3 \times 3 \text{ m}^2$) and 6 seminar rooms (of about $4 \times 5 \text{ m}^2$). These rooms are connected by several corridors of about 3 m wide, while the width of the doors is about 0.8m. Some rooms are marked as exits (in red) and some others as safe (in green), that agents can lock down from inside to be safe from danger until law enforcement arrives. In some experiments we will change the safe rooms in order to test some different scenarios.

Once the division of areas have been decided, we obtain a graph that provides the discrete representation of the space that our model use (see Fig. 4b). By using as reference values used in similar spaces [16], or obtained as mean values in experiments with social forces models [32], the *max-flow* value for the doors was set up to 2.5, and 20 for corridors.

One of the objectives of this work is to provide a tool for comparing results in building evacuation under changes in the configuration of the population and in the type of information available to the agents about an event that is happening (armed attackers in this case). We will use the number of agents who manage to reach an exit, those who manage to reach a safe room, those who have been eliminated by the attackers and those who have suffered a fatal accident due to the pressure of the crowd when trying to escape, as measures to be compared. These accidents are roughly obtained from the density of the node (agents/m^2) and the speed of the agents. In each test we have launched 1000 simulations for each of the possible parameter configurations and present the statistics from them.

In the following subsections we present the several experiments we have carried out to test the influence of the different parameters in the measures obtained from the evacuation processes.

As we will see, we can conclude that the app becomes a good tool in almost any circumstance. However, it will not be the only decisive factor, and visibility and sound transmission conditions in the environment will be a crucial factor, as well as the number of exits and safe rooms in the map.

Additionally, as concluded also by other studies, factors such as the width of the corridors and doorways, as well as the distribution of the rooms, will influence the results obtained. Indeed, these factors directly influence the flow of agents. In our

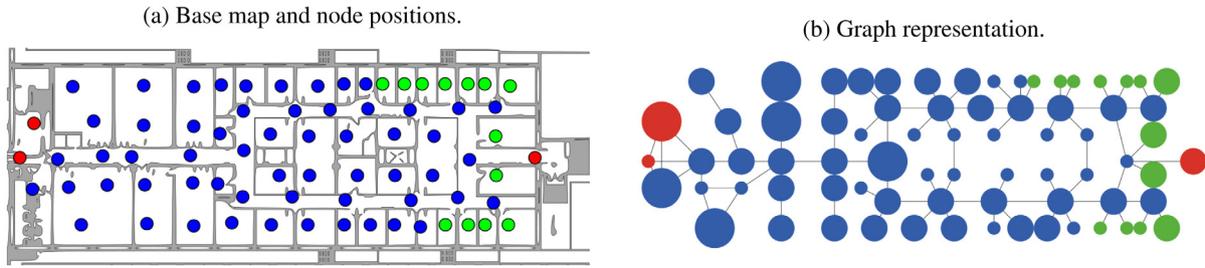


Fig. 4. Example pavilion designed for the experiments. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 6
Configuration of parameters for experiment 1.

Violent parameters		Peaceful parameters		App parameters	
num-violent	2	num-peaceful	From 150 to 600	Agents with installed app	50%
shooting?	On	leaders-percentage	20%		
shoot-noise	0.5	defense-probability	10%		
attack-probability	80%	not-alert-speed	0.5 m/s	App-mode	The App will recommend the nearest option (secure-room or exit)
attack-effectiveness	50%	mean-speed	2.00 m/s		
speed	0.5 m/s	max-speed-deviation	0.15		
initial-positions	Exit nodes				

case, these values have remained unchanged in all the simulations. Also, in all the experiments we will consider two attackers entering from two of the exits.

4.1. Experiment 1: Evacuation process by density of peaceful agents

In this experiment, we are going to study how the evacuation process is affected by gradually increasing the number of agents. The parameter configuration is described in Table 6. Fig. 5 shows the average results using a discontinuous line while the vertical bars indicate the standard deviation for each one of the values in the several simulations performed. In all configurations, we compare the results when the app is used (blue) and when it is not (red). In general terms, the application becomes a good tool that improves the results, it reduces the number of casualties, both those caused by the attackers and those caused by accidents. However, in the latter, there is very little difference between the results, and they seem to depend more on the number of agents present in the world than on whether they have information or not.

4.2. Experiment 2: Noisy and dark environments

In the second experiment we study the effect of a very noisy and dark environment (this can be simulated in our model by applying to *visibility* and *sound* parameters a modifier with value close to 0). In this environment the agents can only be aware of what is happening in their own node and they will not perceive the possible danger signals that may occur in neighboring nodes. The rest of parameters remain the same as Experiment 1 (Table 6).

