A Simulation-Based Behavioral Clustering Method for Crowd Dynamics Evacuation Analysis

Mohamed Chatra[1*], Ahmed Ghenabzia[1], Samir Akhrouf[1], Mustapha Bourahla[1]
[1]Computer Science Department, University of M'Sila, 28000, Algeria Laboratory of Informatics and Its Application of M'Sila, M'Sila, Algeria
*Corresponding author : Mohamed Chatra

Abstract:
Traffic management, urban planning, and emergency management cannot be efficiently done without crowd simulation. This paper proposes a Behavioral Clustering Method (BCM), which tackles the problem of forming crowds in clusters or subgroups based on fundamental behaviors so that congestion is minimized during effective evacuation processes. We designed BCM based on synthetic data obtained from the simulation of the evacuation of a crowd in high-risk situations. Our method regards pedestrians as intelligent agents and predicts key behavioral aspects of future crowd evacuations before they occur. We use cluster analysis on those movement and behavioral data for building as well as evacuation-friendly control strategies by clustering people into subgroups of behavioral similarity. The credibility of the model is validated through Python-based animations to detect and rectify errors. Results from simulation performance evaluations indicate that BCM is successful in modeling the evolution of crowd behavior at the time of evacuation.

Keywords: crowd evacuation behavior, agent-based model simulation, artificial intelligence, pedestrian clustering, behavior animation, effectiveness, and efficiency evaluation.

1. Introduction
Crowd dynamic during evacuation scenarios is one of the most important subject to study collective human decision depth and the movement patterns. Identifying and studying the categories of people with similar behavior provides useful information regarding how people react during under the condition of high stress, which can make a great contribution to designing and planning more effective and safer evacuation procedures [16]. Two main hypotheses account for the development of these patterns of behavior. The first suggests that individuals locally coordinate their actions with their neighbors, resulting in self-organized group movements without external control. This hypothesis highlights the role of proximity and localized interactions in shaping crowd dynamics. The second hypothesis emphasizes collectivism, where individuals exhibit similar behaviors due to shared thought processes or group-level decision-making. This behavior is influenced by various factors, including individual decision-making strategies and the density of the crowd [14].
Clarifying the underlying factors that go into the common behaviors is vital for quantifying and comparing patterns in different crowd situations. By doing so, such studies help advance knowledge of human collective behavior and have real-world implications for crowd control, city planning, and emergency response systems. From the study of these dynamics, we can gain insight into universal principles underlying collective behavior leading to the development of predictive models and effective evacuation procedures [18].
Public emergencies and disasters with significant casualties have become more common in recent years due to the behaviors of individuals. Drills for evacuation are costly and don't accurately simulate how people would act in an emergency. Considerable attention has been drawn to crowd evacuation simulation technology, which may effectively address the drawbacks of evacuation drills and offer guidance for mass evacuation in emergencies [6][7].

Public places like subway stations and bus stops can become very crowded. People usually enter and exit these places in order. However, accidents like crowding and trampling may happen in emergencies (fire or earthquake). It may lead to greater congestion, or more severe accidents, due to large gatherings of people at transport hubs. To design rational evacuation routes and strategies, many scholars have attempted to develop suitable models for real-world scenes and crowd behaviors [1][2].

Model-based simulations are often the main research methodologies used to analyze crowd evacuation rather than real-world experiments, which pose a significant challenge for examining crowd behavior during emergency evacuation [3][4]. Simulation models and clustering of individuals with similar behavior in this case are driven by environmental information, which should be qualitative, not quantitative.

To simulate crowd behaviors, Helbing and Molnar [5] as a combination of forces social, physical, and environmental that guide movement. Clustering is often applied in conjunction with this model to group individuals based on their reactions to these forces, such as people clustering around exits during evacuations.

Multiple studies have explored the use of simulation methods for crowd evacuation analysis. Yue-wen (2014) proposed a model that integrates multi-agent technology and cellular automaton, considering individual differences in behavior. Li (2019) combined the social force model with deep learning for pedestrian detection, resulting in a more realistic simulation of crowd evacuation.

