Multi-agent simulations for emergency situations in an airport scenario

Ana Cristina Bicharra[a], Nayat Sanchez-Pi[a], Luis Correia[b], José Manuel Molina[c]

[a] ADDLabs, Universidade Federal Fluminense. Brazil
[b] Faculdade de Ciências. Universidade de Lisboa. Portugal
[c] Computer Science Department. Universidad Carlos III de Madrid. Spain

KEYWORD

Multi-agent systems
Emergency situations
Simulation

ABSTRCT

This paper presents a multi-agent framework using Net- Logo to simulate human and collective behaviors during emergency evacuations. Emergency situation appears when an unexpected event occurs. In indoor emergency situation, evacuation plans defined by facility manager explain procedure and safety ways to follow in an emergency situation. A critical and public scenario is an airport where there is an everyday transit of thousands of people. In this scenario the importance is related with incidents statistics regarding overcrowding and crushing in public buildings. Simulation has the objective of evaluating building layouts considering several possible configurations. Agents could be based on reactive behavior like avoid danger or follow other agent, or in deliberative behavior based on BDI model. This tool provides decision support in a real emergency scenario like an airport, analyzing alternative solutions to the evacuation process.

1 Introduction

Emergency situation appears when an unexpected event occurs as, for example, earthquake, flood, terrorist attack, burning building or shopping center, sinking ship or an offshore oil platform, etc. In indoor emergency situation, evacuation plans defined by facility manager explain procedure and safety ways to follow in an emergency situation. One of the key issues identified by facility managers is safe egresbased on the layout of the public building and the crowds behavior. The importance of this issue is related with incidents reported regarding overcrowding and crushing in public buildings (Zhan et al. 2008). The definition of safe egress in a public building is an unsolved problem (Pan et al. 2007) that could be improvement by Information Technologies (Filippoupo- litis et al. 2008).

This paper presents an agent based framework using NetLogo to simulate human and collective behaviors during emergency evacuations. Simulation has the objective of evaluating building layouts considering several possible configurations. Simulation considered heterogeneous agents to include the human factor in the collective behavior with levels of interaction as function of the individual capacities. An objective evaluation function, based on the percentage of live people at the end of simulation, is considered.

2 Agent Based Simulation for Emergency Evacuation

In agent simulation, the model specifies behaviors of individuals, in contrast to macro simulation techniques that are based on mathematical models (Davidsson 2002). The use of agent based simulation for modeling emergency evacuation is related with the capacity to analyze collective behavior. In macro simulations, the collective is defined by a number of variables, whereas in micro simulations the collective goes defined by the emergence from the interactions among individuals.

Several approaches in literature model the collective human behavior using agent systems. One of them are centered in simulate a realistic crowd, where the behavior of individuals allow represent different collective behavior similar to a real world. One of the main approaches in this line is Braun and colleagues (Braun et al. 2003). In this work, the multiagent crowd simulation system has individualized agents with particular properties, such as dependence on others and altruism levels, and act according to these behaviors. The simulation try to represent the collective behavior in a realistic way, for example, in a room exiting task, some agents being faster than others and some going back to help others who needed help. The main goal is to generate realistic crowd behavior in a simulation, which can be used in virtual reality or movies.

An application of these simulators is the analysis and the design of buildings and evacuation plans. In (Pan et al. 2007) a multi-agent simulation framework is developed for simulating individual cognitive processes for exploring emergent phenomena such as social or collective behaviors.

The paper presents a Multi-Agent Simulation System prototype for Egress analysis (MASSEgress). The main focus of this work is modeling frequently observed human social behaviors in emergencies, such as competitive, queuing, and herding behaviors, through simulating the cognitive processes of individual agents and interactions among multiple agents in an artificial environment. The MASSEgress tool analyzes these situations on a predefined building design, this mean when and where occur. Then, using cludes the information of reachable and avoidable nodes. The more they see the world, the better there chance to perceive an escape node. At any given time an agent is in one of the 13 possible situations vary- ing from totally free to totally blocked or in imminent death. His sight can see more than just its neighbour, but for simplicity it was a single neighbour distance sight reach. Since they will be moving in the environment, they need an algorithm to trace a path. The shortest path between two nodes is denoted as the distance between the two nodes, dist(u,v) whenever v is reachable from u by the path. Dijkstras shortest path algorithm calculates the distance, according to a path, between two nodes reachable. The time complexity of Dijkstras algorithm is in the order of O(m + nlogn) time for which m is the number of edges and n is the number of vertices of the graph. The world is dynam- ic.

As time goes by, agents change position, the danger spreads meaning the graph connections change. Consequently, even when agent starts knowing the entire world, this knowledge keeps downgrading with time. That fact justifies the helpfulness of exchanging information when meeting other agents even when agents know the world. Danger spots start in cell units that can be either randomly allocated, such as in a forest fire that we never know the focus points, or predefined, such as in a dam river that we know the weak points. In order to run a simulation, it is neces- sary to define spread function. Consequently, at the same time agents are moving and making cells tem- porarily occupied, danger keeps spreading out throughout the environment cells. Our premise is that once in danger, the cell will stay in danger forever during the simulation and the set of blocked nodes in danger keeps increasing according to a pre-defined danger spreading function from a triggered cell. We consider a damaged cell a world unit from which agents should stay away from it to stay alive. The emergency simulation runs in continuous time steps.