Under these circumstances, the app becomes a fundamental tool for improving results. Fig. 6 shows that not only are the casualties caused by the attackers reduced (which have almost tripled with respect to the previous experiment), but this time there is also a decrease in the number of accidents. This happens because, when the app gives information to the agents, they will try to reach safe exits or rooms, thus decongesting the nodes where they were initially. When it is not installed, this mobilization does not take place. Moreover, these nodes may become saturated when they are reached by agents who are fleeing from danger, generating bottlenecks and increasing the probability of accidents.

There is also a huge difference in the agents that manage to reach safe rooms. This is understandable if we take into account the conditions of the simulation. In a noisy and dark environment, the agent is only aware of what is happening in her own node. This causes that an agent in a safe room will only perceive the attacker when she is inside the same safe room, and the agent will try to get away from the attacker by leaving the room. This does not happen when the application provides information about the attack and therefore the agents do manage to reach secure rooms.

4.3. Experiment 3: Leaders' knowledge

In our third experiment, only the leaders know the building and where the exits are. The other agents do not have that knowledge. The question to answer here is: how useful is the app when we increase the number of leaders? The number of agents will be fixed to 350, and we change the percentage of leaders from 0% to 100%. The rest of parameters remain the same as Experiment 1 (Table 6).

As shown in Fig. 7 as the number of leaders increases, the results in terms of rescued and safe rooms are coming closer. Interestingly, the tipping point of the killed plot seems to be around 30% of leaders.

When there are no leaders in the simulation, each agent is going to take its own decisions and move freely through the scenario. However, given that the agents do not know the map configuration, they are exploring nodes randomly until finding a safe room or an exit. When increasing the percentage of leaders, groups of agents that move in the same direction start to generate flows. These groups of agents can have different objectives and some groups hinder others in their displacement and generate jams, which entails an increase of accidents (anyway, the number of accidents is smaller than in previous experiments) and a slowing down of the advance. As a consequence, the attackers have more possibilities of causing casualties. As the figure shows, providing information to agents (also leaders) remains a good option and greatly reduces the number of casualties.

4.4. Experiment 4: Speed of the attacker

This experiment is designed to show how the speed of the attacker influences the results of the simulation (from 0.5 to

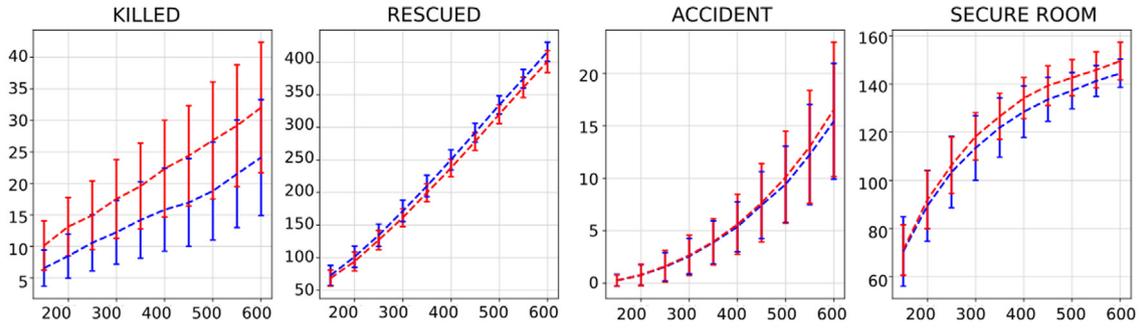


Fig. 5. Experiment 1. Average and deviation of number of agents (y-axis) for different states (Killed, Rescued, Accident, and in Secure Room), when varying the number of peaceful agents (x-axis). Blue - App on, Red - App off. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

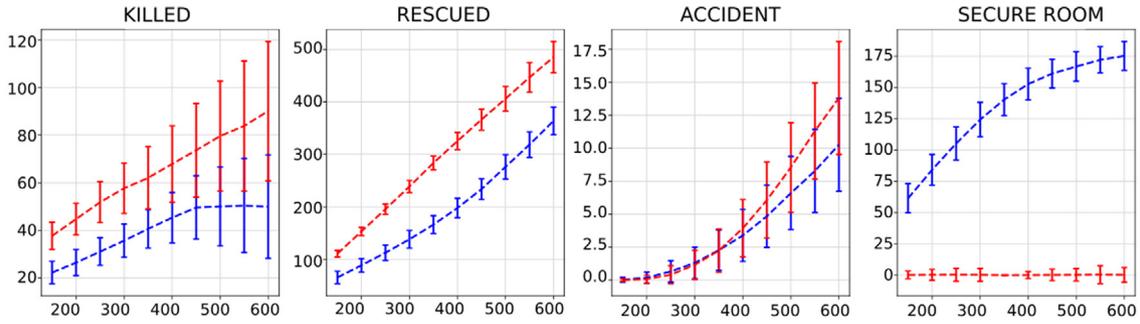


Fig. 6. Experiment 2. Average and deviation of number of agents (y-axis) for different states (Killed, Rescued, Accident, and in Secure Room), when varying the number of peaceful agents (x-axis) in a dark and noisy scenario. Blue - App on, Red - App off. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

