Aalto University
School of Science
Degree Programme of Engineering Physics and Mathematics

Anton von Schantz

Cellular Automaton Evacuation Model Coupled with a Spatial Game

Mat-2.4108 Independent Research Project in Applied Mathematics.

Espoo, April 25, 2014

Supervisor: Professor Harri Ehtamo
Instructor: Professor Harri Ehtamo

Aalto University
School of Science
Degree Programme of Engineering Physics and Mathematics

Author:	Anton von Schantz		
Title: Cellular Automaton Evacuation Model Coupled with a Spatial Game			
Date:	April 25, 2014	**Pages:**	7+ 18
Professorship:	F3008 Systems and operations research	**Code:**	Mat-2
Supervisor:	Professor Harri Ehtamo		
Instructor:	Professor Harri Ehtamo		

For web-based real-time safety analyses, we need computationally light simulation models. In this study, we develop an evacuation model, where the agents are equipped with decision-making abilities. As a starting point, a well-known cellular automaton (CA) evacuation model is used. In a CA, the agents move in a discrete square grid according to some transition probabilities. A recently introduced spatial game model is coupled to this CA. In the resulting model, the strategy choice of the agent determines his physical behavior in the CA. Thus, our model offers a game-theoretical interpretation to the agents' movement in the CA.

Keywords:	Real-time; evacuation simulation; cellular automaton; spatial game.
Language:	English

Contents

Chapter 1

Introduction

To avoid losses, e.g., in evacuation situations, the rescueing authorities should make timely and accurate decisions. A successful operation requires real-time safety analysis to forecast various disasters and accidents that may take place in events involving human crowds. Thus, safety simulations should be computationally light enough to run in real-time, e.g., in the internet. Recent research sites aiming at these goals are [1, 2].

Our ultimate goal is to create a computationally light evacuation simulation model suited for web-based real-time analyses. The most popular computational evacuation models are cellular automaton (CA) model [4, 8, 9] and the social-force model [6]. FDS+Evac is a validated evacuation simulation software based on the social-force model [10]. In FDS+Evac, the agents' decision making is implemented using game theory [7].

Computationally very light CA model is especially suitable to simulate moving agents in traffic jams and evacuation situations. Hence, it could be used to develop web-based tools to simulate these matters as well. Although, agent movement in the CA model is rather realistic resembling granular flow, it lacks agents' explicit decision-making abilities. In the literature, there have been attempts to apply game theory to remove this shortcoming [3, 5, 11, 14, 15]. In them, game theory has been used to solve a conflict situation, i.e., a situation where several agents try to move simultaneously to the same space; hence the agents' decision making is bounded only to these conflict situations.

In this paper, we couple the CA evacuation model [8] with the spatial game model presented in [7]. As a consequence, the agents have more extended decision-making abilities. The choice of strategy results in the agent either walking straight towards the exit or, in situations where it is optimal, following the lead of others. Further, we give a game-theoretical interpretation to the parameters used to describe movement in a CA.

Chapter 2

Cellular Automaton Model

The agents' movement is simulated with a CA introduced by Schadschneider [8]. Next, we give brief overview of the CA model. In the model, the agents are located in a room divided into cells, so that a single agent occupies a single cell. At each time step of the simulation, the agent can move to one of the unoccupied cells orthogonally next to him, i.e., in the *Moore neighborhood*, where the transition probabilities associated with the diagonal cells are set to zero.[1]

2.1 Movement in the CA

The transition probabilities depend on the values of the static and dynamic floor field in the cells. The static floor field S is based on the geometry of the room. The values associated with the cells of S increase as we move closer to the exit, and decrease as we move closer to the walls. On the other hand, the dynamic floor field D represents a virtual trace left by the agents. An agent leaving a cell, causes the value of D in that cell to increase by one unit. Over time, the virtual trace decays and diffuses to surrounding cells. The values of the fields D and S are weighted with two sensitivity parameters $k_D \in [0, \infty)$ and $k_S \in [0, \infty)$.

