



HAL
open science

Impact of group on the evacuation process - Theory and simulation

Xuan Hien Ta, Dominique Longin, Benoit Gaudou, Tuong Vinh Ho

► To cite this version:

Xuan Hien Ta, Dominique Longin, Benoit Gaudou, Tuong Vinh Ho. Impact of group on the evacuation process - Theory and simulation. Symposium on Information and Communication Technology (SoICT 2015), Dec 2015, Hue, Vietnam. pp.350-357, 10.1145/2833258.2833265 . hal-03482611

HAL Id: hal-03482611

<https://ut3-toulouseinp.hal.science/hal-03482611>

Submitted on 16 Dec 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Impact of group on the evacuation process: theory and simulation

Xuan Hien Ta
Toulouse University, UPS-IRIT
118 Route de Narbonne
F-31062 Toulouse CEDEX 9,
France
hientpbk@gmail.com

Dominique Longin
Toulouse University,
CNRS-IRIT
118 Route de Narbonne
F-31062 Toulouse CEDEX 9,
France
dominique.longin@irit.fr

Benoit Gaudou
Toulouse University, UTC-IRIT
2 Rue du Doyen-Gabriel-Marty
31042 Toulouse, France
benoit.gaudou@gmail.com

Tuong Vinh Ho
Vietnam National University
Dai lo Thang Long, Hoa Lac,
Thach That
Ha Noi, Vietnam
ho.tuong.vinh@gmail.com

ABSTRACT

Our main purpose is to build a simulation describing the persons movements in a public area during a crisis situation. Our running example is about the evacuation process of a supermarket during a fire. The main difficulty of such works comes from the fact that it is generally impossible to obtain precise descriptions of persons behaviors. After several works on individual emotions we are convinced that emotion is well adapted to explain such actions in a situation crisis. In the aim to test our hypothesis, we present here a first simulation (without any emotion management) that is a first step of our main purpose. This simulation aims to describe the well-known fact that, in crisis situation, humans tend toward to help each others. Thus, we test the impact of group constitution on survival rate and on average time. The simulation has been implemented with the platform GAMA.

CCS Concepts

•Computing methodologies → Agent / discrete models;

Keywords

MAS; GAMA platform; group; crisis situation; evacuation process; simulation

1. INTRODUCTION

Simulations are interesting for at least two reasons.

Publication rights licensed to ACM. ACM acknowledges that this contribution was authored or co-authored by an employee, contractor or affiliate of a national government. As such, the Government retains a nonexclusive, royalty-free right to publish or reproduce this article, or to allow others to do so, for Government purposes only.

SoICT 2015, December 03 - 04, 2015, Hue City, Viet Nam

ACM ISBN 978-1-4503-3843-1/15/12...\$15.00

DOI: <http://dx.doi.org/10.1145/2833258.2833265>

Firstly, they allow to represent real situations and their dynamics with respect time as, for instance: population movements among a country, rises in river levels in a given region, dynamics of forests, *etc.* To know how to describe in a computer model such situations supposes that the computer model is based on a *possible* mechanism describing the dynamics of these situations with respect time. In other words, simulations allow to validate (when the results of these simulations are in accordance with the results of real situations) a given model as being a plausible model of both those situations and their dynamics.

Secondly, when a simulation gives results that are in accordance with a given real situation, it allows us to make some changes in the simulation in the aim to observe what could be the resulting real situation if these changes were made in the real world. For instance, in a simulation of rises in river levels, what could be in the real world the effects of building a sea wall here or there? In other words, simulations provide a helpful tools allowing to anticipate effects of actions in the real world.

Our main purpose is to build a simulation describing the persons movements in a public area during a crisis situation. Our running example is about the evacuation process of a supermarket during a fire. The main difficulty of such works comes from the fact that it is generally impossible to obtain precise descriptions of persons behaviors. Thus, we cannot build a simulation in a classical way (that is, from precise data that validate the simulation results) and we need to find some paradigms in the aim to explain the actions of persons that are trying to escape from the crisis area. After several works on individual emotions [11, 3, 1, 10, 5, 2] we are convinced that emotion is well adapted to explain such actions in a situation crisis.

In the aim to test our hypothesis, we present here a first simulation (without any emotion management) that is a first step of our main purpose. This simulation aims to describe the well-known fact that, in crisis situation, humans tend toward to help each others [7, 6, 4]. An interesting side of this phenomena is that a lot of humans group together and help each other (see Section 2 for more details). We present

in the following this first simulation and its main results. The rest of this article is organized as follows.

In Section 2 we explain what are humans actions in crisis situations. We also talk about some related works, in particular about humans behavior during a crisis. In Section 3, we present a description of our current simulation (initial parameters and their dynamics). In Section 4, the results of the simulation are given and analyzed. We then talk about future works and conclude.

2. RELATED WORKS

There are a lot of researches about people behaviors in disasters [15]. In such crisis situations, people want to gather together although they don't know each other before the crisis. Or the people in the group try to search for another member in the group before exiting. They try to share information or help each other to move out of the dangerous region.

A lot of studies (see [8] for instance) show that when the danger increases, mutual aid between humans exposed to this danger also increases. The persons share emotions and information, and they help each other, even if they were strangers each other just before the crisis. There are a very few cases of selfish behaviors. One of the faces of this mutual aid is the constitution of groups of persons. People in a group of friends or in a family try to stay together each time it is possible. Sociological studies show that groups increase our chances to be saved [6] (evolutionary condition).

