

Scalable Signal-based Simulation of Autonomous Beings in Complex Environments

Mateusz Paciorek, Agata Bogacz, and Wojciech Turek

AGH University of Science and Technology

Abstract. Simulation of groups of autonomous beings poses a great computational challenge in terms of required time and resources. The need to simulate large environments, numerous populations of beings, and to increase the detail of models causes the need for parallelization of computations. The signal-based simulation algorithm, presented in our previous research, prove the possibility of linear scalability of such computations up to thousands of computing cores. In this paper further extensions of the signal-based models are investigated and new method for defining complex environments is presented. It allows efficient and scalable simulation of structures which cannot be defined using two dimensions, like multi-story buildings, anthills or bee hives. The solution is applied for defining a building evacuation model, which is validated using empirical data from a real-life evacuation drill.

Keywords: Agent-based simulation, Autonomous beings simulation · Scalability · High Performance Computing

1 Introduction

The research on modeling and simulation of groups of autonomous beings, like crowds of pedestrians, cars in traffic or swarms of bees, has a great need for efficient simulation methods. The need for simulating numerous groups of beings in large environment, the growing complexity of models and the desire to collect the results fast increase the complexity of the computational task. Therefore, new, scalable simulation methods are constantly pursued by the researchers.

In our previous research [2], a novel method for parallelization of autonomous beings simulation has been presented. It is based on the concept of information propagation, which replaces the need for searching the required data by individual beings. The signal is modeled in a 2D grid of cells, it is directional and autonomously spreads to adjacent cells in the grid. This simple concept removes the need for analyzing remote locations during model update, and therefore, allows splitting the simulated environment between many computing processes, which communicate with fixed number of neighbors once in a single time step, which is the crucial requirement of super-scalable algorithm defined in [5]. The basic version of the method allows representing two-dimensional environments as grids of cells, which are split into equal fragments and updated in parallel. After each time step, the processes updating neighbor fragments exchange

information about signal and beings located in common borders. The method requires defining dedicated simulation models, however, in return it can provide linear scalability of simulation. Linear scalability has been achieved in several tests performed on HPC hardware, which involved up to 3456 computing cores.

Linearly scalable method for autonomous beings simulation encouraged further research on signal-based models and their validation. In [2] three simple models have been presented in order to demonstrate the capabilities of the method. In [9] a model of pedestrian behaviour based on proxemics rules have been developed and tested. Further work on signal-based models for pedestrians led to the problem considered in this paper: the need for modeling complex, multi-level buildings, which cannot be represented in two dimensions. Similar problem can be encountered in modeling of other species habitations, like anthills, bee hives or mole tunnel systems. This class of modeling problems can be addressed by using three-dimensional models, however, efficiency of such approach would be doubtful. Only a small part of the 3D space is accessible and significant so the 3D model would introduce a huge waste in memory and computing resources.

In this paper we propose the extension of the signal propagation method addressing the requirements of environments that cannot be represented by 2D grid. The extension introduces an abstraction over the relation of adjacency, which enables flexibility in defining the shape of the environment while preserving values of distance and direction. This concept is further explored by implementing an evacuation scenario of a multi-story building as an example of such an environment. Finally, we compare the metrics collected during the simulation with the available real-life data to validate the resulting model.

2 Scalability of Autonomous Beings Simulation

The problem of autonomous beings simulation and computer-aided analysis of phenomena taking place in large groups of such beings has been studied for many decades. For example, first significant result of traffic simulation using computers can be found in the fifties of XX century [6]. Since then countless reports on methods for modeling and simulation of different types of beings and different phenomena have been published. Specific problems, like urban traffic or pedestrian dynamics, attracted so much attention, that several different classifications of models can be found in the literature [10, 12, 15]. The taxonomy is based on the considered level of detail [12] can be found in several different problems, where macro-, mezo- and micro-scale models are being distinguished. In recent years the vast majority of research focus on micro-scale models, which distinguish individual entities and allow differences in their characteristics and behavior.

