J Supercomput (2016) 72:2170–2196
DOI 10.1007/s11227-016-1718-7

Cellular Automata as the basis of effective and realistic agent-based models of crowd behavior

Robert Lubaś[1] · Jarosław Wąs[1] · Jakub Porzycki[1]

Published online: 26 April 2016

Abstract The Cellular Automata (CA) paradigm has been recognized as an effective approach used in the modeling and simulation of complex systems. However, its classical form of a homogeneous and synchronous CA has a limited field of applications. For practical applications, non-homogeneous and asynchronous CAs with hybrid technological construction are especially useful in modeling and simulation. In this article, the authors focus on crowd simulations based on CA and agent-based modeling approaches. Basic technical aspects of large-scale crowd simulations are presented: specifically proposed architecture, our view on synchronization patterns, as well as hierarchy of objects in logic and data layer. A new method of agent conflict resolution is also proposed. Such an approach was successfully applied in the Allianz Arena stadium model, and other large-scale simulations developed by the authors. Thus, finally, practical applications of the models are presented.

Keywords Cellular Automata · Crowd model · Agent-based simulation · Pedestrian dynamics · Data-driven simulation

✉ Robert Lubaś
rlubas@agh.edu.pl

Jarosław Wąs
jarek@agh.edu.pl

Jakub Porzycki
porzycki@agh.edu.pl

[1] Department of Applied Computer Science, AGH University of Science and Technology, 30 Mickiewicza Av, 30-059 Kraków, Poland

1 Introduction

Recent decades have seen a growing interest in crowd modeling. Currently, we can distinguish two different classes of crowd dynamics simulations: those dedicated to entertainment applications and those dedicated to engineering purposes [31,38]. The main aims of the first group are impressive and effective visualization and representation of a variety of entities' behaviors. On the other hand, the second group is devoted to specific applications like evacuation modeling, safety engineering, optimizing architectural plans or managing public gatherings.

Simulations for engineering purposes require compliance of the model with the physical parameters of pedestrian dynamics and are subjected to strict verification and validation using ISO and other international or national standards. The main objective of *engineering* crowd simulation is creating most accurate possible representation of the actual behavior and dynamics of people in specific situations.

It should be stressed that our focus is on the second class of crowd models. For this reason, we define the reliability of a crowd model as compliance of the model with reference data [31]. Each model should be validated and verified using qualitative and quantitative criteria [18,21,29].

Due to the fact that situations involving large gatherings are more and more frequent, architects, safety engineers and other people responsible for safety planning must cope with the growing safety requirements. Therefore, we can observe an upward trend in creating reliable computer crowd simulations, which make it possible to test different scenarios and numerous variants of space arrangement. Fast simulations are also required in order to quickly prepare multi-variant analysis or/and real-time decision support systems [33].

Cellular Automata (CA) have been successfully used in the modeling and simulation of crowd dynamics and behavior in recent years. In particular, non-homogeneous and asynchronous CA are useful for such purposes.

According to Kauffman, a non-homogeneous (inhomogeneous) cellular automaton is defined as a cellular automaton with a lattice made of non-homogeneous cells following different transition rules [20]. This means that the cells are *not qualitatively identical*. Analogically, the cell state updating in the whole lattice can be asynchronous. In such a case, the updating is performed in a predetermined order [17]. We believe that non-homogeneous, asynchronous and probabilistic Cellular Automata provide an excellent basis for agent-based simulations of complex systems. In particular, this kind of CA is useful for large-scale simulations of crowd dynamics and testing complex scenarios [38].

We believe that it is important to distinguish between the classical CA model of crowd dynamics and the agent-based CA model. From our perspective, classical CA models take into account only the operational and tactical level in pedestrian decision-making [7]. It is only possible to represent simple scenarios like large room evacuation, passing through a bottleneck or simple evacuation of a facility, etc. Such models are able to mimic only stereotypical pedestrian reactions like omitting obstacles or predecessor following. That approach is perfectly adequate for many practical applications (standard model of facility evacuation); however, it has been identified as insufficient for some purposes, namely more complicated patterns of behavior (see comments in [30]).

We understand that the agent-based CA model represents the operational (stereotypical reactions), tactical (short-time decisions) and strategic (activity pattern choice) levels of pedestrian decision-making using *high-level description of complex rules of non-homogeneous CA*. In such an approach, we can build more complex scenarios, taking into account the influence of the information spread, unexpected events or differences in agents' perception. Agents/pedestrians are situated in a CA lattice environment (the idea of Situated Cellular Agents [2]). An agent makes decisions, and finally moves according to, respectively, his/her current position on the lattice, the behavior of different agents, the configuration of obstacles and, in general, the local, as well as global, configuration of the lattice. In practice this means that some strategic, tactical or operational level functions are strictly connected with types of non-homogeneity in Cellular Automata (for instance, the action of accessing cell type *stairs* initializes a new, specific local transition function, whilst accessing a cell of the specific type *selection* activates global decision function).

Another important issue is efficiency of simulation, understood as its performance. In [24] we have analyzed the efficiency of the most popular crowd simulations, namely the hydrodynamic–macroscopic approach, the force-based approach and the CA-based approach. The most efficient method is of course the hydrodynamic approach [24], whilst the force-based method of Social Forces is rather inefficient (performance comparison is presented in Fig. 21). It should be stressed that the classical, "pure" CA approach [4] is efficient; however, its capacity to reliably represent complex scenarios is limited. For this reason, we are developing new CA-based models based on social distances representation [36], in order to efficiently simulate complex behavior in crowd movement.

This article is a continuation of previous works by the authors (namely: [37], [38], [24], etc.). The current article concludes our current project. In this article we discuss basic issues of agent-based CA modeling of crowd dynamics, we propose a new method of agent conflict resolution, as well as we present important technical aspects of large-scale crowd simulations (architecture, synchronization patterns and hierarchy of objects in logic and data layer). Finally, we present practical applications of our approach in a multi-variant simulation of evacuation of the Allianz Arena Stadium in Munich, Germany and the Wisla Krakow Stadium, Poland, as well as some experiments performed at the AGH university campus.

The article is organized as follows: Sect. 2 reviews related work on crowd modeling. Section 3 describes the proposed approach using agent-based modeling combined with non-homogeneous Cellular Automata. Section 4 addresses some technical aspects of the developed simulation framework. Sample results are presented and discussed in Sect. 5, while Sect. 6 concludes the article.

2 Related works

The application of Cellular Automata in crowd dynamics modeling is well-known approach. Probably the first application was proposed in 1999 by Fukui and

Ishibashi [12]. They mapped a bidirectional motion in a long corridor, where particles moving in opposite directions were updated alternatingly. Another model was also proposed in 1999—namely the CA-based model proposed by Blue and Adler [3]. This model was similar to the well-known Nagel–Schreckenberg (Na-Sch) model devoted to highway car traffic. In the model, pedestrian movement was considered as analogous to a multi-lane highway, taking into account unidirectional and bidirectional schemes of pedestrian movement. Another pioneering work in the application of CA in crowd simulation was devoted to the evacuation of ships [23]. One should also note a work that raises issues of phase transition and counter flow effect [25].

An agent-based approach in crowd simulations has been widely described in a paper by Musse and Thalmann [26], where a generalized hierarchical approach to crowd dynamics was presented. However, the understanding of crowd simulation presented in the article is much closer to high-level cognitive simulations of crowd than engineering-directed ones.

One proposal for combining Cellular Automata with the multi-agent approach was proposed by Bandini, Manzoni and Vizzari [2] as a framework of Situated Cellular Agents (SCA). The model constitutes a valuable, high-level description of agent-based systems dedicated to crowd models. From our perspective, the SCA model is suitable for a general description of engineering class crowd simulations, because it refers to dedicated layers, floor fields, etc.

An important idea in the field of CA crowd simulation was the introduction of floor fields. This idea was first proposed in [4] and was successfully developed in [27]. In the basic model two kinds of floor fields were applied, static and dynamic. The static floor field is generated from a single POI (point of interest) or a group of POIs to all cells of the lattice. The application of the static floor field constitutes the main mechanism of pedestrian decision-making and movement, while the dynamic floor field is devoted to introducing the mechanism of predecessors following during pedestrian movement. It should be stressed that the dynamic floor field was adapted from a process of chemotaxis (a mechanism of predecessor following in insect population).

In recent years several models were proposed that use a hybrid approach, which combines CA with the application of forces. In 2006 combining Cellular Automata and the influence of forces between agents was proposed in the initial version of the social distances model [36].[1] Recently, agent-based version of the social distances model was proposed [37,38]. Another approach was proposed by Henein and White in 2007, in which pedestrians/agents represented by cellular agents are affected by a floor field modified by forces.

Recently, another kind of floor field—devoted to the smooth avoidance of obstacles was presented in [14]. The proposed solution is based on the generation of a virtual field along obstacles. A pedestrian overcomes the obstacle towards the direction that the field increases its values, leading her/him to avoid the obstacle effectively.

In [10] a process of space acquisition by pedestrians using a Proxemic Floor Field (PFF) is proposed. The authors consider the inflow process of pedestrians (entering a new area). PFF is applied in order to reproduce the effect of a repulsion force between

[1] Currently the model is developed without a force component due to limited efficiency [37].

agents, while in [16] the idea of waiting list and bounds was defined, in order to allow an agent to select an occupied cell. The proposal of blocking agents that want to move into occupied cell reflects the observed line formation and spontaneous queuing, typical for crowd movement with negligible effect of panic.

A comprehensive study devoted to the analysis of different elements of a floor field model of crowd is presented in [15]. The author takes into account different types of applied neighborhood, proxemics-like effects in the context of the model's reliability.

