Fast Evacuation Method: using an effective dynamic floor field based on efficient pedestrian assignment*

Severino F. Galán
Artificial Intelligence Dept. at UNED
C/ Juan del Rosal, 16. 28040 Madrid, Spain
E-mail: seve@dia.uned.es

Abstract

The problem of pedestrian evacuation can be addressed through cellular automata incorporating a floor field that indicates promising movements to pedestrians. The two main types of floor field are the static, which represents the *shortest* path from each cell to an exit (and is usually combined with dynamic measures such as the density or distribution of pedestrians), and the dynamic, which represents the *quickest* path from each cell to an exit. The second type has been widely used recently, since it gives rise to more efficient and realistic simulations of pedestrian dynamics. The goal of these two types of floor field is to minimize the travel time for each pedestrian; however, this paper tackles the evacuation problem from a different perspective: The time taken by the whole evacuation process is optimized. For that purpose, a floor field is constructed by assigning pedestrians to exits such that the estimated time for complete evacuation is minimized. An experimental evaluation is conducted to compare the new fast evacuation method with competitive methods using floor fields based on quickest paths: Flood Fill and the Fast Marching Method. The results show that the new method is effective in terms of the number of time steps for complete evacuation and efficient regarding the total simulation runtime.

Keywords: evacuation, cellular automata, effective floor field, efficient pedestrian assignment.

1 Introduction

The problem of pedestrian evacuation [34, 29] has attracted great interest in the last few years. The safe and efficient movement of crowds, especially under emergency conditions, is an important issue in evacuation situations of aircraft, buildings, concerts, or stadiums, among many others. Specifically, this paper focuses on the version of the problem in which a set of pedestrians move

in (and have complete knowledge of) a 2D room with obstacles and multiple exits. Since real-life evacuations are difficult to perform, computer simulations are usually employed to study evacuation methods.

Different models have been created to simulate and analyze human behavior during evacuation processes. Depending on the level of detail developed, these models can be classified into microscopic, mesoscopic, or macroscopic [9, 37]. The main microscopic techniques for modeling the problem of pedestrian evacuation are social forces [18, 16], lattice gas [39, 40, 17, 20], and cellular automata (CA) [4, 22, 33, 21, 43]. Whereas the social force model represents pedestrians as particles interacting with the environment under continuous space and time, the CA model uses discrete floor fields to determine the motion of pedestrians in discrete time. Precisely, the present paper deals with CA models of evacuation that employ floor fields to indicate promising movements to pedestrians.

In a discrete floor field, space is partitioned into rectangular cells and a weight is assigned to each cell at every time step. The movement of pedestrians is determined by the cell weights and the rules of pedestrian interaction. Generally, a pedestrian moves to the neighboring empty cell with lowest weight. Whereas a fixed weight is assigned to each cell in a static floor field, in a dynamic floor field weights evolve with time and can change by the presence of pedestrians. In the literature, the floor fields for the evacuation problem are based on minimizing the evacuation time for each pedestrian. Specifically, for each CA cell, either the shortest path to exit [33, 3] (typically combined with dynamic measures of pedestrian density or distribution) or the quickest path to exit [24, 25] are calculated. While the shortest path is static and only depends on the room structure, exit location, and obstacle distribution, the quickest path is dynamic since the distribution of pedestrians is also taken into account. Due to the typical appearance of congestions in evacuation processes, considering the quickest path normally leads to more realistic and efficient simulations.

In this paper, a fast evacuation method is introduced that tackles the evacuation problem from a different perspective than minimizing the estimated remaining travel time (or travel distance) for each pedestrian. The new perspective is based on minimizing the estimated duration of the remaining global evacuation process. A floor field is efficiently calculated at each time step by assigning pedestrians to exits such that the time steps for complete evacuation are minimized. In order to do that, a wave is initially propagated from each exit. Every iteration in which a wavefront expands and reaches unassigned pedestrians, the wavefront stops during a number of subsequent iterations. In this way, depending on the pedestrian distribution, some wavefronts can be expanding while others are momentarily stopped. When an occupied or unoccupied cell is first reached by a wavefront, the cell is assigned to the exit generating that wavefront; consequently, from that iteration on, the rest of the wavefronts are ignored by the cell. The propagation of waves across the whole room is performed once for each time step in order to obtain a dynamic floor field.

The advantages of the new fast evacuation method are twofold. On the one hand, the effectiveness of the evacuation process is improved since a lower number of time steps is necessary in general. On the other hand, the efficiency is also improved since the running time employed for each time step is reduced. The $\mathcal{O}(|C| \cdot \log |C|)$ complexity of the best quickest-path methods for constructing the floor field at each time step is reduced to $\mathcal{O}(|C|)$ in the case of the new fast evacuation method, where C is the set of cells in the CA. Both advantages are experimentally evaluated in Section 4 by comparing the new fast evacuation method with competitive methods using floor fields based on quickest paths: Flood Fill and the Fast Marching Method.

The rest of the paper is organized as follows. Section 2 reviews several widely used methods for solving the evacuation problem. Section 3 explains the new fast evacuation method in detail.

Section 4 describes a series of experiments for evaluating the new method. Finally, Section 5 summarizes the main results and enumerates future research directions.

2 Related Work

A CA [44, 42, 48] is formed by a regular grid of cells, denoted as C, where each cell $c \in C$ adopts one of a set of states. The three essential characteristics of CA are that they consist of many identical simple processing cells, that interactions between cells take place in a small neighborhood compared to the grid size, and that discrete time is used. In a two-dimensional square grid, the *von Neumann neighborhood* is formed by a cell and its vertical and horizontal neighbors, whereas the *Moore neighborhood* incorporates the diagonal neighbors.

CA for pedestrian dynamics were introduced in the 1990s [11, 12, 13]. Typically, navigation in the CA employs a floor field representing a scalar function that increases with growing distance from the exits. Thus, pedestrians move by minimizing the floor field values in a local way.

Schadschneider introduced a sophisticated CA model of evacuation [33, 3] that represented a breakthrough in the field of pedestrian dynamics. This model probabilistically combines a static floor field based on shortest paths and long-range pedestrian interactions inspired by the process of chemotaxis. The static floor field reflects the distribution of exits and obstacles, while the interactions among pedestrians are implemented through chemotaxis. Other relevant probabilistic models [28, 7, 27] use a multinomial logit formulation in order to specify the probabilities to select a cell.

Posteriorly to the appearance of Schadschneider's model, the introduction of models incorporating a dynamic floor field based on quickest paths [24, 25] became the standard method for modeling evacuation processes. The values of this dynamic floor field highly depend on the distribution of pedestrians.

Due to their close relationship with the present work, shortest-path and quickest-path floor fields are reviewed in the rest of this section.

2.1 Floor Fields Based on Shortest Paths

The first static floor field based on shortest paths [31, 33, 3] directly relied on Euclidean distances. Instead of explicitly specifying a weight (or floor field value) for each CA cell, a navigation field is constructed in terms of the minimum Euclidean distance to an exit by performing the Dijkstra algorithm [8] on a visibility graph. In a visibility graph, only the nodes that are visible to each other (with no obstacles between them) are linked. Unfortunately, this procedure can be rather slow in complex geometries.

Varas *et al.* [43] developed a simple and efficient method for creating shortest-path static floor fields. The method assigns weight 1 to the exit cells, which constitute the initial layer of cells. Then, according to the costs depicted in Figure 1, the weights of the cells belonging to successive layers are iteratively updated. (A new layer corresponds to the set of all neighboring cells of the current layer by considering the Moore neighborhood.) In case of conflict in a cell, the minimum candidate weight is chosen. The iterative process ends when all the cells have been updated and gives rise to $\mathcal{O}(|C|)$ time complexity, where C is the set of cells. The obstacle cells are considered not to be part of any layer and are assigned a high enough weight that prevents pedestrians from occupying them. As summarized in Table 1, this method has been applied by using different $\lambda \in [1, 2]$ values, where λ is the cost associated to the diagonal links in Figure 1. Other applications of the original Varas *et al.* method (using $\lambda = 1.5$) can be found in [30, 2, 1].