One of the basic decisions which has to be made while defining the model is the method of representing the workspace of the beings. The environment model can be discrete or continuous, it can represent 2 or 3 dimensions. The decision to discretize the workspace significantly simplifies the algorithms for model ex-

ecution, which allows simulating larger groups in bigger environments. In many cases a discrete workspace model is sufficient to represent desired features of the beings and reproduce phenomena observed in real systems, like in the well-recognized Nagel-Schreckenberg freeway traffic model [8]. Many other researchers use inspirations from cellular automata in simulation of different types of beings (swarms of bees [1] or groups of pedestrians [13]) because of simplicity, elegance and sufficient expressiveness.

The common challenge identified in the vast majority of the publications in the area is the performance of the simulations. The need for increasing the complexity of models, simulating larger populations of beings and getting results faster is visible in almost all of the considered approaches. In many cases the performance issues prevent further development, and therefore a lot of effort is being put into creating parallel versions of simulation algorithms and general-purpose simulation frameworks. Scalable communication mechanisms have been added to Repast [4], dedicated frameworks are being built (like Pandora [14]). The results show that by defining models dedicated for parallel execution, scalability can be achieved, like in [3], where almost linear scaling is demonstrated up to 432 processing units with the *FLAME on HPC* framework.

Efficient parallelization of this type of computational task is non-trivial. The algorithm executed in each time step of the simulation operates on a single data structure, which represents the environment state. Parallel access to the data structure requires complex synchronization protocols, which imply significant and non-scalable overhead. Therefore, in our solution presented in [2], we focused on removing the need for accessing the remote parts of the data structure. The modeling method assumes that the information is explicitly pushed to computing units that might need it for updating their state. Implemented *Xinuk* simulation platform proves the scalability of the approach, offering linear scalability up to 3456 cores of a supercomputer. In this paper we present important extensions to the modeling methods supported by the platform, which allow representing complex environments, not possible to model with a 2D grid of cells.

3 Signal-based Model of Complex Environments

The scalable simulation algorithm, implemented by the *Xinuk* framework, is capable of simulating any 2D environment while providing a flexible approach to the interpretation of the grid. Cells might represent a terrain with qualities appropriate for the simulation, actors exclusively occupying a specific place in the grid or a group of actors amassed in one location. Each cell is adjacent to its eight neighbor cells and is able to interact with any of them (e.g. move its contents or a part thereof), if the logic defined by the simulation designer allows such an action. However, the simulations that cannot be represented in simple 2D environment are difficult, if not impossible, to properly model using the framework.

While some 3D layouts can be mapped to 2D grid utilizing simplifications or other compromises (e.g. modelling upward/downward slopes or stairs as spe-

cial cells that modify movement speed or behavior of the agent in such a cell), most terrain configurations would greatly benefit from—or even require—more general solution. One example of such configuration, which will be the main focus of this work, is a multi-story building with staircases located in the center. Each floor can be represented as an independent part of the 2D grid, but the connections between the floors would have to overlap with them. The standard, 2D Moore neighborhood is not sufficient to correctly represent aforementioned environments. One solution would be to generalize the method to represent 3D grid of cells, with each cell having 26 neighbors. However, in the considered class of problems, this approach would result in significant waste of memory and computational resources, as only few of the cells would be important for the simulation results.

From this problem stems the idea of the abstraction of the cell neighborhood. The proposed version of the modeling method introduces a neighborhood mapping for each direction. Each cell can connect to: top, top-right, right, bottom-right, bottom, bottom-left, left and top-left. Given a grid of dimensions $H \times W$, this mapping can be declared as in Equation 1:

$$\begin{aligned} X &: \{0, 1, \dots, H\}, \\ Y &: \{0, 1, \dots, W\}, \\ D &: \{T, TR, R, BR, B, BL, L, TL\}, \\ N &: (X \times Y) \times D \rightarrow (X \times Y) \cup \text{None}, \end{aligned} \tag{1}$$

where $X \times Y$ is a set of all possible coordinates in initial grid, D is a set of mentioned directions and N is a function mapping coordinates and direction to another set of coordinates or *None*, representing the absence of the neighbor in given direction. Likewise, the signal has been updated to be stored as a similar map containing signal strength in given direction. As a result, the signal propagation algorithm required reformulation to make use of the new representation. Firstly, the idea of the function of adjacent direction AD of a direction was necessary, which is presented in Equation 2:

$$AD : D \rightarrow D^2, \quad AD(d) = \begin{cases} \{TL, TR\} & \text{if } d = T, \\ \{T, R\} & \text{if } d = TR, \\ \{TR, BR\} & \text{if } d = R, \\ \{B, R\} & \text{if } d = BR, \\ \{BL, BR\} & \text{if } d = B, \\ \{B, L\} & \text{if } d = BL, \\ \{TL, BL\} & \text{if } d = L, \\ \{T, L\} & \text{if } d = TL \end{cases} \tag{2}$$