Now, for each agent, the transition probabilities p_{ij}, for a move to a neighbor cell (i, j) are calculated as follows

$$p_{ij} = N e^{k_D D_{ij}} e^{k_S S_{ij}} (1 - \xi_{ij}), \tag{2.1}$$

[1] Also called *von Neumann neighborhood*.

where

$$\xi_{ij} = \begin{cases} 1 & \text{for forbidden cells (walls and occupied cells)} \\ 0 & \text{else} \end{cases}$$

and the normalization

$$N = \left[\sum_{(i,j)} e^{k_D D_{ij}} e^{k_S S_{ij}} (1 - \xi_{ij}) \right]^{-1}.$$

The agents' desired movement directions are updated with a *parallel update scheme*, i.e., the directions are updated simultaneously for all agents. In a *conflict situation*, i.e., a situation where several agents try to occupy the same cell, all the agents are assigned equal probabilities to move, and with probability $1-\mu$ one of the agents is allowed to move to the desired cell. Here, $\mu \in [0, 1]$ is a friction parameter, illustrating the internal pressure caused by conflicts. The impact of the friction parameter is depicted in Figure 2.1.

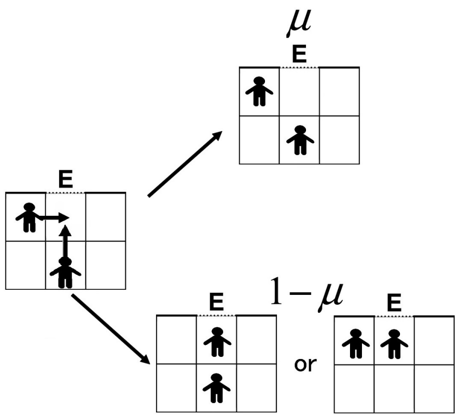


Figure 2.1: The impact of friction parameter on the agents movement. With probability μ neither of the agents get to move, and with probability $1 - \mu$ the other agent moves. Here, E refers to the exit cell.

A cell is assumed to be 40 cm$\times$40 cm. The maximal possible walking velocity for an agent, who does not end up in conflict situations, is one cell per time step, i.e., 40 cm per time step. Empirically the average velocity of a pedestrian is about 1.3 m/s [13]. Thus, a time step in the model corresponds to 0.3 s.

2.2 Different behaviors

In [9], Schadschneider showed that by altering the sensitivity parameters k_S and k_D different behaviors can be observed. He named the different behaviors ordered, disordered and cooperative. In Figure 2.2, the sensitivity parameter combinations responsible for different regimes are plotted in a schematic phase diagram.

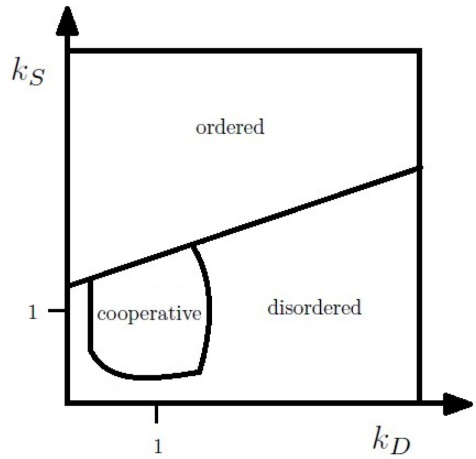


Figure 2.2: Altering the sensitivity parameters k_S and k_D, in the CA model, produces different behaviors.

In the ordered regime, the agents move straight towards the exit. The regime is called ordered, because the movement of the agents is in a sense deterministic. In the disordered regime, the agents just blindly follow each other, whether the agent they are following is heading towards the exit or not. In this study, we are only focusing on ordered and cooperative behavior, as disordered behavior is thought to occur mainly in smoky conditions. Between the ordered and disordered regime, there is the cooperative regime around the values $k_D = k_S = 1$. There, the agents have some knowledge about the location of the exit, but they can also detect regions of higher local flow, and thus minimize their queuing times.

Consequently, for a freely moving agent, ordered behavior makes the agent evacuate fastest. However, a sufficiently large μ causes a faster-is-slower phenomenon, where a crowd of ordered agents will evacuate slowest. The reason is that a large amount of conflicts caused by ordered behavior slows down the evacuation. Conversely, the large μ causes a cooperative crowd to evacuate fastest.

