



# ACoPla: a Multiagent Simulator to Study Individual Strategies in Dynamic Situations

Ana Cristina Bicharra Garcia<sup>a</sup>  
and Adriana Santarosa Vivacqua<sup>b</sup>

<sup>a</sup>Applied Informatics Department at the Federal University of the State of Rio de Janeiro, in Brazil.  
<sup>b</sup>Universidade Federal do Rio de Janeiro. [cristina.bicharra@uniriotec.br](mailto:cristina.bicharra@uniriotec.br), [avivacqua@dcc.ufrj.br](mailto:avivacqua@dcc.ufrj.br)

## KEYWORD

Agents, emergency scenarios, evacuation, Multiagent systems, social simulation, BDI, agent strategies.

## ABSTRACT

*One important issue in multi-agent systems is how to define agents' interaction strategies in dynamic open environments. Generally, agents' behaviors, such as being cooperative/altruistic or competitive/adversarial, are defined a priori by their creators. However, this is a weak premise when considering interaction among anonymous self-interested agents. Whenever agents meet, there is always a decision to be made: what is the best group interaction strategy? We argue that the answer depends on the amount of information required to make a decision and on the deadline proximity for accomplishing the task in hand. In certain situations, it is to the agents' advantage to exchange information with others, while in other situations there are no incentives for them to spend time doing so. Understanding effective behaviors according to the decision-making scenario is still an open issue in multi-agent systems. In this paper, we present a multi-agent simulator (ACoPla) to understand the correlations between agents' interaction strategy, decision-making context and successful task accomplishment rate. Additionally, we develop a case study in the domain of site evacuation to exemplify our findings. Through this study, we detect the types of conditions under which cooperation becomes the preferred strategy, as the environment changes.*

## 1. Introduction

Multiagent systems (MAS) [6] are well suited technologies for simulating social interactions, as they enable the construction of entities with individual characteristics that inhabit an environment [8]. This makes the study of multiple strategies towards goal achievement possible. With multiple agents, partial planning can be undertaken and agents can exchange information. Additionally, agents can adapt to changes in the environment, altering their plans as necessary.

In MAS settings, agents are social entities, capable of interacting with each other and exchanging information to reach common goals [4]. The MAS setting opens up a new range of possibilities for studying complex social systems, as agents are capable to independently acquire complete or partial environment information and decide to share their world assessment to create a more trustworthy mapping of the world. However, agents may



or may not be inclined to do so, not by design, but because of possible conflicts between group interests. Emergency situations are interesting scenarios for which group cooperativeness as well as survival competitiveness may play an extreme role leading to extreme outcomes of life or death.

To conduct our studies, we developed a framework for simulating emergency scenarios, called ACoPla. ACoPla is a multi-agent environment implemented using JADE that allows us to configure the environment and create agents with different social strategies. Communication among agents, whenever it happens, is via message broadcast, following the FIPA protocol. Reasoning follows the Belief-Desire-Intention (BDI) model [2] [15], with a set of beliefs, desires and intentions configured at the start of the simulation and continuously revised as the simulation progresses.

The Belief-Desire-Intention (BDI) model of agency provides the cornerstone for individual agent reasoning and action. Based on beliefs, desires (goals) and intentions, an agent is capable of designing partial plans that will get it closer to its goal. Belief revision and iterative planning can be used to enable learning and intelligent behavior. Each agent has a set of beliefs, desires and intentions that enables it to reason within the environment, with the available information. An agent's preferred social behavior is heavily dependent on the interaction context as well as individual goals and beliefs.

To study agents' social behavior, we have chosen to simulate an emergency scenario, given the potentially high value of successful results. More specifically, we focused on the task of escaping from a fire, in an unknown environment inhabited by other, unknown, agents. This scenario has two desirable characteristics: it is dynamic in nature and it is time constrained. Our goal is to analyze preferred social behaviors and evaluate the survival rate and the average time to escape. We consider five different strategies: randomly act (RANDOM), never communicate (SILENT), always exchange information (COMMUNICATIVE), think first-act later (THINKING) and think-act-learn (LEARNING).

Our emergency case considered a fixed 2D squared matrix as the world. There are a fixed number of exits randomly placed in the world. Agents must find a way out within the allotted time frame, using the information they have. Agents only have partial information about the labyrinth. There are two ways to gain information: through exploration or through the exchange of information with other agents. In this fashion, we design agents that employ different strategies and measure their success rate in different environmental conditions.