Such studies show also that there is not mass panic and that crowds behaviors are meaningful: people in emergency situations try to have the same actions as others (coping mechanism) rather than being passive victims.

In the disaster of September 11, 2001 at World Trade Center in American, a research based on a reading of 745 accounts coming from the 435 survivors shows that the evacuation process happened: with calm and order behavior for a majority of people (57%); with cries and anxiety or nervousness (31%); with pushes and shoves (29%). Moreover, 22% of people found have been helped by others.

To understand more about group behaviors during evacuations, authors in [12] created a simulation and applied the prototype of SAFEgress to estimate the influence of group behaviors in a real case of disaster, a fire in a nightclub in 2003. This fire caused 100 deaths on 452 people in the club. The result of this simulation showed that the evacuation time of group is higher than that one of individual (256s compared to 167s). The duration of evacuation time varies non-linearly. And the behaviors group (stay with or searching for other group members) influence on the choosing of exit door. The people in group choose the regular exit (front entrance, bar exit) more than individual people that have used for instance the available windows to exit. The authors use this argument to demonstrate that agents belonging to a group run away together. The simulation also demonstrates that evacuation time for 352 single persons (that are not belonging in a group) is less than evacuation time of the same number of persons belonging in groups.

Authors in [12] continue with the interesting article [13] after one year. This article gives more details about the behavior of agents in crisis situation. This simulation is still based on the simulation tool SAFEgress with 3 behavioral models (individual, group and crowd) [14]. There are many static attributes corresponding with 3 levels of

agents. For instance, with individual agent, there are physical profile attributes (age, gender, body size, travel speed, personal space), familiarity and knowledge of exit. With group, there are group intimacy level (this value is high for family group), group-seeking, group leader(s), group influence (level influence of member to the others in the group). Crowd has 2 attributes: social order (the exhibit deference behavior) and assigned roles (responsible for executing action like share information). For updating the behavior, an agent pass through 5 stages process: perception, interpretation, decision making, execution and memorization. This agent can base on the knowledge of building, visual perception and social cue to detect the way to go out. A lot of social behaviors are implemented: following building exit signs, following familiar exit (query the new shortest path to known exits), navigating with social group (group seeking), following crowds (follow the way with highest number of neighbors move on that). The result of this simulation demonstrates the significance of knowledge, social behaviors and incorporating perception of occupant in the simulation of disaster.

3. MODEL DESCRIPTION

We present in the following a description of our simulation that have been implemented with the GAMA¹ platform [9]. GAMA is a (open-source) generic agent-based modeling and simulation platform. It provides a lot of powerful tools to develop easily models in particular using geographical data. We use this feature in the model to define the environment. In addition, GAMA allows the modeler to run simulation in either an interactive or a batch mode. This will allow us to launch experiment design in order to explore the model.

In GAMA, the basic assumption to build a model is to consider any entity involved in the model as an agent. The GAMA meta-model does not consider specific entities such as actors, passive entities or resources. So every humans, walls, doors and so on will be modeled and implemented as an agent. This provides a coherent framework for all the entities in the model.

3.1 Environment description

Let AGT be the set of all the agents used in the simulation. $ENV \subseteq AGT$ is the set of all the agents used to describe the environment. $ENV = DOORS \cup WALLS \cup OBSTACLES$. $DOORS = \{door_1, door_2, \dots\}$ is the finite set of doors; $WALLS = \{wall_1, wall_2, \dots\}$ is the finite set of walls; $OBSTACLES = \{obstacle_1, obstacle_2, \dots\}$ is the finite set of obstacles (such as supermarket departments for instance) that can be avoided by agents. The different dimensions of the components of the environment are represented with respect to a dimension scale κ_d : the length l^R in the real world is represented by the length l in the simulation, and $l = \kappa_d \cdot l^R$.

Similarly, for every $n \in \mathbb{N}$, each time point in the real world t_n^R is associated with a time point t_n in the simulation. Suppose that two successive steps happen respectively at t_{n-1} and at t_n , then $\Delta t_n = t_n - t_{n-1}$ is the time used by the simulation for going from the step $n - 1$ to the step n . But we can also say that Δt_n corresponds to some time in the real world (noted Δt_n^R): it is what we call the *simulated real time unit* and $\Delta t_n = \kappa_t \cdot \Delta t_n^R$ where κ_t is the time scale.

¹<http://gama-platform.org>

As we aim to study the influence of group in evacuation process, we do not add fires yet. We just suppose that, at the beginning of the simulation, human agents must find an exit and go out of the supermarket and they just have, for doing that, the simulated real time quantity T_{max}^R (that is equal to the time used by n simulation steps, where n is an integer). We note $T_{max} = \kappa_t \cdot T_{max}^R$. When T_{max}^R is passed, we stop the simulation and consider that the humans remaining in the supermarket are dead (the fire has totally burned it).