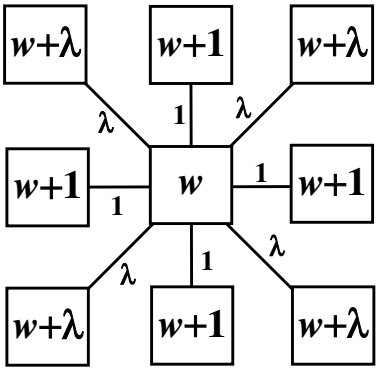

Figure 1: Updating of weights under the method of Varas *et al.* The central cell belongs to the current layer and has weight w. Its vertical and horizontal neighboring cells in the next layer are updated with weight $w + 1$. The diagonal neighboring cells are updated with weight $w + \lambda$, where $\lambda = 1.5$.

Authors	Publication	λ
Tissera *et al.*	[41]	1
Varas *et al.*	[43]	1.5
Huang and Guo	[19]	[1,2]
Kirik *et al.*	[23]	$\sqrt{2}$

Table 1: Review of works applying Varas *et al.* method to calculate static floor fields based on shortest paths. The specific $\lambda \in [1, 2]$ values employed in each work are included, where λ is the cost associated to the diagonal links in Figure 1.

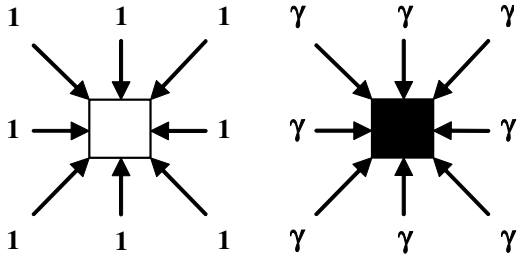

Figure 2: Costs associated to moving to an empty cell (white cell in the left-hand figure) and to an occupied cell (black cell in the right-hand figure) in the FF method. The parameter γ is chosen such that $\gamma > 1$.

Static floor fields indicating shortest paths are independent of pedestrian presence. Therefore, they need to be combined with other dynamic measures that take into account pedestrian distribution [32, 45, 49] in order to produce realistic simulations. A simple and efficient alternative that generates more realistic simulations is to use dynamic floor fields based on quickest paths, which are explained in the next section.

2.2 Floor Fields Based on Quickest Paths

Congestions or jams around exits are usual in evacuations of large crowds. In a congested environment, following the quickest rather than the shortest path to exit is more realistic for pedestrian dynamics simulation. The quickest path from each location to an exit depends on the spatial distribution of pedestrians.

The Flood Fill method (FF) [24, 25] employs a floor field that associates a cost to each CA cell, representing the estimated time spent to move to the cell from any of its neighboring cells. Specifically, a cost equal to unity is associated to empty cells, whereas occupied cells are assigned a cost $\gamma > 1$. This method uses the neighborhood relationships among cells[1] and the Dijkstra algorithm in the induced graph to calculate the quickest path from each CA cell to exit. The Dijkstra algorithm is an efficient algorithm whose time complexity is $\mathcal{O}(|C| \cdot \log |C|)$, where C is the set of cells. This is due to the fact that the list containing the visited cells to be expanded can be implemented through a heap data structure. The operations of extracting the best cell from the heap and inserting its neighboring unvisited cells can be efficiently implemented in $\mathcal{O}(\log |C|)$ time. An advantage of FF is that the first weight calculated by the Dijkstra algorithm for a cell does not need to be reconsidered. This is a consequence of the way FF assigns costs to cells (see Figure 2). However, FF produces somewhat unrealistic movement by pedestrians, which tend to form square jams around exits instead of circular ones. This drawback was partially solved in [26], where the costs associated to the diagonal links in Figure 2 are multiplied by $\sqrt{2}$. This correction to the diagonal costs provokes that the weights calculated by the Dijkstra algorithm for a cell may need to be reconsidered over the course of the execution. The method that corrects the diagonal costs will be denoted as $\text{FF}_{\sqrt{2}}$.

An approach similar to FF that produces more realistic pedestrian dynamics consists in applying the Fast Marching Method (FMM) [35] to obtain the cell weights. The underlying idea, rather than calculating minimum distances in a graph of costs, is to expand a wavefront from

[1]Even though von Neumann or Moore neighborhoods can be used in FF, the Moore neighborhood will be assumed for this method in the rest of the paper.

each exit in the 2D domain of interest. Thus, the cell weights are established as the travel time of the wavefront, which coincides with the shortest distance to exit when the wavefront propagates across empty cells with a velocity equal to unity. The arrival time of the wavefront to point x, $T(x)$, is determined by the Eikonal equation:

$$\begin{array}{rl} |\nabla T(x)| = \frac{1}{F(x)} & \text{for } x \in \Omega \text{ (an open set in 2D)} \\ T(x) = 0 & \text{for } x \in \delta\Omega \text{ (the boundary of } \Omega \text{ or exit points)} \end{array}, \tag{1}$$

where $F(x)$ is the velocity of the wavefront such that $F(x) = 0$ in obstacle cells, $F(x) = 1$ in empty cells, and $F(x) = 1/\gamma$ (with $\gamma > 1$) in occupied cells. FMM efficiently solves the Eikonal equation on a discretized grid (see [6] for a comparison with A* search) through a finite difference approximation of Equation 1 resulting in the following formula:

$$\{\max(0, T_{ij} - T_{i-1,j}, T_{ij} - T_{i+1,j})\}^2 + \{\max(0, T_{ij} - T_{i,j-1}, T_{ij} - T_{i,j+1})\}^2 = \frac{1}{F_{ij}^2}, \tag{2}$$

where T_{ij} and F_{ij} stand for the value at cell (i, j) of T and F respectively. FMM considers the von Neumann neighborhood rather than the Moore neighborhood, for example when applying Equation 2 to calculate a candidate weight T_{ij} for cell (i, j). (T_{ij} is obtained by solving the quadratic equation defined by Equation 2.) Like FF, the order of cell updating in FMM is carried out by using a heap data structure, which gives rise to $\mathcal{O}(|C| \cdot \log |C|)$ time complexity. Nonetheless, unlike FF under the costs depicted in Figure 2, FMM needs to reconsider some of the weight updates over the course of its execution. FMM is widely used in evacuation modeling due to its realistic results.

3 The New Fast Evacuation Method

3.1 Overview

The Fast Evacuation Method (FEM) is a CA model of pedestrian evacuation that aims at minimizing the duration of the whole evacuation process by employing a dynamic floor field that distributes the evacuation workload among exits in an equitable way. The floor field defined by FEM at each time step can be calculated in an efficient manner in time proportional to the number of CA cells and gives rise to a fast and effective evacuation process.

As in typical CA, discrete time and space are assumed. Space is divided into square cells of equal size [2] . Each CA cell can be empty, occupied by a pedestrian (only one at a time), represent an exit, or represent an obstacle. In each time step, the pedestrians are updated asynchronously in a random order, according to the values contained in a dynamic floor field. Every pedestrian moves by occupying the neighboring empty cell with a minimum floor field value. Pedestrians are assumed to know their environment or, equivalently, be in contact with an external agent that provides them with the floor field information.

In each time step, FEM calculates a complete floor field ϕ previously to moving each pedestrian to a neighboring cell.[3] The floor field ϕ is calculated in an iterative way such that a wavefront is initiated from every exit cell at the first iteration. The arbitrary neighborhood relationship

[2]Specifically (see for example [33]), it is usual to consider in the literature that every cell is 40cm x 40cm (typical space occupied by a pedestrian in a dense crowd) and a single pedestrian (not interacting with others) moves at a velocity of one cell per time step. Since the empirical average velocity of a pedestrian is about 1.3 m/s, this gives an estimated time step of 0.3 seconds.