## 2. Application Domain: Emergency Scenario

Emergency situations are dynamic environments in which individuals have little time and partial information to decide their course of actions. Additionally, decisions that may lead to highly negative outcomes such as death [12] [19]. Unfortunately, in emergency scenarios, individual behavior is unpredictable [17]: there is no consensus on the best strategy to increase the survival rate. Whenever possible, workshops and real life simulations are used to train people to recognize situations and react, without thinking, to improve the changes of a successful result.

Individual and group reactions have gained importance with the many recent terrorist attacks around the world [18]. However, these emergency scenarios existed well before the terrorist attacks, in situations such as evacuating a burning building or shopping center, a sinking ship or an offshore oil platform. Therefore, understanding the behavior of individual agents or group of agents working together in emergency scenarios enables predicting the outcomes according to different interaction strategies.

In emergency scenarios, timing is of the essence: the environment changes rapidly, so agents need to move fast before their knowledge of the environment becomes outdated [16]. Additionally, in real emergencies agents have limited time to accomplish their goal. In an environment with many agents and obstacles, calculating full paths for each agent becomes a computationally intensive task. Additionally, as the environment is dynamic, these paths need to be recalculated at each interaction. Perhaps more importantly, in an actual situation, individuals operate based on incomplete knowledge of the environment, making the best choices possible given the information they possess. Accordingly, agents have partial knowledge of the world, and this knowledge is updated as they move about and perceive the world around them and also as they exchange information with other agents, indicating what the state of the world is in other locations.



We are particularly interested in how individual agents can coordinate amongst themselves to achieve their goals. In these settings, communication has a limited range, and information is sometimes incomplete or incorrect, so agents need to calculate the validity of information they receive to decide whether they should act on it. A similar investigation was developed on the correlation between environment layout and chances of survival for different types of agents [5].

In ACoPla, each individual is modeled as an agent, whose goal is to exit the affected area. The world is modeled as a graph, where paths are represented through nodes and links, and at least one of the nodes is an exit. Each agent has a set of beliefs, which is the known subset of the graph (the locations the agent has visited) plus beliefs about other paths, based on information received from other agents. This belief set is valid for a limited time, as changes in the environment may lead to incorrect beliefs. Therefore, an agent needs to revise and update its beliefs about the world with new information as it arrives. The agents' main goal is to exit the labyrinth in the allotted time, and it will act according to this goal.

### 3. ACoPla: Agent Coordination and Planning

ACoPla is a multi-agent system that allows the construction of different types of BDI agents with different social strategies. Agents' basic goal is always the same: to exit the labyrinth. However, we believe agents' social behavior impacts the survival success rate, and ACoPla is built to take that into consideration. The main algorithms are presented in Figure 1. In this section, we present the main elements of the simulation: the Environment, the Agent, the Inter-agent Communication protocol and the Environment Sensing protocol.

#### 3.1. The Environment

The environment is represented as a graph, where nodes are linked to each other. Each node has one of two states:

- Clear (a node the agent can walk through) or
- Blocked (when it is not possible to walk through this node). Blocked nodes represent fire spots or similar obstacles that render the position impassable.

The environment also has a number of exits points (special nodes from which it is possible to exit the labyrinth), usually proportional to the graphs size. Time is counted as rounds to avoid discrepancies due to processor differences.

#### 3.2. The Agent

In addition to the Environment, the framework must allow the agents characteristics to be specified. These will drive their behavior at a later stage. In ACoPla, each agent has the following set of characteristics:

- Starting position: where the agent is when the game starts
- Timeframe: how much time the agent has to reach the exit
- Vision range: how far an agent can see from any given position in the environment (graph node)
- Speech range: how far a message sent by the agent can be heard
- Hearing range: how far the agent can listen a message broadcast by others

---

## Agent Behavior Algorithm

---

```
mainLoop (EnvironmentPartialView, CurrentLocation, PartialGoal, Goal, TimeFrame)
  If any PartialGoal[i] = ExitGoal
    If Not (Null (Path[i]-Dijkstra(CurrentLocation, Goal[i])))
      Then Go(Path[i]) and Go to Action Phase
```