Chapter 3

Spatial Evacuation Game

To equip the agents with decision-making abilities, we couple the CA model with a game-theoretical model by Heliövaara et al. [7]. Remember, that the CA model and the game-theoretical models are two separate models, the first describing the physical movement of the agents, and the second the decision-making process of the agents.

3.1 Game Description

In the game, n_a agents, indexed by i, $i \in I = \{1, ..., n_a\}$, are in an evacuation situation, and located in a discrete square grid. Each agent has an *estimated evacuation time* T_i, which depends on the number λ_i of agents between him and the exit, and on the capacity of exit β. T_i is defined as

$$T_i = \frac{\lambda_i}{\beta}. \tag{3.1}$$

Each agent has a *cost function* that describes the risk of not being able to evacuate before the conditions become intolerable. The cost function $u(T_i)$ is a function of T_i. The shape of the cost function depends on the parameter T_{ASET}, *available safe egress time*, which describes the time, in which the conditions in the building become intolerable. Additionally, a parameter T_0 describes the time difference between T_{ASET} and when the agents start to play the game.

The agents interact with other agents in their Moore neighborhood. Each agent can choose to play either *Patient* or *Impatient*. Let us denote the average evacuation time of agent i and j, $T_{ij} = (T_i + T_j)/2$. In an impatient vs. patient agent contest, an impatient agent i can overtake his patient neighbor j. This reduces agent i's evacuation time by $\triangle T$ and increases j's evacuation time by the same amount. The cost of i is reduced by $\triangle u(T_{ij})$ and increased for j by the same amount. Here

$$\triangle u(T_{ij}) = u(T_{ij}) - u(T_{ij} - \triangle T) \simeq u'(T_{ij})\triangle T. \qquad (3.2)$$

In a patient vs. patient agent contest, the patient agents do not compete with each other, they keep their positions and their costs do not change. In an impatient vs. impatient agent contest, neither agent can overtake the other, but they will face a conflict and have an equal chance of getting injured. The risk of injury is described by a cost $C > 0$, which affects both agents. The constant C is called the *cost of conflict.* We assume that $u'(T_{ASET}) = C$. Also, we assume that $u'(T_{ij}) > 0$. Thus, based on Equation 3.2, we have $\triangle u(T_{ij}) > 0$. Now, an illustration of a quadratic cost function can be drawn (see Figure 3.1).

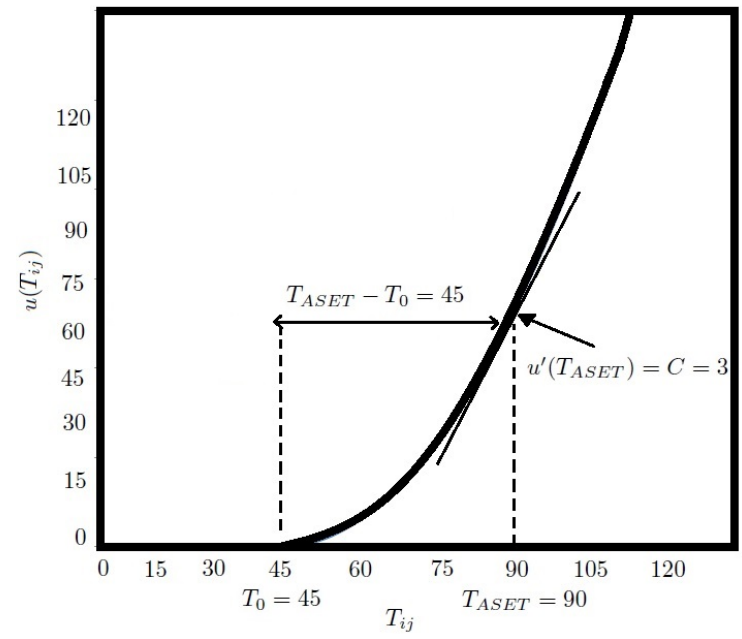


Figure 3.1: Illustration of the parameters of the cost function. The function in the figure has the parameter values: $T_{ASET} = 90, T_0 = 45, C = 3$.

