Implementing Sub Steps in a Parallel Automata Cellular Model

Pablo Cristian Tissera[1], A. Marcela Printista[13] and Emilio Luque Fadon[2]

[1] Laboratorio de Investigación y Desarrollo en Inteligencia Computacional
Universidad Nacional de San Luis -Argentina
[2] Departamento de Arquitectura y Sistemas Operativos
Universidad Autónoma de Barcelona - España
[3] CONICET CCT-San Luis-Argentina
{ptissera, mprinti}@unsl.edu.ar

Abstract.
Computer simulations using Cellular Automata (CA) have been applied with considerable success in different scientific areas. In this work we use CA in order to specify and implement a simulation model that allows to investigate behavioural dynamics for pedestrians in an emergency evacuation.

A CA model is discrete and handled by rules. However several aspects of the crown behaviour should appear as a continuous phenomenon. In this paper we implement the *sub steps technique* in order to solve an unexpected phenomenon: the formation of *holes* (empty cells) both around the exit and mixed in the crowd evacuation. The *holes* occur when individuals of consecutive cells want move towards an exit. Additionally, the methodology allowed us to include, as a new parameter of the model, speeds associated to the pedestrians.

Due to the incorporation of new features, the model complexity increases. We apply a parallel technique in order to accelerate the simulation and take advantage of modern computer architectures.

We test our approaches to several environment configurations achieving important reduction of simulation time and the total evacuation time.

Keywords: Evacuation Simulation, Parallel Cellular Automata, Crown Dynamics.

1 Introduction

Shopping centres, schools and dance halls are some examples of buildings commonly used in different daily activities that involve the meeting of a large number of people within a closed area. The designers of these types of building usually attempt to maximise the productivity of the available space, but also it is necessary to consider a suitable planning for assuring people safety when an unusual behaviour of the crowd occurs. One of the most frequent causes of this kind of behaviour is the emergency evacuation due to the threat of fire. In such situation, a closed area, with a relatively small number of fixed exits, must be evacuated for a large number of people.

In the last years, the interest in models of emergency evacuation processes and pedestrian dynamics has increased.

In this context, models can be characterised according to $resolution$, $fidelity$ and *scale* [1]. The resolution is the level of detail regarding the representation of space. Models used for evaluating the evacuation processes can broadly be categorised in microscopic (high resolution) and macroscopic approaches (low resolution). The macroscopic approaches are based on differential equations that take into account the similarities with systems previously studied like dynamics of fluids. On the other hand, the microscopic approaches allow to consider how the system state evolves during the model running.

This paper considers a microscopic model where the space is represented in a discrete fashion. The special discrete model that we used is called Cellular Automata (CA) model [2] [3] [4]. For microscopic models, the connection between the space (its geometry), population and the propagation of fire and smoke is made via the rules or equations.

A high fidelity model is one with many parameters that directly takes into account all the different influences (e.g., parameters like age, training, weight, etc. in the case of pedestrians). CA allow us to generate “local” and “uniform” behaviours that resemble the dynamics observed in real processes of fire and smoke propagation. However, these local features were not suitable for representing certain aspects of people behaviour that require a more especif and differentiated perspective. We developed a hybrid model where the dynamics of fire and smoke propagation are modelled by means of CA and for simulating people behaviour we are using goal oriented *intelligent agent*.

The scale is the size of the problem with respect to time, space, etc. The scale of a pedestrian simulation depends on the application of course. In our case, a model that scales linearly (with respect to computation time and memory requirement) with the number of persons or the size of the environment is desirable. While agent-based models afford higher complexity and finer granularity they also require more computational resource. This means for large populations, modelling individuals with complex rules becomes infeasible, typically individuals will instead be modelled as simple particles. We used partitioning technique as a solution to this problem, allowing large scale simulations.

The simulation allows to specify different scenarios with a large number of people and environmental features, making easier the study of the complex behaviours that arise when the people interact.

In section 2 we present the Agent-based CA model for the pedestrian motion. In section 3 we developed new features that are needed to adequately represent the dynamics of a crowd. In section 4 multicolumn partitioning technique is explained. In section 5 we describe our work with different instances of the problem at hand and report the performance analysis of each case. This section also includes a few commentaries on the operation of the EVAC and EVAC* simulation systems that support the proposed models. Finally, the section 6 presents the conclusions.