An interesting proposal to applying ambient devices to crowd simulation was suggested in [40]. In the article, the authors augmented the models specifications with behavioral data (evidence) provided by sensing modules attached to individuals. They named these mixed-level models.

In one of the latest works on microscopic crowd modeling [8], a comparison between CA-based and force-based crowd modeling is presented. The authors point out some characteristic features of both types of models and discuss the issue of bridging the gap between them. Specifically, they propose applying continuity to CA models (using different pedestrians' step length) and the parallel application of navigation based on discrete patterns in force-based models. Quantitative comparison of different types of crowd models was presented in [24] (macroscopic, force-based and CA-based), whilst a different set was tested in [32] (Lattice Gas, Social Force and RVO—Reciprocal Velocity Obstacles).

3 Proposed approach

3.1 Representation of pedestrians

In classical Cellular Automata models [30] pedestrians are represented as a particular *State* of a cell (Fig. 1, on the left). Thus, each pedestrian is assigned, in a given time step, to a square cell sized 40 cm (due to maximal crowd density [39]). This approach makes it possible to create different transition functions for pedestrians in a simple manner, but this popular representation is not always enough, due to a relatively coarse representation of space [30].

Alternatively, we have proposed a different approach towards entity representation. This is based on the social distances model [35] and was later developed in [37]. It should be noted that this representation can be generalized by using different sizes of pedestrians (grouped in various classes).

In such a case each pedestrian/agent is assigned to a particular cell of a fine-grained lattice (for instance 25 cm × 25 cm), but due to geometrical dependencies, he/she forms a local configuration of occupied cells (Fig. 1, on the right). In the social distances model [37,38] individuals are represented by ellipses, with reference to WHO data average size, and it comes down to only 14 combinations of local configurations.

Mathematical aspects of CA-based representation of pedestrians are discussed in [9]. In our case we define automaton as a tuple $CA = (L, S, N, f)$: L is the square lattice of CA including different types of cells *Movement Space* (including for instance *Stairs*, *Escalators*, *Sound Fields*, *Visibility Fields Obstacles*), etc.,

S is the set of states including presence of an agent (type of agent) and physical orientation of his/her ellipse [37],
N is the Moore neighborhood,
f is the transition function related with the type of agent, and the type of occupied cell

We define an agent as a following tuple $A_j = (\tau_j, (x_j, y_j), R_j)$, where τ_j is the type of agent j; x_j, y_j are the coordinates of agent j and R_j is the strategic, tactical and operational ability of agent j.

3.2 Decision-making and activity algorithms

Generally, in the presented approach, pedestrian decision-making is executed using a two-level process (as proposed in [38]). The first phase is connected with the strategical level and the second phase is connected with the tactical/operational level (Fig. 2).

3.2.1 Strategic level

In order to develop more complex scenarios in crowd modeling, it is very important to take into account the strategic level of decision-making. A large majority of professional evacuation models only take into consideration simple scenario of pedestrian movement through a system of evacuation routes using only the operational or eventually tactical level [7,30]. However, in real situations, there are many factors affecting agents' decision-making process. This issue was emphasized in 7FP EU Project *Socionical* [40], where the influence of spreading information by mobile devices was taken into account. Therefore, we consider factors like spreading infor-

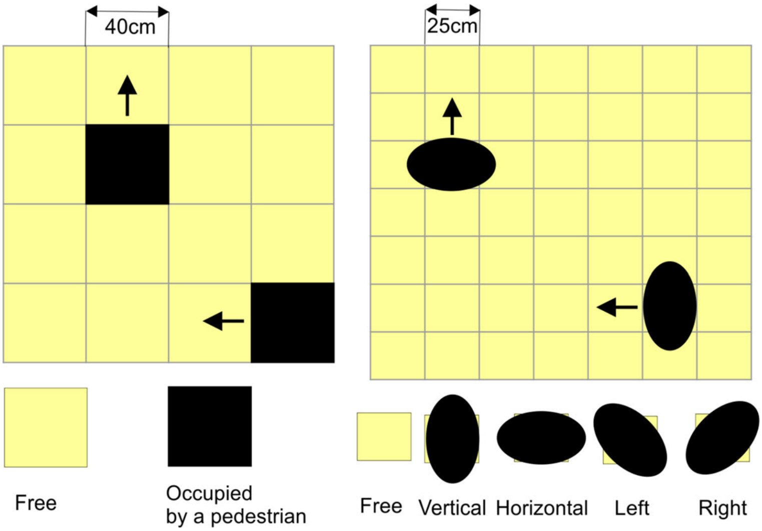


Fig. 1 Physical representation of pedestrians in the classical CA-based model (on the *left*) and the social distances model (on the *right*)

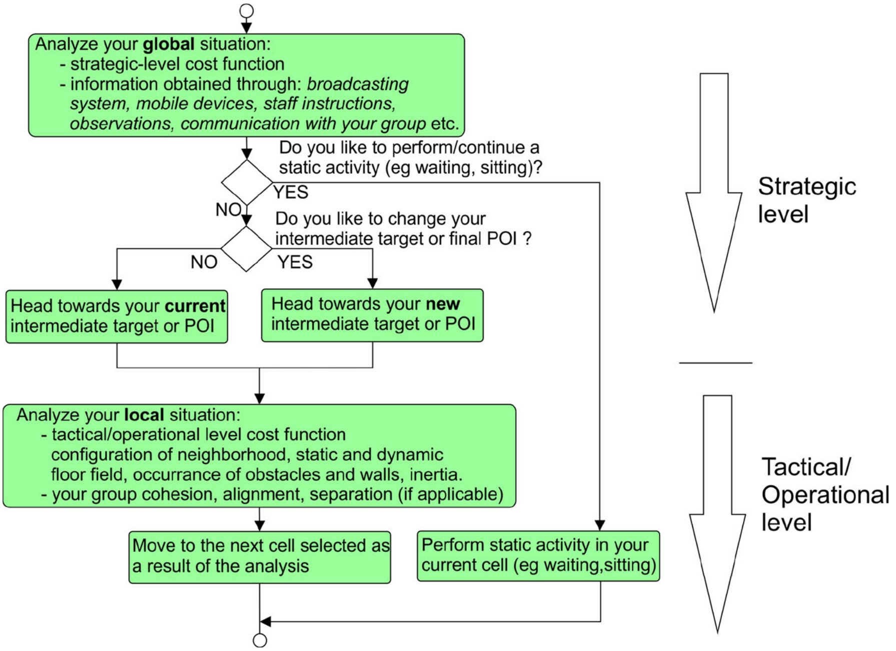

Analyze your **global** situation:
- strategic-level cost function
- information obtained through: *broadcasting system, mobile devices, staff instructions, observations, communication with your group* etc.
Do you like to perform/continue a static activity (eg waiting, sitting)?
YES
NO
Do you like to change your intermediate target or final POI ?
NO
YES
Head towards your **current** intermediate target or POI
Head towards your **new** intermediate target or POI
Analyze your **local** situation:
- tactical/operational level cost function configuration of neighborhood, static and dynamic floor field, occurrance of obstacles and walls, inertia.
- your group cohesion, alignment, separation (if applicable)
Move to the next cell selected as a result of the analysis
Perform static activity in your current cell (eg waiting,sitting)
Strategic level
Tactical/ Operational level

Fig. 2 Global algorithm for decision-making and activity selection by a particular agent, according to [38]

mation through a broadcasting system, instructions given by staff, and communication and coherence of groups.

We strongly believe that a non-homogeneous CA mechanism is very helpful in such a situation. Due to the required efficiency of a simulation, we propose to define special types of cells $\{Selection\}$ which belong to the cells class $\{MovementSpace\}$ [38] on the basis of rules, for instance *at the end of corridor, just after entering the hall, people tend to select the final target*. There is no justification for making global decisions at each time-step of the simulation. For instance, thanks to the non-homogeneity of the cells of a cellular automaton, some cells can be considered as within good audible range of loud speakers. The sound sources have numerous kinds of wave propagation shapes. We assume that a cell-sound generator generates omnidirectional wave shape. The cells in range of sound generator–SF (sound fields) can be defined by a simplified formula:

$$\mathrm{SF}_{x,y} = \{(x', y') : (x' - x)^2 + (y' - y)^2 \leq r^2\} \tag{1}$$

where x, y are the coordinates of sound source cell, r is the $omnidirectional$ wave radius.

When the pedestrian enters into the sound fields area s/he is able to listen to a message with probability P_{SF} and a global decision can be made by agent (e.g., change emergency exit, decrease speed or even stop). The probability P_{SF} can vary and depends on the type broadcast message. For $Type_{SF} = \{noncritical, critical\}$ we define probability as P_{SF}.

Another important possibility for decision-making at the strategic level is communication inside groups of agents or particular agents (direct communication in close proximity, or remotely using mobile devices), and finally a global decision can be changed/made.

We define group G_i as a following tuple (L_i, R_i, S_i):

$$L_i = (x_i, y_i) \tag{2a}$$

$$R_i = (r_1, r_2, \ldots, r_n)_i \tag{2b}$$

$$S_i \in \{2, 3, 4, \ldots, \} \tag{2c}$$

where L_i is the leader of group i; x, y are the coordinates of the leader of the group; R_i is the rule of group i; S_i is the size of group i; i is the group index, $i \in \{1, 2, 3, \ldots, m\}$; r is the rule and n is the number of rules.

A pedestrian is assigned to a group with probability P_{GB} at the beginning of the simulation, only if the distance D between him/her and leader is less than some assumed value. If the distance D is more than assumed value with probability P_{GR} pedestrian remains in the group. Depending on the scenario (the degree of congestion of the environment) the distance D may vary:

$$D = f(\rho) \tag{3}$$

where $f(\rho) = \frac{\text{Dens}_{\text{max}}}{\rho}$; ρ is the density of pedestrians per square meter around the leader of the group; $\rho \in < 0; \text{Dens}_{\text{max}} >$ and Dens_{max} is the maximal density allowed by model.