Agents resources are mapped to time units of their life. Each type action differently decreases units of agents lifetime. For instance, depending of the scenario, moving from one cell to another may require less lifetime resource than exchanging information to other agents. The simulation runs in two different modes: exhaustive and bounded by time frame. In the first scenario, the simulation runs until either agents escape, die or get locked. In the later scenario, the simulation runs for a specified amount of time.

MASSEgress tool and a visual inspection of the simulation, an expert in the field could determine which the best building design for evacuation por- poises is.

Finally, other works are centered in the possibility to apply simulation in real time. These tools give decision support in a real emergence, analyzing alternative solutions as the evacuation evolves. (Filippoupolitis et al. 2008) present a augmented reality simulation system to operate in an emergency disaster to evaluate evacuation strategies in real-time, named Building Evacuation Simulator. The system is able to evaluate evacuation policies for a specific building. Authors show the effect of individual and collective behaviors in an evacuation procedure, including grouping behaviors and the inclusion of the leadership role.

4 Problem formulation

People, represented by agents, move in an indoor building defined in a layout. When an emergency occurs (that is propagate all over the layout, as the fire, with a speed and a certain topology), agents can move to search the exit with a certain speed or warn others about the emergency and about exits. In that case, agents will form a kind of a network quickly spreading the warnings over possibly all agents and the whole environment. This will strongly depend on the communication range of the agents.

4.1 World definition

The world is a 2D matrix of cell position, in which agents act upon, is represented as a directed graph G=(V;E; -), for which V is a non-empty set of nodes; E is a set of edges, one for each link; and - is a weight function from path E reflecting nonzero positive real numbers. The number of nodes is denoted by n, and the number of the directed edges m. A directed edge in E is denoted by an ordered pair of nodes from V. If directed edge e = uv 2 E , node v is said to be reachable from node u in E. The weight of edge uv is denoted by '(uv). A path between two nodes v0 and vk is nite sequence p = v0; v1:::vk of nodes such that for each 0 < i < k; vivi+1 2 E, and the weight of the path is -(p) = P 0<i<k; -(vi; vi+1).

According to this representation of the world, agents are always located in a node cell, moving from node to node to find an escape route. They may have a partial or total view of the world, a sub graph that I cludes the information of reachable and avoidable nodes. The more they see the world, the better there chance to perceive an escape node. At any given time an agent is in one of the 13 possible situations varying from totally free to totally blocked or in imminent death. His sight can see more than just its neighbour, but for simplicity it was a single neighbour distance sight reach. Since they will be moving in the environment, they need an algorithm to trace a path. The shortest path between two nodes is denoted as the distance between the two nodes, dist(u,v) whenever v is reachable from u by the path. Dijkstras shortest path algorithm calculates the distance, according to a path, between two nodes reachable. The time complexity of Dijkstras algorithm is in the order of O(m + nlogn) time for which m is the number of edges and n is the number of vertices of the graph. The world is dynamic.

As time goes by, agents change position, the danger spreads meaning the graph connections change. Consequently, even when agent starts knowing the entire world, this knowledge keeps downgrading with time. That fact justifies the helpfulness of exchanging information when meeting other agents even when agents know the world. Danger spots start in cell units that can be either randomly allocated, such as in a forest fire that we never know the focus points, or pre-defined, such as in a dam river that we know the weak points. In order to run a simulation, it is necessary to define spread function. Consequently, at the same time agents are moving and making cells temporarily occupied, danger keeps spreading out throughout the environment cells. Our premise is that once in danger, the cell will stay in danger forever during the simulation and the set of blocked nodes in danger keeps increasing according to a pre-defined danger spreading function from a triggered cell. We consider a damaged cell a world unit from which agents should stay away from it to stay alive. The emergency simulation runs in continuous time steps.

Agents resources are mapped to time units of their life. Each type action differently decreases units of agents lifetime. For instance, depending of the scenario, moving from one cell to another may require less lifetime resource than exchanging information to other agents. The simulation runs in two different modes: exhaustive and bounded by time frame. In the first scenario, the simulation runs until either agents escape, die or get locked. In the later scenario, the simulation runs for a specified amount of time.

4.2 Emergency Model

Emergency is represented as a set of events originated by external agents, such as a fire spot, that destroy world cell units and may damage agents whenever in contact with them. There may be one or more source of these events that spreads into the world with time according to some spreading function. The emergency changes the status of the cell from available to destroyed. An emergency event (Ev) has a degree of severity that reflects the degree of damage on an agent according to the distance between them, varying from 0 (no damage) to 1(kill agent whenever touches it). In order to simulate our crowd evacuation scenario, it is important to define the world density

(AD), in terms of number of agents per squared cell unit. Additionally, for each agent in play, we have to define its characteristics (Agi) including its initial position cell (Cs), the time when it started playing (Ts), its cognitive skills (Think), such as Reactive, Cognitive or Follower, for reaching their goal of escaping or saving somebody else, their physical skills such as its motion (Move), vision (See) and hearing abilities (Hear), and its role (Role) in the world such as being a civilian, a villain or a hero. Agents initial position can be specifically defined, such as for a fireman starting at an exit cell, or randomly determined.