2 Agent based CA Model for Evacuation Simulation

The cellular automata (CA) are discrete dynamic systems that offer an appealing alternative to deal with this kind of situations due to their capacity to develop complex behaviours from a simple set of rules . Basically, these rules allow to specify the new state of a cell based on the state of the neighbouring cells. In this way, it is possible to model complex dynamic systems from the specification of the local dynamics of its components. A further advantage of these systems is the support usually provided for displaying the results in a graphical way, allowing an easier comprehension of the dynamics of the system under study.

We use the CA as basis of our simulation model for to investigate the pedestrian dynamics in emergency situations. Below, we describe the main features of the proposed model:

Cellular Space: it is a finite bi-dimensional array (grid) with closed boundaries. Each cell of the cellular space represents $40 \times 40\ cm^2$. This is the space usually occupied by a person in a crowd with maximal density [5,6,7]. The dimensions of the cellular space are specified in meters. So, one grid of $10 \times 10\ m^2$ will contain 25 cells by side.

States: a cell can be in one of the states of the set $Q = \{W, E, P, O, S, SF, PS\}$ where:

W : External wall cell.
E : Empty cell.
P : Cell with a person.
O : Internal obstacle cell.
SF : Cell with smoke and fire.
PS : Cell with a person and smoke.
S : Cell with smoke.

Neighbourhood: the neighbourhood considered in the model is *Moore's Neighbourhood*, that includes the eight cells surrounding the central cell. With this choice we aim to provide to each individual in the system with all possible movement directions.

Initial Configuration: before the simulation starts diverse information related to the outer walls, inner obstacles, individuals, combustible locations, cell with fire and arrangement of the exits must be specified.

Virtual clock: taking into account recent studies [8,5], an updating time of 0.3 seconds by time-step was specified for our model. This value is the estimated time required by a pedestrian for walking 0.4 m (size of a cell side).

Model Evolution Rules [9]:

1. Rules about the *building:* a cell in state W or O (outer wall or obstacle) will not change its state throughout the simulation.
2. Rules about *smoke propagation:* a cell with smoke (S, SF or PS) at time t, also will have smoke in time $t + 1$. If at time t the central cell does not have smoke, but some of its adjacent cells have smoke, the central cell also

will have smoke at time $t+1$ with a probability proportional to the number of adjacent cells with smoke. For example, if one cell has four adjacent cells with smoke on a total of eight adjacent cells, that cell will have smoke in the next time-step with probability $\frac{1}{2}$.

3. Rules about *fire propagation:* these rules are analogous to the rules for smoke propagation.
4. Rules about the *people motion:* a simple reflex intelligent agent represents the pedestrian capability to decide about its desire to move to the cell depending on its environment. To solve the collisions that arising when two or more people simultaneously attempt to occupy the same physical location belonging to an exit way, we changed the approach commonly used in other works. Instead of being the pedestrian who decides which cell to move, the current cell is in charge of choosing between the people in the adjacent cells, which pedestrian will move to the cell. The policy distributes the decision between neighbouring agents. For this, each agent builds a *beliefs matrix* containing the distances from its neighbour to the exit. This information is relevant to the decision that it must take:

   If the distance from the requirer cell to the exit is greater than the current distance, then the agent does not move.

   If the distance from the requirer cell to the exit is less than the current distance, then the agent replies its desire of moving.

   If both distances are the same, the agent checks from its neighbourhood, cells that offer better placement:

   – If the number of empty cells around the agent with a better position is 0 or 1, the agent will respond positively to the move with a 50% probability.

   – If the number of empty cells around the agent with a better position is 2, the agent will respond positively to the move with a 25% probability

   – If the number of empty cells around the agent with a better position is greater than 3, the agent says no wish to move. With increasing number of empty positions around a cell, the desire to move decreases.