This study introduces a Behavioral Clustering Method (BCM) designed to group individuals in a crowd based on shared behavioral patterns during evacuation scenarios. Synthetic data generated through simulations serves as the basis for this method, enabling the prediction of dominant behavioral trends before actual evacuation events occur. Features such as movement patterns and decision-making tendencies are extracted and analyzed to form behavior-based subgroups.

To validate the proposed method, an animation framework implemented in Python is used to visually examine the clustering results and identify potential model issues. Performance assessments reveal promising results, highlighting BCM's potential to enhance evacuation planning and congestion mitigation in critical situations.

The remainder of this paper is organized as follows. Section 2, presents a brief overview of previous work on crowd simulation. In section 3, we present our approach to solving the problem posed and contained the description of the crowd evacuation model based on artificial intelligence, and the BCM model is established. Section 4 describes how to validate the proposed model using model animation and how to produce simulation results, with reports of performance evaluation of the proposed model is presented by analysis of simulation results. Conclusions of this research and perspectives are presented in section 5.

2. Related Works

Various simulation models have been developed over the past few decades to analyze the dynamics of crowd evacuation in both regular and emergencies. We will organize the discussion by modeling methodologies: social force, cellular automaton, fuzzy logic, artificial intelligence, fluid dynamic methods, and psychological and emotional models.

Helbing and Molnar [11] have suggested a social force model that views pedestrians as a combination of forces social, physical, and environmental that guide movement. Clustering is often applied in conjunction with this model to group individuals based on their reactions to these forces, such as people clustering around exits during evacuations. The model simulates group dynamics for Reynolds, C. W. [12] based on three simple rules: alignment, separation, and cohesion. These rules naturally form clusters of agents within the simulation. Behavioral clustering methods enhance these groups by introducing individual or subgroup variations. The BCM goes beyond the rigid, rule-based clustering of social force models by dynamically grouping individuals based on data-driven behavioral similarities. This enhances adaptability and accuracy in predicting future evacuation patterns.

The rule-based model [4][5] and the social-force based model [8][21] are the most common. Xiong et al. [4] proposed a set of man-made sampling and evaluation rules based on the partial model. The author applied it to the evacuated crowd to improve the efficiency of evacuation. Liu [8] used the social power model to study the crowd evacuation in public places by terrorist attacks. The author discussed the efficiency of crowd evacuation when there were different numbers of exits or when the attackers had different locations. While cellular automaton models focus on movement at a macroscopic level, BCM emphasizes clustering based on individual behavioral data, offering a finer resolution for understanding evacuation dynamics.

Nasir et al. [23] presented a genetic fuzzy system. Fuzzy perception and fear are ingrained in human thinking, and Dell'Orco et al. [24] proposed a behavioral model for crowd evacuation based on fuzzy logic and accounting for these aspects. Furthermore, several fuzzy inference systems are made to provide escape, egress delay, and motion direction [7][8]. The BCM avoids the complexity of fuzzy inference systems by using clustering techniques that are computationally efficient and scalable. It focuses on extracting actionable insights from behavioral data rather than modeling subjective psychological factors.

The model developed by Chatra M. and Bourahla M. [34] integrates artificial intelligence with deep learning to predict future trajectories based on observed movement patterns. It combines linguistic variables with reinforcement learning to adapt individual behaviors dynamically in response to changing environmental conditions, such as the emergence of new obstacles or shifting goals. Notably, studies of pedestrian dynamical behaviors in both calm and frenzied situations have been carried out through modeling and simulation based on artificial intelligence [25][26], whereby artificial intelligence techniques can be used to predict human spirit and perception. To mimic and model the guiding behavior of pedestrians in constructed environments, Wang et al. [27] presented a study on pedestrian movement dynamics under emergency evacuation using machine learning. In the work presented in the study by Yao et al. [28], a reinforcement learning method is used to produce a data-driven model for crowd evacuation. Li et al. [29] have combined the techniques of deep learning and social to develop a simulation model for crowd evacuation.