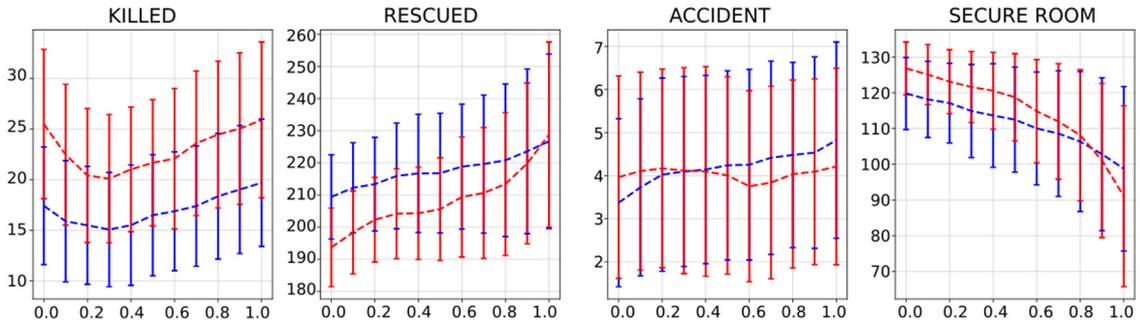


Fig. 7. Experiment 3. Average and deviation of number of agents (y-axis) for different states (Killed, Rescued, Accident, and in Secure Room), when varying the number of leaders (x-axis) in a dark and noisy scenario. Blue - App on, Red - App off. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

1.5 m/s). In this case, the attackers are going to make only melee attacks, since when they have the option to attack from the distance, the speed at which they move is not particularly relevant. The rest of parameters remain the same as Experiment 1 (Table 6).

As shown in Fig. 8, there is an increase in casualties as the attacker's speed increases. Moving faster allows them to reach a greater number of targets throughout the simulation. Improvements can still be seen when the app has informed the agents of the danger, although the results in terms of accidents seem to have slightly worsened. However, the accidents that occur will be influenced mainly by the number of agents in the simulation, the percentage of leaders and also by the configuration of the map.

4.5. Experiment 5: Safe rooms number

Previous experiments have shown that the app works as a good tool under almost any circumstance, but especially in scenarios where the environment is noisy and dark (Experiment 2).

In that situation, there was an improvement in both the number of casualties and accidents and rescues, but above all, in the number of agents who manage to reach safe rooms. The next question can be raised: how can these rooms be improved? What results can we expect in a map that has many safe rooms compared to the same map without safe rooms? Our last experiment will tackle these questions.

To this end, and similarly to Experiment 2, a modifier is applied to the edges to prevent them from transmitting visibility and sound. Additionally, the original safe rooms are removed from the map and, instead, all rooms at the top and the bottom of the map become into safe rooms. The rest of parameters remain the same as Experiment 1 (Table 6).

Fig. 9 shows that, under these circumstances, the results when using the app are really good, where the average of casualties is much lower than those when the app is not in use. Having many safe rooms reduces the distance of the agents' movements, thus reducing the number of accidents and increasing the number of

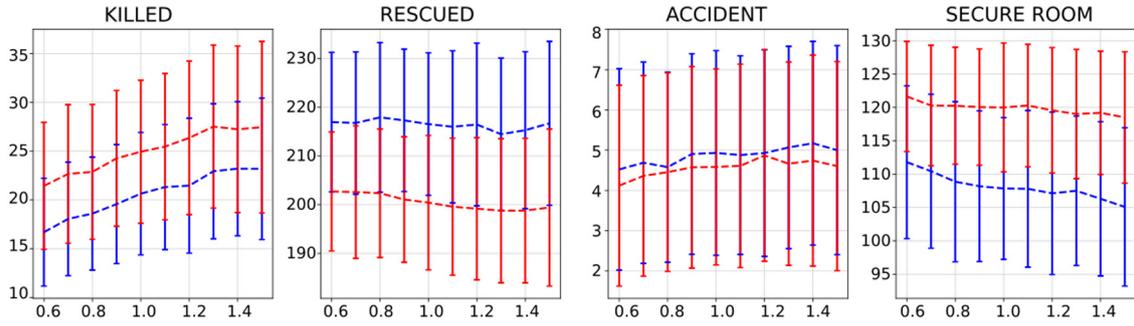


Fig. 8. Experiment 4. Average and deviation of number of agents (y-axis) for different states (Killed, Rescued, Accident, and in Secure Room), when varying the attackers' speed (x-axis) in a dark and noisy scenario. Blue - App on, Red - App off. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