   The above percentage values were obtained empirically, using those that best fit the reality. For calculating the distances from each cell to an exit, the Dijkstra's algorithm was used. This algorithm solves the single-source shortest-paths problem on a weighted graph [10]. The cellular space is considered as a graph, where each cell represents a node and all the edges connecting adjacent cells have weight *1*. Cells with state W (wall) or O (obstacle) are not considered to build the graph.[4]. If the building have more than one exit, the distance computation takes into account the exit nearest to the cell.

   The cell builds a list of candidates with all the agents who want to move. Later, it must select a single pedestrian. The used process for solving this point is the following: a) if more than one individual remains as candidate to occupy the cell, the one with the minor damage grade (parameter specified in the model) will be selected; b) if the conflict persists, the pedestrian will be selected at random. The decision is concentrated in the empty cell.

---

[4] An individual can not cross a wall and therefore he should surround the wall to reach the exit

3 Implementation of Crown Dynamics

Whether to use a discrete or continuous representation of space is closely connected to the implementation. A strong argument in favour of discrete models is that they are simple and can be used for large scale simulations. Additionally, for pedestrian motion there is a finite reaction time, which introduces a time scale. The time is chosen to be discrete in the model and this naturally leads to a discrete representation of space.

It is possible to discretize the progress of an individual pedestrian, however the movement of a crowd should appear as a continuous phenomenon. Consider the situation in Fig. 1, which shows a mass of people around the exit is competing to quickly leave the building. Graphically, it is possible to observe the formation of holes (empty cells) both around the exit and mixed in the crowd.

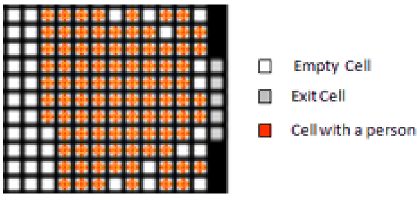


Fig. 1. Evacuation Simulation

In order to analyse this phenomenon, the Fig. 2 (top) examines in detail the progress of two people who are in adjacent cells. The time-step 0, only B can move a cell towards the exit. At the same time-step, A must remain in place because no rule can move it a cell forward. This is because A, based on their environment, can not determine that the cell will be abandoned by the current individual. At this point a hole is formed which propagate back across the crowd.

In the next time-step, 1, A can move one cell forward, and the time-step 2 it will exit of building. Although both individuals were in adjacent cells (and more generally belong to a crowd), they could not evacuate in consecutive time-steps. The Fig. 2 (bottom) shows the effect of continuity should be seen in an evacuation of a crowd.

This parallel update (inherent in CA) leads to blocking of cells used during one time-step [11]. An alternative is the introduction of sub steps among two consecutive time-steps of CA, as you can see in Fig. 3(a). Clearly in the case of more than two individuals, we only need to process more than one sub step to

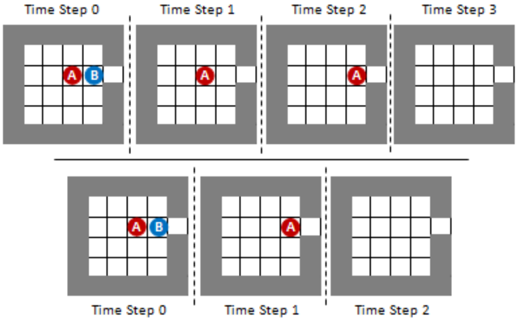
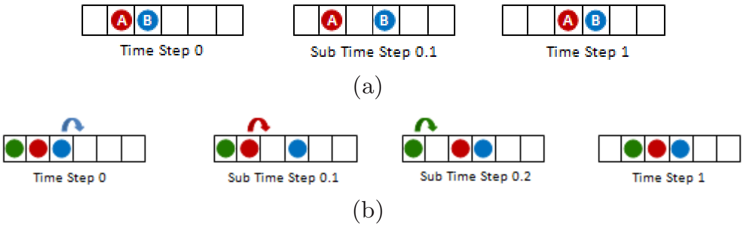


Fig. 2. Advance of two adjacent agents in a CA Model (top). Advance in a Real Evolution (bottom)

resolve the situation. The Fig. 3(b) shows that the procedure is similar to shift all the elements of an array one position forward. In this case, with 3 pedestrian, it is necessary 2 sub steps.