With use of this function, and assuming the function S that returns the current signal in the cell at given coordinates in given direction (Equation 3), the new signal propagation function SP can be described as in Equation 4:

$$S : (X \times Y) \times D \rightarrow \mathbb{R}, \quad (3)$$

$$SP : (X \times Y) \times D \rightarrow \mathbb{R},$$

$$SP((x, y), d) =$$

$$SPF \cdot \begin{cases} S(N((x, y), d), d) & \text{if } d \in \{TR, BR, BL, TL\}, \\ S(N((x, y), d), d) + \\ \sum_{ad \in AD(d)} S(N((x, y), d), ad) & \text{if } d \in \{T, R, B, L\} \end{cases} \quad (4)$$

where $SPF \in [0, 1]$ is a global suppression factor of the signal.

It is worth noting that, should the need arise, this mechanism can be extended to any collection of directions, as long as the signal propagation function is updated to properly represent distribution in all directions.

Introduction of the new neighbor resolution allows seamless adaptation of the previous approach: by default, all the neighbors of the cell are the adjacent cells. The neighbor mapping function for such a case is defined as in Equation 5, with an exception of the grid borders, where neighbors are nonexistent (*None*, as in (1)):

$$N((x, y), d) = \begin{cases} (x - 1, y) & \text{if } d = T, \\ (x - 1, y + 1) & \text{if } d = TR, \\ (x, y + 1) & \text{if } d = R, \\ (x + 1, y + 1) & \text{if } d = BR, \\ (x + 1, y) & \text{if } d = B, \\ (x + 1, y - 1) & \text{if } d = BL, \\ (x, y - 1) & \text{if } d = L, \\ (x - 1, y - 1) & \text{if } d = TL \end{cases} \quad (5)$$

Additionally, in the step of the grid creation any neighbor relation can be replaced to represent non-grid connection between cells. While the concept of remote connections appears trivial, it is critical to consider the possibility that in the process of the grid division neighbors are distributed to the separate computational nodes. Such a situation is certain to occur on the line of the division. In our previous approach, we applied buffer zones mechanism as a solution. The new concept required this part of the framework to be redesigned, to allow more flexible cell transfer. As a result, new type of cells was introduced: *remote cells*. Each represents a cell that is not present in the part of grid processed in this worker and contains information regarding:

- the identifier of the worker responsible for processing the part of grid containing target cell,
- the coordinates of target cell,
- the contents of the cell awaiting the transfer to the target cell.

Following each simulation step, contents of all remote cells are sent to their respective workers, using the logic previously associated with the buffer zones.

The modification of the framework did not introduce any alterations in the overall complexity of the simulation process. Communication between processes was not altered and utilizes the same methods as the previous synchronization of the buffer zones. Creation of the grid containing large number of non-standard neighborhood relations does introduce additional cell contents that need to be transmitted to the target worker, however it is the minimal volume of data required for the simulation to be processed.

Summarizing, as a result of all the mentioned modifications, the simulation algorithm acquired the ability to model environments that are not possible to be represented on the 2D grid. Buffer zones mechanism has been abstracted to allow more flexible communication without assuming that the communication can only occur at the grid part borders.

The scalability of the framework has been preserved, since the amount of the data sent between workers remains unchanged for the same simulation model represented in the previous and the proposed approach. It is possible to define more complex communication schemes, however, the number of communication targets remains fixed and relatively low for each worker.

4 Building Evacuation Model

Signal-based methods can be used to simulate evacuations of people from buildings. In such a signal-based model, it is enough to place a signal sources in exits, so beings will move accordingly to egress routes created by signal emitted by the sources, leaving the building. A negative signal can be used to make beings stay away from potential threats, for instance fire or smoke. In the presented model, the repelling signal was used for representing the reluctance for creating large crowds when alternative routes are possible.