[3]In this work, the Moore neighborhood is considered for the movement of pedestrians.

established for wavefront propagation determines the wavefront shape. (The Moore and von Neumann neighborhoods give rise to square wavefronts; however, in Section 3.3 two more convenient neighborhoods are defined for the evacuation problem which produce wavefronts more similar to circles.) A wavefront freely propagates across empty cells and, when occupied cells are reached, the wavefront stops for a number of iterations equal to the number of pedestrians reached at the current iteration. Thus, it is possible that some wavefronts keep propagating while others remain stopped. The value of ϕ for a cell is determined by the iteration in which the first wavefront arrives in the cell.

3.2 Pseudocode

Figure 3 contains the pseudocode of the FF_{FEM} algorithm, which calculates the floor field ϕ used by FEM at each time step. First, lines 1-8 in Figure 3 initialize the exit cells. Likewise, the rest of non-obstacle cells in the grid are initialized in lines 9-11. Then, an iterative process takes place in lines 12-38. This iterative process corresponds to the propagation of wavefronts across the grid.

The following variables and functions are used by FF_{FEM}:

- ***c*.assignedExit:** Variable that contains the exit assigned to a cell c. This exit is the origin of the wavefront that first arrived in the cell.
- ***c*.isUpdated?:** Variable that stores whether or not the floor field value of a cell c has already been calculated.
- **delayOfExit[]:** Array indexed by an exit that specifies the number of remaining iterations that the wavefront originated at the given exit will be stopped.
- **wavefront[]:** Array indexed by an exit that stores the cells that are part of the current wavefront originated at the given exit.
- **goOn?:** Variable used for the termination condition. The algorithm ends after the current iteration (see line 35 in Figure 3) if no wavefront is delayed and no wavefront has propagated at the current iteration.
- **activeCells:** Variable that contains those cells in **wavefront**[] whose assigned exit is not delayed.
- **updateExitDelays1:** Function that carries out the normal updating of delays for the exits after one iteration (see line 17 in Figure 3). For example, if there are five exits whose delays are stored in the array [0 5 0 4 1], the new delays for the exits will be [0 4 0 3 0]. Therefore, the delays equal to 0 are not changed, while the rest of delays are decremented by 1.
- $\mathcal{N}_{\mathbf{FEM}}$: Function that returns the cells that belong to the neighborhood of **activeCells** and have not been updated yet (see line 15 in Figure 3). Several types of neighborhood can be arbitrarily used in this function. In Section 3.3, two convenient alternatives to the von Neumann and Moore neighborhoods are introduced that give rise to wavefronts more similar to circles.
- **newCells:** Variable that stores the cells generated after **activeCells** expansion through $\mathcal{N}_{\mathbf{FEM}}$. Note that each of these cells is assigned an exit in line 21. If the cell belongs to the Moore neighborhoods of wavefront cells with different assigned exits, it is finally assigned the exit of its nearest wavefront cell. (The Euclidean distance $\delta_{\text{Euclidean}}$ from the cell to a neighboring wavefront cell can only be 1 or $\sqrt{2}$.)

Floor Field Method	Time Complexity
Varas *et al.* (shortest path)	$\mathcal{O}(\|C\|)$
FF (quickest path)	$\mathcal{O}(\|C\| \cdot \log \|C\|)$
FMM (quickest path)	$\mathcal{O}(\|C\| \cdot \log \|C\|)$
FEM	$\mathcal{O}(\|C\|)$

Table 2: Time complexity of several floor field methods.

- ***c*.isOccupied?:** Variable that determines whether or not a cell c is occupied by a pedestrian.
- **updateExitDelays2:** Function that develops the updating of delays when none of them is equal to zero (see line 29 in Figure 3). For example, if there are five exits whose delays are stored in the array [2 5 6 4 2], the new delays for the exits will be [0 3 4 2 0]. Thus, the delays are decremented by the minimum of them. This operation improves the efficiency of the algorithm, since it makes no sense that all the wavefronts are stopped at the same time.
- **updateExitDelays3:** Function that carries out the updating of delays when **newCells** is empty and at least one exit delay is greater than zero (see line 33 in Figure 3). For example, under the mentioned conditions, if there are five exits whose delays are stored in the array [0 0 0 4 2], the new delays for the exits will be [0 0 0 2 0]. Therefore, the delays greater than zero are decremented by the minimum of them, which makes it possible that a wavefront assigned to a new exit is checked for expansion at the next iteration.

The time complexity of FF_{FEM} is $\mathcal{O}(|C|)$, where C is the set of CA cells. As shown in Table 2, this constitutes an improvement of efficiency in comparison with the floor fields based on quickest paths (see Section 2.2), whose time complexity is $\mathcal{O}(|C| \cdot \log |C|)$. The additional $\log |C|$ factor is a consequence of applying the Dijkstra algorithm. Figure 4 contains an example of application of FF_{FEM} to a 16x11 grid with pedestrians depicted in black, obstacles in dark gray, exits in light gray, and empty locations in white. For the sake of simplicity, the Moore neighborhood is employed in Figure 4 for wavefront propagation.

3.3 Neighborhood for Wavefront Propagation in FEM

In line 15 of algorithm FF_{FEM} (see Figure 3), $\mathcal{N}_{\text{FEM}}$ calculates the neighborhoods of the propagating wavefronts. The most immediate option is to use either the von Neumann neighborhood or the Moore neighborhood; however, these two options tend to produce wavefronts and jams of square shape, which is not realistic. Furthermore, some preliminary experiments conducted in the context of this work show that wavefronts more similar to circles allow both more realistic and faster evacuations to be obtained. In this way, two interesting options were initially considered: (1) nearly octagonal wavefronts generated by alternately applying the von Neumann neighborhood and the Moore neighborhood (see Figure 5) and (2) nearly circular wavefronts generated by applying a probabilistic neighborhood. The latter turned out to produce results somewhat better than those produced by the former; thus, this paper assumes that a probabilistic neighborhood is employed for wavefront propagation in FF_{FEM}. The probabilistic neighborhood is introduced in the rest of this section and, to the best of the author's knowledge, is a novel contribution of this work.

The *probabilistic neighborhood* of a cell $c \in C$ is denoted as $\mathcal{N}_\sigma(c)$, where $\sigma \in [0, 1]$. A cell

Algorithm $\mathbf{FF_{FEM}}$: ***Generation of the FEM Floor Field***

Input: A CA with non-obstacle cells $M \subset C$ and exit cells $E \subset M$
Output: A floor field ϕ defined over M

```
    % Initial updating of E
 1: i ← 0
 2: for c ∈ E
 3:         φ(c) ← i
 4:         c.assignedExit ← c
 5:         c.isUpdated? ← true
 6:         delayOfExit[c] ← 0
 7:         wavefront[c] ← {c}
 8: end-for

    % Initial updating of M \ E
 9: for c ∈ M \ E
10:         c.isUpdated? ← false
11: end-for

    % Main loop
12: goOn? ← true
13: while goOn? = true
14:         ActiveCells ← {x ∈ wavefront[c] | delayOfExit[c] = 0} ∀c ∈ E
15:         updateExitDelays1
16:         newCells ← {x ∈ N_FEM(ActiveCells) | x.isUpdated? = false}
17:         if newCells ≠ ∅
18:             i ← i + 1
19:             for c ∈ newCells
20:                 φ(c) ← i
21:                 c.assignedExit ← c'.assignedExit | c' = arg min_x {δ_Euclidean(c, x)} ∀x ∈ N_Moore(c) ∩ ActiveCells
22:                 c.isUpdated? ← true
23:                 if c.isOccupied?
24:                     delayOfExit[c.assignedExit] ← delayOfExit[c.assignedExit] + 1
25:                 end-if
26:                 wavefront[c.assignedExit] ← wavefront[c.assignedExit] ∪ {c}
27:             end-for
28:             if ∄c ∈ E | delayOfExit[c] = 0
29:                 updateExitDelays2
30:             end-if
31:         else
32:             if ∃c ∈ E | delayOfExit[c] > 0
33:                 updateExitDelays3
34:             else
35:                 goOn? ← false
36:             end-else
37:         end-else
38: end-while
```

Figure 3: Pseudocode for the $\mathrm{FF_{FEM}}$ algorithm that generates the floor field used by FEM in each time step.