Let $n \in \mathbb{N}$ an integer describing the total number of simulation steps. The simulation steps sequence can be formalized as follows:

$$s_0 \rightarrow s_1 \rightarrow \dots \rightarrow s_{n-1} \rightarrow s_n \quad (1)$$

where s_n is the simulation state after the execution of n steps, s_0 is the initial state, s_n is the final state. We suppose that each step s_n happens at the simulated real time t_n^R , and that for every $n \in \{1, \dots, n-1\}$, $\Delta t_n^R = \Delta t_{n+1}^R = \Delta t^R$. So, for every $n \in \{0, 1, \dots, n\}$:

$$t_n^R - t_0^R = n \cdot \Delta t^R \quad (2)$$

Similarly, we have:

$$t_n - t_0 = n \cdot \Delta t \quad (3)$$

A corollary is that $T_{max}^R = n \cdot \Delta t^R$ and $T_{max} = n \cdot \Delta t$.

In the following, we only use dimension and time point of the simulation. When we refer to simulated real dimension or time point, we will explicitly specify it.

3.2 Humans description

Let HUM be the set of human agents such that $HUM \subseteq AGT$. In the following, i, j, k are three human agents of HUM . Each agent $i \in HUM$ is associated with a 8-tuple $State_i^t = \langle spd_i^t, spd_i^{norm}, spd_i^{max}, \rho_i^t, \rho_i^{max}, dist_i^t, D_i^t, CD_i^t \rangle$. $State_i^t$ describes the state of agent i at time t . So, $State_i^{t_0}$ is the initial state of agent i . The components of $State_i^t$ that do not depend on time are constants (their value does not change during the execution of the simulation).

- spd_i^t is the current walking speed of agent i at time t , spd_i^{norm} is the normal walking speed of agent i when this agent is walking, spd_i^{max} is its maximal walking speed (agent i cannot walk faster), and we have both $0 \leq spd_i^{t_n} \leq spd_i^{max}$ for every $n \in \{0, 1, \dots, n\}$ and $0 \leq spd_i^{norm} \leq spd_i^{max}$.

- $spd_i^{t_0} = 0$ means that agent i does not walk at time t . $spd_i^{t_0}$ is randomly initialized (the real speed limit has been fixed to 10 km/h). (Note that spd_i^{norm} is constant and is used for giving a walking speed to the agent when this agent has previously walked slower or faster.)

- ρ_i^t is the visual field radius of agent i at time t . So, at time t , agent i can only perceive the objects (walls, obstacles) and the persons that are at a distance d such that $0 \leq d \leq \rho_i^t$. In normal situations, $\rho_i^t = \rho_i^{max}$. But it is possible that the visual field radius of agent i is lower than ρ_i^{max} , due to some obstacles or smokes for instance. So, for every $n \in \{0, 1, \dots, n\}$ we have $0 \leq \rho_i^{t_n} \leq \rho_i^{max}$.

- The maximal distance covered by agent i at time t_n is $spd_i^{t_n} \cdot \Delta t_n$. But as an agent may meet with obstacles, the distance $dist_i^{t_n}$ that it has really covered may be different, and $0 \leq dist_i^{t_n} \leq spd_i^{t_n} \cdot \Delta t_n$. More precisely, if there is an obstacle on the way followed by agent i , and if the distance between i and this obstacle is d , then $dist_i^{t_n} = \min(d, spd_i^{t_n} \cdot \Delta t_n)$.

- D_i^t is the set of exit doors known by agent i at time t . During the simulation execution, it is possible that agent i discovers some other doors and then, $D_i^{t_0} \subseteq D_i^{t_n} \subseteq DOORS$.

- As soon as agent i learns that a known door in D_i^t is closed, this door is added to the set $CD_i^t \subseteq DOORS$ (the set of closed exit doors known by agent i at time t). Conversely, each times that agent i detects that a door has been opened, this door is removed from CD_i^t if the agent believed that this door was closed.

Note that agent i believes that each door in $OD_i^t = D_i^t \setminus CD_i^t$ is open although it may not be the case because at time t , agent i may be unaware of all the open doors. Moreover, if for some reasons an exit door has been blocked, an agent can search another doors.

The above properties apply for every human agent. We now distinguish between two kinds of human agents: leaders and customers. Let $LDR \subseteq HUM$ be the set of human leaders and let $CUST \subseteq HUM$ be the set of customers. These both sets of humans as initially built as follows:

$$LDR \cap CUST = \emptyset \quad (4)$$

$$LDR \cup CUST = HUM \quad (5)$$

In other words, an agent i is a leader (respectively, a customer) if and only if $i \in LDR$ (respectively, $i \in CUST$).

Leaders are human agents with special knowledge. They are typically security agents that know both all the existing doors of the supermarket and how each door can be reached with a minimal time. Conversely, the set of exit doors known by a customer may be empty, it may contain some doors, or every doors (if this customer knows the supermarket very well or if the customer has discovered some doors during the execution of the simulation). Leaders can be heads of customers groups (but customers cannot).

- In crisis situation, the leaders can guide the customers towards the exit doors. If and only if agent i is a leader, we extend $State_i^t$ with the set $FOLL_i^t \subseteq CUST$ that contains every agent that follows i . We define $LState_i^t$ (the state of the leader i at time t) as the extension of $State_i^t$ by the set $FOLL_i^t$.

- If and only if agent i is a customer, we extend $State_i^t$ with the agent $leader_i^t \in LDR$ (the leader that agent i follows at time t). When $leader_i^t$ is undefined we write that $leader_i^t = \emptyset$.