In *Xinuk* framework, there are two basic cell types:

- *Obstacle* - a cell that does not let signal through,
- *EmptyCell* - an empty cell traversable by a signal and accessible by any being.

In the proposed model, *Obstacle* type of cells was used to create walls. In addition, the following cells were added to create the evacuation model:

- *PersonCell* - representing a person that was to evacuate,
- *ExitCell* - representing an exit from a building,
- *TeleportationCell* - a remote cell, described in the previous section, that was moving a being to a destination cell,
- *EvacuationDirectionCell* - source of a static signal.

A being of *PersonCell* type was able to move to any accessible adjacent cell, that is an *EmptyCell*, an *ExitCell* or a *TeleportationCell*. Movements were

possible in 8 directions if there were no walls or other people in the neighborhood, as shown in Fig. 1.

In the created model, two types of signals were used:

- static signal field - a snapshot of a signal propagated in a particular number of initial iterations, where cells of *EvacuationDirectionCell* type were the signal sources. Propagated signal created egress routes that were static during the simulation,
- dynamic signal field - signal emitted by moving *PersonCell* beings.

The static signal field can be compared to a floor field described in [11] or potential field [13].

Two different models of evacuating people behaviors were implemented and tested:

- *Moving variant* - always move if a movement is possible.
- *Standing variant* - move only when the best movement is possible.

In the *moving variant*, a person's destination was calculated as follows:

1. signals in directions of neighbor cells were calculated based on dynamic and static signals by summing both signals,
2. calculated signals were sorted in a descending order,
3. from sorted destinations, a first destination was chosen that was currently available (the cell was not occupied by any other person).

In the *standing variant*, a person did not move if the best direction was not available, preventing unnatural movements to directions further away from targeted exit. Thus the 3rd step from the *moving variant* algorithm was changed as follows:

3. from sorted destinations, a first destination was chosen. If the destination was not available, the being would not move.

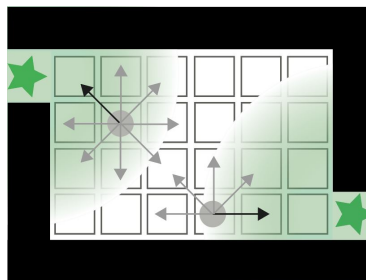


Fig. 1. Possible directions of beings. Beings marked as circles, signal sources marked as stars. Round gradient around signal sources symbolizes propagation of signal - the further from a signal source, the weaker the signal is

In a high congestion of beings trying to get to a particular location, conflicts are highly possible. One solution to this problem is to let two beings enter one cell, creating a *crowd*. A simpler solution, implemented in our model, is to check if a destination cell chosen by a being is empty both in current and in the next iteration. This way, all conflicts will be avoided.

Each floor of a building was mapped onto a 2D grid reflecting its shape and dimensions. To simulate a multi-story buildings using 2D grids, a special *TeleportationCell* was created. A being that entered the *TeleportationCell* at the end of a staircase on a floor N , was moved to a cell corresponding to the beginning of a staircase at floor $N - 1$.

5 Model Validation

To validate the evacuation model, a simulation of a real-life evacuation described in [7] was implemented. The evacuation drill took place in a 14-story tower connected to a 3-story low-rise structure. The highest floor of the tower was ignored in the research, thus in the implementation there is no *XII* floor in the tower, as shown on the building plan (Fig. 2). The highest floor of a low-rise structure was ignored as well and the data focused on the tower, which is a reason why the low-rise structure is shown as one floor on the building plan. Each floor was connected to two staircases, each 0,91m wide, allowing only a single line of pedestrians to be created. The staircase length was not mentioned in the paper.

Evacuation started earlier on 3 floors: *V*, *VI* and *VII*. After 3 minutes, there was a general alarm on the remaining floors. Pre-evacuation time curve was approximately linear (Fig. 6 in [7]). Evacuation rate was over one person per second for the first 75% of people in the building. Afterwards, the rate was slightly smaller because of discontinuation of use of one of exits (the reason not explained in the article). Results from the drill can be seen in Fig. 6.