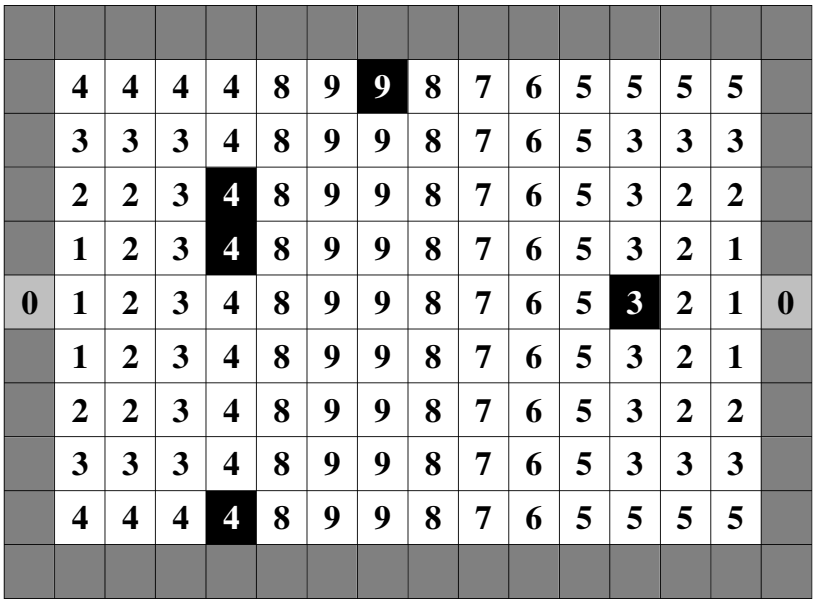

Figure 4: Example of application of $\mathrm{FF}_{\mathrm{FEM}}$, the algorithm in Figure 3 that is executed at each time step to calculate the dynamic floor field used by FEM. For the sake of simplicity, the Moore neighborhood is employed for wavefront propagation.

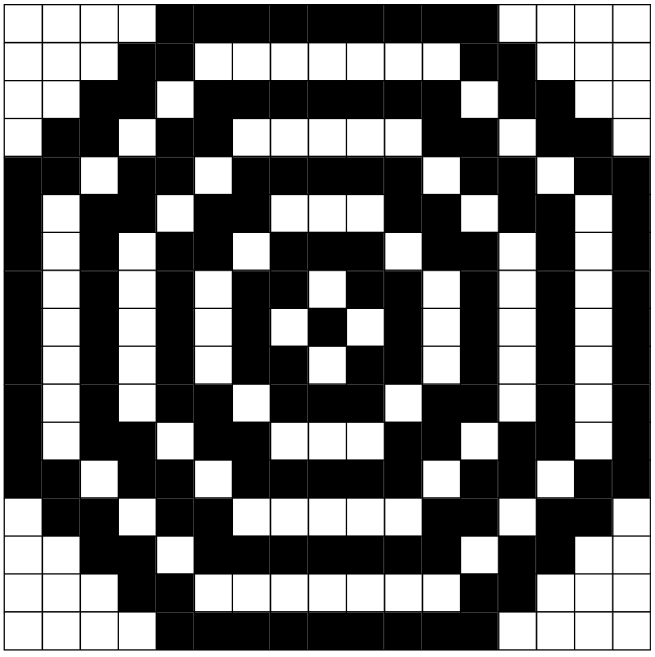

Figure 5: Nearly octagonal wavefront generated by alternately applying the von Neumann neighborhood (in white) and the Moore neighborhood (in black).

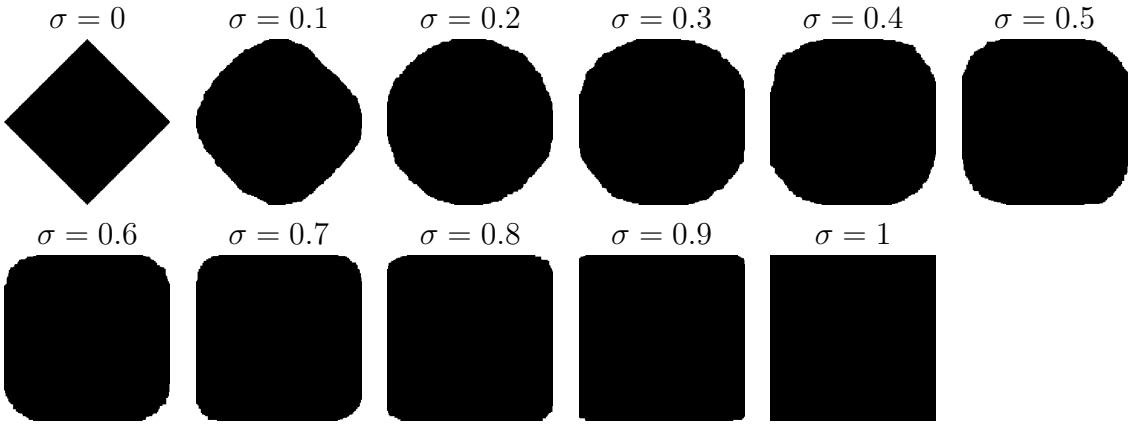

Figure 6: Examples of wavefront generated through $\mathcal{N}_\sigma$ for $\sigma \in \{0, 0.1, \ldots, 0.9, 1\}$. Each wavefront corresponds to one hundred iterations using a 201x201 grid.

$c' \in C$ belongs to $\mathcal{N}_\sigma(c)$ with a probability defined by the following expression:

$$P(c' \in \mathcal{N}_\sigma(c)) = \begin{cases} 0 & \text{if } c' \notin \mathcal{N}_{\mathrm{M}}(c) \\ 1 & \text{if } c' \in \mathcal{N}_{\mathrm{N}}(c) \\ \sigma & \text{if } c' \in \mathcal{N}_{\mathrm{M}}(c) \setminus \mathcal{N}_{\mathrm{N}}(c) \end{cases},$$

where $\mathcal{N}_{\mathrm{N}}(c)$ and $\mathcal{N}_{\mathrm{M}}(c)$ represent the von Neumann and Moore neighborhoods of c respectively. Note that $\mathcal{N}_\sigma$ is a generalization of $\mathcal{N}_{\mathrm{N}}$ and $\mathcal{N}_{\mathrm{M}}$, since $\mathcal{N}_\sigma \equiv \mathcal{N}_{\mathrm{N}}$ if $\sigma = 0$ and $\mathcal{N}_\sigma \equiv \mathcal{N}_{\mathrm{M}}$ if $\sigma = 1$. Figure 6 contains several examples of wavefront generated by applying $\mathcal{N}_\sigma$ for one hundred iterations and $\sigma \in \{0, 0.1, \ldots, 0.9, 1\}$. In this paper, $\mathcal{N}_{\sigma=0.2}$ is the neighborhood selected for wavefront propagation with promising results.

3.4 Example of Application of FEM

Consider a 225x150 grid with the distribution of pedestrians, exits, and obstacles illustrated in Figure 7 for $t = 0$. In this figure, empty cells are depicted in white, exits in light gray, obstacles in dark gray, and pedestrians in black. There are 584 pedestrians distributed in nine groups. Two exits are located at coordinates $(20, 75)$ and $(205, 77)$ respectively, where $(0, 0)$ corresponds to the coordinates of the bottom left cell. The border of the grid is occupied by obstacles.

In Figure 7, the competitive FMM method explained in Section 2.2 and the FEM method are compared for time steps $t \in \{10, 50, 125, 300\}$. Note that FMM is sensitive to parameter γ (the cost for each occupied cell), and establishing a satisfactory γ value is problem-dependent and time-consuming for the user. This is an important advantage for FEM, which requires no parameter fine-tuning. In the problem presented in Figure 7, $\gamma = 50$ constitutes an appropriate value for FMM.

It can be observed in Figure 7 that all the pedestrians head to the left exit at $t = 10$ in the case of FMM. This is due to the fact that, at the first ten time steps, the quickest path for every pedestrian ends at the left exit. At $t = 50$, the pedestrians are near the left exit, where a congestion is starting to form. At $t = 125$, the jam around the left exit provokes that some peripheral pedestrians head to the right exit, which is now more promising in terms of evacuation time. Finally, at $t = 300$, both exits are congested and 301 pedestrians remain to be evacuated.