Fluid Dynamic models [17][19] and the model based on the method of potential fields [9][10] are two of the most common. In Fluid Dynamic models, Partial Differential Equations of a man-made setting are often used to describe the dynamic properties of a crowd such as velocity and constant density. For example, Mao et al. [20] studied the process of crowd evacuation based on Fluid Dynamic models, which improved the efficiency of crowd evacuation.

3. Our Approach

The description of the crowd evacuation model is based on the description of the environment or physical space in which the crowds move, consisting of the following elements: obstacles, boundaries, exit door and, pedestrians, which are represented by physical positions (indicated by location coordinates) and behaviors performed by pedestrians. We are using Python [30], an all-purpose high-level programming language to develop a simulation model for analyzing crowd evacuation using artificial intelligence (full Python code is available on request). Behavioral Clustering Method (BCM) plays a fundamental role in the analysis and interpretation of the behavior of the crowd during evacuation simulations. Since pedestrians can be classified into different categories according to their behavioral patterns, BCM is capable of offering valuable information on vital crowd dynamics that can be utilized to enhance evacuation plans. Follows a detailed description of the algorithms implemented through the stages of the BCM and flowcharts depicting each process.

3.1. Global Architecture

To achieve our goal, we propose this three steps architecture (figure 1):

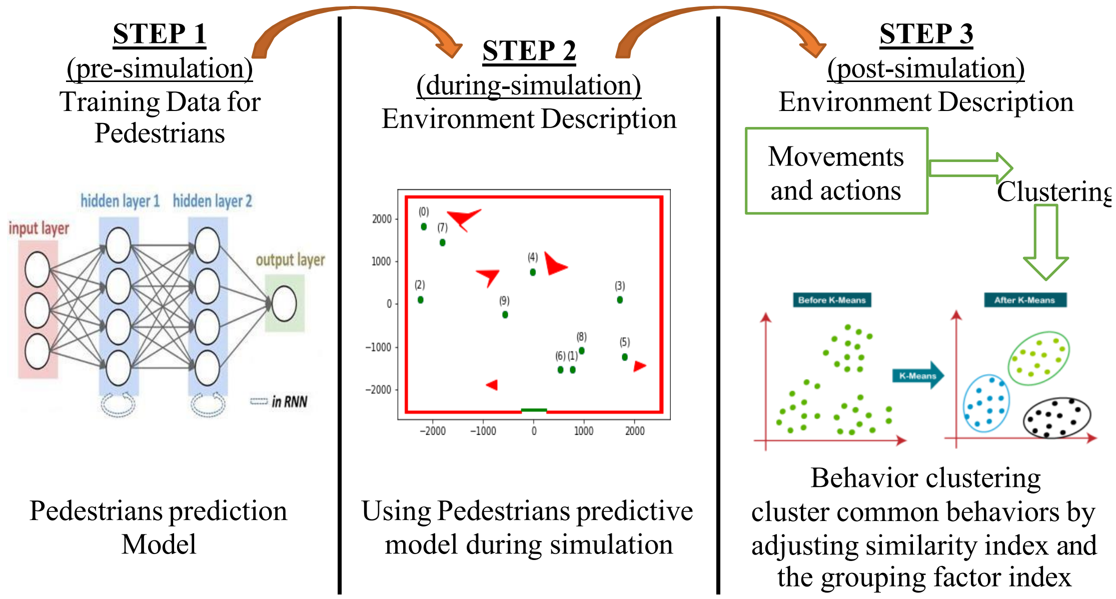


Figure 1: Flowchart of the proposed crowd evacuation simulation methodology: pedestrian prediction, simulation, and behavioral clustering

Step1: pedestrian model creation – Pre-Simulation-

Before going to the simulation, the main idea is to create a model that can achieve and simulate real pedestrian behavior to extract this knowledge, linguistic variables and label encodings are used to implement it, the big challenge in this step is: how can we get a data that represent real human behavior? We propose to use data from different simulation platforms and try to convert it to a discrete representation – linguistic- instead of continuous representation –coordinates, angles, speed –.