Based on the above data, an implementation of evacuation model described in the previous section was created in *Xinuk* platform. Crucial parameters were set as follows:

- *gridSize* = 264; size of a grid. The whole grid was a square of 264 x 264 cells, each cell representing a square with side length of 0,91m.
- *iterationsNumber* = 1000; number of iterations completed in a single simulation. 1 iteration was corresponding to 1 second.
- *evacuationDirectionInitialSignal* = 1; signal intensity that was emitted by *EvacuationDirectionCell*.
- *personInitialSignal* = -0.036; signal intensity that was emitted by *PersonCell*. In contrast to *evacuationDirectionInitialSignal*, the value was negative so people were repelling each other slightly.

In the simulation, 1 iteration corresponds to 1 second of an evacuation. Number of people on each floor was set accordingly to *Table 1* from the source paper. The

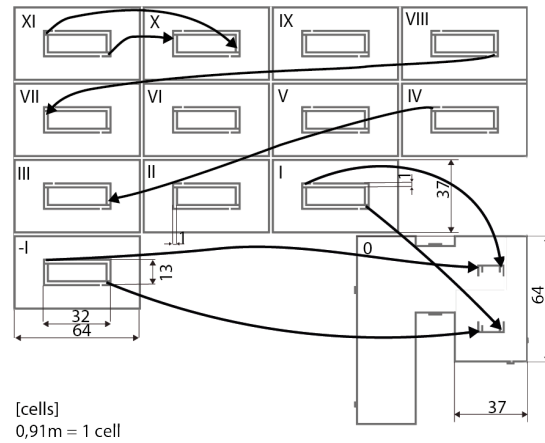


Fig. 2. Plan of the building described in [7] mapped onto a 2D grid. Arrows show transitions of people at the end of corridors - once a person reaches a corridor's end cell, it is moved to the corresponding corridor on the neighboring floor. For clarity, only a few arrows are drawn - remaining transitions were analogous.

number of people placed on low-rise structure's floor was equal to the number of all people in low-rise structure stated in the source paper.

The implemented simulation had three phases:

- *Phase 1* - 1st to 18th iteration - creating static signal field by placing signal sources in the cells corresponding to the exits and corridors' ends,
- *Phase 2* - 19th to 199th iteration - evacuation after the initial alarm, evacuating people from *V*, *VI* and *VII* floors,
- *Phase 3* - 200th to 1000th iteration - evacuation after the general alarm, evacuating people from the whole building.

To achieve linear pre-evacuation times, all of the people were not placed on a grid in the first iteration of the 2nd and 3rd phase of the simulation, but they were placed on a grid linearly - one person per iteration on each floor (relying on the Fig. 6 in [7]).

Validation of the model was based on visual analysis of simulation results and on the *evacuation curve* metric which is a dependency between the number of people evacuated from the building and time. Empirical data is shown in Fig. 6.

During the observations, two anomalies were visible:

1. **Problem 1:** Crowds of people next to the upper and lower entrance of staircases on each floor were consistently different—*lower* entrance seemed to generate larger crowds—and an evacuation using *upper* stairs tended to take longer (Fig. 3),

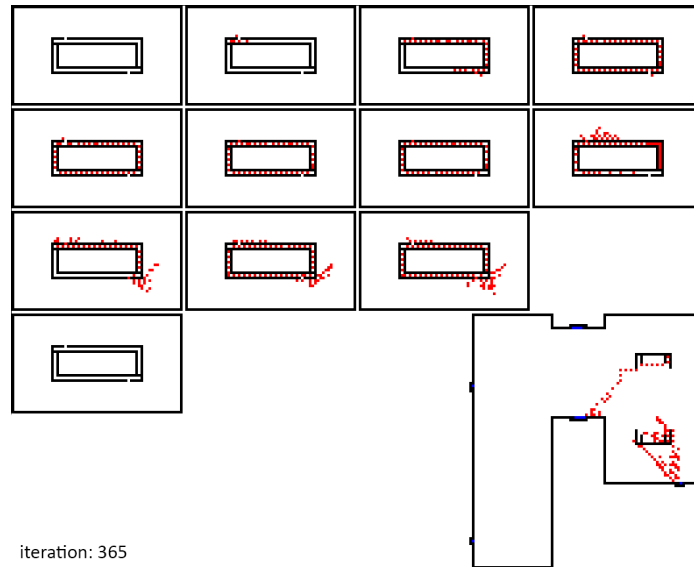


Fig. 3. Arrangement of people in the building using a sequential order of cells updating (with moving variant). Iteration: 365. Link to the full video: <https://bit.ly/2Q081HA>