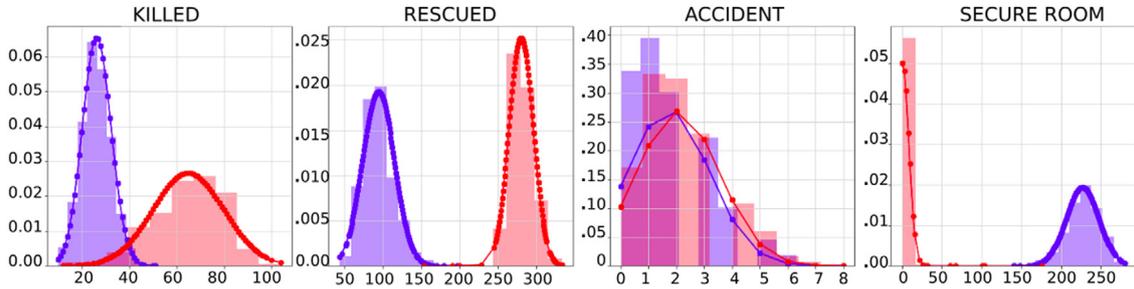


Fig. 9. Experiment 5. Distribution of number of agents for different states (Killed, Rescued, Accident, and in Secure Room) with different secure rooms. Blue - App on, Red - App off. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

agents who manage to reach a safe room. In the figure, y-axis shows the percentage of experiments reaching the number of individuals in the x-axis for the different cases.

5. Conclusions

The main objective of this work is to provide a tool that allows the modeling and simulation of hazardous situations in large spaces, such as work areas, malls, theaters, concert areas, educational centers, etc. To this end, features that are specific to continuous multi-agent models have been transferred to a discrete model as aggregated parameters. This has been key to be able to check and compare different conditions both in the physical attributes of the space and in the population configurations that occupy the space, or the different types of hazard to which the occupants of the space may be subjected. As a use case, we have analyzed the influence that a mobile information app could have on different cases that can occur under attack situations in a work/educational center. In this sense, we have been able to check how the simulations could help to improve the protocols of action in these special conditions.

The followed process to obtain a viable model has consisted of several phases. Firstly, it has been necessary to build a discrete space that must maintain a balance between the need for spatial description detail and the reduction of complexity required for its computational manipulation. To do this, and by using ideas from the CA discrete models, a virtual space is built in the form of a graph that represents the spatial units necessary for a complete description of the real space considered. Through this network, the model can transmit information about the various events that occur, as well as reflect physical conditions that affect the behavior of individuals who occupy the space (such as the capacity of the rooms and corridors, transitivity of doors and corridors, visibility of areas, pressure exerted in areas determined by the presence of crowds, etc.).

By using the appropriate parameters in nodes, edges and agents, the model enables us to simulate very diverse situations

with relative efficiency. We can also simulate dark or noisy environments by (dynamically) adjusting the values of parameters such as visibility and sound of the edges, or both at the same time, what is very common at music festivals.

Secondly, when modeling situations where agents must reflect the real behavior of human beings, the key point is to use an appropriate behavioral model that takes into account the complexity of this problem. In this work we get inspiration from the BDI paradigm, adapting the behavior of the agents, mainly, from the signals that they are able to perceive from their environment and the personal objectives and features that each one of them has. Through fuzzy logic, each agent processes the signals that she receives to obtain a personal estimate of the risk that believes she is exposed to. Her risk assessment (that changes at every moment of the simulation), her personal characteristics, the other agents around her, and her knowledge of the environment will generate individual behaviors that will influence the decision making process to solve the situation she face, emulating the behavior of real individuals under similar situations. The discrete subdivision of the decision-making process makes it easier to extend. Consequently, for example, the repertoire of behaviors of the agents can be expanded, given that certain signals influence the agents in a particular way, contributing certain nuances to the behavior that they adopt. By expanding the ways in which signals influence the agent, creating new states and actions, we can introduce more varied behaviors, providing greater diversity in the actions of agents.