The results of this step are a model that can behave like a human, we will use this knowledge in the next step of crowd evacuation simulation (figure 1). the algorithm that summarizes step 1 is defined below.

Algorithm Step1

Start
  Simulate Crowd Evacuation.
  Collect Data (Position, Speed, Proximity, etc.)
  Normalize Data.
  Feature Engineering.
  Preprocessed Data Ready for Clustering.
End.

Step2: crowd evacuation simulation

This step plays the main role in understanding the crowd evacuation actions. The challenge we faced in the beginning of our research is: how we build a logic that can predict this evacuation actions, we found 2 paths:

- Gathering data from crowd evacuation experiences and trying to use it to predict actions behind the evacuation. This solution costs a lot of resources and need enormous number of experiences and different situations to converge to the reality. Also need to update this data over the time and updating this logic for covering all scenarios and environment is also a big challenge.

- The solution we select is to build a pedestrian model that can predict the individual behavior and use this model for each pedestrian in the simulation environment and track the crowd simulation over the time to get realistic scenarios.

At the end of this step; we will get a realistic crowd evacuation and this is the time to start the Behavioral clustering step to analyze crowd behaviors and actions for discovering problems, optimizing the environment, and creating good policies to avoid bad scenarios during the evacuation (figure 1). the algorithm that summarizes step 2 is defined below.

```
Algorithm Step2
Start
  Preprocessed Data.
  Apply Clustering Algorithm (K-means)
  Identify Behavioral Subgroups.
  Behavioral Clusters Identified.
End.
```

Step3: behavioral clustering step

After we got all data related with the crowd evacuation, this data is used to analyze and create sub-groups from the crowd based on their features. The goal from getting this data is to analyze the crowd evacuation and convert the data to a good knowledge about remarkable behaviors and critical actions on the crowd. This knowledge helps the decision maker to understand action in behaviors point of view instead of statistical point of view.

The Behavioral Clustering Method (BCM) is the name we choose for the logic used on this step and it will be explained more in the BCM section.
In the next section, we will start deep diving on environment and pedestrian description and describing all details used on linguistic variables, pedestrian model building and the Crowd simulation (figure 1).

3.2 Environment Description

First, we present a method of representing the environment that plays a main role in modeling and simulation. The proposed pedestrian model, we have displayed the target that is set in the center of the wall and a scale *SpaceWidth* × *SpaceLength*, will be used for simulation in a rectangular hall *'s'*. The physical environment consists of boundaries (walls), internal obstacles of various shapes and one exit. For a good illustration the following is part of the algorithm used to generate a model of the physical space *'s'*.

```
Algorithm generate physical space:
Inputs: width, length, goal
Outputs: positions, obstacles, borders, goal
  Generates the goal region based on the provided width
  Creates positions for entities, potentially distributed randomly
  Creates obstacles based on predefined vertices and the positions of entities
  Constructs the borders of the environment
```

The exit (goal) location is first created by running the following code: *goal_position = generate_goal(goal_width)*, where goal is the exit polygon (a rectangle). To represent the starting locations of pedestrians in *'s'*, a collection of *pnumber* positions (points) within the space will be generated at random by calling the function *positions = generate_positions(pnumber)*. When *obstacles = generate_obstacles(positions, nobstacles_vertexes)* is executed, a random set of *obstacles_filled* polygons, each with multiple vertexes will be produced. The number of vertexes for each obstacle is contained in the list *nobstacles_vertexes*. The method *borders =*

generate_borders(goal_position, goal_width) is called to create the borders, which are external barriers that have the shape of rectangles.