2. **Problem 2:** People in corridors that could not move forward but could go back, were moving back and forth waiting to go further or were choosing to go back to staircase entrances.

Problem 1 was a result of a sequential order of updating cells. A person that was trying to enter an upper staircase was above the people in the staircase, thus the person was updated before the people in the corridor. This way, the person's movement was favored and congestion was visible in the corridors and there were no crowds close to the upper entrances to the upper staircases. Similarly, trying to enter a lower staircase, the movement of people already in the staircase was favored. A person trying to enter the staircase was updated after people in the staircase managed to occupy all empty spaces, preventing the person from entering the staircase. Fig. 5 shows both of the situations. A simple solution to this problem was to update locations of people in a random order. Fig. 4 shows the results of such an approach. Crowds are distributed equally close to both of staircase entrances and evacuation in both staircases progresses similarly.

Problem 2 was a result of the decision making algorithm. According to the algorithm, people would make any movement if it was possible, even a bad one, rather than stay in place. A solution was to use the *Standing variant*: choose only the most attractive destination or not to move at all.

The visual analysis of the final model behavior is satisfactory. The formation of crowds and selection of the exits resembled the expected phenomena. The people were eager to follow the shortest path towards the exits, while avoiding excessive crowding which might lead to trampling.

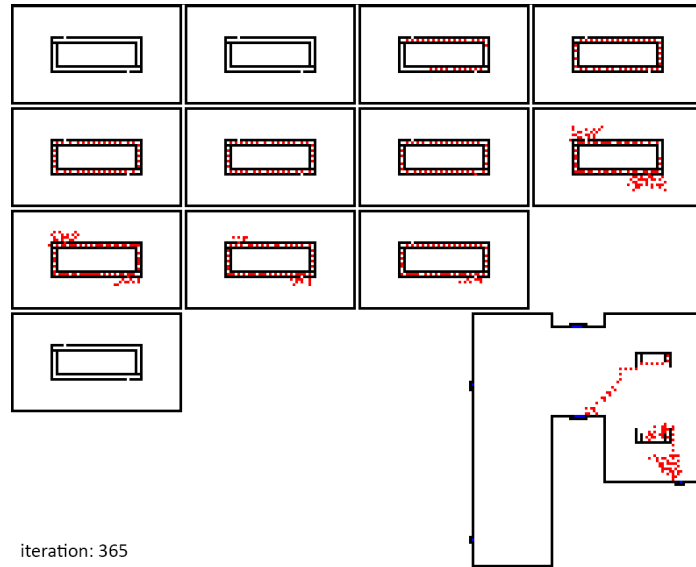


Fig. 4. Arrangement of people in the building using a random order of cells updating (with moving variant). Iteration: 365. Link to the full video: <https://bit.ly/2MX6S08>

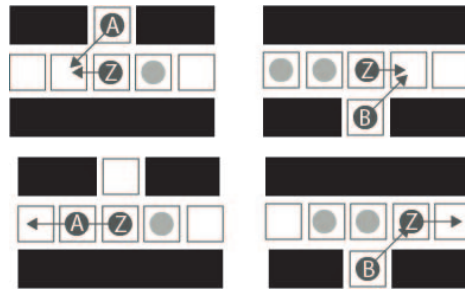


Fig. 5. Favoring movements of people that are upper in a grid in sequential way of updating cells. On the left - a situation when person A is trying to enter a staircase that is below. On the right - a situation when person B is trying to enter a staircase that is above