Each agent occupies exactly one unit. Each unit has exactly one agent at a time, except when and agent is carrying another. We are interested in studying individual behavior for individual as well as population survival in emergency scenarios. As for the population, we will look at averages and standard deviations of duration to escape. For individual analysis, we will be looking at: time to escape (Te), starting point (Cs), stating time (Ts) and chosen exit (Exit). As measurements, or outputs, of the experiments we defined the following observables:

Ndi: number of deaths, per agent type.

Tei: average time survivors took to escape the environment, per agent type.

T0: number of iterations completed until no more agents in the environment

We also record, for each experiment, the location of the fire breakout so that we can latter correlate its location with the escape results. For instance, fire breakouts near an exit are prone to produce much worse results than most of the other breakout locations.

4.3 Agents Architecture

We are simulating crowd behavior in emergency scenario using two different approaches. In the first approach, we model individual agents varying their cognitive skills as the building block unit to create the society. In the second approach, we model the crowd as a compact unit based on swarm theory.

In order to model each individual agent in the world scenario, it is necessary to describe the way they perceive and perform on the environment as well as their reasoning abilities. Our agents present the following skills:

1. Agents initial position: reflect its coordinates in the world graph;

2. Agents resources: reflect the available resource to perform the actions to achieve its goals. Different types of resources can be mapped to a single one. In our case we are mapping to time units of life.

3. Agents interaction abilities: reflect its abilities to perceive and act on the environment.

(a) Perception skills include:

i. Sight range defines from its current location the depth of the subgraph of the world it is able to see.

ii. Hearing range defines from how far the agent can listen to messages. Similarly to sight range, it defines the maximum distance to others an agent can be in order to transfer information (communicate) about the world.

(b) Acting skills include:

i. Speed to move defines how fast the agent can move in the environment meaning how many cells per unit of time simulation the agent can move.

ii. Communication to others defines the ability agent has to transmit and receive information in a shared communication language.

4. Agents learning skills: reflect the amount of information the agent can incorporate in its memory.

5. Agents reasoning ability defines the way they decide its next action.

Agents vary in their reasoning skills from no reason to full alternative generation and evaluation. No matter the reasoning process, agents act perceiving the environment though their sensors, such as hearing and sight sensors, choosing what to do next and acting in the environment though their actuators, such as moving, communicating or planning what to do next. We are considering three main types of agents inhab iting the environment: purely reactive agents, followers and cognitive agents.

Reactive agents: Purely reactive agents randomly choose their next action, just avoiding the immediate danger. There is neither memory from the past nor a rational decision making process associated with their next move. Their inference algorithm is similar to a blind search with no memory of previous states as described in Algorithm 1.

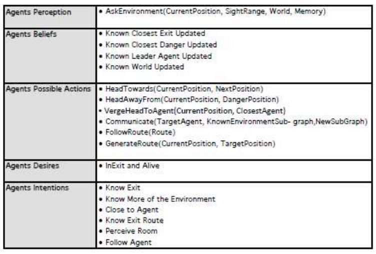
```
Algorithm 1: Reactive Agent reasoning algorithm
    Input: Agent State, Agent Goal
    Output: Reactive reasoning
1   AskEnvironment(CurrentNode, EnvironmentSubGraph,
    SightRange)
2   if Reached(CurrentNode, Goal) then
        Exit with Success
3   if AgentRemainingLifetime = 0 then
        Exit with Success
4   Case
a   Reachable(Exit, CurrentNode, EnvironmentSubGraph):
    HeadTo(Closest(Exit), CurrentNode,
    EnvironmentSubGraph)
b   Reachable(DangerNode, CurrentNode,
    EnvironmentSubGraph):
    if Reachable(LocalExit, CurrentNode,
    EnvironmentSubGraph) and CloserTo(LocalExit,
    DangerNodeS) then
        HeadTo(LocalExit, CurrentNode,
        EnvironmentSubGraph)
    HeadAway(DangerNodeS, CurrentNode,
    EnvironmentSubGraph)
c   Reachable(LocalExit,CurrentNode,
    EnvironmentSubGraph): HeadTo(LocalExit,
    CurrentNode, EnvironmentSubGraph)
d   Otherwise StepRandomly(CurrentNode,
    EnvironmentSubGraph)
5   DecreaseLifetime(ActionCost)
```

Follower agents: Followers are agents that react differently. Their systematic behavior consists in following the others strategy. They have a glint of reasoning when needed. They follow the group with more followers. Whenever there is no one to follow, they behave just as the purely reactive agents. They also have no memory of previous history.

Their inference algorithm is mostly blind search based, with a cheap reasoning when meeting others, as described in Algorithm 2.