In the case of FEM, Figure 7 shows that a subset of the pedestrians head to the right exit at $t = 10$. This is because, even at the first ten time steps, both exits have been assigned pedestrians.

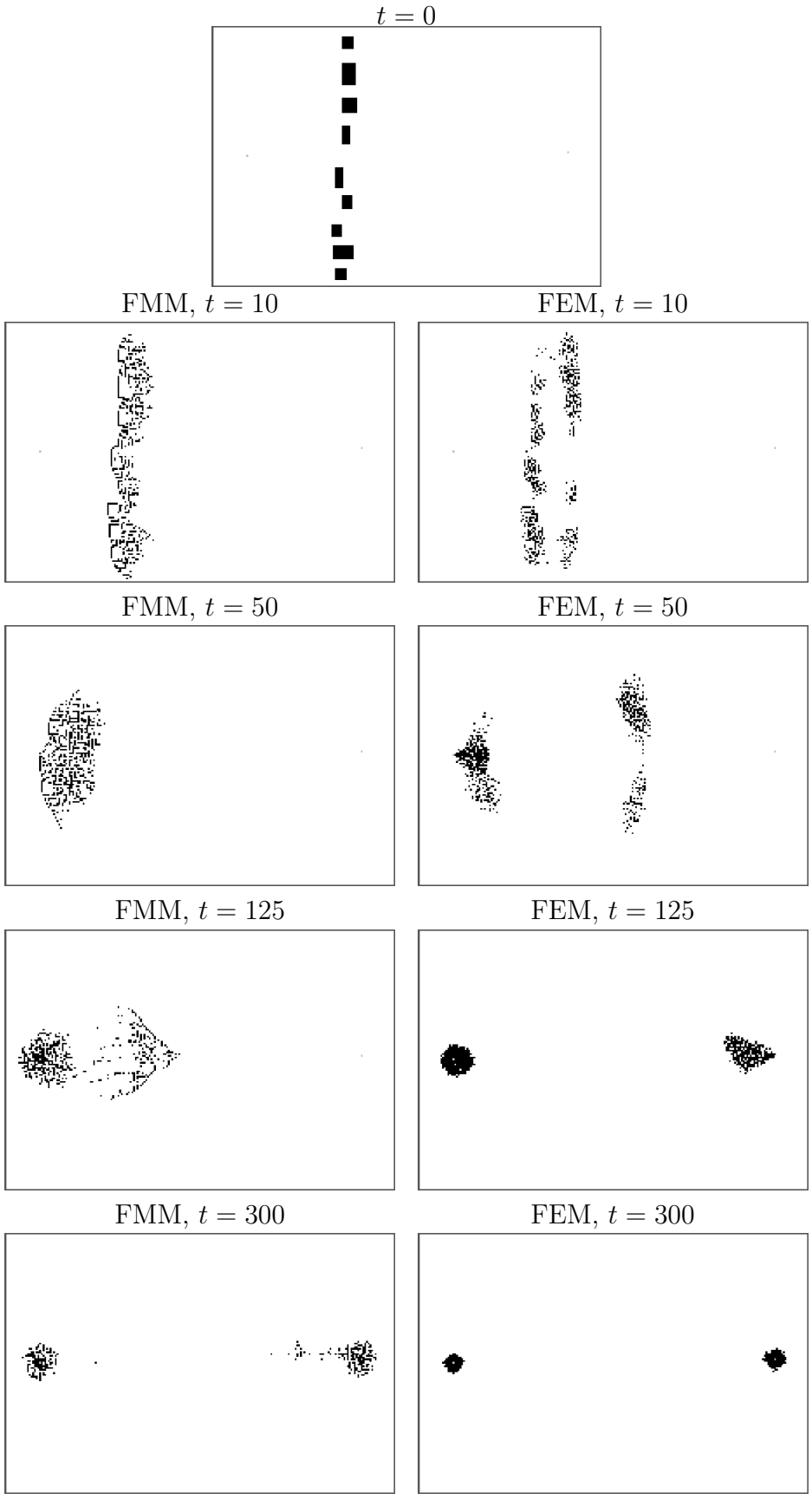

Figure 7: Example of evacuation simulation for 584 pedestrians, distributed in 9 groups, through two exits (top figure). FMM (with cost $\gamma = 50$ for each occupied cell) and FEM are compared for time steps $t \in \{10, 50, 125, 300\}$.

Evacuation Problem	Grid	Pedestrians	Exits	Obstacles
`9groups`	225x150	584	2	746
`lecture-hall`	225x150	8760	33	9473
`corner`	225x150	8436	109	9182
`u-turn`	225x150	7210	77	849
`random500`	225x150	500	4	746
`random500bottlenecks`	225x150	500	4	1210

Table 3: Structural characteristics of the six evacuation problems illustrated in Figure 8.

At $t = 50$, some pedestrians are on their way to the right exit, much earlier than under FMM. At $t = 125$, a jam around the right exit starts appearing, again much earlier than under FMM. Finally, at $t = 300$, only 198 pedestrians remain to be evacuated (301 under FMM at this time step) and are equally distributed between both exits.

In summary, for the evacuation problem illustrated in Figure 7, FEM develops a more effective evacuation process compared to FMM. At $t = 300$, 103 out of the 584 initial pedestrians have been evacuated under FEM, but remain on the grid under FMM. In the next section, a comparative evaluation is carried out of FEM, FMM, and other evacuation methods. Evacuation effectiveness and simulation runtime are analyzed over a set of evacuation problems.

4 Experimental Evaluation

This section comparatively evaluates FEM with three competitive evacuation methods using dynamic floor fields based on quickest paths (see Section 2.2): FF, $\text{FF}_{\sqrt{2}}$, and FMM. The four evaluated algorithms are implemented within NetLogo [47], an agent-based modeling and programming environment well suited for modeling and inspecting complex systems developing over time. The experiments are carried out on an Intel Core i5 processor (2.67 GHz) with 8 Gb of memory.

Six different evacuation problems are used for the comparative evaluation: (1) the problem presented in Section 3.4, (2) three problems widely used in the literature for evacuations in a lecture hall, a corner, and a u-turn, and (3) two problems with initially randomly distributed pedestrians. Figure 8 illustrates the six evacuation problems, and Table 3 shows their main structural characteristics. These problems cover a wide range of crowd motion features as classified in [5, 10, 9, 14, 15, 36, 38], with the peculiarity that the present work deals with evacuation processes.

The best values for parameter γ in FF, $\text{FF}_{\sqrt{2}}$, and FMM are selected via manual fine-tuning. This constitutes a costly and problem-dependent task for the user and is an important disadvantage of these three algorithms with respect to FEM, which requires no parameter fine-tuning. The selected γ values (see Table 4) are those leading to the best results regarding the average number of time steps necessary to evacuate a pedestrian in a simulation, which is called "mean evacuation time steps" later in Section 4.1.

There are two main aspects to be considered when the performance of an evacuation method is evaluated: on the one hand, the effectiveness of the evacuation process and, on the other hand, the efficiency of the simulation. These two aspects are explored in the following two sections, in which every single considered measure is averaged over ten evacuation simulations.