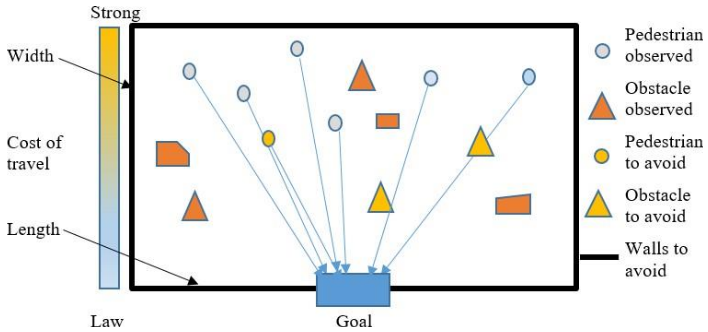


Figure 2: Environment space

the results of our algorithm give a description of the physical space and the physical entities that plays an important role in our simulation (figure 2).

3.3 Pedestrian Description

The suggested model includes a crowd of pedestrians that need to be evacuated from a hazardous scenario in addition to the actual surroundings. A pedestrian's actions in reaction to their surroundings are the primary basis for evacuation. Because the knowledge utilized to describe the intelligent model is based on human experience in identifying the values of distances, directions (angles), and speeds, this human behavior gives our model intelligence.

Several linguistic classes (Categories) are produced by their classifications, and they are employed as data to help make informed decisions in order to avoid a variety of stationary and mobile impediments. The accuracy and temporal complexity of this structurally intelligent model rise sharply as the number of classes increases. Therefore, in order to balance accuracy and computing efficiency, we use several language classes to define state variables (figure 3).

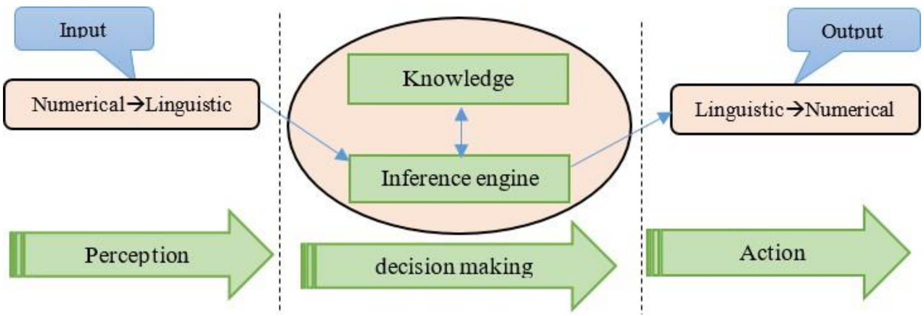


Figure 3: Reasoning of humans

"Near" and *"far"* are two categories (sets of knowledge) that indicate distances based on human experience. If the distance between his position and any other position in the physical environment is between 0 and 100 units, the pedestrian will experience it as being close; if it is within the interval [100, visual distance], he will perceive it as being far.

By using the method *perceiveDistance(self, d)*, where self is the reference to the pedestrian object, we have constructed a Multi-Layer Perceptron Classifier called *DistanceModel*, which is trained with specific data to make each pedestrian sense a distance value *d* (figure 4).

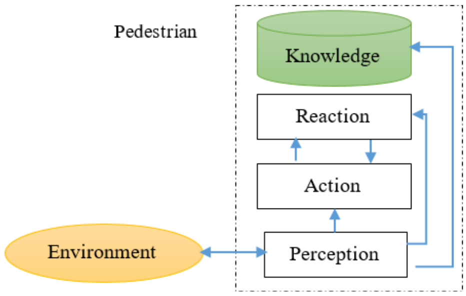


Figure 4: Decision maker of our pedestrian

Direction angle is defined as the angle between the pedestrian's position and the goal's (exit) position. The pedestrian (decision maker) can perceive this angle as belonging to one of the following classes (knowledge sets): *"Zero", "SmallPos", "LargePos", "SmallNeg"*, or *"LargeNeg",* where the commands *"Neg"* and *"Pos"* turn pedestrians left or right, respectively (figure 5).
This data is used to build and train a Multi-Layer Perceptron Classifier named *DirectionModel*, where *DirectionY* is the concatenation of category (class) codes and *DirectionX* is the concatenation of all direction value intervals.
When someone is walking in the other way, pedestrians may interpret their speed as falling into one of three categories: *"Stop", "Slow" or "Fast"*. The provided data is used to construct and train a multi-layer perceptron classifier known as *SpeedModel*, in which *SpeedX* is the concatenation of all speed value intervals and *SpeedY* is the concatenation of category (class) codes.
The *direction_crisp()* and *speed_crisp()* methods must be defined in addition to the prediction functions of the direction model *DirectionModel.predict()* and the speed model *SpeedModel.predict()* in order to update the pedestrian's state during the simulation. These routines use the random selection approach to determine the apparent (crisp) value of the relevant class direction angle and velocity, respectively.
An object belonging to the Python class *"Pedestrian"* is a pedestrian *"p"* moving in an environment *"s"*. It has three features:

- movement speed (the speed at which it goes from one position to another),
- direction information (the angle between the pedestrian's position and the goal's (exit) position), and
- location information (a point with coordinates (x_p, y_p) to trace its movements).

MovementsNbre, *MovementsDistance*, and *MovementsSpeed* it's the number of pedestrian movements (defined as a change in direction, speed, or both) as well as their total velocities are taken into account when calculating the distance that pedestrians must walk to get to their destination. The pedestrian energy needed to leave the physical region is calculated using these.
The data component of the *"Blocked"* information indicates that the pedestrian may be blocked due to physical obstacles preventing him from moving forward; if the pedestrian has already accomplished the goal, the data member of the *"ReachedGoal"* information will reflect this.
By using the method member *update(self, sp, speed)*, pedestrian objects also known as decision-makers can update their data members during the model's simulation steps, changing the pedestrians' locations, directions, and speeds. The decision maker is represented by the parameter self, the pedestrian's specific movement speed is denoted by speed, and the visual sector's position to identify the pedestrian's direction of movement toward the goal is denoted by *sp*.
Using their member techniques, pedestrians can sense information about *direction (perceiveDirection(self, a))*, *velocity (perceiveSpeed(self, s))*, and *distance (perceiveDistance(self, d))*. These senses can then trigger the relevant model prediction algorithms.
We need to create a collection of pedestrians that don't cross over in order to complete the model description. The function *generate_pedestrians(n)* is called to construct *n* pedestrians for a simulation, with the i-th pedestrian situated at position positions[i] in the space *"s"*.

A pedestrian's visual field is modeled as a circular area with a central angle of 360°, a radius R unit, and a central point representing the pedestrian's position. This visual field is divided into n sectors using sector angles $Sector\_Angle_i, i = 1, \dots, n$ $(\sum_{i=1,\dots,n} Sector\_Angle_i = Central\_Angle)$, such that the sum of all sector angles satisfies. These sectors are comparable in terms of their functional representation but are not necessarily congruent, as shown in Figure 5.

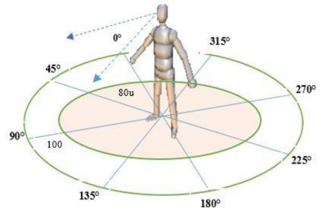


Figure 5: Pedestrian's visual field

The number of sectors in the pedestrian's visual field is determined by balancing model complexity with prediction accuracy. This division allows for the integration of objective information with pre-collected environmental data from each sector of the visual field. Using this information, pedestrian motion states are dynamically updated, supported by a predefined spatial representation strategy.

3.4 Behavior Description

In order to survive, pedestrians steer clear of hazardous circumstances and surroundings. In sections 3.1 and 3.2, we introduced a structural model of pedestrian behavior in a crowd that is based on artificial intelligence. To make the suggested model more complete and clearer, the following presumptions are made:

- Although every pedestrian is aware of the nuances and subtleties of their goals as well as the local knowledge inside their field of vision, they are not aware of the global knowledge regarding their surroundings.
- During an ignored rest period, the pedestrian may choose to alternate between any two preset states.

The artificial intelligence-based model is built on a framework that integrates obstacle-avoidance and goal-seeking behaviors to predict pedestrian motion. The input data primarily consists of information about pedestrians, obstacles, and goals. For example :

- A pedestrian’s proximity to a barrier determines their avoidance behavior.
- The distance and speed of individuals moving in the opposite direction influence their goal-seeking decisions and pathfinding behavior.