Note that initially, $CD_i^{t_0} = \emptyset$ for customers, but for leaders, it possible that $CD_i^{t_0}$ is not empty.

3.3 Groups description

As we have said, a group is defined by a leader and a set of agents. Formally, for every $i \in LDR$:

$$Group_i^t = FOLL_i^t \cup \{i\} \quad (6)$$

is the group of agents at time t with the leader i . By definition, $i \in Group_i^t$ and then, $Group_i^t \neq \emptyset$.

It follows from above definitions that for every $i \in CUST$:

$$leader_i^t \neq \emptyset \text{ if and only if } i \in Group_{leader_i^t}^t \quad (7)$$

In other words, the customer i has a leader at time t if and only if i is a member of the group led by this leader at time t .

We have defined above several walking speeds for each agent, but what is the walking speed of a group? (Generally, the walking of each agent of a group is different.) We

consider here that the “group effect” entails both that some persons will walk faster than their current walking speed and that some other persons will walk slower than their current walking speed. An explanation is that the slowest persons will be helped and motivated by others. Conversely, the fastest persons will spend time to help the slowest persons. So, we define the group walking speed $spd_{Group_i}^t$ of a group $Group_i^t$ at time t as an average speed:

$$spd_{Group_i}^t = \frac{\sum_{j \in Group_i^t} spd_j^t}{\#(Group_i^t)} \quad (8)$$

where $\#(Group_i^t)$ is the number of agents in $Group_i^t$. Note that the group can evolve during the evacuation process (agents can join or leave the group), and then, the walking speed of a group can also evolve.

Now, we could suppose that the current walking speed of agents that are member of a group is the group walking speed. But what happens if we suppose that this speed is greater than the maximal speed of some agents? We adopt the following heuristics. For every $i \in LDR$:

$$spd_i^t = \begin{cases} spd_i^{norm} & \text{if } leader_i^t = \emptyset \\ \min(spd_i^{max}, spd_{Group_i^t}^t) & \text{else} \end{cases} \quad (9)$$

and for every $i \in CUST$:

$$spd_i^t = \begin{cases} spd_i^{norm} & \text{if } leader_i^t = \emptyset \\ \min(spd_i^{max}, spd_{Group_{leader_i^t}}^t) & \text{else} \end{cases} \quad (10)$$

On one hand, it follows from these definition that some agents may walk slower than other members of the group and then, these slowest agents can lost the group (they are removed from this group). On the other hand, the faster an agent walks, the more it will probably be a survival. An important consequence of that is to find the “good” group walking speed, that is, the speed that will allow to save the most of humans as possible. This quest is at the core of this work: we have tested a lot of group walking speed by using $\lambda \cdot spd_{Group_i}^t$ instead of $spd_{Group_i}^t$ for $\lambda \in [0.0, 2.0]$. (See below for the results.)

3.4 Obstacle avoidance algorithm description

As said above, agent can avoid obstacle when they meet one. The algorithm to avoid obstacle is described in Figure 1 and Figure 2.

The main idea is as follows. We define $\vec{V}_{obstacle}$ by the two following points: the *current position* of the agent and the *intersection position* (that is defined by the intersection between the current path followed by the agent towards the target, and the obstacle).

So, we are now able to define \vec{V}_{normal} . This vector is defined from the *current position* and from a point defined by: the fact that the angle between this vector and $\vec{V}_{obstacle}$ is equal to 90 degrees; and the fact that its norm is equal to a fixed value (the same for all the human agents).

We can then define $\vec{V}_{direction}$ as the vector composition of $\vec{V}_{obstacle}$ and \vec{V}_{normal} . The *guess position* depends on the speed of each agent.

In Figure 1 the guess position has no intersection with the obstacle and then the next position of the agent is the guess position. But it is shown in Figure 2, it is possible

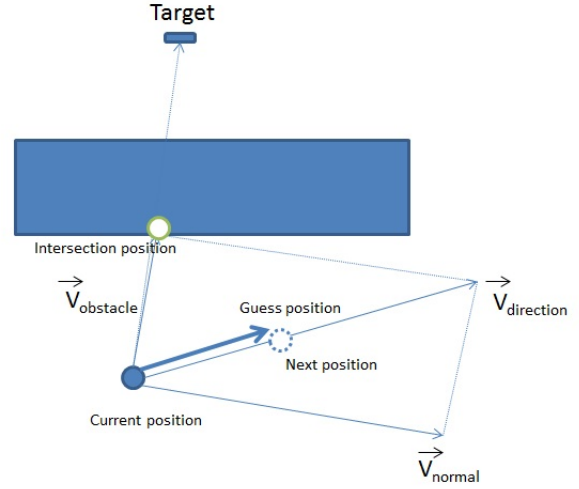


Figure 1: Obstacle avoidance algorithm in case guess position has no intersection with the obstacle.

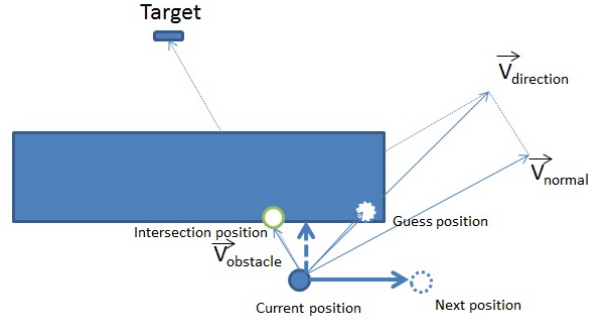


Figure 2: Obstacle avoidance algorithm in case guess position has an intersection with the obstacle.

that the guess position intersects with the obstacle again. In this case, we arbitrarily choose $\vec{V}_{direction}$ as being parallel with the obstacle.