From the aforementioned assumptions, a 2×2 game matrix can be constructed:

		Agent 2	
		Impatient	Patient
Agent 1	Impatient	$C/\triangle u(T_{ij}), C/\triangle u(T_{ij})$	$-1, 1$
	Patient	$1, -1$	$0, 0$

.

Here, all the elements of the more intuitive form of the game matrix have been divided by $\triangle u(T_{ij})$. When a particular pair of strategies is chosen, the costs for the two agents are given in the appropriate cell of the matrix. The cost to agent 1 is the first cost in a cell, followed by the cost to agent 2.

Because this is a cost matrix, the agents want to minimize their outcome in the game. Depending on the number $C/\triangle u(T_{ij})$, the matrix game, considered as a one-shot game, is a Prisoner's Dilemma game or a Hawk-Dove game. In addition to pure Nash equilibria (NE) the latter has mixed strategy NE. These equilibria are analyzed in detail in [7].

3.2 Update of Strategies

The total cost for an agent is the sum of the costs against all of his neighbors, and the agent's *best-response strategy* is a strategy that minimizes his total cost. The agents are *myopic* in the sense that they choose their strategies based on the previous iteration period of the game, not considering what possibly will happen in the future iteration periods.

At each iteration period, all the agents' strategies are updated once. The agents' strategies are updated with a *shuffle update scheme*. In the scheme, the order in which the strategies are updated is randomized. The best-response strategy $s_i^{(t)}$ of agent i on iteration period t is given by his best-response function BR_i, defined by

$$s_i^{(t)} = BR_i(s_{-i}^{(t-1)}; T_i, T_{-i}) = \arg\min_{s_i' \in S} \sum_{j \in N_i} v_i(s_i', s_j^{(t-1)}; T_{ij}). \qquad (3.3)$$

Here, N_i is the set of agents in agent i's Moore neighborhood. The function $v_i(s_i', s_j^{(t-1)}; T_{ij})$ gives the loss defined by the evacuation game to agent i, when he plays strategy s_i', and agent j has played strategy $s_j^{(t-1)}$ on iteration period $(t-1)$. That is, $v_i(s_i', s_j^{(t-1)}; T_{ij})$ is equal to the corresponding matrix element. Here, $s_{-i}^{(t-1)}$ is used to denote the strategies of all other agents than agent i on iteration period $t-1$, and T_{-i} includes the estimated evacuation times of these agents.

3.2.1 Implementation into FDS+Evac

The game is implemented into FDS+Evac [10]. There, playing the game actually changes the physical behavior of the agents. Impatient agents do not avoid contacts with other agents as much; they accelerate faster to their target velocity, and move more nervously. Whereas, patient agents avoid contact with other agents.

Chapter 4

Cellular Automaton Evacuation Model Coupled with a Spatial Game

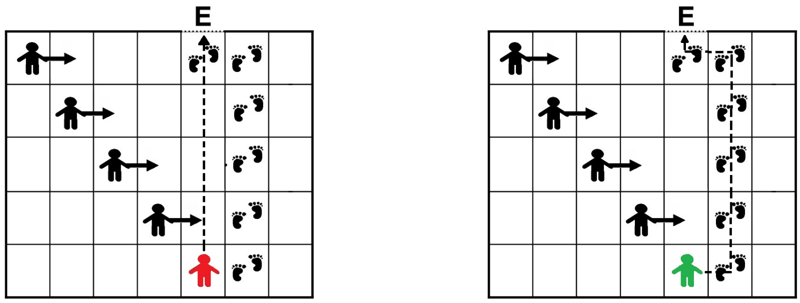
There are similarities between the presented spatial game and CA model. As noted above, patient agents avoid conflicts whereas impatient agents end up in conflicts by competing with other agents. Also, there is a cost associated with conflicts.

By coincidence, in the CA model, situations, where several agents try to move simultaneously to the same cell, are also called conflicts. Also, in these conflicts, there is a cost for those agents that have to wait to move to their target cell, provided we regard the waiting time as a cost.