In addition, the concept of role has been introduced within the agents, so that certain agents have a specific behavior that fits the pattern we want to simulate (attackers, peaceful, leaders, policemen, firemen, room managers, etc.). They can be defined/added to the model at will.

Regarding the experiments with the developed tool, they have allowed us to validate, for example, that in noisy spaces with poor visibility (where the occupants are not able realize that an attack is taking place since a clear perception of the surroundings cannot

be obtained), the fact that the agents have a warning app becomes a critical factor. This application warns them about the event taking place and the possible escape routes. The results showed that the application considerably modifies the final obtained results, determining the success or failure of the evacuation of the site. In general, regardless of the environmental conditions, it is possible to minimize the evacuation time of the scenario when the percentage of users using the application is high. This results in a decrease in the number of casualties and an increase in the number of evacuees. Therefore, the information offered through the app becomes a good tool to improve the evacuation of a space.

The experiments have also showed that few agents using the application are enough in order to improve the results. This is due to the fact that the users of the app are quickly warned about hazards, which causes them (in most cases) to flee and alerts other agents, who also begin to flee. This reaction chain reduces the evacuation time and improves the number of evacuees and casualties. Consequently, we can conclude that providing information to agents through an app is a very effective way of minimizing damage in an emergency situation.

As a continuation of this work, we plan to add BDI logic features to the model [53]. They will allow for more complex decision making while maintaining the bounded rationality of the individuals. We will also expand the battery of experimental tests required to adjust the behaviors to the variety of cases that we can find in real-life scenarios under similar circumstances. Since aggregated values for parameters from social forces simulations or real data is a necessary step to save higher computational demands while maintaining realistic behavior in the discrete model, we still need a big amount of continuous simulated environments in order to cover a sufficiently wide range of situations. In this sense, we are also working on alternative approaches to pre-calculate the aggregated parameters by approximating social forces results through Deep Learning models. Moreover, we are also working on efficient simulation software to speedup the real-time visualization and statistics calculation by using compiled languages and parallel architectures.

CRedit authorship contribution statement

Enrique J. López-Ortiz: Software, Validation, Data curation, Writing - original draft. **Fernando Sancho-Caparrini:** Conceptualization, Formal analysis, Methodology, Software, Writing - original draft, Writing - review & editing, Supervision. **Miguel Á. Martínez-del-Amor:** Writing - original draft, Writing - review & editing. **Luis M. Soria-Morillo:** Writing - review & editing. **Juan A. Álvarez-García:** Conceptualization, Funding acquisition, Project administration, Writing - review & editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research is partially supported by The Spanish Ministry of Economy and Competitiveness MINECO/FEDER R&D, UE through the project VICTORY (grant no. : TIN2017-82113-C2-1-R).