### Planning phase

```
For each partial goal i do
  Pathi- Dijkstra(CurrentLocation, Goal[i])
  AgentPath- SelectShortest(Path)
While Null(AgentPath) keep
  AskEnvironment (CurrentNode, EnvironmentSubGraph)
  UpdateEnvironmentView (EnvironmentSubGraph)
  If there is still Time and the environment knowledge changed
    Then Go to Planning Phase
```

### Action Phase

```
For each node[i] in AgentPath
  CheckStatusNode(Node[i] status)
  Talk(AgentId, Node[i], Status)
  If Status(Node[i]) is ok
    Then Go(Node[i] from Node[i-1])
  Else UpdateEnvironmentView(link(Node[i-1],Node[i]), LinkStatus)
    Go to Evaluation Phase
```

### Environment Interaction Phase

```
AskEnvironment (CurrentNode, EnvironmentSubGraph)
UpdateEnvironmentView (EnvironmentSubGraph)
```

### Agent Interaction Phase

```
ListenMessages (CurrentPosition, HearingReach, Messages)
Select Messages to believe
UpdateEnvironmentView (EnvironmentSubGraph)
If there is still Time and the environment knowledge changed
  Then Go to Planning Phase
  Else If there is still time
    Then return to Environment Interaction Phase
  Else Die
```

### Learning Phase

```
EvaluateOutcome (Action, CurrentKnowledge, GainedKnowledge)
If PositiveOutcome
  Then Reinforce (Action)
Else Reduce (Action)
  Go to Planning Phase
```

---

*Fig. 1: Agent Behavior Algorithm*

To achieve their goals and learn more about the environment, agents can exhibit the behaviors listed below. Each action has a cost in terms of time. Reasoning about what to do also has a time cost, which is a function of the amount of information the agent has. At each round, the agent can perceive the environment (seeing the surrounding environment within sight range) and then decide what to do. Action's cost is expressed in terms of time needed to execute it. Possible agent behaviors are:

- Move (position): agent moves to the target position.
- Run (position): this action is actually a set of moves to a given position, but the agent moves without thinking and doubles its speed.
- Exit: exit the map (only possible when at an exit point).
- Ask (agent, information): ask another agent if it has information to exchange.
- Tell (agent, information): provide information to another agent (exchange information).
- Ignore (agent): ignore another agent's request for information.

Action's cost varies and can be configured according to the situation being simulated.

Agents begin with no knowledge of the environment (empty maps), and each agent's knowledge is constructed as it moves about the map or exchanges messages with other agents. The variables that influence the choice of behavior at each round are:

- Current position: the agents current position;
- Remaining time: the amount of time the agent has left before expiring;
- Vision window: how far the agent can see;
- Exit route: whether a route to the exit is known;
- Target agent: agent to be followed;
- Agent Trust: based on previous encounters, the degree of trust among agents (not explored in this paper);
- Current knowledge: set of known agents and set of visited nodes, i.e. current world map.

### 3.3. Agents' Communication Protocol

Communication is normally undertaken one-to-one, as agents speak to each other privately. However, it is also possible to broadcast information within a short range. In this case, agents within range will hear the message. Agents can differ in complexity through their reasoning and decision making capabilities. The map is subdivided into sections (north, South etc.) so that agents can exchange information based on general directions and not absolute positions. This means that an agent may tell another that it has information concerning a specific section, as requested, in exchange for something, another agent may choose not to exchange information because it knows what it needs or even broadcast all it knows about the place for free. This introduces a level of uncertainty, since agents never know exactly what they will learn, and also creates a need for better decision-making, as agents must consider whether they want to know about a certain region or if they already know enough about it. Communication is undertaken using the following protocol:

- Ask (agent, region, exchange?): the agent asks a neighboring agent whether it wants to exchange information, and volunteers what region it knows about.
- Tell(agent, [reply]): the agent identifies itself and replies in one of three ways:
- Exchange = N
- Exchange = Y; ExitLocation
- Exchange = Y; RegionMap

At this point, we consider that an agent will always provide full and correct information (to the best of its knowledge). However, an agent may choose not to exchange information if that would not be in its best interests. Issues of incorrect information and trust will be introduced at a later stage in the research.

### 3.4. Environment Sensing Protocols

As mentioned before, the environment is represented as a directed graph in which nodes are possible locations the agent can go to and links are paths they can traverse to move from one location to the other. The Agent-Environment Interaction Protocol is:

- AskEnvironment (CurrentNode, VisionReach, EnvironmentSubGraph): the agent checks the environment, looking at the world and acquiring knowledge considering its vision constraint.