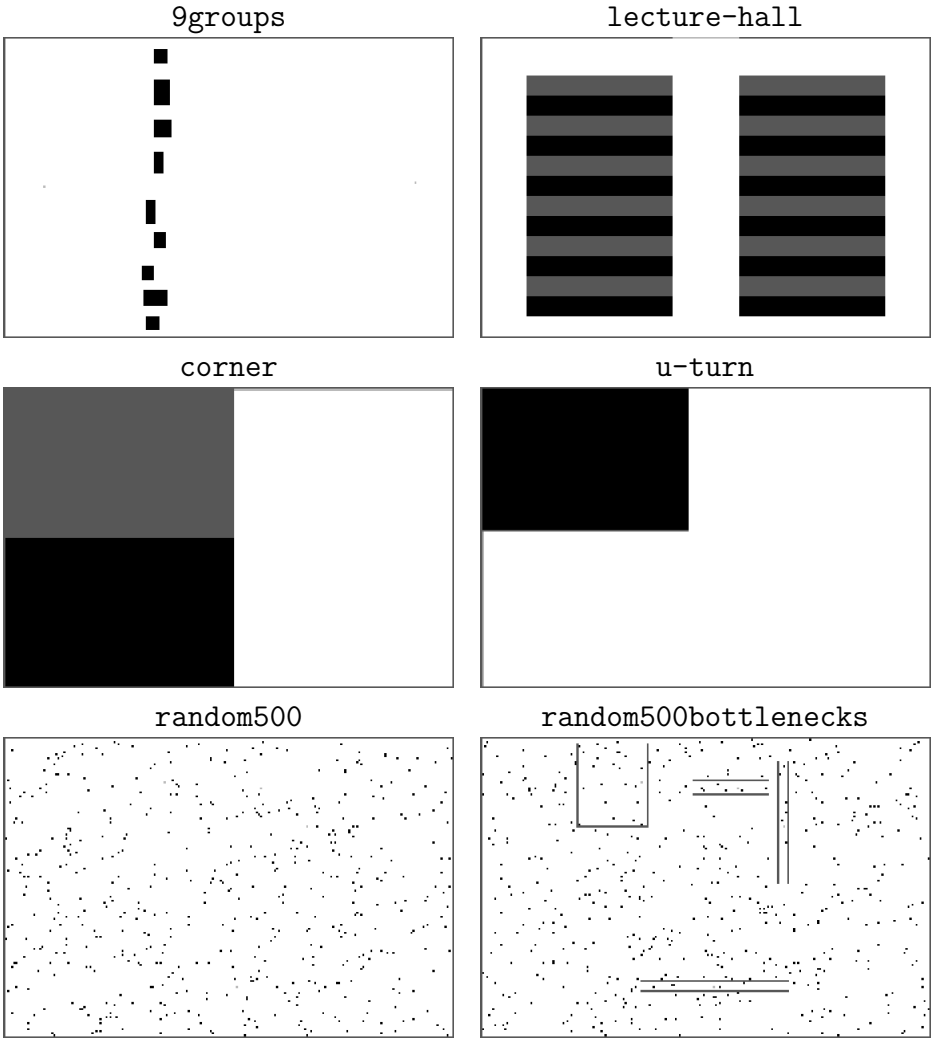

Figure 8: The six evacuation problems used in the experimental evaluation. The empty cells of the 225x150 grid are depicted in white color, the exits in light gray, the obstacles in dark gray, and the pedestrians in black.

Evacuation Problem	γ (FF)	$\gamma\left(\mathrm{FF}_{\sqrt{2}}\right)$	γ (FMM)
`9groups`	51	53	50
`lecture-hall`	1.75	1.5	1.5
`corner`	3.75	3.25	3.5
`u-turn`	7.5	7.25	6
`random500`	32	30	31
`random500bottlenecks`	32	24	30

Table 4: Manually fine-tuned best γ values for each evacuation problem in Figure 8 under FF, $\mathrm{FF}_{\sqrt{2}}$, and FMM.

Evacuation Problem	*mets*(FF)	*mets*($FF_{\sqrt{2}}$)	*mets*(FMM)	*mets*(FEM)
`9groups`	311.707	353.317	291.197	**246.808**
`lecture-hall`	291.123	286.966	**282.136**	286.714
`corner`	359.721	327.936	296.405	**271.820**
`u-turn`	469.697	452.804	398.804	**397.668**
`random500`	86.020	88.318	83.430	**77.172**
`random500bottlenecks`	101.940	101.530	102.928	**97.740**

Table 5: *Mean evacuation time steps* obtained for the evaluated methods over the six evacuation problems defined in Table 3. The best result in each row appears highlighted in bold.

Evacuation Problem	*gets*(FF)	*gets*($FF_{\sqrt{2}}$)	*gets*(FMM)	*gets*(FEM)
`9groups`	551	674	469	**427**
`lecture-hall`	587	557	**533**	548
`corner`	598	564	491	**454**
`u-turn`	739	719	**624**	660
`random500`	202	222	217	**168**
`random500bottlenecks`	233	236	252	**231**

Table 6: *Global evacuation time steps* obtained for the evaluated methods over the six evacuation problems defined in Table 3. The best result in each row appears highlighted in bold.

4.1 Evacuation Effectiveness

In this section, the *mean evacuation time steps* (*mets*) are the measure used to evaluate evacuation effectiveness. It is defined as the average number of time steps that it takes a pedestrian to exit the grid of cells in an evacuation simulation. In each time step, the floor field is calculated for the whole grid and, subsequently, each pedestrian moves to one of the empty neighboring cells according to the floor field values calculated for that time step. Table 5 contains the *mets* obtained for FF, $FF_{\sqrt{2}}$, FMM, and FEM when applied to the evacuation problems defined in Table 3.

As a complementary measure, the *global evacuation time steps* (*gets*) are included in Table 6. This measure represents the number of time steps that it takes the grid to become empty in a simulation.

The example described in Section 3.4 showed that, for the `9groups` evacuation problem, FEM develops a more effective evacuation process compared to FMM. This preliminary result is now confirmed by the measures reported in Tables 5 and 6 for the `9groups` problem.

From the results in Tables 5 and 6, FMM and FEM clearly outperform FF and $FF_{\sqrt{2}}$ in terms of effectiveness. Furthermore, regarding the *mets* measure in Table 5, FEM is the best method in five out of the six evacuation problems.

4.2 Simulation Efficiency

This section employs the simulation runtime to evaluate the efficiency of the evaluated evacuation methods. Table 7 includes the running times obtained for FF, $FF_{\sqrt{2}}$, FMM, and FEM when applied to the evacuation problems defined in Table 3.

Additionally, it is interesting to calculate the runtime (see Table 7) divided by the evacuation time steps (see Table 6). Table 8 shows these results for each problem simulation, which give an idea of the efficiency of the methods to carry out the tasks included in a time step.

Evacuation Problem	*runtime*(FF)	*runtime*($FF_{\sqrt{2}}$)	*runtime*(FMM)	*runtime*(FEM)
`9groups`	306.996	499.337	444.941	**96.263**
`lecture-hall`	238.542	285.889	348.135	**86.660**
`corner`	242.668	291.510	325.499	**88.848**
`u-turn`	401.885	498.482	562.977	**153.309**
`random500`	117.942	165.926	201.833	**34.727**
`random500bottlenecks`	132.516	171.931	235.217	**47.847**

Table 7: Runtimes in seconds obtained for the evaluated methods over the six evacuation problems defined in Table 3. The best result in each row appears highlighted in bold.

Evacuation Problem	$\frac{runtime}{gets}$(FF)	$\frac{runtime}{gets}$($FF_{\sqrt{2}}$)	$\frac{runtime}{gets}$(FMM)	$\frac{runtime}{gets}$(FEM)
`9groups`	0.557	0.741	0.949	**0.225**
`lecture-hall`	0.406	0.513	0.653	**0.158**
`corner`	0.406	0.517	0.663	**0.196**
`u-turn`	0.544	0.693	0.902	**0.232**
`random500`	0.584	0.747	0.930	**0.207**
`random500bottlenecks`	0.569	0.728	0.933	**0.207**

Table 8: Running times in Table 7 divided by the *global evacuation time steps* in Table 6. The best result in each row appears highlighted in bold.

Table 7 shows that, for the global evacuation process, FEM is the most efficient method with FF in second place. It is interesting to observe that FEM turns out to be more efficient than FMM. Exclusively in terms of one time step, these same conclusions can be obtained from Table 8 as well.

4.3 Visual Evaluation

Systems are also evaluated by visualizing snapshots of them at particular time steps. In order to do that, Figure 9 illustrates a comparison of FEM and FMM in which an evacuation snapshot is included for both of them when applied to the six problems defined in Table 3. FMM is employed in this section because it constitutes the most widely used method for evacuation simulation, due to its realistic and competitive results. The specific number of time steps selected for each snapshot in Figure 9 corresponds to half the time steps reported for *gets* in Table 6.