Below we present sample rules r_i:

- "Follow the leader"—r_1 A pedestrian in a group following this rule select the next destination cell in movement algorithm as follows: try to move to the cell where the leader was or to the cell in the radius of 2 (in leader's Moore neighborhood).
- "No overtaking"—r_2 Members of the group have at most such desired speed as leader.

Such functionality and proposed application of CA with an agent-based approach makes it possible to develop more complex scenarios in crowds [2,38]. It should be stressed that after the strategic phase of decision-making (e.g., selecting a new POI), the behavior of particular agents is mimicked according to operational/tactical patterns explained in the next section.

As an example of decision-making at the strategic level, we present the changing behavior of pedestrians in a specific stadium environment, applied in a simulation of evacuation from the Allianz Arena stadium. During the evacuation simulation, agents were informed that the nearest exit is blocked and they should slow down or even stop the evacuation process until new information arrived. This situation is presented in Fig. 3. Agents changed behavior by slowing down from mean value equals 1.34 to 0.5 m/s.

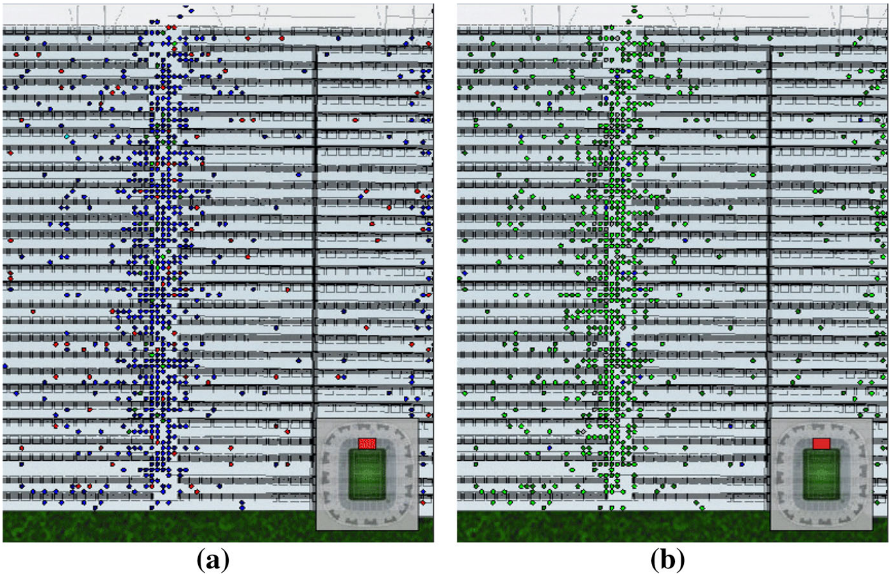


Fig. 3 A view on evacuated sectors located in the lowest level of Allianz Arena stadium. An example of decision-making at the strategic level in the simulation of Allianz Arena evacuation. Agents' desired velocities were decreased due to information about problems on the evacuation route. The desired velocity was coded using color: in **a** *darker colors* represent higher velocities and in **b** *lighter colors* mean lower velocities (color figure online)

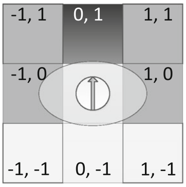
3.2.2 Tactical and operational level

This subsection describes in detail an algorithm for the tactical and operational level. The algorithm is composed of two main stages: decision-making and conflict resolving.

The first stage consists of an agent decision-making process about the next step in the simulation (choosing the next cell $c_{i,j}$). In this process, the following elements are taken into account: static S and dynamic D floor fields, neighborhood configuration, and the presence of walls, obstacles, etc. An asynchronous method of updating cellular automaton stages is applied. We build the priority list based, mainly, on distance to the global POI. We can apply different agent/pedestrian scheduling rules in the list. For instance: the smaller the distance to the POI the higher priority the pedestrian receives. Different priorities (staff, firefighters, etc.) can also be applied. The decision-making process about the movement to the next cell starts from the pedestrian with the highest priority. If there is more than one pedestrian with the same priority, the updating precedence is selected randomly.

The gradient of the static floor field leads a pedestrian to his/her selected point of interest (POI), e.g., the nearest emergency exit. The pedestrian obtains information about the approximate direction of motion. The dynamic floor field introduced by Nishinari in [27] provides knowledge for the pedestrian about his predecessors in the simulation (a "virtual tail"). In particular, we have applied a dynamic floor field with some modification to reproduce the zipper-effect, and to achieve better, closer to reality, space acquisition in movement around corners. These modifications depend

Fig. 4 Pedestrian occupies center cell (0, 0), *dark gray* cell is one with the smallest potential value (dx, dy) = (0, 1), light and *dark gray* cells are visibility fields computed by Eq. 4



on changing the neighborhood from von Neumann to Moore. The process of diffusion and decay provides values which are used in the cost function (Eq. 5). Thanks to this, the pedestrians tend to avoid directly following (overlapping) their predecessors.

The pedestrian receives information about the current neighborhood configuration from a subset of the Moore neighborhood with radius 1, restricted (limited) by the pedestrian's field of vision. The concept of a field of vision and corresponding visibility fields *VF* are briefly described in [37], below we introduce formalism and visualization (see Fig. 4) of this concept:

$$VF_{x,y} = \left\{ (x', y') : |x' - x| \leq r \wedge |y' - y| \leq r \wedge \arccos\left(\frac{a \cdot b}{|a||b|} \right) \leq 90^\circ \right\} \qquad (4)$$

where x, y are the coordinates of center cell; r is the Moore neighborhood radius; dx, dy are the coordinates of cell with the smallest potential value; a is the vector $(x'y')$ and b is the vector (dx, dy).

Until this point, the agent has knowledge about a set of visible fields and a choice of movement directions. The next phase is the decision about which cells in the *VF* set are available (without walls or obstacles, and without the presence of another pedestrian) with respect to available/forbidden configurations [35,36].

The decision-making process includes the selection of the next cell (a cell c_{ij} from a Moore neighborhood) during movement towards the chosen emergency exit. The proposed cost function (Eq. 5) calculates the cost of potential movement from the current cell to all available cells from the *VF* set. The cell with the smallest cost is chosen for the next step in the simulation.

$$\begin{aligned} \mathrm{Cost}(c_{ij}) &= S^a_{ij} + (\mathrm{dens}(c_{ij}) + \alpha \cdot \mathrm{dist}(c_{ij}, POI)) \cdot W \cdot I \\ \mathrm{Dens}(c_{ij}) &= e^{\delta \cdot D_{ij}} \end{aligned} \qquad (5)$$

where S^a_{ij} is the value of a static potential field, where a is an index of a potential field leading to a POI (for instance an emergency exit); $a \geq 1$ is the number of POIs (for instance emergency exits); $\alpha \in [0, \infty)$ is the weight parameter; $\mathrm{dens}(c_{ij})$ is the density in a Moore neighborhood of cell c_{ij}; D_{ij} is the value of a dynamic field; $\delta \in [0, \infty)$ is the weight parameter; $\mathrm{dist}(c_{ij}, POI)$ is the distance between a cell with coordinates i, j and the closest POI; W is the wall avoidance parameter which takes into account distance to the nearest wall $W = 1.0 + \sigma_W$ and increasing values in walls neighborhood, $\sigma_W \in (0, 1]$ (more detailed description in [37]) and I is the inertia parameter, agents prefer to maintain movement direction as long as possible $I \in (0, 1)$ (more detailed description in [37]).

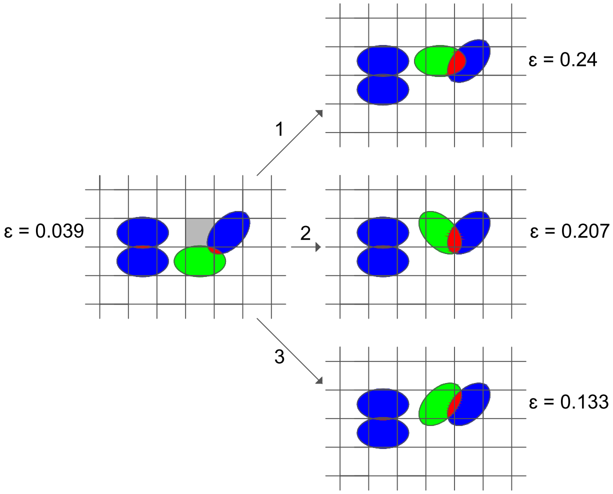


Fig. 5 Possible scenarios in conflict resolution. On the *left* is the initial phase of conflict situation, on the *right* the possible variants of resolving the conflict. The *green* pedestrian makes a move to a *gray* cell, body position can change by 45 ° to the *right* or *left*. If $\varepsilon \leq 0.21$ the second and third variants are allowed (color figure online)

The second stage of the movement algorithm is dedicated to conflict resolution. The conflict resolution methods depend on update types of the cells state, in the bibliography one can fine following schemes: shuffled sequential, parallel update, ordered sequential update [22]. If a simulation contains more than one pedestrian, two or more of them may select the same cell as the destination of their next step. A second type of conflict situation is related to 14 combinations of local configurations defined by the social distances model. Depending on the so-called compressibility coefficient ε, forbidden states may occur. The ε coefficient determines which local configurations are allowed. In Fig. 5, we present three possibilities of pedestrian configuration during the second stage of conflict resolution.

In the process of conflict resolution, we firstly deal with a classical conflict. As a solution, we assigned all conflicted pedestrians to one queue and assign to them priority according to waiting time. The waiting time is the number of the pedestrians' motionless simulation steps from the last movement. The conflict is won by the pedestrian with the longest waiting time or if pedestrian waiting times are equal then by random selection.