Also, the description of impatient agents resembles the movement of agents in the ordered regime. Agents in the ordered regime are set to move straight towards the exit, and thereby have a tendency to cross paths with other agents. On the other hand, the description of patient agents resembles the movement of agents in the cooperative regime. Agents in the cooperative regime are set to avoid conflict situations, by rather following other agents than crossing paths with them.

4.1 Effect of Strategy Choice on Behavior

From the aforementioned observations, we propose a model, where we couple the CA model with the spatial evacuation game. In our model, we let the strategy choice of playing Impatient result in ordered behavior in the CA model, and playing Patient in cooperative behavior. The effect of strategy choice on the agent's movement is depicted in Figure 4.1.



(a) If the agent plays Impatient, he moves straight towards the exit, regardless of the awaiting conflict situation.

(b) If, on the other hand, the agent plays Patient, he follows the virtual trace, avoiding the conflict situation.

Figure 4.1: Effect of strategy choice on the agent's movement.

It should be noted, that the strategy choice the agent makes, does not reflect an optimal route towards the exit, i.e., it is not an optimal strategy for the whole evacuation over time. Rather, the strategy choice is optimal in a snapshot of the evacuation against his immediate neighbors (actually the whole crowd is in an NE in a snapshot [7]).

4.2 Model Description

Next, a step-by-step description of our model is given. In the beginning of the simulation, the agents are located randomly in the room. None of the agents play the game, and all agents are considered patient.

Step 1. At the beginning of each time step, T_i is calculated for $i = 1, ..., n_a$. If $T_i > T_{ASET} - T_0$, the agent i plays the game.

Step 2. The agents' strategies are updated with the shuffle update scheme. The agents observe the strategies of the other agents in their Moore neighborhood, and choose a best-response strategy according to Equation 3.3.

Step 3. The agents' behavior is updated in the CA model, to correspond to their strategy choice. This is done by altering the agents' sensitivity parameters as follows:

(a) Playing Impatient results in ordered behavior. The agents sensitivity parameters are set to $k_D = 1.0$ and $k_S = 10.0$.

(b) Playing Patient results in cooperative behavior. The agents sensitivity parameters are set to $k_D = 1.0$ and $k_S = 1.0$.

Step 5. The agents move in the CA.

Step 6. Go to Step 1. This procedure is repeated until all agents have evacuated the room.

Remark 1: Here, a time step refers to a time step in the CA, i.e., the agents are able to move once.

Remark 2: In Step 3, the shuffle update scheme is repeated multiple times, to ensure that the agents are in an equilibrium configuration all the time. Figure 4.2 illustrates a snapshot of the evacuation in such a configuration. More such simulations, with different patient and impatient agent densities, can be found in [7, 12].

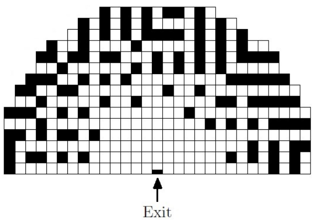


Figure 4.2: An equilibrium configuration for 378 agents with parameter values $T_{ASET} = 450$ and $T_0 = 400$. Black cells represent impatient agents and white patient.

Remark 3: The sensitivity parameter values chosen to represent ordered and cooperative behavior are arbitrary. But still, they are chosen to be such that they are clearly inside the appropriate regimes in Figure 2.2.

Chapter 5

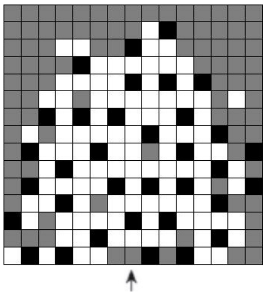
Numerical Results

We have presented an evacuation model, where the agents' sensitivity parameters appear as a result from the game the agents play. In the following, we illustrate how the agents behave in a typical evacuation simulation. Additionally, we show that the faster-is-slower effect, already found in the original formulation [8], now appears as a result of the game the agents play. The result is compared to a similar analysis made by Heliövaara et al. with FDS+Evac [7].