It is important to note that the implementation allows us to associate different speeds to individuals. Suppose the evacuation of Fig 4 in a one-dimensional environment. Consider that the individuals A, C and D have an associated speed of $1.2m/s$ (3 cells per s.) and individual B of $0.8m/s$ (2 cells per s.). Individuals A and D were able to move 3 cells (maximum speed / time-step), B advanced 2 cells (maximum velocity / time-step) and C was blocked by the individual B, being able to move only 2 cells. Note that in the two-dimensional cellular space, the individual C might then have found an alternative path and achieves at some point move to 3 cells per time-step.

In this work, the parameter sub steps/time-step has been established empirically. However the Figs. 3 and 4 give an indication of how complex it can be dynamically determine this parameter.



(a)

(b)

Fig. 3. Sub Steps among two consecutive Time-Step

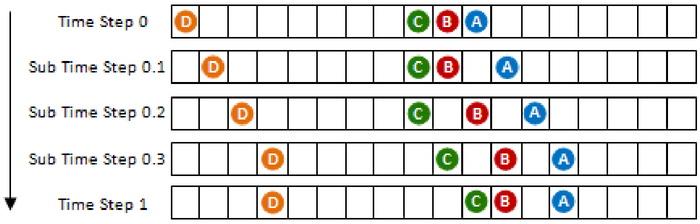


Fig. 4. Evacuation of 4 individuals with different speeds

4 Multi-column Partitioning

The CA are discrete systems that evolve over time. As a general rule, the evolution of CA is performed by repeated update of the complete set of cells, where the new state of each cell depends on the existing state of its neighbourhood. CA simulation techniques used in this work, are too slow if the space to simulate is large or complex.

The aim of this work is to search for techniques to accelerate simulations exploiting the parallelism available in current multicomputers.

One of the most common methods used for resolving this problem is the Ghost Cell Pattern [12]. In this technique, the grid is geometrically divided into chunks that are processed by different processors. One challenge with this approach is that the update of points at the periphery of a chunk requires values from neighbouring chunks. If a cell and its neighbours are in the same node, the update is easy. On the other hand, when nodes want to update the border cells, they must request the values of the neighbouring cells on other nodes. The solution to this problem is to allocate space for a series of ghost cells around the edges of each chunk. For every iteration, each pair of neighbour exchanges a copy of their border and places the received borders in the ghost cell region (see Fig. 5). The ghost cells form a halo around each chunk that contains replicates of the borders of all immediate neighbour. These ghost cells are not updated locally, but provide stencil values when the borders of this chunk are updated.

In our proposal, the cellular space of the automaton is represented by an bi-dimensional array, which contains $X \times Y$ cells. Inside of a cluster based on distributed memory system, the parallel execution using P processors (denoted $p_0, p_1, ...p_{P-1}$) is performed by applying the transition function simultaneously to P chunks in a $SPMD$ way. Due to the particular characteristics of the model, the proposed parallelization differs a bit from the traditional approach [13,14]. The Fig. 6 shows the decomposition applied in our proposal where the cellular space is divided in one-dimensional chunks (multi-column).

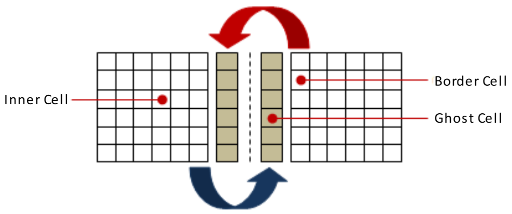


Fig. 5. Border Exchange

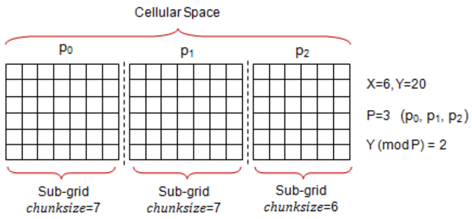


Fig. 6. Multi-column Decomposition: if P and Y are multiples then $chunksize = Y/P$; if P and Y are not multiples and $rank < Y \pmod{P}$ then $chunksize = Y/P + 1$ else $chunksize = Y/P$

In each time-step, the algorithm updates each cell in the lattice. If a cell contains an individual, three things can happen in the next state: (1) the individual may leave the border cell of a chunk and move into the other chunk, (2) it can change of chunk, and (3) it can move from inside a chunk to its border. The Fig. 7 illustrates these movements. Situations 1 and 2 can lead to inconsistencies in the next state.