Secondly, the pedestrian checks if the move can be done according to the compressibility coefficient in the Moore neighborhood of the cell chosen for the next move. If the compressibility parameter does not allow that configuration, the pedestrian will try to fit in the chosen cell by turning 45 °. If the above operations fails, the pedestrian will not move in the next simulation step.

However, when testing the conflict resolution process the authors have observed that the available space for the movement is not fully utilized, especially under low-density conditions. In order to address this issue, we employed an additional step in conflict resolution. If it turns out that the move to the cell chosen by the pedestrian is impossible because of the compressibility coefficient, the second cell with the smallest

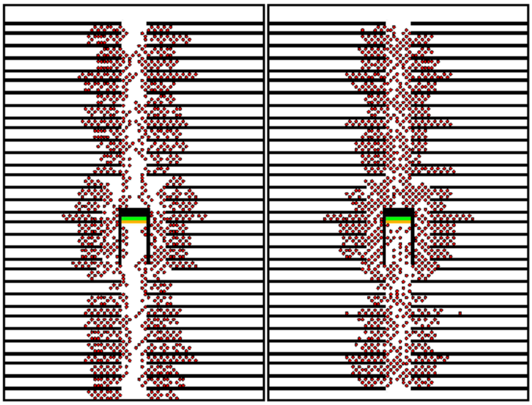


Fig. 6 Sample simulation of a single sector of Wisła Kraków Stadium without (on the *left*) and with (on the *right*) the additional step in the conflict resolution. The simulation's parameters in both cases are exactly the same

cost from the available *VF* set is selected and the conflict resolution is repeated. The results of this mechanism are presented in Fig. 6.

This additional step improves the quality of representation of reality in the model. The available space near the exit (green line in the middle) and between the left and right rows of seats is better utilized. For visual comparison please see Fig. 17.

4 Simulator

This section describes the architecture and the most important parts of the pedestrian evacuation simulator. In our opinion, such an architecture can be used as the basis for developing such applications. Subsequently, the logic and data layer, the controller layer and the presentation layer will be briefly described.

4.1 Architecture of the application

The application developed by the authors implements the proposed model. It has been designed based on a multi-layered architecture [11]. Three types of layers can be distinguished:

1. *The logic and data layer* Implements pedestrian dynamics model, and data storing—package *core*,
2. *The controller layer* Handling user requests and utility classes—package *engine* and *tools*,
3. *The presentation layer* Graphical user interface, package *gui*.

In Fig. 7, we schematically shown the application packages in which components are grouped.

4.2 Logic and data layer

The logical structure of the pedestrian evacuation model consists of: *Global Object*, *Objects* and *Levels* (e.g., stadium, separate stands and its sections). Let us consider

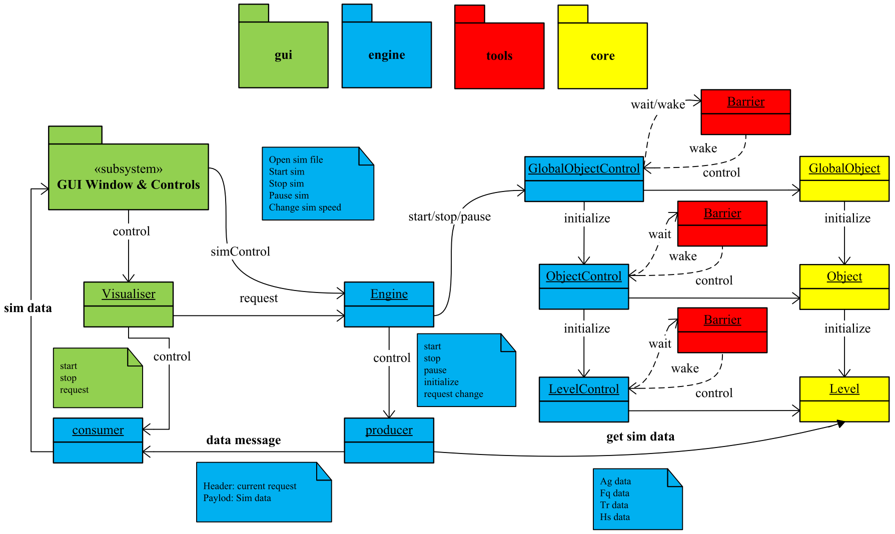


Fig. 7 The overall scheme of components, that can be considered part of application architecture

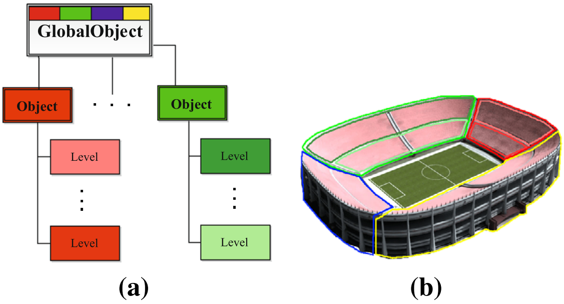


Fig. 8 The hierarchy of object in the logic and data layer (**a**). A proposal for the logical division of a football stadium (**b**). The corresponding *colors* represent object type (color figure online)

an example of Wisla Krakow stadium which part (east stand) was used as application of a proposed model to real case. This stadium has four stands (east, west, north and south) which corresponds to Objects. Each stand has sections which can be treat as Levels. At the end, we have Wisla Krakow stadium as *Global Object*, each stand as *Object* and each section of stand as *Levels* (see Fig. 8a).

Such a structure is scalable and it takes into account diverse facilities (e.g., football stadiums, high-rise buildings or even a single rooms). The minimal and full implementation of the model is included in *Level*. Among the most important components in *Level* one can distinguish: movement algorithm, static and dynamics floor field, geometry layout and simulation setup. In fact each *Level* is an autonomous simulation with all necessary data structure and algorithms.

The static floor field is a 3-dimensional array $S[x][y][POIs]$ which leads pedestrians to the POI (e.g., chosen emergency exit). We are generating the static floor field for each POI, so when the pedestrian wants to change the POI, s/he simply starts to using the static field which corresponds to chosen POI.

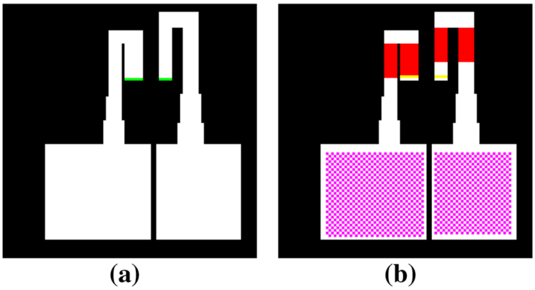


Fig. 9 Two input files used to generate geometry layout. In input **a** *color black* means inaccessible space, *white* accessible space and *green* is point of interest. In input **b** *color pink* means pedestrians' starting point, *red* is reduction of speed (stairs) and *yellow* is measured area (evacuation time) (color figure online)

The dynamic floor field is a 2-dimensional array $D[x][y]$ which changes after each simulation step. The dynamic of this structure is described by diffusion and decay rules.

The geometry layout is a 3-dimensional array with information about accessible/inaccessible space, free/occupied cells, placement of the POIs and parameters such as pedestrians' starting points, speed reduction (stairs) and measurement areas. We create this layout by analyzing colors of two inputs files (see Fig. 9).

4.3 Controller layer

Increasing demand for large-scale crowd simulation drives proper parallelization. Every object from the logic and data layer has its own facade object, which is responsible for communication with the controller [13].

In terms of synchronization three facade objects are differentiated:

- *Level Control* Threads responsible for a simulation on given levels (e.g., one floor in section of the building),
- *Object Control* Threads responsible for data flow between levels (e.g., whole building sections or stadium stands),
- *Global Control Object* Thread that controls the whole simulation and collects results.

Such a division better encapsulates the situation and helps the develop of solutions. The facade object mediates the exchange of data between the logic layer and the controller layer. All facade objects run in a separate thread and for obvious reasons the threads must be synchronized. The thread synchronization method is based on the authors' implementation of synchronization design pattern—Barrier [19] (see Fig. 10a), and contains five global memory barrier objects: input and output barriers for the Level Control and Object Control components and one barrier for Global Object Control.

The input barrier for a Level Control object is released directly by a Global Control Object. When each Level Control Object finishes its activity, two elements are released: the Level Control output barrier and the Object Control input barrier. The object then starts waiting at its input barrier. Object Control executes the pedestrian's transitions between levels and then stops at the Object Control output barrier. When all Object Controls finish their activities, they release their output barrier, as well as the Global

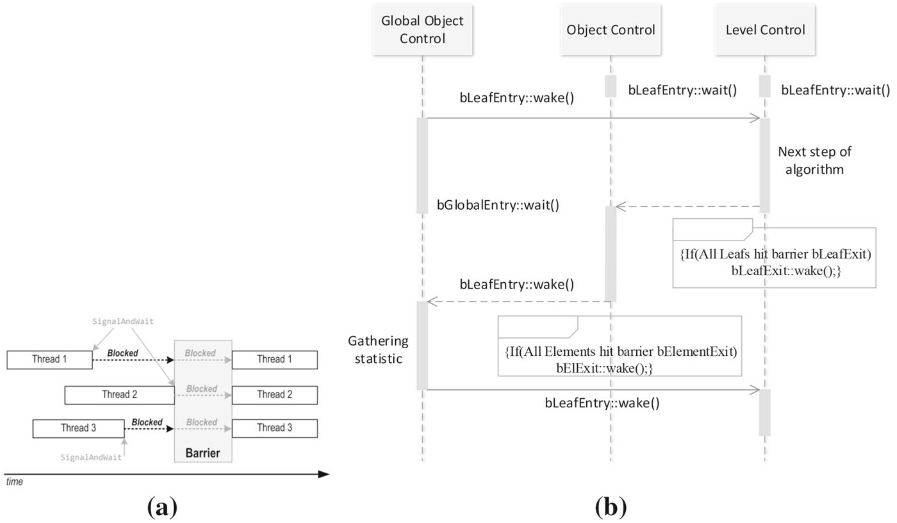


Fig. 10 Thread synchronization using the barrier design pattern [1] in **a**. Sequence diagram depicts the order of execution and synchronization of threads in simulator in **b**

Object Control barrier. After of the execution of the Global Object Control is finished, it releases the Level Control input barrier and starts a new iteration. The Mechanism described above is also introduced in Fig. 10b in the form of a sequence diagram.