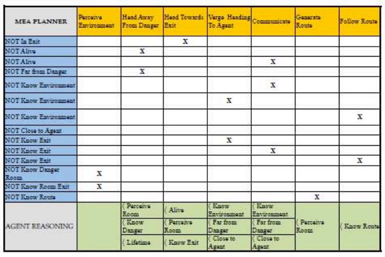
Cognitive agents: Cognitive agents are the ones that follow a rational decision making process to choose their next action. They also learn as they act upon the environment. They have memory and consider their history of interactions to choose the best move considering what they have already learnt from the world. There are many approaches to rational agents. Here, we take a practical reasoning approach to rep resent cognitive agents considering they reason towards actions (Bratman and Intentions) . Agents will act according to plans they create plans to fulfill Intentions to accomplish Desires considering a set of Beliefs about the environment and their own abilities, i.e. cognitive agents will be represented as BDI agents (Rao, Georgeff, and others 1995). Cognitive agents reason to decide the behavior for achieving adequate performance when deliberation is subject to resource bounds (Kinny and George 1991) . The be- liefs keep updating as times goes by. While beliefs remain, agents keep following their plans. The plan will be executed consuming agents lifetime according to the cost involved to execute each action of the plan. See Figure 1.

Each action consumes a certain amount of agents lifetime resource that should be configured to better reflect the world being modeled. We have considered all actions as consuming 1 unit of resource, except for the communication action.



Agents Perception	• AskEnvironment(CurrentPosition, SightRange, World, Memory)
Agents Beliefs	• Known Closest Exit Updated • Known Closest Danger Updated • Known Leader Agent Updated • Known World Updated
Agents Possible Actions	• HeadTowards(CurrentPosition, NextPosition) • HeadAwayFrom(CurrentPosition, DangerPosition) • VergeHeadToAgent(CurrentPosition, ClosestAgent) • Communicate(TargetAgent, KnownEnvironmentSub- graph,NewSubGraph) • FollowRoute(Route) • GenerateRoute(CurrentPosition, TargetPosition)
Agents Desires	• InExit and Alive
Agents Intentions	• Know Exit • Know More of the Environment • Close to Agent • Know Exit Route • Perceive Room • Follow Agent

Figure 1: Agents' perceptions, Beliefs, Desires, Intentions.



MEA PLANNER	Perceive Environment	Head Away From Danger	Head Towards Exit	Verge Heading To Agent	Communicate	Generate Route	Follow Route
NOT In Exit			X				
NOT Alive		X					
NOT Alive					X		
NOT Far from Danger		X					
NOT Know Environment					X		
NOT Know Environment				X			
NOT Know Environment							X
NOT Close to Agent							
NOT Know Exit				X			
NOT Know Exit					X		
NOT Know Exit							X
NOT Know Danger Room	X						
NOT Know Room Exit	X						
NOT Know Route						X	
AGENT REASONING		(Perceive Room (Know Danger (Lifetime	(Alive (Perceive Room (Know Exit	(Know Environment (Far from Danger (Close to Agent	(Know Environment (Far from Danger (Close to Agent	(Perceive Room	(Know Route

Figure 2: MEA planner for the cognitive agent.

```
Algorithm 2: Follower Agent reasoning algorithm
    Input: Agent State, Agent Goal
    Output: Follower reasoning
1   AskEnvironment(CurrentNode, EnvironmentSubGraph,
    SightRange)
2   if Reached(CurrentNode, Goal) then
        Exit with Success
3   if AgentRemainingLifetime = 0 then
        Exit with Success
4   Case
a   Reachable(Exit, CurrentNode, EnvironmentSubGraph):
    HeadTo(Closest(Exit), CurrentNode,
    EnvironmentSubGraph)
b   Reachable(DangerNode, CurrentNode,
    EnvironmentSubGraph):
    if Reachable(Agent, EnvironmentSubGraph) and
    CloserTo(Agent, DangerNodeS) then
        VergeHeading(Agent, CurrentNode,
        EnvironmentSubGraph)
    else if Reachable(LocalExit,CurrentNode,
    EnvironmentSubGraph) AND CloserTo(LocalExit,
    DangerNodeS) then
        HeadTo(LocalExit, CurrentNode,
        EnvironmentSubGraph)
c   Reachable(Agent, EnvironmentSubGraph):
    VergeHeading(NEAREST Agent, CurrentNode,
    EnvironmentSubGraph)
d   Reachable(LocalExit,CurrentNode,
    EnvironmentSubGraph):HeadTo(NEAREST
    LocalExitNodeS, CurrentNode, EnvironmentSubGraph)
e   Otherwise StepRandomly(CurrentNode,
    EnvironmentSubGraph)
5   DecreaseLifetime(ActionCost)
```

In this last case, we assume the effort is a function of the amount to be communicated. The agent assumes the other agent has the same amount of information to communicate too. Consequently, the communicative act will take from both agents the number of nodes they know together multiplied by an adjustment factor since communicate should be much faster than move. Agents create a plan based on the meansends analysis (Newell and Simon 1961) planning proce- dure described below in Figure 2.

The MEA technique (Newell and Simon 1972) is a strategy to control search in problem-solving. Given a current state and a goal state, an action is chosen which will reduce the difference between the two. The action is performed on the current state to produce a new state, and the process is recursively applied to this new state and the goal state.