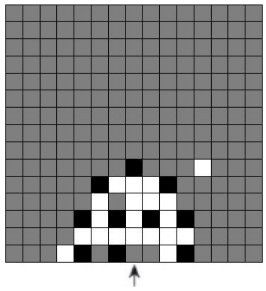
5.1 Evacuation of a Large Room

Here, we simulate a typical evacuation situation, i.e., the evacuation of a large room. In Figure 5.1 there are three snapshots from different stages of this evacuation simulation. The black squares represent impatient agents and the white patient.

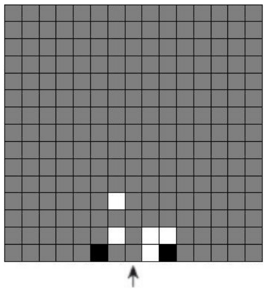
As can be seen, the agents form a half-circle rather quickly in front of the exit. Notice that the agents play their equilibrium strategies at each snapshot of the simulation. At these snapshots, the impatient agents have decided that they will evacuate fastest by moving straight towards the exit, whereas the patient agents have decided to follow other agents.



(a) Early stages



(b) Middle stages



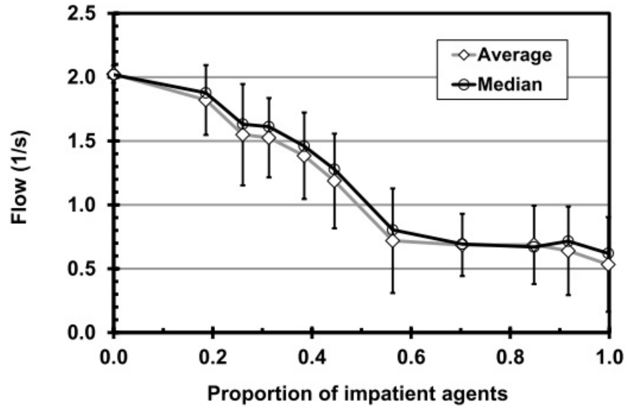
(c) Late stages

Figure 5.1: Snapshots of the simulation in different stages of the evacuation process. The black squares represent impatient agents and the white patient.

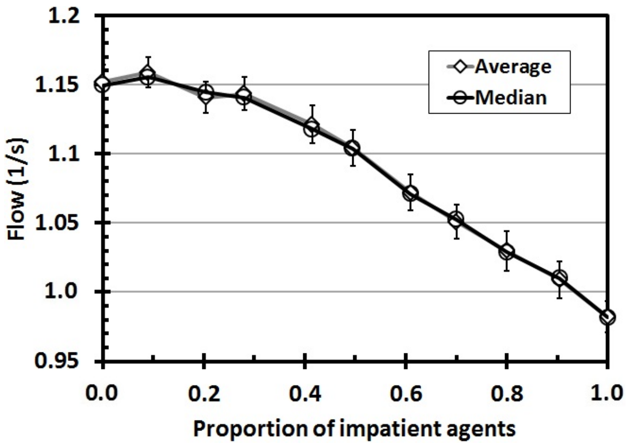
5.2 Faster-is-Slower Effect

Heliövaara et al. studied the dependence of the proportion of impatient agents on egress flow in FDS+Evac [7]. The agents were set in a half-circle in front of the exit, and they updated their strategies until equilibrium was reached. Afterwards, the agents' strategies were fixed, the exit was opened and the agents start to evacuate. The same simulations were run with our model. The results of the simulations with these two models can be seen in Figure 5.2.

It is clearly seen, from both Figures 5.2 (a) and (b), that the more agents behave impatiently, the smaller the egress flow is. Since the effective velocity of an impatient agent is larger than that of a patient, a faster-is-slower effect can be distinguished. In FDS+Evac, this is caused by impatient agents pushing harder towards the exit, which results in jams and reduced flows [7]. In our model, it is caused by impatient agents moving straight towards the exit, resulting in more conflict situations and slowing down the evacuation. The quantitative differences can be explained by the different geometries of



(a) Simulations with FDS+Evac [7] (a 0.8 m wide exit).



(b) Simulations with our model (a 0.4 m wide exit).

Figure 5.2: Average egress flow for 200 agents with different proportion of impatient agents in the population. In the simulations, 11 different values of T_{ASET} were used.

both the agents and the exits. Also, the velocities of the agents are different in the two models.