Figure 9 shows that FEM forms jams around the exits earlier than FMM. For example, this can be clearly observed in the case of `9groups` and `random500bottlenecks`. Furthermore, FEM develops a pretty homogeneous use of all the available exits in the evacuation process. In particular, this is more evident in the case of `corner`.

5 Conclusion and Future Research

This paper introduces a novel evacuation method called FEM which is based on distributing pedestrians among exits in a homogeneous way, so that the evacuation workload is shared by the exits in an effective fashion. FEM efficiently creates a dynamic floor field by expanding from each exit a wavefront that stops when new pedestrians are reached. The new method is experimentally

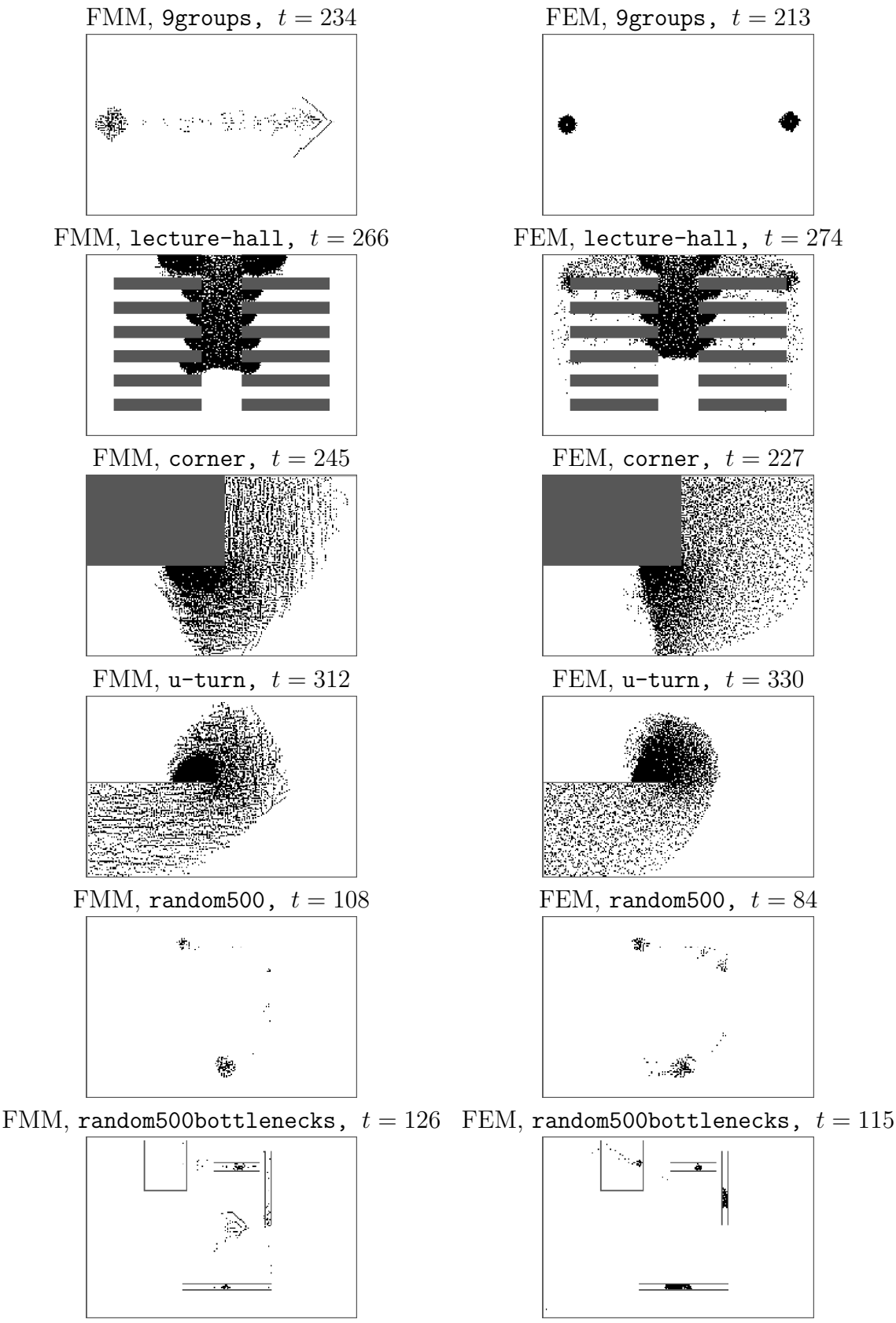

Figure 9: Visual comparison between FEM (right) and FMM (left) for the six evacuation problems defined in Table 3. For each case, a snapshot is included correponding to half the number of time steps reported in Table 6.

evaluated and compared with other competitive evacuation methods with promising results. In particular, FEM outperforms FMM in most of the tested evacuation problems. This is a remarkable result since FMM is one of the most widely used methods for evacuation simulation, due to its efficiency and realistic results. An important advantage of FEM is that, contrarily to FMM, no parameter fine-tuning is required.

This paper also introduces the concept of *probabilistic neighborhood*, which generalizes the typical von Neumann and Moore neighborhoods in CA. The application of the probabilistic neighborhood in wavefront propagation allows FEM to simulate circular wavefronts in a very simple way, even in the presence of any kind of obstacles. Circular wavefronts give rise to realistic evacuation dynamics and are also simulated in competitive methods like FMM.

The present work opens up the following research directions:

- It would be interesting to investigate whether FEM can be easily adapted to the context of the Social Force Model for pedestrian dynamics.
- FEM can be easily applied to dynamic environments where exits and obstacles change over time. This could produce important applications with a high impact on society.
- The viability and benefits of the static version of FEM for some evacuation problems could be investigated. Static FEM would use the floor field created in the initial time step as the floor field for the whole evacuation process. This static version would be much faster than FEM, although it would be subject to some inaccuracies.
- In order to perform a thorough evaluation of FEM, its simulation results should be compared with existing empirical data [36, 46].
- Regarding the practical application of FEM, it remains to study how this new method can be achieved in real-life evacuation scenarios [46].

References

[1] L. E. Aik and T. W. Choon. Simulating evacuations with obstacles using a modified dynamic cellular automata model. *Journal of Applied Mathematics*, 2012. Article ID 765270, 17 pages.

[2] R. Alizadeh. A dynamic cellular automaton model for evacuation process with obstacles. *Safety Science*, 49:315–323, 2011.

[3] C. Burstedde, A. Kirchner, K. Klauck, A. Schadschneider, and J. Zittartz. Cellular automaton approach to pedestrian dynamics - applications. In M. Schreckenberg and S. D. Sharma, editors, *Pedestrian and Evacuation Dynamics*, pages 87–98. Springer, 2002.

[4] C. Burstedde, K. Klauck, A. Schadschneider, and J. Zittartz. Simulation of pedestrian dynamics using a two-dimensional cellular automaton. *Physica A: Statistical Mechanics and its Applications*, 295(3-4):507–525, 2001.

[5] X. Chen, M. Treiber, V. Kanagaraj, and H. Li. Social force models for pedestrian traffic - state of the art. *Transport Reviews*, 38(5):625–653, 2018.

[6] C. H. Chiang, P. J. Chiang, J. Fei, and J. S. Liu. A comparative study of implementing Fast Marching Method and A* SEARCH for mobile robot path planning in grid environment: Effect of map resolution. In *IEEE Workshop on Advanced Robotics and Its Social Impacts (ARSO 2007)*, pages 1–6, 2007.

[7] C. Dias and R. Lovreglio. Calibrating cellular automaton models for pedestrians walking through corners. *Physics Letters A*, 382(19):1255–1261, 2018.

[8] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959.

[9] D. C. Duives, W. Daamen, and S. P. Hoogendoorn. State-of-the-art crowd motion simulation models. *Transportation Research Part C*, 37:193–209, 2013.