### 3.5. Agent Strategies

Different types of agent's strategies can be instantiated through a combination of the aforementioned behaviors. For initial tests of our framework, we instantiated the following agent strategies:

- Random (RANDOM): walks randomly without thinking or communicating.
- Silent Explorer (SILENT): never exchanges information with other agents. Tries to increase its knowledge of the environment by moving to new parts of the map. Reasons whether or not to follow other agents or run. When it finds an exit, it moves towards the exit does not broadcast this information.
- Communicative Explorer (COMMUNICATIVE): always exchanges information. Tries to increase its knowledge of the environment by moving to new parts of the map. Reasons whether or not to move randomly or run. When an exit is found, broadcasts this information.
- Thinking Explorer (THINKING): tries to increase its knowledge of the environment by moving to new parts of the map. It may display any of the behaviors. It decides which behavior to execute by trying to maximize its utility function.
- Learning (LEARNING): the most complex agent, it is a step up from the Thinking Explorer. It may also display any of the behaviors. It not only reasons about which behavior to execute, but also evaluates the outcomes after executing it. The learning capability reinforces the behaviors that led to the better outcomes. Learning is a function of the amount of information gained with each action.
- After executing an action, the agent evaluates how much information it gained with it. The action executed is reinforced proportionally to the gain, so that, if there was a good outcome, there will be an incentive to perform it again next time.

## 4. Experiments

To check our framework and analyze the effect of different environment variables on the performance of our agents in finding an exit in a dynamic environment, we constructed a simulation. Agents move around in a graph, following links between nodes while looking for an exit point. We instantiated environment graphs and ran simulations with different times to exit, number of exits and map sizes to see how well they would do. Table I shows the list of possible agent behaviors and their cost in terms of time, for this instantiation of the model.

Table I: Agent behavior - cost structure

Primitive Behavior (Action)	Cost	Composite behavior	Cost
Move (position)	1	Explore = Think + Move	2
Run (position)	1	Follow (agent) = Think + Move	2
Exit	0	Exchange Information = Think + Ask/Tell	2
Ask (agent, information)	1	Sprint (position) = Think (decide to run) + Run (x positions)	1 + x
Tell (agent, information)	1	Broadcast (information) = Tell (within range)	1
Ignore (agent)	0		

### 4.1. Experimental Setup

To provide all agents with the same chances, agents all started from the same position in the map. They had constant sight (vision range) and speaking and hearing range (for broadcast purposes) and they always had the same time to exit. The simulation was run considering the following run configuration:

- Initial position = random
- Vision range = 2
- Hearing range = 0 (must be in a position within speaking range of the other agent)

- Speaking range = 4
- Map size: 100x100
- Number of Agents: 50

The independent variables, i.e. number of exits, max time to exit, were set at:

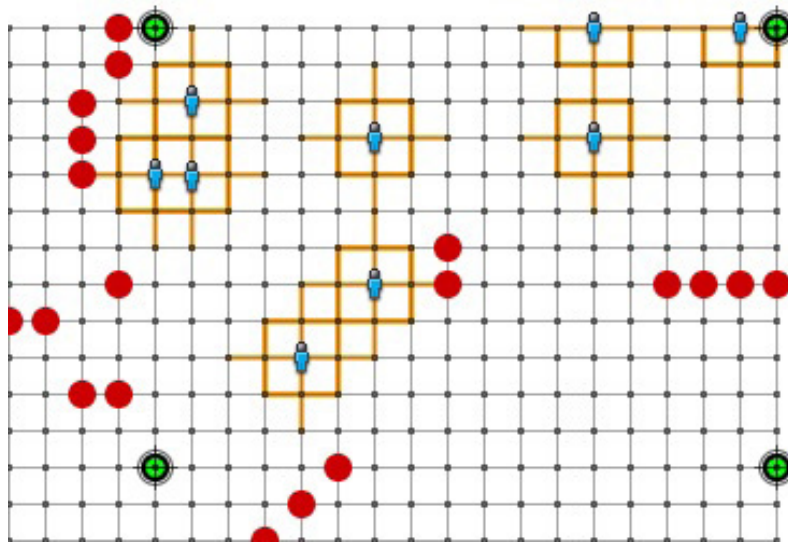
- Maximum time to exit: 10, 50, 100 rounds
- Number of Exits: 10, 20, 40, 80

The dependent variable was:

- Number of survivors

## 4.2. Experimental Run

Each iteration was run 10 times, and consisted of a map of the given size, number of exits and time, inhabited by 50 agents of the same type (homogeneous setting). At the end of each run we measured the number of survivors to see how well the strategy worked. Figure 2 shows a partial view of the map. Yellow squares represent the agents vision.