We consider simple cognitive agents planning as following the means-ends analysis problem-solving technique (MEA) (Newell and Simon 1972). Problem solving with MEA requires agents to represent the states the world assumes at each iteration time. Given a current and a goal state, an action is chosen which will reduce the difference between the two. The action is performed on the current state to produce a new state, and the process is recursively applied to this new state and the goal state. The MEA table, illustrated in Figure 2 , represents the reasoning strate- gy for the cognitive agents acting in the emergency world. Column in blue represents the difference be- tween current and goal states to be removed. The line in orange means the operators capable of removing differences and, finally, the line in yellow represents the set of pre-conditions for applying a specific oper- ator, as illustrated in Figure 2. Duplicate lines reflect there is more than one way to remove a difference. More than one X in the same line means more than one operator must be applied.

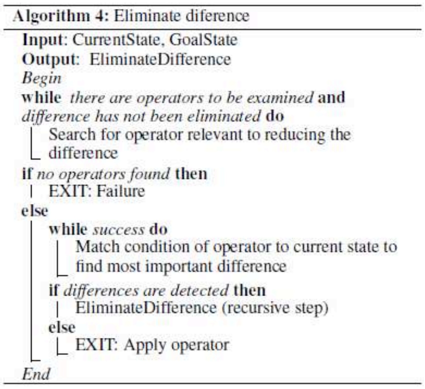
We use Andersons algorithm (Anderson and Moore 1985) to execute MEA as described below:

Apply the operator that will make the most important difference to the current state. In selecting the operator to apply, match the conditions of the operator to the current state to identify the most important difference. In this paper, we consider the following decision-making strategy in case of conflict. In any circumstances, survival is the most important goal, consequently head away from the danger will take over. Rational agents always prefer to head to an available exit whenever they know a route towards it, except when a fire is close to the exit. Otherwise they need to decide upon the alternative actions: explore the world in a rational way (following a previously generated route), verge heading to another agent, exchange information about the world with others and generate a route.

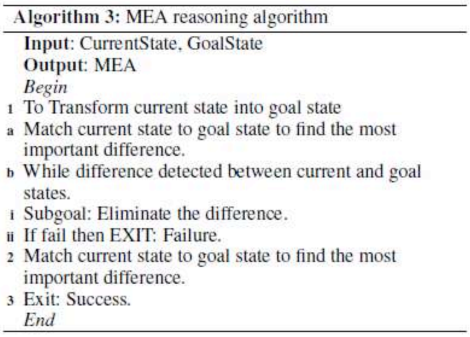
Whenever agent knowledge is insufficient to rationally create a route, it can randomly choose between two options: head to any local exit or verge heading to another agent. As the agent gains knowledge it makes sense to plan its own route trying to find an exit. Since we are neither considering agents reputation nor information truthfulness, communication is the preferred operation whenever meeting other agents, whenever the expected gain of information is greater than the expected gain of information exploring the world.

Agents expectation about others agents knowledge of the world is directly proportional of their own current knowledge.

This heuristic is based on the idea that all agents think they are alike. Consequently they believe eve- rybody is acquiring information at the same rate. This is a reasonable assumption with homogeneous agents, since the size of the explored world tends to be the same from each of them. Notice that, when the amount of knowledge the agent has about the envi- ronment is very large, the expected gain in exchang- ing information with other agents decreases. At this point, agent falls naturally into planning its own route stopping communication.



```
Algorithm 4: Eliminate diference
  Input: CurrentState, GoalState
  Output: EliminateDifference
  Begin
  while there are operators to be examined and
  difference has not been eliminated do
      Search for operator relevant to reducing the
      difference
  if no operators found then
      EXIT: Failure
  else
      while success do
          Match condition of operator to current state to
          find most important difference
      if differences are detected then
          EliminateDifference (recursive step)
      else
          EXIT: Apply operator
  End
```



Algorithm 3: MEA reasoning algorithm

```
Input: CurrentState, GoalState
Output: MEA
Begin
1 To Transform current state into goal state
  a Match current state to goal state to find the most
    important difference.
  b While difference detected between current and goal
    states.
      i  Subgoal: Eliminate the difference.
      ii If fail then EXIT: Failure.
2 Match current state to goal state to find the most
  important difference.
3 Exit: Success.
End
```

5 Experiments

A set of experiments was prepared to assess the behaviour of agents with different capabilities under the proposed model. The general approach was to use populations with different compositions in the number of agents of each type. Results are analysed in order to identify to what extent we can draw conclu sions from the model. In this set of experiments we considered fire as the cause of emergency. Depending on the exposition of the agents to fire they can be injured and they can die. Injuries are represented by a decrease of health points from the maximum corresponding to perfect health that agents start with.

An agent moves about in a closed environment, representing a building floor, until it detects a fire breakout in its vicinity. In that situation it tries to escape the environment, with its characteristic strategy. That happens when the agent exits through one of the doors that allow a passage between the closed environment and the external world. Each run may complete when one of the two following conditions is met: all agents have either escaped or died; a limit number of iterations is completed.