The controller is responsible for realizing of demands from the presentation layer (e.g., start, stop, pause or change current view). In addition, it coordinates the delivery of data to the presentation layer using the well-known producer–consumer design pattern.

The synchronization method described above can also be applied to hybrid solutions, which combine continuous and discrete models of pedestrian dynamics. To maintain the consistency of that connection, the simulation time in all methods must be synchronized. Adapting the sync method to a hybrid model relies on redefining the tasks for Global, Object and Level Control, respectively, to control the whole hybrid simulation. It makes it possible taking care when transforming the pedestrians from one type of model to another and the performed simulation of given region using given models (included in a hybrid solution).

4.4 Presentation layer

This layer is composed of elements from the GUI package—and implements a user graphical interface—including dialogue windows, buttons and widgets. It is worth noting that the visualization of the simulation results is displayed after every 1/8 of a simulation second (this is the main unit of time in proposed simulations) and has almost negligible impact on performance [37]. The presentation layer is responsible for two main tasks. The first is the accurate drawing of the graphical scenes, while the second user interaction, controlling the simulation and visualization view.

The four levels of details of representing pedestrians are shown in Fig. 11 (on the left), with statistics such as pedestrians' trajectories (on the right). For performance reasons at minimal zoom the pedestrians are drawn as squares, at maximum zoom we can observed more details, such as elliptical shape, head and front of the body. The frequency matrix (Fig. 12, left) can be helpful in the visual detection of bottlenecks, potentially dangerous places in building geometry and the assessing the occurrence of line formation in pedestrian flow. The evacuation times (Fig. 12, right) present how

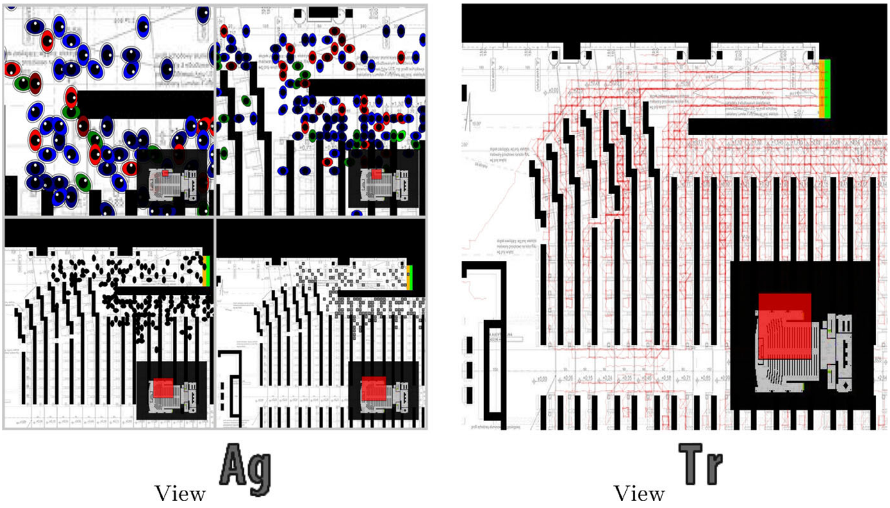


Fig. 11 Representation of pedestrian in different level-of-details (on the *left*) and pedestrians' traces-marked in *red* (on the *right*) (color figure online)

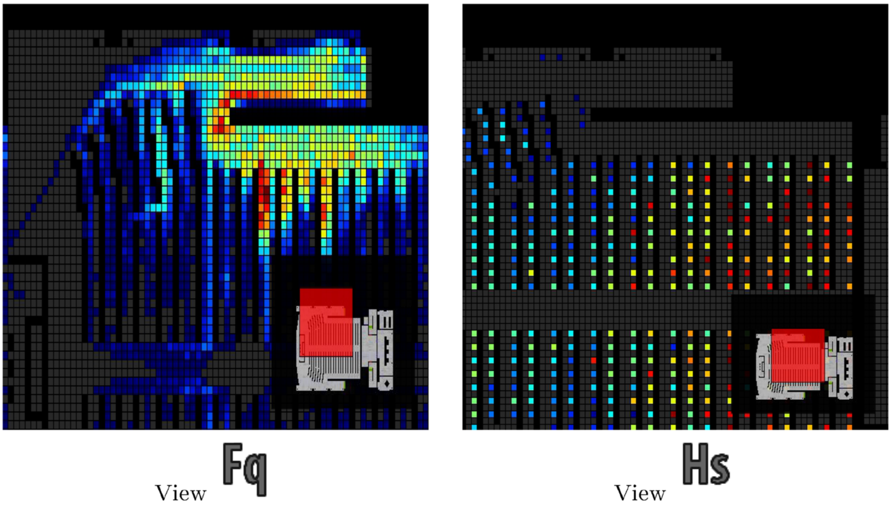


Fig. 12 On the left the frequency of visits to each cell is visualized. A *warmer color* means higher number of visits. On the *right* the evacuation time for each pedestrian is presented. *Colored squares* represent the starting point. The warmer the color, the longer the evacuation time (color figure online)

evacuation time distribution depends on the distance to exit. All four views can be previewed during the simulation.

5 Applications of the model to real cases

In this section, we present sample results of evacuation simulations from a lecture room, a lecture hall, the Wisła Kraków Stadium and the Allianz Arena Stadium, as well as an application of the model in a simple data-driven simulation. The data obtained from simulation of the lecture room and the lecture hall were compared with empirical results from experiments, and simultaneously, some issues of validation were discussed. It should also be stressed that the simulation results of the two football stadiums are good examples of scalability and efficiency.

5.1 Lecture room

In [34], the author presented experiments of pedestrians' evacuation in different situations. A group of 31 students took part in a lecture room evacuation experiment. The average simulation results from 50 trials were compared with empirical data obtained from the experiments of the controlled (meaning non-competitive) evacuation. The authors regards the controlled evacuation as a pattern of cooperation between pedestrians: during the experiments the aim of each participant was to achieve the best evacuation time for the whole group.

According to the empirical data the evacuation time was 6.24 s, while in the simulation it was 6.60 s. The average speed in the simulation was 1.25 ms^{-1}. In Fig. 13, we can compare the simulation results with empirical with respect to the individual evacuation as well as the chosen exit. Figure 14a presents the comparison between the empirical data and the simulation data, Fig. 14b shows the average velocity and distance covered from starting point to the exit for each pedestrian.

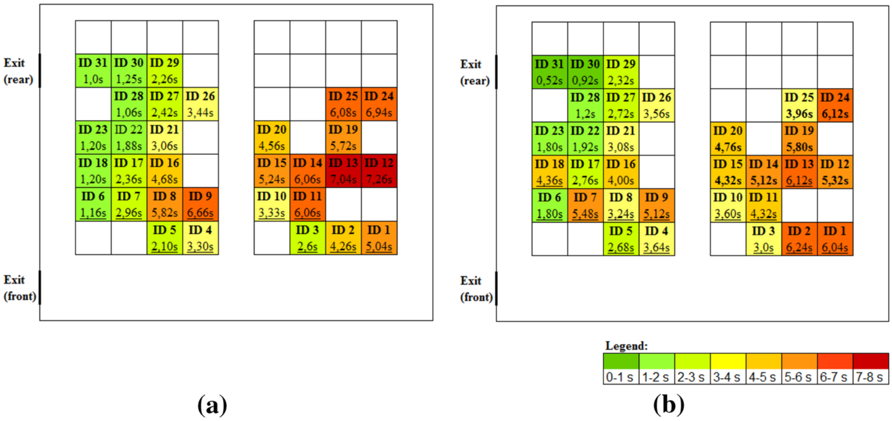


Fig. 13 The initial allocation of pedestrians in the lecture room. At the *top* of each cell is a pedestrian ID. At the *bottom* of each cell is an exit time score (*underlined* front exit, *not underlined* rear exit). The simulation data are in **a** and the empirical data are in **b**