*Fig. 2: Agent Environment: red dots are fire spots; green dots are exit places; yellow squares are vision range*

Our analysis indicates that, when little time is available, randomly walking around is the best option. The graphs in Figure 3 shows the outcomes of the simulation when maximum time = 10 rounds. In this situation, agents that exit do so by luck, because they happened to be close to an exit.

## 4.3. Experimental Results

Even though the thinking and random agents had similar results, thinking agents actually took longer to reach the exits, as shown in Figure 4, so we consider this worsens the results. Thinking takes time, and, in an emergency situation, it might be beneficial to act instinctively.

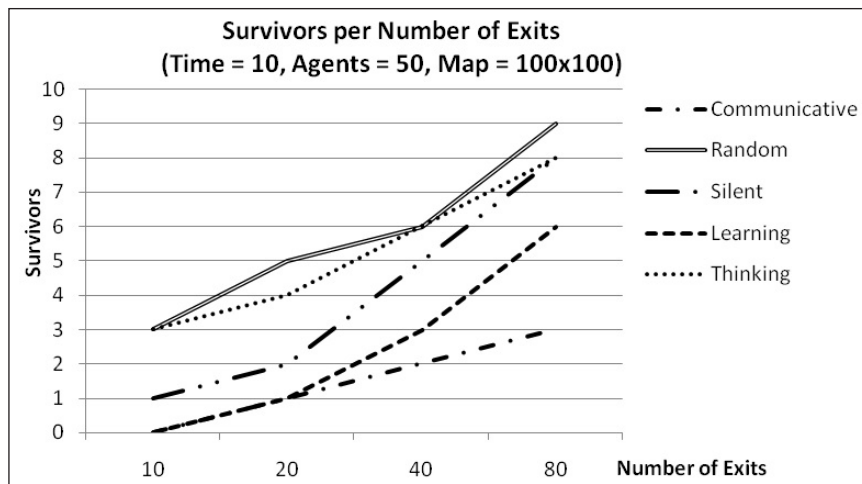


Fig. 3: Number of survivors when maximum run time = 10

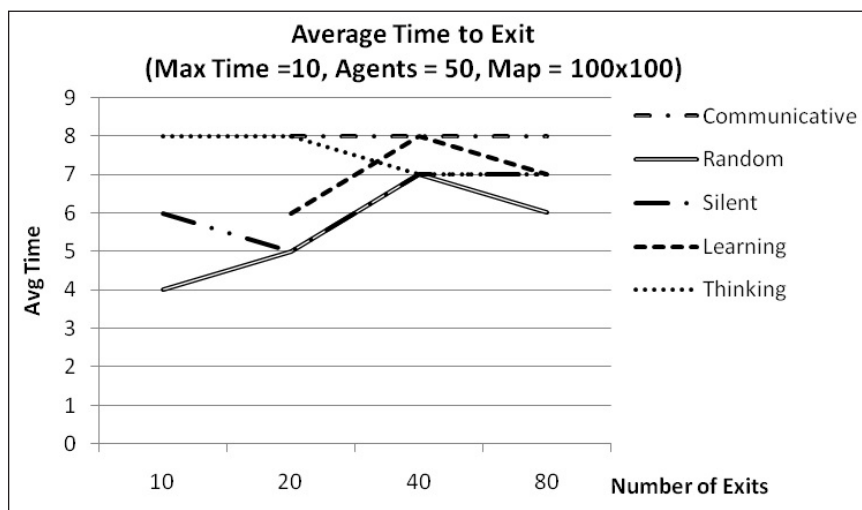


Fig. 4: Average time to exit, when maximum run time = 10

Even though random searching is good in short amounts of time, another analysis reveals that, with time, thinking becomes a better option, as shown in Figure 5.

The learning agent follows the results of thinking, and we believe it will grow to the same level with some tweaking of the algorithm. It should be noted, that the thinking agent encodes our preferred reasoning, and acts as a non-panicked actor able to rationally decide what the best option is, so it functions as our baseline. We expect the learning agent would be able to grow to that point given enough time.