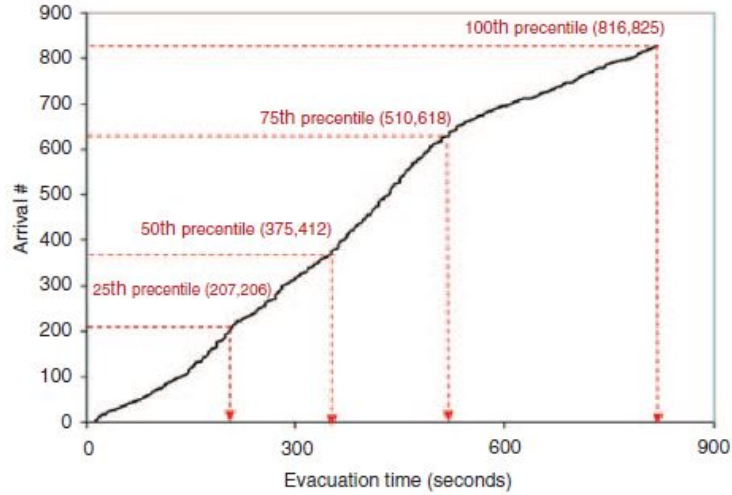


Fig. 6. Number of people evacuated in time, empirical results from [7]

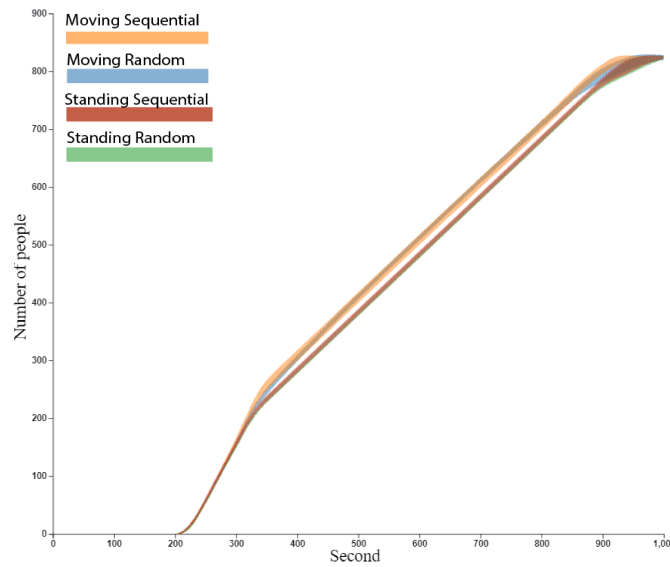


Fig. 7. Evacuation curves for four tested model update algorithms. The ribbons represent the median of 30 runs.

The four combinations of the model update algorithm, *moving/standing variant* and *sequential/random variant*, were executed 30 times. We used the Prometheus supercomputer (part of PL-Grid infrastructure¹), which allowed completion of all computation within several minutes. The resulting chart (Fig. 7) shows that this particular metrics is not influenced significantly by the selected variant. An average rate of people leaving the building is a little over 1 person per second, which matches the data on 6. After an evacuation of 75% of people, the rate did not change, as in the simulation people continued using two exits. On a source chart (Fig. 6) people have already reached exits in first seconds while on the resulting chart (Fig. 7) the first person left the building after 200th second of evacuation. According to [7], people on the floors other than V, VI and VII did not evacuate before the general alarm. Thus, it is not clear why the chart shown in Fig. 6 suggests that some people have left the building in such a short time overcoming at least 5 floor long stairs.

The results of the experiments are not perfectly matching the empirical data from [7]. Nonetheless, taking into consideration the ambiguity, contradictions or lack of some details of the described evacuation, we conclude that the resulting model yielded realistic and consistent outcomes. In the context of validation of the presented method as a building block of similar environments, the results are satisfactory.

6 Conclusions and Further Work

In this paper we proposed an extension to the signal propagation modeling method and the Xinuk framework, addressing the limitations of the existing approach in the context of complex 3D environments. We introduced the alternative to the standard, grid-based neighborhood as a means to generalizing the idea of buffer zones present in the framework. The new mechanism greatly increased the flexibility of the method, while avoiding the massive growth of complexity that would result from switching to full 3D grid environment. At the same time, the scalability of the original solution remained unhindered, as the alteration did not increase the amount of data exchanged between computational nodes in the synchronization step. The evacuation scenario used as a demonstration of the new capabilities of the method provided results confirming that such a scenario can be accurately represented in the resulting framework.