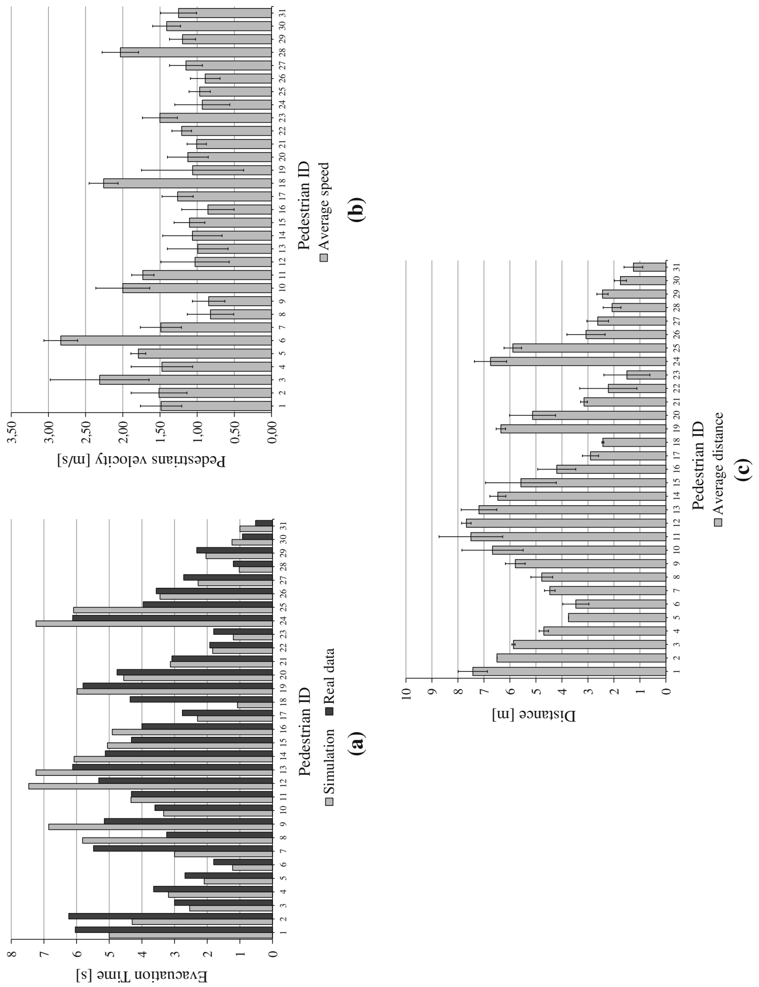


Fig. 14 A lecture room evacuation. Comparison of the simulation results and the real data (**a**). Average speed and an average distance covered by each pedestrian in the simulation (**b**) and (**c**)

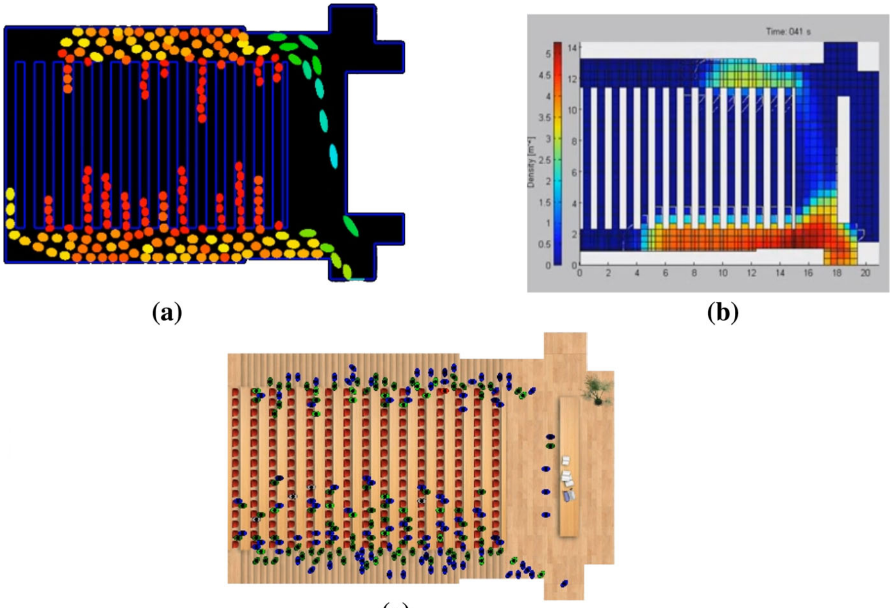
It is noticeable that the exit time in both simulation and in experiment are similar for almost all occupants. The main differences are caused by different exit choice—occupants 8, 13, 18 change the evacuation exit during the simulation. Depending on the strategy, pedestrian in simulation is able to change the exit several times taking into account the current situation in front of the exit.

5.2 Lecture hall scenario

In [24] an analysis of a lecture hall evacuation was conducted, in order to confirm model reliability and efficiency. Reliability in this experiment is understood as the similarity of results to experimental data, while efficiency is equated with performance. The Social Distance Model [37], was compared with two popular models:

- *The Generalized Centrifugal model* [5] A force-based approach, an extension of the social force model. Pedestrian movement is determined by a combination of three forces: destination driving force, obstacle repulsive force and repulsive force between pedestrians.
- *The hydrodynamic approach* A macroscopic model, based on [6]. A crowd is considered in terms of locally averaged density and velocity, rather than a set of individuals.

Visualization of all three methods is presented at Fig. 15.



(a)

(b)

(c)

Fig. 15 Three methods of crowd dynamics simulation in a lecture hall experiment: **a** Generalized Centrifugal model, **b** hydrodynamic approach and **c** proposed approach based on Social Distance Model

Outflows produced by the tested models were compared with experimental results. Both the centrifugal model and the Social Distance Model were able to simulate overall tendencies of outflow changes, while the outflow produced by hydrodynamic approach differs significantly from the experimental data (see details in [24]).

Efficiency was investigated through an analysis of the execution time of consecutive seconds of the simulation. The average execution time of a simulation second for the Macroscopic, the examined version of social distances and Generalized Centrifugal was, respectively, 1.86, 6.946 and 8825.67 ms. For the hydrodynamic approach execution time was almost constant for each simulation second. In the two other methods the execution time of a simulation second depends on the number of moving pedestrians. It rises from 1st to 25th s, due to pre-movement time distribution, then decreases as pedestrians leave the simulation area. The peak in execution time can also be associated with the highest number of interactions between pedestrians.

It is worth noting, that in this scenario the average execution time for the Social Distance Model [37] is of the same order of magnitude as in the macroscopic model. On the other hand, the obtained results are similar to the ones produced by the force-based model, and are not far from the experimental data.

5.3 East stand of Wisła Kraków Stadium

The following results include data from a simulation of a non-competitive evacuation scenario of the eastern stand of Wisła Kraków Stadium. The evacuation time for 95 % of initial pedestrian capacity was 653 s (10 min and 54 s).

Table 1 presents the basic statistics from the evacuation simulation. Figure 16 depicts other data gathered during the simulation. A notable fact is that average speed (Fig. 16b) decreased during the entire simulation. This is caused by the fact that most individuals clog nearby exits. Such a result can be treated as a symptom of errors in stadium design.

A section of a stand of the Wisła Kraków stadium during normal egress condition is shown at Fig. 17. A visual comparison of simulation results (Fig. 6, right), with real data shows that the model is able to properly reproduce specific clogging patterns.

Table 1 Summary of the simulation of the non-competitive evacuation scenario of the East Stand of Wisła Kraków Stadium

Name	Evac. threshold 95 %
Initial agent number	11,808
Current agent number	583
Evacuation time	653 s
Quickest agent evac. time	3 s
Slowest agent evac. time	653 s
Desired population speed	1.36 ms^{-1}
Average agents speed	0.316 ms^{-1}

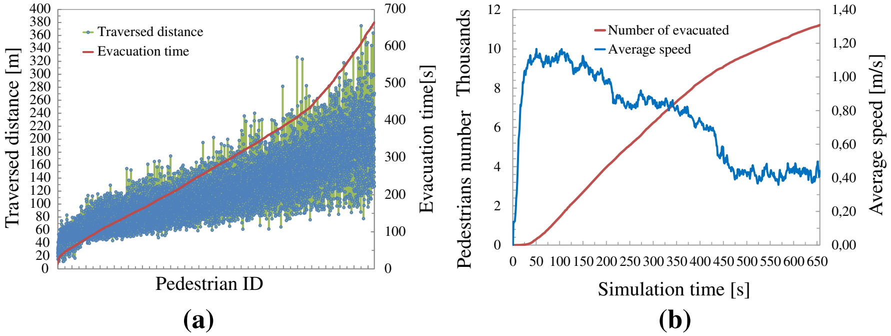


Fig. 16 A non-competitive evacuation of the eastern Wisła Kraków Stadium eastern stand. **a** Sample statistics of covered distance and evacuation time for each pedestrian. **b** Number of evacuated over time and average velocity

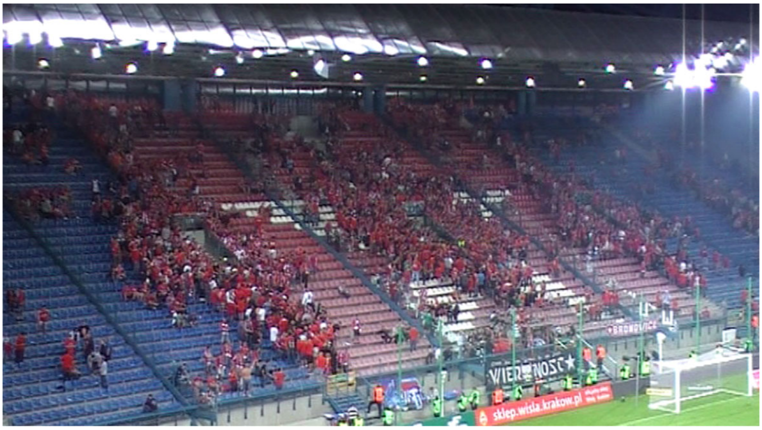


Fig. 17 Fans leaving one of section of Wisła Kraków stadium (under normal egress condition). One should observe the characteristic structures created by pedestrians, when they try to reach the exit from the section. Similar structures can be observed in our simulation (see Fig. 6, *right*)

5.4 Allianz Arena Stadium

Another example is a simulation of the Allianz Arena Stadium in Munich. Simulation of this complex facility with 70,000 seats was a real challenge especially in terms of an integrated simulation scenario (as a part of 7FP EU-Socionical).

Table 2 and Fig. 18 contain the quantitative results of the simulation of a non-competitive evacuation scenario. The evacuation time for 95 % of initial pedestrian capacity was 1117 s (18 min and 37 s). The average agents speed after first 300 s has stabilized at around 0.7 ms^{-1}—for almost 70 % of evacuation time, pedestrians were able to maintain stable and quite high outflow.