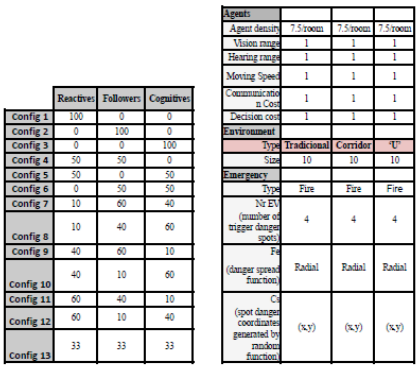
5.1 Dependent and Independent Variables

The number of variables is quite large and therefore we fixed most of the parameters to keep the experiments in a reasonable size. We varied the composition of the population from 100% of each type of agent to identical percentages for all types.We also varied the fire breakout locations, from a concentration in one single room to random breakouts, over a set of three types of environment configurations.

Three types of environments were defined. One in the form of a regular lattice of square rooms, in which one has communication doors between all adjacent rooms (Scenario Tradicional) and the other has only doors between rooms and corridor (Scenario Corridor). The other environment characterizes configurations where rooms are not uniformly distributed. The rooms form a kind of a U configuration (Scenario U).

In one set of experiments the fire breakout was on a fixed room, in a corner of the environment to test for the sensitivity of the model to the random positioning of agents, in a situation in which the fire could take more time to percolate through the whole environment. A subsequent set of experiments considered four fire breakouts in and around that same corner, to analyze the influence of small variations in fire breakout position. All other simulations used random positioning of one fire breakout. The initial density of agents was constant with the value of 7.5/room, over all the experiments.

Since one of the main goals of this work is to study the influence of the population composition inthe escaping results, we varied the initial percentage of different types of agents according to Figure .



	Reactives	Followers	Cognitives
Config 1	100	0	0
Config 2	0	100	0
Config 3	0	0	100
Config 4	50	50	0
Config 5	50	0	50
Config 6	0	50	50
Config 7	10	60	40
Config 8	10	40	60
Config 9	40	60	10
Config 10	40	10	60
Config 11	60	40	10
Config 12	60	10	40
Config 13	33	33	33

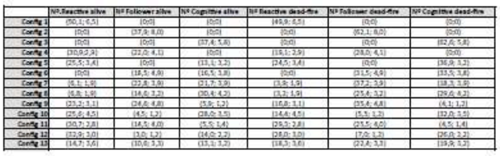
Agents			
Agent density	7.5/room	7.5/room	7.5/room
Vision range	1	1	1
Hearing range	1	1	1
Moving Speed	1	1	1
Communication Cost	1	1	1
Decision cost	1	1	1
Environment			
Type	Tradicional	Corridor	'U'
Size	10	10	10
Emergency			
Type	Fire	Fire	Fire
Nr EV (number of trigger danger spots)	4	4	4
Fe (danger spread function)	Radial	Radial	Radial
Cs (spot danger coordinates generated by random function)	(x,y)	(x,y)	(x,y)

Figure 3: Agent composition and parameters for experiments.

5.2 Results

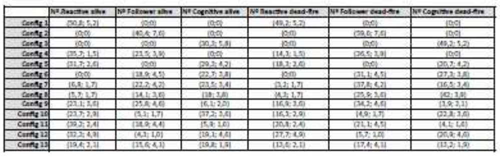
For each parameter configuration, 30 runs were made, with random initial agent positions, to obtain statistically significant results. Result of computing mean and standard deviation (;) are presented for each scenario . See Figure 4, 5, 6.

Following are the results of computing data acquired by the simulation. Graphics resulting of computing 13 different configurations let us conclude about the influence of the population composition in each type of scenario. We take into account the confidence interval drawn by the graphics. See Figures: 7, 8, 9. In an Airport scenario with “Tradicional” configuration, cognitive agents has a better performance, followed by reactive agents. There are less agents deads, so when there is the case of this kind of scenario, it would be good to invest on training people for this kind of situations. On the other hand, we can appreciate that for Scenario Corridor, reactive agents end the simulation with less deaths, and it can be explained by the distribution of the scenario. And for Scenario U, followers have a better disengagement.