As a followup of this work, we intend to further explore the possibilities arising from the flexible neighborhood declaration, especially the environments that would benefit from additional directions, e.g. layers of an anthill utilizing the vertical directions. Research dedicated to further reduction in the communication might yield interesting results as well, e.g. investigating a trade-off between accuracy and performance of the system while performing the synchronization after several simulation steps.

¹ <http://www.plgrid.pl/en>

Acknowledgments

The research presented in this paper was supported by the Polish Ministry of Science and Higher Education funds assigned to AGH University of Science and Technology. The authors acknowledge using the PL-Grid infrastructure.

References

1. Becher, M.A., Grimm, V., Thorbek, P., Horn, J., Kennedy, P.J., Osborne, J.L.: *Beehave: a systems model of honeybee colony dynamics and foraging to explore multifactorial causes of colony failure*. *Journal of Applied Ecology* **51**(2), 470–482 (2014)
2. Bujas, J., Dworak, D., Turek, W., Byrski, A.: *High-performance computing framework with desynchronized information propagation for large-scale simulations*. *Journal of Computational Science* **32**, 70–86 (2019)
3. Coakley, S., Gheorghe, M., Holcombe, M., Chin, S., Worth, D., Greenough, C.: *Exploitation of high performance computing in the flame agent-based simulation framework*. In: *2012 IEEE 14th International Conference on High Performance Computing and Communication & 2012 IEEE 9th International Conference on Embedded Software and Systems*. pp. 538–545. IEEE (2012)
4. Collier, N., North, M.: *Repast hpc: A platform for large-scale agent-based modeling*. *Large-Scale Computing* pp. 81–109 (2012)
5. Engelmann, C., Geist, A.: *Super-scalable algorithms for computing on 100,000 processors*. In: *International Conference on Computational Science*. pp. 313–321. Springer (2005)
6. Gerlough, D.L.: *Simulation of freeway traffic on a general-purpose discrete variable computer*. Ph.D. thesis, University of California, Los Angeles (1955)
7. Gwynne, S., Boswell, D.: *Pre-evacuation data collected from a mid-rise evacuation exercise*. *Journal of fire protection engineering* **19**(1), 5–29 (2009)
8. Nagel, K., Schreckenberg, M.: *A cellular automaton model for freeway traffic*. *Journal de physique I* **2**(12), 2221–2229 (1992)
9. Renc, P., Bielech, M., Pęcak, T., Morawiecki, P., Paciorek, M., Turek, W., Byrski, A., Wąs, J.: *Hpc large-scale pedestrian simulation based on proxemics rules*. In: *3th International Conference on Parallel Processing and Applied Mathematics*. Springer (2020)
10. Sargent, R.G.: *Verification and validation of simulation models*. In: *Proceedings of the 2010 Winter Simulation Conference*. pp. 166–183. IEEE (2010)
11. Varasa, A., Cornejoa, M.D., Mainemera, D., Toledob, B., Rogan, J., Munoz, V., Valdivia, J.A.: *Cellular automaton model for evacuation process with obstacles*. *Physica A: Statistical Mechanics and its Applications (PHYSICA A)* **382**, 631–642 (2007)
12. van Wageningen, F., Van Lint, H., Vuik, K., Hoogendoorn, S.: *Genealogy of traffic flow models*. *EURO Journal on Transportation and Logistics* **4**(4), 445–473 (2015)
13. Wąs, J., Lubaś, R.: *Towards realistic and effective agent-based models of crowd dynamics*. *Neurocomputing* **146**, 199–209 (2014)
14. Wittek, P., Rubio-Campillo, X.: *Scalable agent-based modelling with cloud hpc resources for social simulations*. In: *4th IEEE International Conference on Cloud Computing Technology and Science Proceedings*. pp. 355–362. IEEE (2012)
15. Zhan, B., Monekosso, D.N., Remagnino, P., Velastin, S.A., Xu, L.Q.: *Crowd analysis: a survey*. *Machine Vision and Applications* **19**(5-6), 345–357 (2008)