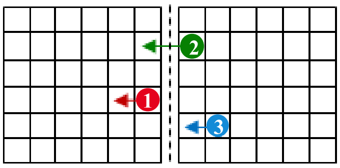


Fig. 7. Possible Movements

Suppose in the Fig. 8-left that the cell marked with X is being updated by the process 1. That cell offers a better position than the pedestrian currently possesses (filled circle), so after applying the transition rules it decides to move there. The resulting state would be that the individual has taken the new position, leaving the ghost cell of process 2 outdated, since in the cell still contained the pedestrian who is no longer in that position (Fig. 8-right).

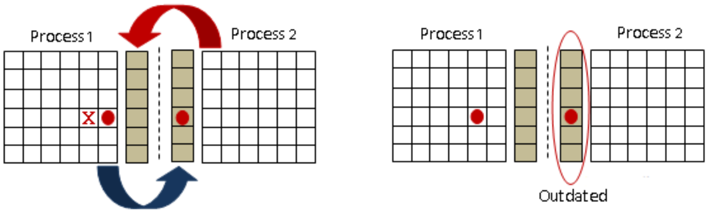


Fig. 8. Status Inconsistency

At first this does not imply any kind of error, but if the individual was placed in a position equidistant from two possible exits, the agent could take a different decision in both processes (the replicated individual moves toward two possible exits).

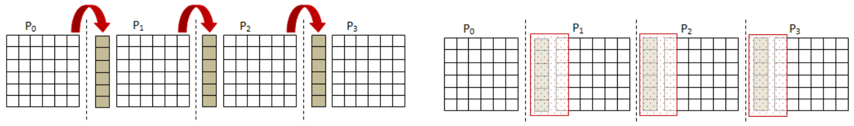
To keep the two processes with the updated status on the decision of an agent by making use of ghost cell pattern technique requires multiple message exchange during the same time-step. If we want to model large environments (airports, convention centres, stadiums, etc..) and simulate the evacuation of thousands of people, the communication overhead would be significant, implying a degradation of performance of the model.

The Algorithm 4.1 shows the proposed methodology that alleviates the impact of the interaction of processes against the decision to an agent.

Algorithm 4.1: EVAC*()

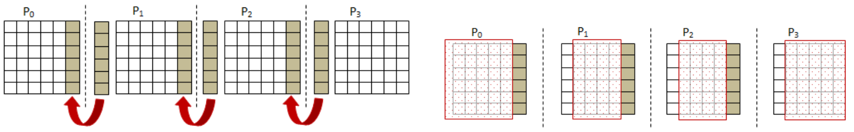
```
comment: $l$ : leftmost column $r$ : rightmost column
while People to evacuate
  do
    if $rank < P - 1$
      then
        Send local $chunk_r$ to process $rank + 1$
        Receive local $chunk_r$ from process $rank + 1$
    if $rank > 0$
      then
        Receive $chunk_{ghost}$ from process $rank - 1$
        Evolve $chunk_{ghost}$ and local $chunk_l$ (contiguous columns)
        Send evolved $chunk_{ghost}$ to process $rank - 1$
    Evolve remaining cells of $chunk$
    Actualise number of people to be evacuated
End.
```

The algorithm is illustrated by means of the Fig. 9.



(a) Send rightmost column of *chunk* to process $rank + 1$

(b) Evolve ghost and leftmost column



(c) Return evolved ghost column to process $rank - 1$

(d) Evolve remaining cells

Fig. 9. Multi-column Partitioning Algorithm for CA Model

5 Result of the Simulation

In this section, we present the simulation result of the explained research.

The experiments were carried out with EVAC* Simulator [15], an simulation system based on parallel cellular automata. EVAC* is a system developed in C and MPI for passing message and uses the graphical interface of EVAC Simulator. EVAC is a system developed in Java that allows the design and simulation of spatial environments in an sequential way. EVAC simulator offers a friendly graphical interface which can be easily used by non expert users [9].