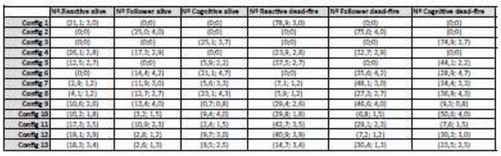
	Nº.Reactive alive	Nº Follower alive	Nº Cognitive alive	Nº Reactive dead-fire	Nº Follower dead-fire	Nº Cognitive dead-fire
Config 1	(50,1; 6,5)	(0;0)	(0;0)	(49,9; 6,5)	(0;0)	(0;0)
Config 2	(0;0)	(37,9; 8,0)	(0;0)	(0;0)	(62,1; 8,0)	(0;0)
Config 3	(0;0)	(0;0)	(37,4; 5,8)	(0;0)	(0;0)	(62,6; 5,8)
Config 4	(30,9;2,9)	(22,0; 4,1)	(0;0)	(19,1; 2,9)	(28,0; 4,1)	(0;0)
Config 5	(25,5; 3,4)	(0;0)	(13,1; 3,2)	(24,5; 3,4)	(0;0)	(36,9; 3,2)
Config 6	(0;0)	(18,5; 4,9)	(16,5; 3,8)	(0;0)	(31,5; 4,9)	(33,5; 3,8)
Config 7	(6,1; 1,9)	(22,8; 3,9)	(21,7; 3,9)	(3,9; 1,9)	(37,2; 3,9)	(18,3; 3,9)
Config 8	(6,8; 1,9)	(14,6; 3,2)	(30,4; 4,2)	(3,2; 1,9)	(25,4; 3,2)	(29,6; 4,2)
Config 9	(23,2; 3,1)	(24,6; 4,8)	(5,9; 1,2)	(16,8; 3,1)	(35,4; 4,8)	(4,1; 1,2)
Config 10	(25,6; 4,5)	(4,5; 1,2)	(28,0; 3,5)	(14,4; 4,5)	(5,5; 1,2)	(32,0; 3,5)
Config 11	(30,7; 2,8)	(14,5; 4,0)	(5,5; 1,4)	(29,3; 2,8)	(25,5; 4,0)	(4,5; 1,4)
Config 12	(32,9; 3,0)	(3,0; 1,2)	(14,0; 2,2)	(28,0; 3,0)	(7,0; 1,2)	(26,0; 2,2)
Config 13	(14,7; 3,6)	(10,6; 3,3)	(13,1; 3,2)	(18,3; 3,6)	(22,4; 3,3)	(19,9; 3,2)

Figure 4: Experiment results $(\mu;\sigma)$ for Scenario Traditional configuration.



	Nº.Reactive alive	Nº Follower alive	Nº Cognitive alive	Nº Reactive dead-fire	Nº Follower dead-fire	Nº Cognitive dead-fire
Config 1	(50,8; 5,2)	(0;0)	(0;0)	(49,2; 5,2)	(0;0)	(0;0)
Config 2	(0;0)	(40,4; 7,6)	(0;0)	(0;0)	(59,6; 7,6)	(0;0)
Config 3	(0;0)	(0;0)	(30,2; 5,8)	(0;0)	(0;0)	(49,2; 5,2)
Config 4	(35,7; 1,5)	(23,5; 3,9)	(0;0)	(14,3; 1,5)	(26,5; 3,9)	(0;0)
Config 5	(31,7; 2,6)	(0;0)	(29,3; 4,2)	(18,3; 2,6)	(0;0)	(20,7; 4,2)
Config 6	(0;0)	(18,9; 4,5)	(22,7; 2,8)	(0;0)	(31,1; 4,5)	(27,3; 2,8)
Config 7	(6,8; 1,7)	(22,2; 4,2)	(23,5; 3,4)	(3,2; 1,7)	(37,8; 4,2)	(16,5; 3,4)
Config 8	(5,7; 1,7)	(14,1; 3,6)	(18; 3,6)	(4,3; 1,7)	(25,9; 3,6)	(42; 3,8)
Config 9	(23,1; 3,6)	(25,8; 4,6)	(6,1; 2,0)	(16,9; 3,6)	(34,2; 4,6)	(3,9; 2,1)
Config 10	(23,7; 2,9)	(5,1; 1,7)	(37,2; 3,6)	(16,3; 2,9)	(4,9; 1,7)	(22,8; 3,6)
Config 11	(39,2; 2,4)	(18,9; 4,4)	(5,9; 1,6)	(20,8; 2,4)	(21,1; 4,5)	(4,1; 1,6)
Config 12	(32,3; 4,9)	(4,3; 1,0)	(19,1; 4,6)	(27,7; 4,9)	(5,7; 1,0)	(20,9; 4,6)
Config 13	(19,4; 2,1)	(15,6; 4,1)	(19,8; 1,9)	(13,6; 2,1)	(17,4; 4,1)	(13,2; 1,9)

Figure 5: Experiment results $(\mu;\sigma)$ for Scenario Corridor configuration.



	Nº.Reactive alive	Nº Follower alive	Nº Cognitive alive	Nº Reactive dead-fire	Nº Follower dead-fire	Nº Cognitive dead-fire
Config 1	(21,1; 3,0)	(0;0)	(0;0)	(78,9; 3,0)	(0;0)	(0;0)
Config 2	(0;0)	(25,0; 4,0)	(0;0)	(0;0)	(75,0; 4,0)	(0;0)
Config 3	(0;0)	(0;0)	(25,1; 3,7)	(0;0)	(0;0)	(74,9; 3,7)
Config 4	(26,1; 2,8)	(17,3; 2,9)	(0;0)	(23,9; 2,8)	(32,7; 2,9)	(0;0)
Config 5	(12,5; 2,7)	(0;0)	(5,9; 2,2)	(37,5; 2,7)	(0;0)	(44,1; 2,2)
Config 6	(0;0)	(14,4; 4,2)	(21,1; 4,7)	(0;0)	(35,6; 4,2)	(28,9; 4,7)
Config 7	(2,9; 1,2)	(11,9; 3,0)	(5,6; 3,3)	(7,1; 1,2)	(48,1; 3,0)	(34,4; 3,3)
Config 8	(4,1; 1,2)	(12,7; 2,7)	(23,1; 4,3)	(5,9; 1,2)	(27,3; 2,7)	(36,9; 4,3)
Config 9	(10,6; 2,6)	(13,4; 4,0)	(0,7; 0,8)	(29,4; 2,6)	(46,6; 4,0)	(9,3; 0,8)
Config 10	(10,2; 1,8)	(3,2; 1,5)	(9,4; 4,0)	(29,8; 1,8)	(6,8; 1,5)	(50,6; 4,0)
Config 11	(17,3; 3,5)	(10,9; 2,3)	(2,4; 1,5)	(42,7; 3,5)	(29,1; 2,3)	(7,6; 1,5)
Config 12	(19,1; 3,9)	(2,8; 1,2)	(9,7; 3,0)	(40,9; 3,9)	(7,2; 1,2)	(30,3; 3,0)
Config 13	(18,3; 3,4)	(2,6; 1,3)	(9,5; 2,5)	(14,7; 3,4)	(30,4; 1,3)	(23,5; 2,5)