Figure 19 presents graphical statistics for a section of the lowest level of the stadium. The *frequency matrix* of moving pedestrians can be useful to detect possible bottlenecks or places with abnormally high densities. A graphical representation of

Table 2 Summary of the simulation of non-competitive evacuation scenario for the Allianz Arena Stadium

Name	Evac. threshold 95 %
Initial agent number	58,000
Current agent number	2900
Evacuation time	1117 s
Quickest agent evac. time	2 s
Slowest agent evac. time	1117 s
Desired population speed	1.34 ms^{-1}
Average agents speed	0.726 ms^{-1}

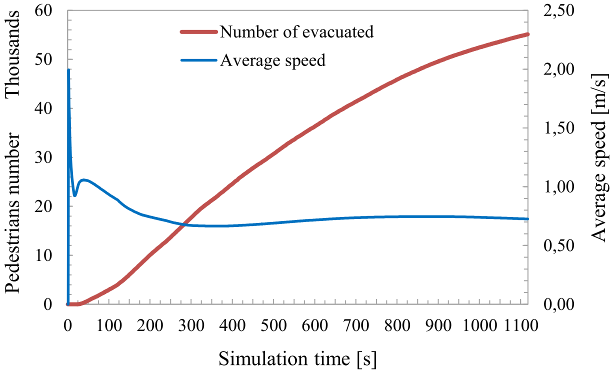


Fig. 18 Sample statistics of number of people evacuated over time and average velocity for the Allianz Arena Stadium simulation

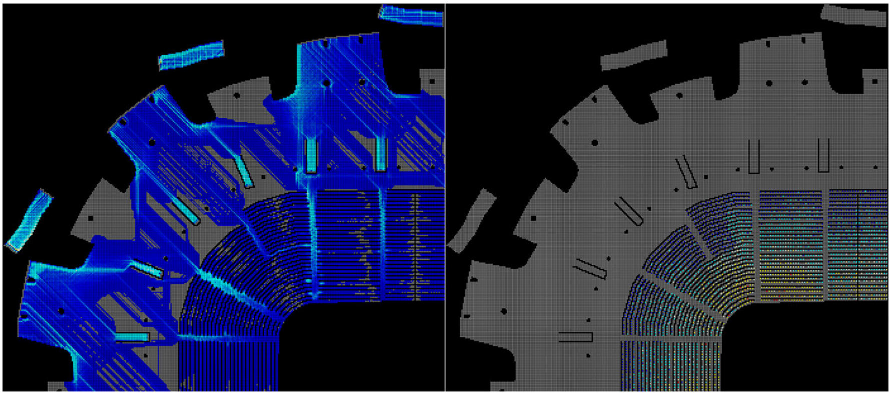


Fig. 19 Sample graphical statistics for the lower part of the Allianz Arena Stadium. The frequency matrix is on the *left* and evacuation time on the *right*. *Warmer colors* represent higher visit frequency and larger evacuation times, respectively (color figure online)

the evacuation time for each pedestrian shows, that it in fact closely corresponds to the distance from emergency exits (in most cases: the longer the distance the bigger the evacuation time).

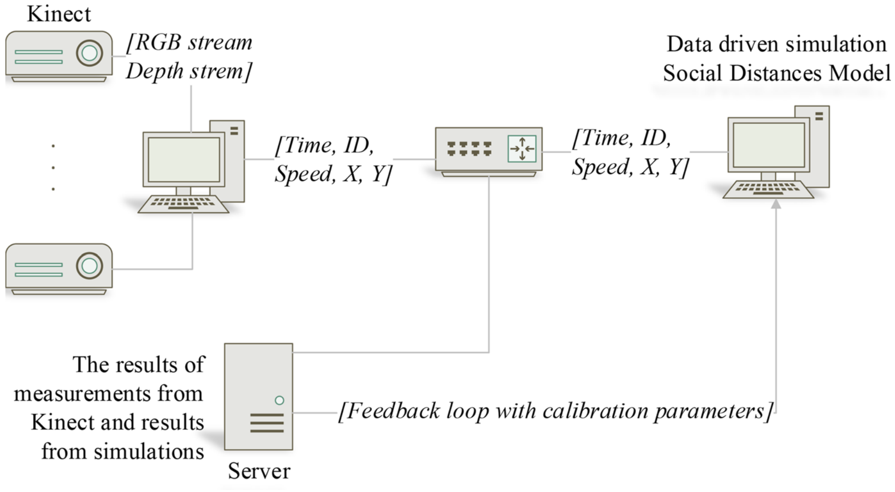


Fig. 20 The architecture of an indoor crowd simulation system, created by the authors as a demonstration of the usage of the presented model in data-driven simulations. The system uses a set of Microsoft Kinect devices as input sensors. The proposed architecture includes a feedback loop that allows automatic validation and simple re-calibration of the simulation

5.5 Model application in data-driven simulation

Recent years have seen growing demand for online data-driven multi-scale simulations—pedestrian simulation system integrated with many different sensors, e.g., CCTV camera, depth sensors, GPS-positioning. Due to its ability to be automatically customized with real data, data-driven simulation can be a valuable tool for supporting LEA (law enforcement agencies, e.g., police) actions during mass crowd events.

The presented model has a number of crucial features that make it a good tool for such systems. First of all, the simulation model has to be fast enough to be executed in real time and to process current data. As the tests show, the presented model is able to simulate up to 12,000 pedestrians even on a typical personal computer.[2] Secondly, this model has a simple, well-defined interface; therefore, it can be easily integrated with other elements of the system. Moreover, using such an agent-based system makes it possible to customize each agent in the simulation according to real data (e.g., its size or desired maximum speed).

As a demonstration of concept, the authors proposed a simple indoor crowd simulation system driven by data gained from Microsoft Kinect devices [28]. The system contains three main application components (see Fig. 20):

- *Server application* Provides information exchange between system components, resolves synchronization issues, allows a feedback information loop.

[2] The setup of a performance test and a brief analysis of computational and memory complexity has been presented in [37].

- *Sensor data analyzer* Handles data from a Kinect device and generates pedestrian trajectories.
- *Crowd simulation* Receives actual data and calculates short-time pedestrian position predictions using the Social Distance Model.

It is worth mentioning that all components—client applications, sensor data analyzer and crowd simulation—use the same communication protocol. In fact, there is no difference between flow data from sensors and exits from simulated areas. The considered area is divided into two groups: the measurement area covered by sensors and the simulation area. The presented system includes a feedback loop, output data from the simulation can be compared with real data and used for validation and simple re-calibration of the model.

Such properties are easy to obtain due to the usage of agent-based modeling. Agent-based approach allows for customization of pedestrian attributes (e.g., speed, size and movement direction). Detected individuals are automatically and immediately represented in the simulation as an agent, using its own set of parameters based on real data. On the other hand, the fact that the social distances model is based on CA provides high efficiency. In our opinion, CA agent-based models are currently the most appropriate type of model for online data-driven large-scale simulations due to their effectiveness and accuracy.

6 Conclusions

On the basis of our experience and several finished projects, we have used Cellular Automata as the basis for agent-based crowd simulations. In general, in contrast to the vast majority of CA-based simulations designed for engineering purposes, we have a applied strategic level in decision-making process, more complex scenarios of crowd behavior and patterns for different classes of situations.

Projects were carried out devoted to the following case studies: an integrated simulation of the Allianz Arena Stadium Munich (FP7 Project of EU Socionical), evacuation of Wisla Krakow Stadium, an evacuation of GKS Tychy Stadium, a simulation of pedestrians in the Didactic Center of AGH University, a simulation of the Galeria Krakowska shopping center in Cracow, as well as other projects.

In comparison to classical CA-based crowd models [4,22,27], we have used a finer representation of the environment—represented as a square lattice of a non-homogeneous cellular automaton. On the other hand, we have used an elliptic representation of a pedestrian (from the social distances model), where the center of a pedestrian coincides with the center of a cell, and affects neighboring cells according to the permissible degree of compressibility. Finally, the proposed changes have caused the introduction of new elements into our approach, with respect to classical models: a two-level decision-making function (with a strategic and a tactical/operational level), a new cost function on the tactical/operational level, a new definition of visibility fields, a modified method of resolving conflicts, etc.

In the article we proposed a new method of resolving conflicts between pedestrians (Sect. 3.2.2), which is an important element of the tactical and operational level of agent decision-making. Next, we introduced the architecture of our system dedicated

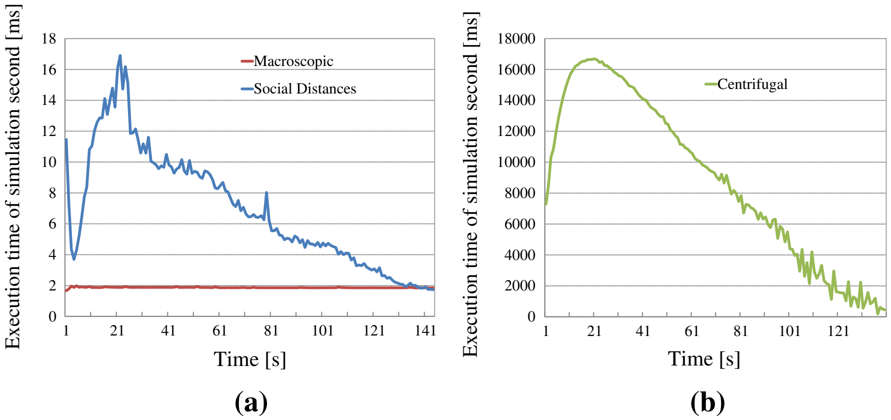


Fig. 21 Performance tests for all models. The *curves* show execution time for each consecutive second of simulation [24]

to pedestrian simulations (Sect. 4.1). Finally, we discussed a practical, data-driven application on the basis of our experiences (Sect. 5.5).