The experiments consider three environment configurations of the building to be evacuated (A,B,C), addressed those situations where several exits exists, varying the different occupation densities of the environment (number of individuals distributed evenly) and the size of the exit:

- A, 20 × 20 m^2, two exits of 1.2 m. each, 831 pedestrians.
- B, 40 × 40 m^2, three exits of 1.2 m. each, 1397 pedestrians.
- C, 80 × 50 m^2, three exits of 2.4 m. each, and 1888 pedestrian.

The experiments are designed to compare the performance of the AC model which runs sub steps between successive time-steps, called the SS experiment, with the original model called TS experiment. For the SS experiment, we set the speed to all pedestrians in 3 cells/s. The corresponding total evacuation time (TET) (seconds) and mean travelled distance (MD) (meters) per individual to the exit were obtained. The comparative results are shown in Table 1.

Case	TET (Time-Step Experiment)	TET (Sub Step Experiment)	MD
A	70.38	18.30	9.99
B	105.00	25.05	22.39
C	138.83	37.08	40.87

Table 1. EVAC*: Total Evacuation Time and Mean Travelled Distance

The significant reduction of the evacuation times experienced for the sub step experiment is because individuals are able to move without blocking the exit at the same time-step. While our project is in development, the empirical values obtained for the evacuation time are comparable to other implementations, which have validated their results against real evacuation exercises [16] [11].

For the parallel experiments, we used a cluster equipped with 16 nodes of 64 bits with Intel Q9550 Quad Core 2.83GHz processors and RAM memory of 4GB DDR3 1333M. The nodes are interconnected by a Switch Linksys SLM2048.

With both growing environment size and the occupation densities the simulation time increases. Using a partitioning strategy where the cellular space is divided in one-dimensional chunks, we can observe that as the degree of parallelism grows, the simulation performance improves (to see Fig. 10) . In addition, this new version of evacuation simulator requires the largest computational effort, since the tests have to be carried out with the execution of multiple *sub steps*.

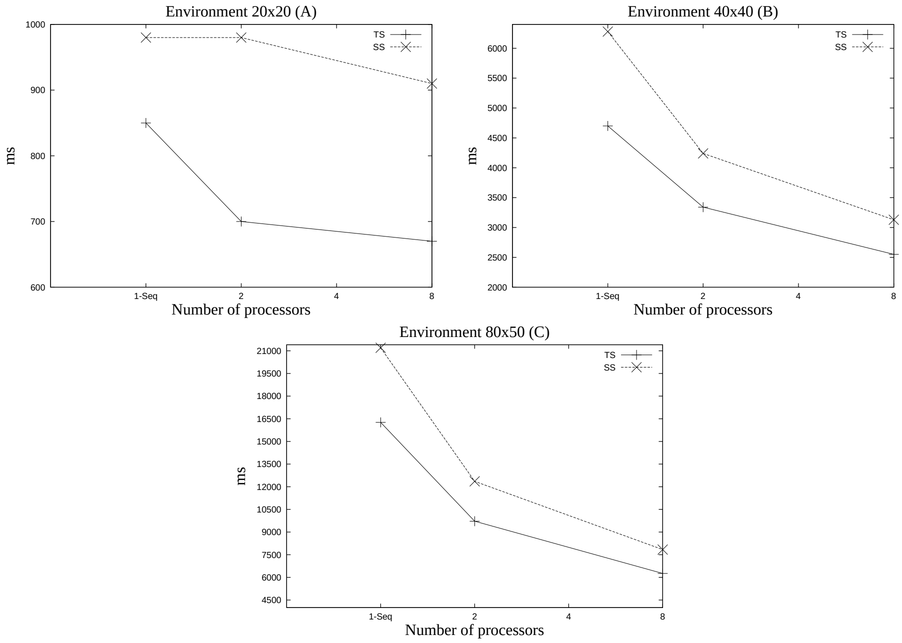


Fig. 10. EVAC*: Parallel Simulation Time

6 Conclusions

In this work, we used Cellular Automata for developing and implementing a simulation model of emergency evacuations due to the fire threat.