Figure 6: Experiment results $(\mu;\sigma)$ for Scenario U configuration.

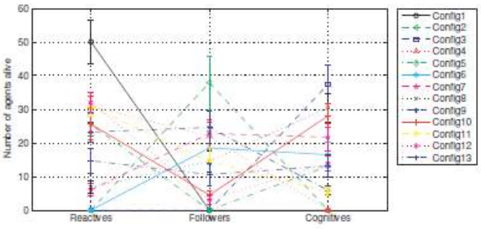


Figure 7: Agents behaviour- Scenario Tradicional

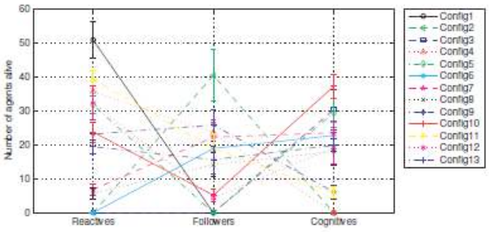


Figure 8: Agents behaviour-Scenario Corridor

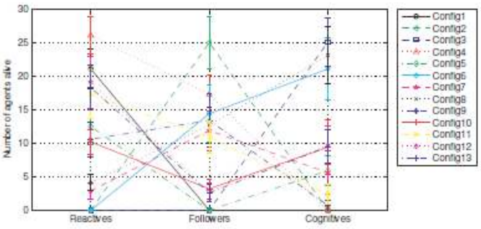


Figure 9: Agents behaviour- Scenario U

6 Conclusions

The simulation results help us to establish the direct relation between population of the crowd and type of scenarios. It also provides us with an estimate for an Airport scenario of where we should put the effort depending on the number of people for every scenario considered. Experiments and graphics resulting of computing 13 different configurations let us conclude about the influence of the population composition in each type of scenario. It allows us to decide where to put the efforts, i.e: investing money and time training people for these emergency situations; or putting effort choosing an adequate scenario for the specific domain of application.

8 References

[ANDERSON, G. *et al.* 1985] ANDERSON, G., and Moore, G. 1985. A linear algebraic procedure for solving linear perfect foresight models. Eco-nomics letters 17(3):247–252. Spain

[BRATMAN, M. 1987] BRATMAN, M., and Intentions, P. Practical reason. 1987.

[BRAUN, A. *et al.* 2003] BRAUN, A.; Musse, S.; de Oliveira, L.; and Bodmann, B. 2003. Modeling individual behaviors in crowd simulation. In Computer Animation and Social Agents, 2003. 16th International Conference on, 143–148. IEEE.

[DAVIDSSON, P. 2002] DAVIDSSON, P. Agent based social simulation: A computer science view. Journal of artificial societies and social simulation 5(1). 2002.

[FILIPPOUPOLITIS, A. *et al.* 2002] FILIPPOUPOLITIS, A.; Hey, L.; Loukas, G.; Gelenbe, E.; and Timotheou, S. Emergency response simulation using wireless sensor networks. In Proceedings of the 1st international conference on Ambient media and systems, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering). 2008.

[KINNY, D., *et al.* 2002] KINNY, D., and George, M. Commitment and e effectiveness of situated agents. In Proceedings of the twelfth international joint conference on artificial intelligence (IJCAI-91), 82–88. 1991

[NEWELL, A., *et al.* 1961] NEWELL, A., and Simon, H. 1961. Computer simulation of human thinking. Rand Corp.

[NEWELL, A., *et al.* 1972] NEWELL, A., and Simon, H. 1972. Human information processing. Annu. Rev. Psychol. l974 25.

[PAN, X., *et al.* 2007] PAN, X.; Han, C.; Dauber, K.; and Law, K. A multiagent based framework for the simulation of human and social behaviors during emergency evacuations. AI & Society 22(2):113–132. 2007.

[RAO, A., *et al.* 1995] [ZHAN, RAO, A.; Georgeff, M.; et al. 1995. Bdi agents: From theory to practice. In Proceedings of the first international conference on multi-agent systems (ICMAS-95), 312–319. San Francisco. 1995.

A., *et al.* 2008] Zhan, B.; Monekosso, D.; Remagnino, P.; Velastin, S.; and Xu, L. Crowd analysis: a survey. Machine Vision and Applications 19(5):345–357. 2008.