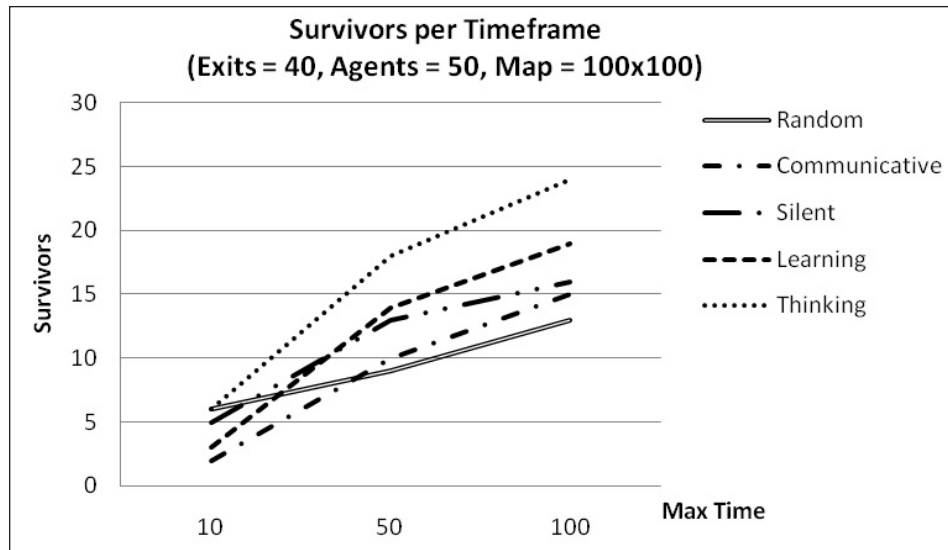


Fig. 5: Number of survivors in different timeframe

## 5. Discussion and Further Work

Simulations of evacuation situations have been used before to help with planning, building design and emergency preparedness (e.g. [10]). Zhang [20] discusses modeling urban expressways emergency conditions using petri nets as a basis for simulation, while Baster [1] uses rules to achieve human-like decision-making in agent-based simulations. Meshulam, Felner and Kraus [9] explore multi agent navigation where agents decide between exploratory behavior and the use of a known suboptimal path to find an exit. Agents are able to exchange information about the environment. Their approach outperformed a greedy approach. Full and partial knowledge sharing provided better results in respect to time elapsed and number of messages needed for the agent to find better paths.

Braun and colleagues [3] describe a multiagent crowd simulation system in which individualized agents have particular properties, such as dependence on others, and altruism levels that influence their social behavior. In a “room exiting” task, agents behaves realistically, with some being faster than others and some going back to help others who needed help. They take a descriptive approach trying to mimic actual crowd social behavior to generate a believable crowd background for virtual reality and cinema domains. BDI Agents have also been successfully used for crowd simulation, as seen in [14]: a set of BDI agents is used to simulate a terrorist attack and evacuation scenario. Their experiment was built in a virtual reality environment. Most agents had “shelter-seeking” or “running-from-the-scene” behaviors, while others (officials and first responders) were able to instruct victims, directing them to viable exits. Their objective was to use this virtual reality environment for training.

On a different theme, Pan [13] introduces computer simulations with a focus on the study of human behavior, and Chu [7] builds a multiagent framework to simulate different behaviors based on social studies. Noh and Gmytrasiewicz [11] present a large-scale agent simulation under time pressure, but are more concerned with system performance than strategy development.

Our study follows on the footsteps of previous ones, but we wish to create a framework where we can model multiple different situations, especially decision making under uncertainty. Through this study, we wish to stress the importance of observing certain variables before selecting a strategy. Environmental factors influence the outcomes, such that it is important to verify the influence of these variables on each strategy before selecting one.

This framework can be applied to other domains, for instance map size can be translated to uncertainty in economics or other decision theory (how much of the domain is unknown). Time is another variable that appears frequently: pressing decisions that must be made in fast paced environments. Number of agents can be translated to the number of players/competitors involved.

In the tests we ran, noticed a few interesting points: in some situations, panic is the best option. Additionally, we see that our learning agent does learn, given time. This appearance of a learning curve to be overcome before the agent can make good decisions is unexpected and an interesting side effect, which makes the simulation more realistic. By modeling more complex agents and behaviors, we can generate fairly realistic scenarios, and we can then try out different ways of intervening in the situation to get to the best results for the whole group.