In spite of the assumptions introduced for obtaining a more simple model, the CA resulted to be very suitable tools for modelling this class of problems achieving in many cases, results very similar to those expected to occur in a real evacuation situation. The techniques and results reported in this work are part of a research project in the long term. To date, our strategies are only empirically tested. But as future work we wish to compare our research with real evacuation exercises or previous work to validate the proposed strategies.

Simulate the movement of a crowd in a state of emergency is a complex process, not only for its mathematical modelling but also because it should be formalised in the model the behaviour that arises from both the natural interaction between individuals and the reaction group (crowd) against a threat that arises in a particular building.

A primary objective of this work is to develop and select techniques of high performance computing (HPC) in order to execute and perform large-scale complex simulations. In this way, we used partitioning techniques for allowing large scale simulations. The simulation is then divided up with the set of environment and its agents being distributed equally amongst the computer nodes. However, distribution introduced its own challenges. The multicolumn partitioning method reduces the communication and synchronisation time, while ensuring each agent did not perceive contradictions in the environment.

Finally, we believe that the EVAC* system is a good start point for analysing and designing preventive safety policies and therefore, we wish to investigate further to optimise both the model and its parallel implementation.

References

1. Nagel Kai. Particle hopping models and traffic flow theory. *Phys. Rev. E*, 53(5):4655–4672, May 1996.
2. von Neumann Jhon. *The Theory of Self-reproducing Automata*. Univ. of Illinois Press, Urbana, IL, 1966.
3. Gardner Martin. *Mathematical Games: The fantastic combinations of John Conway's new solitaire game "Life"*. Scientific American, 1970.
4. Wolfram Stephen. *Cellular Automata and Complexity*. Addison Wesley, USA, 1994.
5. C Burstedde, K Klauck, A Schadschneider, and J Zittartz. Simulation of pedestrian dynamics using a 2-dimensional cellular automaton. *Physica A: Statistical Mechanics and its Applications*, 295(3-4):507525, 2001.
6. Schreckenberg Michael, Meyer-Knig Tim, and Klpfel Hubert. Simulating mustering and evacuation processes onboard passenger vessels: Model and applications. In *In The 2nd International Symposium on Human Factors On Board (ISHFOB*, 2001.
7. Timmermans Harry. *Pedestrian Behavior: Data Collection and Applications*. Emerald Group Publishing Limited, 2009.
8. Klupfel Hubert, Schreckenberg Michael, and Meyer-Konig Tim. Models for crowd movement and egress simulation. In Serge P. Hoogendoorn, Stefan Luding, Piet H. L. Bovy, Michael Schreckenberg, and Dietrich E. Wolf, editors, *Traffic and Granular Flow 03*, pages 357–372. Springer Berlin Heidelberg, 2005.
9. Tissera Cristian. Simulador de evacuaciones basado en automatas celulares. Informe de tesis de Licenciatura. UNSL, Julio 2006.
10. Brassard G. and Bratley P. *Fundamentos de Algoritmia*. Prentice Hall, 2000.
11. Klupfel Hubert Ludwig. *A Cellular Automaton Model for Crowd Movement and Egress Simulation*. PhD thesis, Universitat Duisburg-Essen, 2003.
12. Kjolstad Fredrik Berg and Snir Marc. Ghost cell pattern. In *Proceedings of the 2010 Workshop on Parallel Programming Patterns*, ParaPLoP '10, pages 4:1–4:9, New York, NY, USA, 2010. ACM.
13. Nishinari K., Kirchner A., Namazi A., and Schadschneider A. Extended floor field ca model for evacuation dynamics. In *IEICE Transactions on Information and Systems, E87-D*, pages 726–732, 2004.
14. Blue Victor J. and Adler Jeffrey L. Cellular automata microsimulation for modeling bi-directional pedestrian walkways. *Transportation Research Part B: Methodological*, 35(3):293–312, March 2001.
15. Tissera Cristian, Printista Marcela, and Errecalde Marcelo. Multi-column partitioning for agent-based ca model. In *HPC Proceeding. JAIIO*. SADIO-Argentina, 2011.
16. Aik Lim Eng. Exit-selection behaviors during a classroom evacuation. *International Journal of the Physical Sciences*, 6(13):3218–3231, 2011. http://www.academicjournals.org/IJPS.