There certainly exists more finest algorithm of obstacle avoidance. But in this simulation, both the number of agent and the number of obstacle can be great. So, the number of avoidance computing can greatly increases and we must solve all the conflictual paths in real time. So, we think this algorithm is a good compromise solution.

3.5 Evacuation algorithm description

In crisis situations, leaders and customers will try to reach an exit door. The leaders that have more information about both the supermarket and its current situation (for instance: where is the fire, what could be the best direction to go out of the supermarket, etc.) will be the first out. With the help of a built-in GAMA algorithm, they can easily move by avoiding obstacles and reach exit doors.

Because their knowledge is both more unclear and more incomplete than the knowledge of leaders, customers have less probability to find an exist door before the fire deadline.

So, their action policy is more complex. A simple evacuation algorithm for customers is described below:

```

perceivedDoors = doorsDetect ();
if (!empty(perceivedDoors)) {
    target = closestDoor(perceivedDoors);
} else {
    perceivedLeaders = leadersDetect ();
    if (!empty(perceivedLeaders)) {
        target = closestLeader(perceivedLeaders);
    } else {
        if !empty(knownDoors) {
            target = closestDoor(knownDoors);
        } else {
            if (isAlongWall) {
                target = nextStepAlongWall ();
            } else {
                if (!isTargetWall()) {
                    perceivedWalls = wallsDetect ();
                    if (!empty(perceivedWalls)) {
                        target = closestWall(perceivedWalls);
                        detectFirstCornerOnTheWall ();
                    } else {
                        target = nextStep ();
                    }
                }
            }
        }
    }
}

isAvoidObstacle = checkToAvoidObstacle ();
if (isAvoidObstacle) {
    nextPosition = getPositionAvoidObstacle ();
} else {
    nextPosition = target;
}

```

```

// move
goTo(nextPosition) ;

```

Note that this algorithm is for customers only. Some methods (such as `doorsDetect()` or `leadersDetect()` for instance) are related to the visual field radius of each agent i at time t (ρ_i^t). So, detection process may be different for two customer staying at the same place.

GAMA supports many useful functions to implement this algorithm like function `getClosestAgent` from a list or `getClosestPointOnObstacle`. The code in the function `reflex` will be executed each step of simulation. Here is the simple code for evacuation of customer in GAMA.

```

reflex detectTarget {
    target <- detectClosestDoor ();
    if (target != nil) {
        targetLocation <- closestPointObj(self, target);
        do quitFromSuperMarket;

        if (currentLeader != nil) {
            do removeFromGroup;
        }
    } else {
        target <- detectClosestLeader ();
        if (target != nil) {
            targetLocation <- target.location;

            // In case change closest leader
            if (currentLeader != target) {
                if (currentLeader != nil) {
                    do removeFromGroup;
                }
                do joinInGroup;
            }
        }
    }
}

```

```

} else {
    if (isInGroup) {
        do removeFromGroup;
    }

    if (length(knownExitDoorList) > 0) {
        target <- closestKnownExit ();
        targetLocation <- target.location;
    } else {
        if (isAlongWall) {
            targetLocation <- nextStepAlongWall ();
        } else {
            target <- detectClosestWall ();
            if (target != nil) {
                isDetectWall <- true;
                targetLocation <- listClosestPoint ();
                do detectFirstCornerWall;
            } else {
                targetLocation <- nextStep ();
            }
        }
    }
}

isAvoidObstacle <- checkToAvoidObstacle ();
if (isAvoidObstacle) {
    nextPosition <- getPositionAvoidObstacle ();
} else {
    nextPosition <- targetLocation;
}

do goto target:nextPosition speed:speedCustomer;
}

```

3.6 Additional simulation parameters

As said above, the simulation needs to use some additional parameters such as the sets of agents, customers and leaders, the different scales, *etc.* (see above) or the norm of \vec{V}_{normal} for instance (see Section 3.4). Moreover, we have defined the following parameters in order to explore their impact on the results of the simulation.

- n is the total steps number of the simulation. The default value is 400.

- `spdGroupEnabled` has a Boolean value enabling or disabling the fact that walking speed of group members is set by the group walking speed (see Principle (10)). A false value means that every agent walk with its own current speed, even if it is a member of a group. The default value is true.

- As said at the end of Section 3.3, $\lambda \in [0.0, 2.0]$ allows us to control how much the group walking speed influences the current walking speed of persons. $\lambda = 1$ (the default value) means that the individual speed is 100% of the group speed.

- $\#(LDR)$ and $\#(CUST)$ are the number of leaders and of customers respectively. The default value of $\#(CUST)$ is 100. Generally, $\#(LDR)$ has several values during a test.

- r is the ratio of customers knowing all the doors. So, if agent i is one on these agents then $D_i^{t_0} = DOORS$; else $D_i^{t_0} = \emptyset$. The default value is 0.2.

Moreover, the algorithm described in Section 3.5 can be modified with the three following parameters:

- `leadersEnabled` has a Boolean value enabling or disabling the fact that customers follow leaders. The default value is true.