[10] D. C. Duives, W. Daamen, and S. P. Hoogendoorn. Continuum modelling of pedestrian flows - Part 2: Sensitivity analysis featuring crowd movement phenomena. *Physica A*, 447:36–48, 2016.

[11] M. Fukui and Y. Ishibashi. Traffic flow in 1D cellular automaton model including cars moving with high speed. *Journal of the Physical Society of Japan*, 65(6):1868–1870, 1996.

[12] M. Fukui and Y. Ishibashi. Jamming transition in cellular automaton models for pedestrians on passageway. *Journal of the Physical Society of Japan*, 68(11):3738–3739, 1999.

[13] M. Fukui and Y. Ishibashi. Self-organized phase transitions in cellular automaton models for pedestrians. *Journal of the Physical Society of Japan*, 68(8):2861–2863, 1999.

[14] D. Helbing. Traffic and related self-driven many-particle systems. *Reviews of Modern Physics*, 73(4):1067–1141, 2001.

[15] D. Helbing, L. Buzna, A. Johansson, and T. Werner. Self-organized pedestrian crowd dynamics: Experiments, simulations, and design solutions. *Transportation Science*, 39(1):1–24, 2005.

[16] D. Helbing, I. Farkas, and T. Vicsek. Simulating dynamical features of escape panic. *Nature*, 407(6803):487–490, 2000.

[17] D. Helbing, M. Isobe, T. Nagatani, and K. Takimoto. Lattice gas simulation of experimentally studied evacuation dynamics. *Physical Review E*, 67(6):067101, 2003.

[18] D. Helbing and P. Molnár. Social force model for pedestrian dynamics. *Physical Review E*, 51(5):4282–4286, 1995.

[19] H. J. Huang and R. Y. Guo. Static floor field and exit choice for pedestrian evacuation in rooms with internal obstacles and multiple exits. *Physical Review E*, 78(2):021131, 2008.

[20] M. Isobe, D. Helbing, and T. Nagatani. Experiment, theory, and simulation of the evacuation of a room without visibility. *Physical Review E*, 69(6):066132, 2004.

[21] A. Kirchner, H. Klüpfel, K. Nishinari, A. Schadschneider, and M. Schreckenberg. Simulation of competitive egress behavior: comparison with aircraft evacuation data. *Physica A: Statistical Mechanics and its Applications*, 324(3-4):689–697, 2003.

[22] A. Kirchner and A. Schadschneider. Simulation of evacuation processes using a bionics-inspired cellular automaton model for pedestrian dynamics. *Physica A: Statistical Mechanics and its Applications*, 312(1-2):260–276, 2002.

[23] E. Kirik, T. Yurgel'yan, and D. Krouglov. The shortest time and/or the shortest path strategies in a CA FF pedestrian dynamics model. *Journal of Siberian Federal University. Mathematics & Physics*, 2(3):271–278, 2009.

[24] T. Kretz. Pedestrian traffic: on the quickest path. *Journal of Statistical Mechanics: Theory and Experiment*, 3:P03012, 2009.

[25] T. Kretz. The use of dynamic distance potential fields for pedestrian flow around corners. In *Proceedings of the 1st International Conference on Evacuation Modeling and Management (ICEM 2009)*, 2009.

[26] T. Kretz, C. Bönisch, and P. Vortisch. Comparison of various methods for the calculation of the distance potential field. In *Proceedings of the 9th International Conference on Pedestrian and Evacuation Dynamics (PED 2008)*, pages 335–346, 2010.

[27] R. Lovreglio, C. Dias, X. Song, and L. Ballerini. Towards microscopic calibration of pedestrian simulation models using open trajectory datasets: The case study of the Edinburgh Informatics Forum. In *Proceedings of the Conference on Traffic and Granular Flow (TGF 2017)*, 2017.

[28] R. Lovreglio, C. Dias, X. Song, and L. Ballerini. Investigating pedestrian navigation in indoor open space environments using big data. *Applied Mathematical Modelling*, 62:499–509, 2018.

[29] F. Martínez-Gil, M. Lozano, I. García-Fernández, and F. Fernández. Modeling, evaluation and scale on artificial pedestrians: a literature review. *ACM Computing Surveys*, 50(5):Article No. 72, 2017.

[30] M. J. Mrowinski, T. M. Gradowski, and R. A. Kosinski. Models of pedestrian evacuation based on cellular automata. *Acta Physica Polonica A*, 121(2B):95–100, 2012.

[31] K. Nishinari, A. Kirchner, A. Namazi, and A. Schadschneider. Extended floor field CA model for evacuation dynamics. *IEICE Transactions on Information and Systems*, E87-D(3):726–732, 2004.

[32] L. A. Pereira, D. Burgarelli, L. H. Duczmal, and F. R. B. Cruz. Emergency evacuation models based on cellular automata with route changes and group fields. *Physica A*, 473:97–110, 2017.

[33] A. Schadschneider. Cellular automaton approach to pedestrian dynamics - theory. In M. Schreckenberg and S. D. Sharma, editors, *Pedestrian and Evacuation Dynamics*, pages 75–86. Springer, 2002.

[34] A. Schadschneider, D. Chowdhury, and K. Nishinari. *Stochastic Transport in Complex Systems: from Molecules to Vehicles*, chapter Pedestrian dynamics, pages 407–460. Elsevier, 2011.

[35] J. A. Sethian. *Level Set Methods and Fast Marching Methods*. Cambridge University Press, 1999.

[36] X. Shi, Z. Ye, N. Shiwakoti, and O. Grembek. A state-of-the-art review on empirical data collection for external governed pedestrians complex movement. *Journal of Advanced Transportation*, 2018. Article ID 1063043, 42 pages.

[37] N. Shiwakoti, M. Sarvi, and G. Rose. Modelling pedestrian behaviour under emergency conditions - State-of-the-art and future directions. In *Proceedings of the 31st Australasian Transport Research Forum (ATRF 2008)*, pages 457–473, 2008.

[38] N. Shiwakoti, X. Shi, and Z. Ye. A review on the performance of an obstacle near an exit on pedestrian crowd evacuation. *Safety Science*, 113:54–67, 2019.

[39] Y. Tajima and T. Nagatani. Scaling behavior of crowd flow outside a hall. *Physica A: Statistical Mechanics and its Applications*, 292(1-4):545–554, 2001.

[40] Y. Tajima, K. Takimoto, and T. Nagatani. Scaling of pedestrian channel flow with a bottleneck. *Physica A: Statistical Mechanics and its Applications*, 294(1-2):257–268, 2001.

[41] P. C. Tissera, M. Printista, and M. L. Errecalde. Evacuation simulations using cellular automata. *Journal of Computer Science and Technology*, 7(1):14–20, 2007.

[42] T. Toffoli and N. Margolus. *Cellular Automata Machines: A New Environment for Modeling*. MIT Press, 1987.

[43] A. Varas, M. D. Cornejo, D. Mainemer, B. Toledo, J. Rogan, V. Muñoz, and J. A. Valdivia. Cellular automaton model for evacuation process with obstacles. *Physica A: Statistical Mechanics and its Applications*, 382(2):631–642, 2007.

[44] J. von Neumann. *Theory of Self-Reproducing Automata*. University of Illinois Press, 1966. Edited and completed by A. W. Burks.

[45] A. von Schantz and H. Ehtamo. Cellular automaton evacuation model coupled with a spatial game. In *Proceedings of the 10th International Conference on Active Media Technology (AMT 2014)*, pages 371–382, 2014.

[46] N. Wijermans, C. Conrado, M. van Steen, C. Martella, and J. Li. A landscape of crowd-management support: An integrative approach. *Safety Science*, 86:142–164, 2016.

[47] U. Wilensky. NetLogo. `http://ccl.northwestern.edu/netlogo/`. Center for Connected Learning and Computer Science, Northwestern University, Evanston, IL, 1999.

[48] S. Wolfram. *A New Kind of Science*. Wolfram Media Inc., 2002.

[49] W. Yuan and K. H. Tan. An evacuation model using cellular automata. *Physica A*, 384:549–566, 2007.