We observe a number of advantages associated with the use of Cellular Automata as the basis for agent systems in modeling pedestrian dynamics and behavior. The main advantage is the scalability of the developed solutions. In this paper, we have presented an example of a simulated evacuation performed for a group of 31 people initially located in a small room, as well as a large-scale simulation of the Allianz Arena Stadium developed for 70,000 pedestrians allocated on different levels. It should be stressed that using CA as a basis enables high efficiency and extensive possibility for parallelization (due to massive parallelism of CA).

Another advantage is that we can reproduce the compressibility of the crowd in a more detailed way (when compared to classical CA methods). We also observe that due to the proposed elliptic representation of pedestrians, we are able to reproduce densities of crowd with greater precision than in classical CA models. Thanks to the application of the strategic level in pedestrian decision-making, we are able to build more complex scenarios when compared to classical CA models. It should be stressed that the presented approach constitutes a good compromise between continuous methods and the basic CA approach. Despite the use of more complex transition functions and the elliptic representation of pedestrians, the measured effectiveness of our method is good in comparison to other methods (Fig. 21b).

We have also noticed several limitations of our approach. The first issue is connected with discretization. In our model, larger errors of environment representation and positions of agents occur with regard to the real data and continuous methods of simulation. One can also observe polygonal paths of agent trajectories. It should be noted that the proposed method is an approximate one, although the level of mapping accuracy is greater for our approach when compared to the classical CA models.

Summing up, we recognize a strong potential in terms of the fine representation of various objects like: different sizes of pedestrians, prams, luggage carts and wheelchairs in our CA representation base. On the basis of the performed simulations coupled with the data gathered from MS Kinect, we also notice a strong potential in the application of our model in building real-time systems based on the concept of data-driven modeling.

References

1. Albahari J, Albahari B (2010) C# 4.0 in a Nutshell: The Definitive Reference, 4th edn. OReilly Media, Inc., Newton
2. Bandini S, Manzoni S, Vizzari G (2004) Situated cellular agents: a model to simulate crowding dynamics. Trans Inf E 87D(3):669–676
3. Blue V, Adler J (1999) Cellular automata microsimulation of bidirectional pedestrian flows. J Transp Res Board 1678(1):135–141
4. Burstedde C, Klauck K, Schadschneider A, Zittartz J (2001) Simulation of pedestrian dynamics using a two-dimensional cellular automaton. Phys A 295(3–4):507–525
5. Chraibi M, Seyfried A, Schadschneider A (2010) Generalized centrifugal-force model for pedestrian dynamics. Phys Rev E 82:046111. doi:10.1103/PhysRevE.82.046111
6. Coscia V, Canavesio C (2008) First order macroscopic modelling of human crowd dynamics. Math Models Methods Appl Sci 18:1217–1247
7. Daamen W (2004) Modelling passenger flows in public transport facilities. Ph.D. thesis, Delft University of Technology, The Netherlands
8. Dietrich F, Koester G, Seitz M, von Sivers I (2014) Bridging the gap: from cellular automata to differential equation models for pedestrian dynamics. J Comput Sci 5(5):841–846. doi:10.1016/j.jocs.2014.06.005
9. Dudek-Dyduch E, Was J (2006) Knowledge representation of pedestrian dynamics in crowd: formalism of cellular automata. In: Artificial Intelligence and Soft Computing—ICAISC 2006, 8th International Conference, Proceedings. Zakopane, Poland, pp 1101–1110
10. Ezaki T, Yanagisawa D, Ohtsuka K, Nishinari K (2012) Simulation of space acquisition process of pedestrians using Proxemic Floor Field Model. Phys A 391(1–2):291–299
11. Fowler M (2002) Patterns of enterprise application architecture. Addison-Wesley Longman Publishing Co. Inc, Boston
12. Fukui M, Ishibashi Y (1999) Self-organized phase transitions in cellular automaton models for pedestrians. J Phys Soc Jpn 68(8):2861–2863. doi:10.1143/JPSJ.68.2861
13. Gamma E, Helm R, Johnson RE, Vlissides J (1995) Design patterns: elements of reusable object-oriented software. Addison-Wesley, Reading
14. Georgoudas IG, Koltsidas G, Sirakoulis GC, Andreadis IT (2010) A cellular automaton model for crowd evacuation and its auto-defined obstacle avoidance attribute. In: Proceedings of the 9th international conference on Cellular automata for research and industry, ACRI'10. Springer, Berlin, Heidelberg. pp 455–464. http://dl.acm.org/citation.cfm?id=1927432.1927488
15. Gwizdalla TM (2015) Some properties of the floor field cellular automata evacuation model. Phys A 419:718–728. doi:10.1016/j.physa.2014.10.070
16. Hrabak P, Bukacek M, Krbalek M (2013) Cellular model of room evacuation based on occupancy and movement prediction: Comparison with experimental study. J Cellular Autom 8(5–6):383–393
17. Ingerson T, Buvel R (1984) Structure in asynchronous cellular automata. Phys D 10(1–2):59–68
18. ISO 16730:2008(E) Fire safety engineering—Assessment verification and validation of calculation methods. Tech Rep ISO
19. Karmani RK, Chen N, Su BY, Shali A, Johnson R (2009) Barrier synchronization pattern. Computer Science Department University of Illinois and EECS Department University of California
20. Kauffman S (1984) Emergent properties in random complex automata. Phys D 10:145–156
21. Kirik E, Malyshev A (2014) On validation of sigmaeva pedestrian evacuation computer simulation module with bottleneck flow. J Comput Sci 5(5):847–850. doi:10.1016/j.jocs.2014.05.002
22. Kluepfel H (2003) A cellular automaton model for crowd movement and egress simulation. Ph.D. thesis, University Duisburg–Essen
23. Klüpfel H, Meyer-König T, Wahle J, Schreckenberg M (2000) Microscopic simulation of evacuation processes on passenger ships. In: Bandini S, Worsch T (eds) ACRI 2000. Springer, London, pp 63–71

24. Lubaś R, Miller J, Mycek M, Porzycki J, Wąs J (2013) Three different approaches in pedestrian dynamics modeling—a case study. In: Zamojski W, Mazurkiewicz J, Sugier J, Walkowiak T, Kacprzyk J (eds) New results in dependability and computer systems, advances in intelligent systems and computing, vol. 224. Springer International Publishing, New York, pp 285–294
25. Masakuni M, Tunemasa I, Takashi N (1999) Jamming transition in pedestrian counter flow. Phys A 267(3–4):487–498
26. Musse SR, Thalmann D (2001) Hierarchical model for real time simulation of virtual human crowds. IEEE Trans Visual Comput Graphics 7:152–164
27. Nishinari K, Kirchner A, Namazi A, Schadschneider A (2004) Extended floor field CA model for evacuation dynamics. IEICE Trans Inf Syst E87(D):726–732
28. Porzycki J, Lubaś R, Mycek M, Wąs J (2014) Dynamic data—driven simulation of pedestrian movement with automatic validation. Traffic and Granular Flow '13 proceedings (in print)
29. Ronchi E, Kuligowski ED, Reneke PA, Peacock RD, Nilsson D (2013) Nist technical note 1822, the process of verification and validation of building fire evacuation models. Tech. rep, NIST
30. Schadschneider A, Klingsch W, Klupfel H, Kretz T, Rogsch C, Seyfried A (2011) Evacuation dynamics: empirical results, modeling and applications. In: Extreme Environmental Events. Springer, New York, pp 517–550
31. Schadschneider A, Seyfried A (2009) Validation of CA models of pedestrian dynamics with fundamental diagrams. Cybern Syst 40:367–389
32. Viswanathan V, Lee C, Lees M, Cheong S, Sloot P (2014) Quantitative comparison between crowd models for evacuation planning and evaluation. Eur Phys J B 87(2):27. doi:10.1140/epjb/e2014-40699-x
33. Wagoum K, Ulrich A, Steffen B, Seyfried A (2012) Runtime optimisation approaches for a real-time evacuation assistant. Parallel Processing and Applied Mathematics, Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, pp 386–395
34. Wąs J (2010) Experiments on evacuation dynamics for different classes of situations. In: Klingsch WWF, Rogsch C, Schadschneider A, Schreckenberg M (eds) Pedestrian and Evacuation Dynamics. Springer, Berlin, Heidelberg, pp 225–232. doi:10.1007/978-3-642-04504-2-17
35. Wąs J, Gudowski B, Matuszyk PJ (2006) New cellular automata model of pedestrian representation. Cellular Automata, Lecture Notes in Computer Science, vol 4173. Springer, Berlin, Heidelberg, pp 724–727
36. Wąs J, Gudowski B, Matuszyk PJ (2006) Social distances model of pedestrian dynamics. In: Proceedings of the 7th international conference on Cellular Automata for Research and Industry, ACRI. Springer, Berlin, Heidelberg, pp 492–501
37. Wąs J, Lubaś R (2013) Adapting social distances model for mass evacuation simulation. J Cellular Autom 8:395–405
38. Wąs J, Lubaś R (2014) Towards realistic and effective agent-based models of crowd dynamics. Neurocomputing 146:199–209. doi:10.1016/j.neucom.2014.04.057
39. Weidmann U (1992) Transporttechnik der Fussganger—Transporttechnische Eigenschaften des Fussgangerverkehrs, Literaturauswertung, Schriftenreihe des IVT. Tech. Rep. 90, Institut fur Verkehrsplanung und Transportsysteme, Zurich
40. Zia K, Ferscha A, Riener A, Wirz M, Roggen D, Kloch K, Lukowicz P (2010) Pervasive computing in the large: the Socionical approach. In: Proceedings of the 8th International Conference on Pervasive Computing, Helsinki, Finland. Springer, New York