References

- [1] A. Schadschneider, W. Klingsch, H. Klüpfel, T. Kretz, C. Rogsch, A. Seyfried, Evacuation dynamics: Empirical results, modeling and applications, 2008, arXiv preprint [arXiv:0802.1620](https://arxiv.org/abs/0802.1620).
- [2] D. Helbing, P. Mukerji, Crowd disasters as systemic failures: analysis of the love parade disaster, *EPJ Data Sci.* 1 (1) (2012) 7.
- [3] A.M. Ibrahim, I. Venkat, K. Subramanian, A.T. Khader, P.D. Wilde, Intelligent evacuation management systems: A review, *ACM Trans. Intell. Syst. Technol. (TIST)* 7 (3) (2016) 1–27.
- [4] N. Ahmed, A. Ghose, A.K. Agrawal, C. Bhaumik, V. Chandel, A. Kumar, Smartevactrak: A people counting and coarse-level localization solution for efficient evacuation of large buildings, in: 2015 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops), IEEE, 2015, pp. 372–377.
- [5] I. Moya, M. Chica, J.L. Saez-Lozano, O. Cordon, An agent-based model for understanding the influence of the 11-m terrorist attacks on the 2004 spanish elections, *Knowl.-Based Syst.* 123 (2017) 200–216.
- [6] G. Mishra, S. Mazumdar, A. Pal, Improved algorithms for the evacuation route planning problem, in: *Combinatorial Optimization and Applications*, Springer, 2015, pp. 3–19.
- [7] Y. Hayakawa, K. Mori, Y. Ishida, K. Tsudaka, T. Wada, H. Okada, K. Ohtsuki, Development of emergency rescue evacuation support system in panic-type disasters, in: 2012 IEEE Consumer Communications and Networking Conference (CCNC), IEEE, 2012, pp. 52–53.
- [8] Y. Ortakci, I.R. Karas, U. Atila, E. Demiral, Intelligent mobile indoor navigation system for fire evacuation based on artificial neural network, *Int. J. Comput. Sci. Inf. Secur.* 14 (11) (2016) 980.
- [9] E. Gelenbe, F.-J. Wu, Future research on cyber-physical emergency management systems, *Future Internet* 5 (3) (2013) 336–354.
- [10] T. Wada, T. Takahashi, Evacuation guidance system using everyday use smartphones, in: 2013 International Conference on Signal-Image Technology & Internet-Based Systems, IEEE, 2013, pp. 860–864.
- [11] M. Bratman, et al., *Intention, Plans, and Practical Reason*, Vol. 10, Harvard University Press Cambridge, MA, 1987.
- [12] D. Helbing, P. Molnár, Social force model for pedestrian dynamics, *Phys. Rev. E* 51 (1995) 4282–4286, [http://dx.doi.org/10.1103/PhysRevE.51.4282](https://doi.org/10.1103/PhysRevE.51.4282).
- [13] S. Bandini, F. Rubagotti, G. Vizzari, K. Shimura, A cellular automata based model for pedestrian and group dynamics: Motivations and first experiments, in: V. Malyszhkin (Ed.), *Parallel Computing Technologies*, in: *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011, pp. 125–139, [http://dx.doi.org/10.1007/978-3-642-23178-0_11](https://doi.org/10.1007/978-3-642-23178-0_11).
- [14] F. Sancho-Caparrini, Hybrid agent-based methodology for testing response protocols, 2021, <https://www.comses.net/codebases/0478941a-af4f-49c5-b08f-f21956c0a58e/releases/1.0.0/> (Online; CoMSES Computational Model Library; accessed 03-February-2021).
- [15] D. Helbing, I. Farkas, T. Vicsek, Simulating dynamical features of escape panic, *Nature* 407 (2000) 487–490, [http://dx.doi.org/10.1038/35035023](https://doi.org/10.1038/35035023).
- [16] Q. Xie, S. Lu, D. Kong, J. Wang, The effect of uncertain parameters on evacuation time in commercial buildings, *J. Fire Sci.* 30 (1) (2012) 55–67, [http://dx.doi.org/10.1177/0734904111424258](https://doi.org/10.1177/0734904111424258), arXiv:<https://doi.org/10.1177/0734904111424258>.
- [17] H. Wang, A. Mostafizi, L.A. Cramer, D. Cox, H. Park, An agent-based model of a multimodal near-field tsunami evacuation: Decision-making and life safety, *Transp. Res. C* 64 (2016) 86–100, [http://dx.doi.org/10.1016/j.trc.2015.11.010](https://doi.org/10.1016/j.trc.2015.11.010).
- [18] S. Li, J. Zhuang, S. Shen, A three-stage evacuation decision-making and behavior model for the onset of an attack, *Transp. Res. C* 79 (2017) 119–135, [http://dx.doi.org/10.1016/j.trc.2017.03.008](https://doi.org/10.1016/j.trc.2017.03.008).
- [19] F. Enríquez, L.M. Soria, J.A. Álvarez-García, F. Sancho-Caparrini, F. Velasco, O. Deniz, N. Vallez, Vision and crowdsensing technology for an optimal response in physical-security, in: J.M.F. Rodrigues, P.J.S. Cardoso, J. Monteiro, R. Lam, V.V. Krzhizhanovskaya, M.H. Lees, J.J. Dongarra, P.M. Slood (Eds.), *Computational Science – ICCS 2019*, in: *Lecture Notes in Computer Science*, Springer International Publishing, Cham, 2019, pp. 15–26, [http://dx.doi.org/10.1007/978-3-030-22750-0_2](https://doi.org/10.1007/978-3-030-22750-0_2).
- [20] G.-Y. Wu, M. Mizuno, The numerical analysis of mass evacuation in taipei 101 with control volume model, *Simul. Model. Pract. Theory* 96 (2019) 101937, [http://dx.doi.org/10.1016/j.simp.2019.101937](https://doi.org/10.1016/j.simp.2019.101937), URL: <https://www.sciencedirect.com/science/article/pii/S1569190X19300620>.
- [21] R. Liu, D. Jiang, L. Shi, Agent-based simulation of alternative classroom evacuation scenarios, *Front. Archit. Res.* 5 (1) (2016) 111–125, [http://dx.doi.org/10.1016/j.foar.2015.12.002](https://doi.org/10.1016/j.foar.2015.12.002).
- [22] S. Tisue, U. Wilensky, Netlogo: A simple environment for modeling complexity, in: *International Conference on Complex Systems*, Vol. 21, Boston, MA, 2004, pp. 16–21.
- [23] Y. Li, M. Chen, Z. Dou, X. Zheng, Y. Cheng, A. Mebarki, A review of cellular automata models for crowd evacuation, *Physica A* 526 (2019) 120752, [http://dx.doi.org/10.1016/j.physa.2019.03.117](https://doi.org/10.1016/j.physa.2019.03.117).