- `wallsEnabled` has a Boolean value enabling or disabling the fact that, when a customer is close to a wall, this agent will follow this wall (if neither door nor leader are detected).

The default value is true.

- As it has been explained in Section 3.5, when an agent does not detect anything (door, leader, or wall) this agent chooses a random direction and walk towards this direction as long as possible. *keepDirectionEnable* has a Boolean value enabling or disabling the fact that this direction is kept as long as possible or not. When this parameter is false, the agent randomly chooses a new direction at each step of the simulation. The default value is true.

4. SCENARIOS AND RESULTS

In the following, we fix the value of some parameters and give the results that we analyze. The follows from 20 to 200 iterations of the simulation for a given set of parameters.

Influence of the three algorithm parameters.

Here, we want to test the effects on the survival rate, of some changes in the behavior algorithm of the customers and of some leaders numbers.

The eight different scenarios that we have tested are generated by enabling (\top value) or disabling (\perp value) the three algorithm parameters (*leadersEnabled*, *wallsEnabled*, and *keepDirectionEnable*); see Table 1.

Table 1: Case for testing algorithm

Algorithm number	1	2	3	4	5	6	7	8
<i>leadersEnabled</i>	\perp	\perp	\perp	\perp	\top	\top	\top	\top
<i>wallsEnabled</i>	\perp	\perp	\top	\top	\perp	\perp	\top	\top
<i>keepDirectionEnable</i>	\perp	\top	\perp	\top	\perp	\top	\perp	\top

For each scenario number (1 to 8), $\#(LDR) \in \{0, 4, 8\}$ for a total of 24 scenarios. (The other parameters have their default value.) In all cases, the survivals rate is presented in Figure 3.

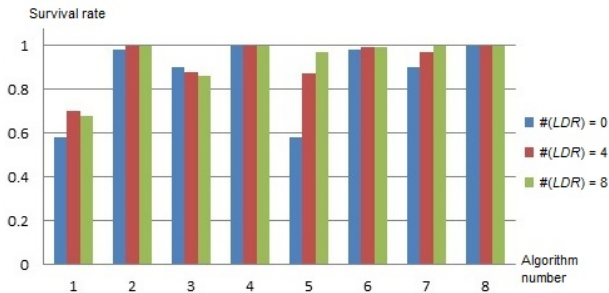


Figure 3: Relation between algorithm parameters, leaders number, and the survivals rate.

Following the results presented in Figure 3 we can observe that scenarios 2, 4, 6, and 8 give the better results. In all this scenarios, *keepDirectionEnable* is true. So, randomly walking decreases the chances of agent to be saved.

Moreover, if agents do not follows leaders (scenario 1 to 4), to increase the leaders number does not proportionally increase the rate of survival, which it is not the case in scenarios 5 to 8.

Finally, both *wallsEnabled* and *keepDirectionEnable* are true (scenarios 4 and 8) the survivals rate is maximal.

Ratio of initially known doors and leader number.

Here: $\#(LDR) \in \{0, 2, 4, 6, 8, 10\}$; for every $i \in CUST$, $r \in \{0, 0.2, 0.4, 0.6, 0.8, 1.0\}$. So, there are different 36 scenarios. In all cases, the ratio of survival is almost 100%, but the average time to exit of customers is different. The results are presented in Figure 4.

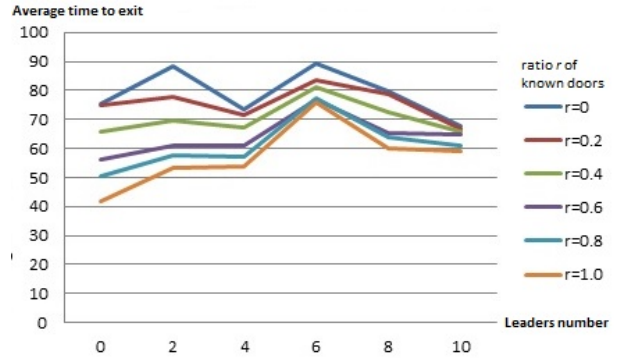


Figure 4: Impact of known doors ratio and leaders number on the average time to exit.

When *doorsRatio_i* increases, the average time to exit decreases. Moreover, when $\#(LDR)$ increases, the average time to exit does not necessarily decreases, and in some cases it increases. It mean that the value of *doorsRatio_i* is more important to exit than $\#(LDR)$.

Study of the group walking speed influence.

We analyse first the influence of the group walking speed on the survival rate. So, $\lambda \in \{0.4, 0.8, 1.0, 1.2, 1.6\}$ and $\#(LDR) \in \{0, 2, 4, 6, 8, 10\}$; $n = 50$. The other parameters have their default value. (There 30 different scenarios.) Results are given in Figure 5.



Figure 5: Survival rate according to the group speed and the leaders number.

The main result of this scenario is that, when the group walking speed is too slow ($\lambda < 1.0$) the survival rate decreases. We explain this result by the fact that, when $\lambda < 1.0$, the current walking speed of customers is slower than their normal walking speed, and then it takes more time to evacuate the area, and then the survival rate decreases. In other cases ($\lambda \geq 1.0$), the more there are leaders, the more the survival rate increases. In other words, it is better for the

survival rate to be alone rather than to be in a slow-moving group.