## 6. Acknowledgment

Not included by maintaining blind revision.

## 7. References

- [1] B. Baster, J. Duda, A. Maciol, and B. Rbiasz. Rule-based approach to human-like decision simulating in agent-based modeling and simulation. In *2013 17th International Conference on System Theory, Control and Computing (ICSTCC)*, pages 739-743, Oct 2013.
- [2] Michael Bratman. Intention, plans, and practical reason. 1987.
- [3] Adriana Braun, Soraia Raupp Musse, Luiz Paulo Luna de Oliveira, and Bardo EJ Bodmann. Modeling individual behaviors in crowd simulation. In *Computer Animation and Social Agents, 2003. 16th International Conference on*, pages 143-148. IEEE, 2003.
- [4] Davide Brugali and Katia Sycara. Towards agent oriented application frameworks. *ACM Computing Surveys (CSUR)*, 32(1):21, 2000.
- [5] Ana C. B. Garcia, Nayat Sanchez-Pi, Luis Correia and José M. Molina. Multi-agent simulations for emergency situations in an airport scenario. *Advances in Distributed Computing and Artificial Intelligence* (1):69-78, 2012.
- [6] Victor R Lesser. Multiagent systems: An emerging subdiscipline of ai. *ACM Computing Surveys (CSUR)*, 27(3):340-342, 1995.
- [7] Mei Ling Chu and Kincho Law. Computational framework incorporating human behaviors for egress simulations. *27:699-707*, 11 2013.
- [8] Charles M Macal. Everything you need to know about agent-based modelling and simulation. *Journal of Simulation*, 10(2):144-156, 2016.
- [9] Ram Meshulam, Ariel Felner, and Sarit Kraus. Utility-based multi-agent system for performing repeated navigation tasks. In *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pages 887-894. ACM, 2005.
- [10] Y. Murakami, K. Minami, T. Kawasoe, and T. Ishida. Multi-agent simulation for crisis management. In *Proceedings. IEEE Workshop on Knowledge Media Networking*, pages 135-139, 2002.
- [11] Sanguk Noh and P. J. Gmytrasiewicz. Flexible multi-agent decision making under time pressure. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 35(5):697-707, Sept 2005.
- [12] Juliane Noveanu, Felix Amsler, Wolfgang Ummenhofer, Thomas von Wyl, and Mathias Zuercher. Assessment of simulated emergency scenarios: Are trained observers necessary? *Prehospital Emergency Care*, 21(4):511-524, 2017.
- [13] Xiaoshan Pan, Charles S. Han, Ken Dauber, and Kincho H. Law. Human and social behavior in computational modeling and analysis of egress. *15:448-461*, 07 2006.
- [14] Ameya Shendarkar, Karthik Vasudevan, Seungho Lee, and Young-Jun Son. Crowd simulation for emergency response using BDI agents based on immersive virtual reality. *Simulation Modelling Practice and Theory*, 16(9):1415-1429, 2008.



- [15] Young-Jun Son. An integrated human decision making model under extended belief-desire-intention framework. In *Proceedings of the 2017 ACM SIGSIM Conference on Principles of Advanced Discrete Simulation*, pages 39-39. ACM, 2017.
- [16] David Szpilman, Billy Doyle, Jenny Smith, Rachel Griffiths, and Mike Tipton. Challenges and feasibility of applying reasoning and decision making for a lifeguard undertaking a rescue. *International Journal of Emergency Mental Health and Human Resilience*, 19(4):1-9, 2018.
- [17] Ashutosh Trivedi and Shrisha Rao. Agent-based modeling of emergency evacuations considering human panic behavior. *IEEE Transactions on Computational Social Systems*, 2018.
- [18] William L Waugh. Terrorism as hazard and disaster. In *Handbook of Disaster Research*, pages 123-143. Springer, 2018.
- [19] Chao Zhang, Jiansong Wu, Chao Huang, and Bo Jiang. A model for the representation of emergency cases. *Natural Hazards*, 91(1):337-351, 2018.
- [20] Yaying Zhang, Yuefeng Fu, and Wei Qiang. Modeling and control of urban expressways with emergency using hybrid petri nets. In *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 001483-001489, Oct 2016.