Chapter 6

Discussion and Conclusions

We introduced a CA evacuation model, where the agents are equipped with decision-making abilities. For the simulation of the agents' movement, we used Schadschneider's simulation platform [8]. In it, ordered and cooperative behaviors can be obtained by altering the sensitivity parameters k_D and k_S. To provide the agents with decision-making abilities, we coupled it with a spatial game introduced by Heliövaara et al. [7].

In our model, the choice of strategy actually changes the behavior of the agent in the CA. Patient agents follow other agents, i.e., are cooperative, when there is a possibility for a conflict situation. On the other hand, impatient agents walk straight towards the exit, risking ending up in a conflict, i.e., are ordered.

In Schadschneider's original model, the values of the sensitivity parameters should be fixed before simulation starts. In our formulation, the agents' sensitivity parameters depend on their strategy choice in the spatial game. Moreover, the agents' parameters change dynamically according to their perception of the surrounding conditions, i.e., the risk of not being able to evacuate in time, and the behavior of neighboring agents. Ultimately, our model changes the evacuating crowd, from an unintelligent granular flow, to a crowd of intelligent decision-makers.

In the end of the numerical section, we noticed that our model in some aspects behaves qualitatively similarly to the validated FDS+Evac evacuation simulation software [10]. To map the full potential of our model, further comparisons with the evacuation simulation software should be done. Since our model is computationally light, it could be used for web-based real-time safety analyses.

Bibliography

[1] http://2013.nycbigapps.com/project/113/mesa-the-massive-evacuation-simulation-application (accessed april 24, 2014).

[2] http://www.emergencymgmt.com/training/online-evacuation-tool-plan.html (accessed april 24, 2014).

[3] Bouzat, S., and Kuperman, M. N. Game theory in models of pedestrian room evacuation. *Physical Review E 89* (2014), 032806.

[4] Burstedde, C., Klauck, K., Schadschneider, A., and Zittartz, J. Simulation of pedestrian dynamics using a two-dimensional cellular automaton. *Physica A: Statistical Mechanics and its Applications 295* (2001), 507–525.

[5] Hao, Q. Y., Jiang, R., Hu, M. B., Jia, B., and Wu, Q. Pedestrian flow dynamics in a lattice gas model coupled with an evolutionary game. *Physical Review E 84* (2011), 036107.

[6] Helbing, D., Farkas, I., and Vicsek, T. Simulating dynamical features of escape panic. *Nature 407* (2000), 487–490.

[7] Heliövaara, S., Ehtamo, H., Helbing, D., and Korhonen, T. Patient and impatient pedestrians in a spatial game for egress congestion. *Physical Review E 87* (2013), 012802.

[8] Kirchner, A., Nishinari, K., and Schadschneider, A. Friction effects and clogging in a cellular automaton model for pedestrian dynamics. *Physical Review E 67* (2003), 056122.

[9] Kirchner, A., and Schadschneider, A. Simulation of evacuation processes using a bionics-inspired cellular automaton model for pedestrian dynamics. *Physica A: Statistical Mechanics and its Applications 312* (2002), 260–276.

[10] Korhonen, T., and Hostikka, S. Fire dynamics simulator with evacuation: Fds+evac. Tech. rep., VTT Technical Research Centre of Finland, 2009.

[11] Shi, D. M., and Wang, B. H. Evacuation of pedestrians from a single room by using snowdrift game theories. *Physical Review E 87* (2013), 022802.

[12] von Schantz, A. Modeling egress congestion using a cellular automaton approach. Master's thesis, Aalto University School of Science, 2014.

[13] Weidmann, U. Transporttechnik der fussgänger. *Schriftenreihe des IVT, ETH Zürich 80* (1992).

[14] Zheng, X., and Cheng, Y. Conflict game in evacuation process: A study combining cellular automata model. *Physica A: Statistical Mechanics and its Applications 390* (2011), 1024–1050.

[15] Zheng, X., and Cheng, Y. Modeling cooperative and competitive behaviors in emergency evacuation: A game-theoretical approach. *Computers & Mathematics with Applications 62* (2011), 4627–4634.