Number of customers.

Here we analyze how the number of customers may influence the average time to exit. $\#(LDR) \in \{0, 2, 4, 6, 8, 10\}$ and $\#(CUST) \in \{50, 100, 200, 400, 800, 1600\}$. Here, the survival rate is always equal to 1. See Figure 6 for the results. It seems that the average rate to exit should not be influenced by the number of customers.

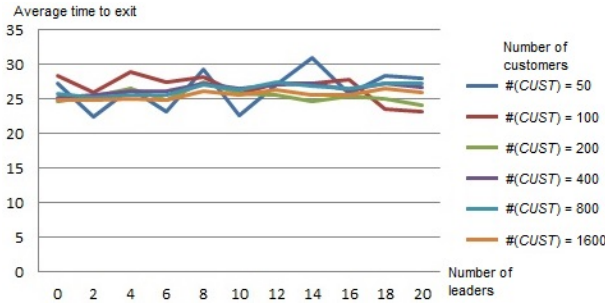


Figure 6: Influence of the customers number on the average time to exit.

Influence of the visual field radius.

In crisis situations, the visual field radius can greatly fluctuate (obstacles, lack of light, fire smoke, etc.). In the following, we analyze how the visual field radius influences the survival rate and the average time to exit. We consider that the visual field radius is the same for all the humans, and that it does not change during the simulation: $\rho_i^t = \rho_j^t$ for $i, j \in HUM$, and $\rho_i^t = \rho_i^{t-1}$ for every $t \in \{1, 2, \dots, n\}$. We note ρ this unique visual radius.

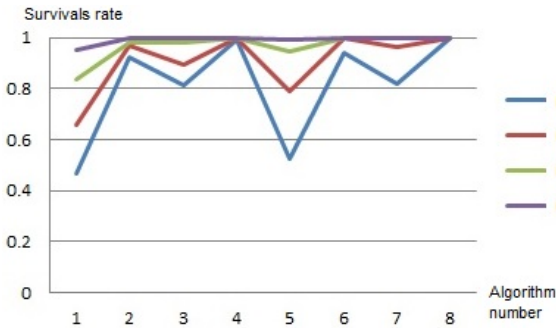


Figure 7: Influence of the algorithm type and of the visual field radius on the survival rate.

In the first test, we analyze the effects on the survival rate of both the eight changes in the customer algorithm (see Table 1) and several values for the visual field radius: $\rho \in \{15, 30, 45, 60\}$ (with respect to the internal units of length of the simulation). The results are presented in Figure 7. The survival rate increases when the visual field radius increases. The worst results are when the visual field radius is minimal and the algorithm number is 1 or 5 (that

is, when customers that do not detect anything have a random direction at each step of the simulation and that they do not go along the walls). But when the visual field radius increases, the negative effects of algorithm changes decrease, because the detection of doors and of leaders increases with visual field radius.

In the second test, we analyze the influence of the same parameters on the average time to exit (see Figure 8).

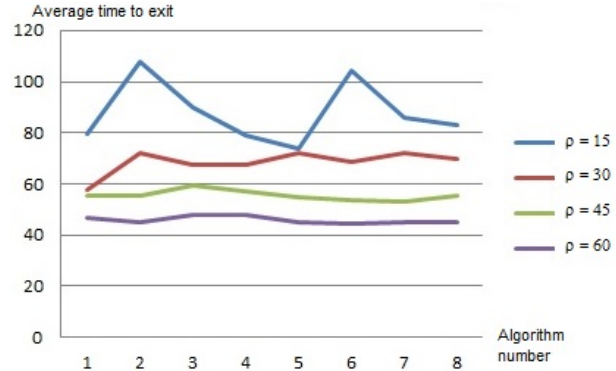


Figure 8: Influence of the algorithm type and of the visual field radius on the average time to exit.

As expected, the average time to exit decreases when the visual field radius increases.

Finally, the last test concerns the influence of both several visual field radii and several numbers of customers, on the average time to exit. (See Figure 9.)

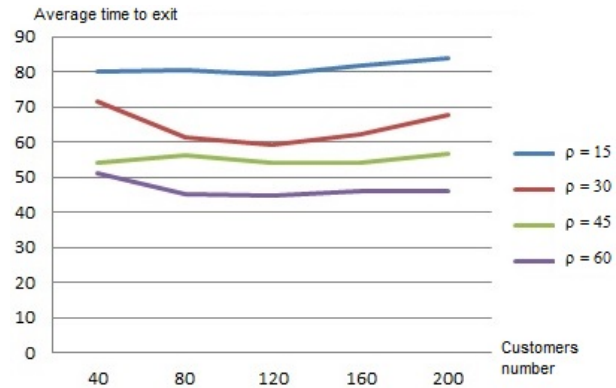


Figure 9: Influence of the number of customers and of the visual field radius on the average time to exit.

Here again, as expected, the average time to exit decreases when the visual field radius increases. Note that the customer number does influence significantly the results.

5. CONCLUSION AND FUTURE WORKS

In this paper, we have presented the first results of our works on evacuation process in crisis situation. We built a simulation that should be as realistic as possible. It concerns in particular a model based on real distances (size of the supermarket, visual field radius, dimensions of obstacles, walking speeds, etc.).