- [24] J.B. Dadová, A. Ferko, *Doctoral Dissertation: Cellular Automata and Particle Systems for Crowd Simulation in Selected Environments* (Ph.D. thesis), Univerzity Komenského Bratislava, 2014.
- [25] A. Pluchino, C. Garofalo, G. Inturri, A. Rapisarda, M. Ignaccolo, Agent-based simulation of pedestrian behaviour in closed spaces: A museum case study, *J. Artif. Soc. Soc. Simul.* 17 (1) (2014) 16, <http://dx.doi.org/10.18564/jasss.2336>, URL: <http://jasss.soc.surrey.ac.uk/17/1/16.html>.
- [26] M. Muramatsu, T. Irie, T. Nagatani, Jamming transition in pedestrian counter flow, *Physica A* 267 (3) (1999) 487–498, [http://dx.doi.org/10.1016/S0378-4371\(99\)00018-7](http://dx.doi.org/10.1016/S0378-4371(99)00018-7).
- [27] R.-Y. Guo, H.-J. Huang, Route choice in pedestrian evacuation: formulated using a potential field, *J. Stat. Mech. Theory Exp.* 2011 (04) (2011) P04018, <http://dx.doi.org/10.1088/1742-5468/2011/04/p04018>.
- [28] I. Georgoudas, P. Kyriakos, G. Sirakoulis, I. Andreadis, An FPGA implemented cellular automaton crowd evacuation model inspired by the electrostatic-induced potential fields, *Microprocess. Microsyst.* 34 (7) (2010) 285–300, <http://dx.doi.org/10.1016/j.micpro.2010.06.001>.
- [29] P. Zhang, X.-X. Jian, S.C. Wong, K. Choi, Potential field cellular automata model for pedestrian flow, *Phys. Rev. E* 85 (2012) 021119, <http://dx.doi.org/10.1103/PhysRevE.85.021119>.
- [30] I. Koumis, I.G. Georgoudas, G.A. Trunfio, J. Ws, G.C. Sirakoulis, A GPU implemented 3f cellular automata-based model for a 2D evacuation simulation pattern, in: 2017 25th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP), 2017, pp. 497–504, <http://dx.doi.org/10.1109/PDP.2017.93>.
- [31] V.Q. Ha, Master Thesis: Agent-Based Modeling of Emergency Building Evacuation (Ph.D. thesis), University of Connecticut, 2014.
- [32] T. Korhonen, S. Hostikka, Technical Reference and User's Guide for Fire Dynamics Simulator with Evacuation, FDS+ Evac, (FDS 5.5. 0, Evac 2.2. 1), VTT Technical Research Center of Finland, 2010, May.
- [33] X. Yang, H. Dong, Q. Wang, Y. Chen, X. Hu, Guided crowd dynamics via modified social force model, *Physica A* 411 (2014) 63–73, <http://dx.doi.org/10.1016/j.physa.2014.05.068>, URL: <http://www.sciencedirect.com/science/article/pii/S0378437114004580>.
- [34] B. Liu, H. Liu, H. Zhang, X. Qin, A social force evacuation model driven by video data, *Simul. Model. Pract. Theory* 84 (2018) 190–203.
- [35] H. Klüpfel, T. Meyer-König, M. Schreckenberg, Microscopic modelling of pedestrian motion – comparison of simulation results with an evacuation exercise in a primary school, in: *Proceedings of Traffic and Granular Flow, 2004*.
- [36] X. Pan, C.S. Han, K. Dauber, K.H. Law, A multi-agent based framework for the simulation of human and social behaviors during emergency evacuations, *Ai Soc.* 22 (2) (2007) 113–132.
- [37] R.H. Wills, *Human Instincts, Everyday Life, and the Brain: A Paradigm for Understanding Behavior*, The Book Emporium, 1998.
- [38] J. March, *Primer on Decision Making: How Decisions Happen*, Free Press, 1994, URL: <https://books.google.es/books?id=cFagB2qj2jwC>.
- [39] R. Sommer, *Personal Space: The Behavioral Basis of Design*, in: *Spectrum book*, Bosko Books, 2007, URL: <https://books.google.es/books?id=VMrSPQAACAAJ>.
- [40] R. Cialdini, *Influence: The Psychology of Persuasion*, in: *Collins Business Essentials*, HarperCollins e-books, 2009, URL: <https://books.google.es/books?id=5dfv0HJ1TEoC>.
- [41] S. SFPE, *Engineering Guide to Human Behavior in Fire*, SFPE Gaithersburg, MD, US, 2002.
- [42] T. Shields, G. Proulx, The science of human behaviour: past research endeavours, current developments and fashioning a research agenda, *Fire Saf. Sci.* 6 (2000) 95–113.