We evaluated survival rate and average time of exit of customers in several situations. These situations fit with different sets of parameters values: some changes in the algorithm of evacuation, the management of groups of customers by leaders, the impact of the group walking speed, the number of customers, the visual rate radius, the numbers of doors known by the customers.

It is interesting to note that the role of leaders is not really high here as it is in the reality. In this simulation, they only help customers to go out with the shortest way, but in reality their role is more complex: they guide customers in the supermarket, but they also give advices or a psychological help, *etc.* As it has been shown, if a group walking speed is not great enough, customers should not joint this group (because they average time to exit will be greater than if they were alone).

Future works will concern at least three different tasks. Firstly, we will refine the concept of group. In our opinion, there are preformed groups (such as families, friends, *etc.*) whose the nature is different from spontaneous groups (the group that are formed during the evacuation process). A preformed group is not just a group of agents that are physically close. So, agents' priorities are different according as they are members of preformed groups or spontaneous groups.

Secondly, we will integrate a dynamics in the simulation environment. In this paper, a fire is break out at the beginning of the simulation (at time t_0). We will modify the simulation in the aim to see to it that the fire spreads in the supermarket. So, some area are dangerous at some instant, and are not dangerous anymore at some other instant. Some doors can be blocked by fire, or can be opened, *etc.*

Finally, the next step will be to add emotion. Emotion plays an important role in decision making. It allows agents to react more quickly (for instance, the stress generally increases the walking speed) but can also modify the reaction of agents. For instance, they should be more social and more helpful to others. We expect that emotion should increase survivals rates. Moreover, emotion provides a concrete explanation of the actions of the agents. It is also true for groups. In particular, emotion does not only concern individuals, single agents, but also groups. Some works analyze the emotion spreading among a group of agents, and the influence of this emotion on this group global behavior.

We also need to add some realistic details such as fire in particular positions, the fact that some exits have been blocked by the fire, or the fact that the visual field radius may change during the evacuation process. Finally, some groups may exist before the beginning of a crisis (family, group of friends). Some works show that these groups have an impact on the end of the story.

6. REFERENCES

- [1] C. Adam, B. Gaudou, D. Longin, and E. Lorini. Logical modeling of emotions for Ambient Intelligence. In F. Mastrogiovanni and N.-Y. Chong, editors, *Handbook of Research on Ambient Intelligence and Smart Environments: Trends and Perspectives*, pages 108–127. IGI Global, <http://www.igi-global.com>, 2011.
- [2] C. Adam, A. Herzig, and D. Longin. A logical formalization of the OCC theory of emotions. *Synthese*, 168(2):201–248, 2009.
- [3] C. Adam and D. Longin. Shame: when emotion and reasoning are linked (short paper). In E. Lorini, editor, *European Workshop on Multi-Agent Systems (EUMAS), Toulouse, France, 12/11/2013-13/12/2013*, volume 1113, page (on line), <http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS>, décembre 2013. CEUR Workshop Proceedings.
- [4] V. Bondarev and O. Bondareva. Psychological features of seamen's activity in emergency situations. *TransNav - International Journal on Marine Navigation and Safety of Sea Transportation*, 4(4):453–457, 2010.
- [5] J.-F. Bonnefon, D. Longin, and M. H. Nguyen. A Logical Framework for Trust-Related Emotions. *Electronic Communications of the EASST, Formal Methods for Interactive Systems 2009*, 22:1–16, 2009.
- [6] J. Drury and C. Cocking. The mass psychology of disasters and emergency evacuations: A research report and implications for practice. Research report, University of Sussex, 2007.
- [7] J. Drury, C. Cocking, and S. Reicher. Everyone for themselves? a comparative study of crowd solidarity among emergency survivors. *British Journal of Social Psychology*, 48:487–506, 2009.
- [8] J. Drury, C. Cocking, and S. Reicher. The nature of collective resilience: Survivor reactions to the 2005 london bombings. *International Journal of Mass Emergencies and Disasters*, 27(1):66–95, 2009.
- [9] A. Grignard, P. Taillandier, B. Gaudou, D. A. Vo, N. Q. Huynh, and A. Drogoul. Gama 1.6: Advancing the art of complex agent-based modeling and simulation. In *PRIMA 2013: Principles and Practice of MAS*, pages 117–131. Springer, 2013.
- [10] N. Guiraud, D. Longin, E. Lorini, S. Pesty, and J. Rivière. The face of emotions: a logical formalization of expressive speech acts. In *Proc. of AAMAS'11*, pages 1031–1038, <http://www.acm.org/>, mai 2011. ACM Press.
- [11] E. Lorini, D. Longin, and E. Mayor. A logical analysis of responsibility attribution: emotions, individuals and collectives. *Journal of Logic and Computation*, 24(6):1313–1339, Dec. 2014.
- [12] C. Mei Ling and L. Kincho. Computational framework incorporating human behaviors for egress simulations. 2013.
- [13] C. Mei Ling, P. Paolo, L. Jean-Claude, and L. Kincho H. Safegress: a flexible platform to study the effect of human and social behaviors on egress performance. 2014.
- [14] C. Mei Ling, P. Paolo, L. Jean-Claude, and L. Kincho H. Simulating effects of signage, groups, and crowds on emergent evacuation patterns. 2014.
- [15] F. Rita F, P. GuylÁlne, and A. Lata. 'panic' and human behaviour in fire. 2009.