- [43] J.M. Chertkoff, R.H. Kushigian, *Don't Panic: The Psychology of Emergency Egress and Ingress*, Praeger Publishers, 1999.
- [44] S. Li, J. Zhuang, S. Shen, J. Wang, Driving-forces model on individual behavior in scenarios considering moving threat agents, *Physica A* 481 (2017) 127–140, <http://dx.doi.org/10.1016/j.physa.2017.04.011>.
- [45] C. Adam, B. Gaudou, *Bdi agents in social simulations: a survey*, *Knowl. Eng. Rev.* 31 (3) (2016) 207–238.
- [46] P.C. Tissera, A. Castro, A.M. Printista, E. Luque, Evacuation simulation supporting high level behaviour-based agents, *Procedia Comput. Sci.* 18 (2013) 1495–1504, <http://dx.doi.org/10.1016/j.procs.2013.05.317>, 2013 International Conference on Computational Science.
- [47] H. Jones, J. Saunier, D. Lourdeaux, Fuzzy rules for events perception and emotions in an agent architecture, in: *Proceedings of the 7th Conference of the European Society for Fuzzy Logic and Technology*, Atlantis Press, 2011/08, pp. 657–664, <http://dx.doi.org/10.2991/eusflat.2011.34>.
- [48] T.V. Arredondo, Fuzzy motivations in behavior based agents, in: *Smart Information and Knowledge Management*, Springer, 2010, pp. 247–272.
- [49] A.J. Park, H.H. Tsang, M. Sun, U. Glässer, An agent-based model and computational framework for counter-terrorism and public safety based on swarm intelligence a, *Secur. Inform.* 1 (1) (2012) 23.
- [50] M. Goyal, Attitude based teams in a hostile dynamic world, *Knowl.-Based Syst.* 18 (6) (2005) 245–255.
- [51] J. Patrix, A.-I. Mouaddib, S. Gatepaille, Detection of primitive collective behaviours in a crowd panic simulation based on multi-agent approach, *Int. J. Swarm Intell. Res.* 3 (3) (2012) 50–65, <http://dx.doi.org/10.4018/jsir.2012070104>.
- [52] J.L. Salazar, L.M. Soria-Morillo, J.A. Álvarez-García, F. Enríquez, A.R. Jiménez, Energy-efficient indoor localization wifi-fingerprint system: An experimental study, *IEEE Access* 7 (2019) 162664–162682.
- [53] A. Herzig, E. Lorini, L. Perrussel, Z. Xiao, BDI Logics for BDI architectures: old problems, new perspectives, *KI-Künstliche Intell.* 31 (1) (2017) 73–83.

MSc. Enrique López-Ortiz is a researcher at the Department of Computer Science and Artificial Intelligence at the University of Seville. His research focuses in Multi-Agent Systems and Artificial Intelligence.

Dr. Fernando Sancho-Caparrini is an Associate Professor at the Department of Computer Science and Artificial Intelligence at the University of Seville. His research focuses in complex systems through ABM and AI techniques. He holds a Ph.D. degree in Mathematics in the same university.

Dr. Miguel Ángel Martínez-del-Amor is an Assistant Professor at the Department of Computer Science and Artificial Intelligence at the University of Seville. His research focuses in the interplay between parallel computing, artificial intelligence and bio-inspired computing. He holds a Ph.D. degree in Computer Science and AI from the University of Seville.

Dr. Luis Miguel Soria-Morillo is a Lecturer and Researcher at the Department of Languages and Computer Systems at the University of Seville. His research focuses in activity recognition, human computer interaction and healthcare solutions. He holds a Ph.D. degree in Software Engineering from the University of Seville.

Dr. Juan Antonio Álvarez-García is an Associate Professor at the Department of Languages and Computer Systems at the University of Seville. His research focuses in computer vision, deep learning and human activity recognition. He holds a Ph.D. degree in Software Engineering in the same university.