Input data of model, namely the turning angles, move directions and move velocities, lead to final states of motions regarding pedestrians. The above outputs integrate local obstacle-avoidance and goal-driven behavior for simulating the crowd evacuation behavior. Pedestrians need to reach their destinations with a certain velocity, while trying to avoid collision with other pedestrians, obstacles and walls being in their vision. Figure 8 illustrates this behavior.

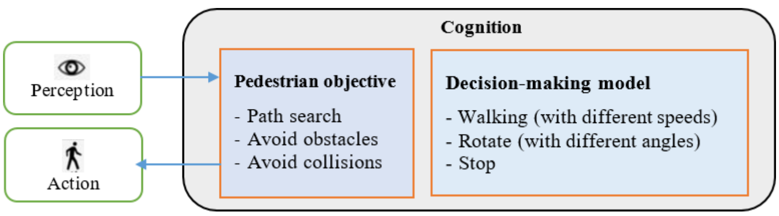


Figure 6: AI-based pedestrian movement model combining obstacle-avoidance and goal-seeking behaviors

Pedestrian behavior is strongly influenced by the location of the destination, the distribution of similar pedestrians, and the presence of obstacles. To ensure effective navigation, the goal's location must be known in advance by all pedestrians. The visible field for each pedestrian is modeled as a circle with a radius of R units (default value: 50 units), defining the area within which they perceive and respond to their surroundings.

The pedestrian's viewing field's center angle the *central_angle* is separated into sector angles that are identical and have the same value. Then the sectors that comprise the visual field are *n=Central_Angle/Sector_Angle*. each sector specifies a path from the pedestrian location to the sector position *sp*.

In general, pedestrians base their choices on the following factors. As the decision-maker, the pedestrian first examines his visual field, sector by sector, and chooses which sector to proceed through in order to reach the exit (target). He then moves in the direction of the chosen sector position (*sp*) to accomplish the goal while maintaining the proper speed and direction of travel to prevent running into oncoming traffic and frontal impediments.

The algorithms inputs are the nearest pedestrian-obstacle distances in each sector, while the outputs are the turning angle and movement speed. Goal-seeking and obstacle-avoiding behavior are combined to establish the decision maker's ultimate turning angle and movement speed. The decision maker will avoid the obstacle in front of the goal before pursuing it. Therefore, avoiding obstacles is more crucial than pursuing your goals. As a result, the pedestrians are able to accomplish their objective while avoiding potential roadblocks and other pedestrians.

To control pedestrian movement, it is necessary to comprehend the following information. The decision maker (pedestrian) "*p*", position (x_p, y_p), and goal "*g*" location is at goal distance "*dg*" from the pedestrian position "*p*", goal angle "γ_g" (the angle between the pedestrian position "*p*" and goal position "*g*", also called the direction angle of the pedestrian), and position (x_g, y_g) that represent the location that pedestrians wish to reach in "*s*".

The method behavior(self), defined below, will invoke the *get_distance_perceptions(self)* method (self is the pedestrian's reference) to determine the pedestrian's perception of the minimum distances between his location and surrounding obstacles, borders, and other pedestrians approaching from opposite directions in different sectors of his visual field during model simulation.

A list called distance_perceptions contains these senses of distance. Its longest point is equivalent to *Central_Angle/Sector_Angle*, which is *Sector_Angle*, the sector's angle, which by default is 45 degrees (figure 5).

The information about the distances that the subject perceives himself in each sector of his visual field will be present in the list *distance_perceptions* after *get_distance_perceptions(self)* has been executed.

The initial part of all sector information, which is triples, consists of the sector's position and the minimum perceived distance between the pedestrian's position and any barriers, boundaries, and other pedestrians visible inside the sector's window of the visual field.

Use the third piece of information in the triple check to determine whether the blockage is another obstacle or a person approaching from the opposite direction. Based on the information obtained about him, the pedestrian who is now making decisions will choose to change positions at the proper speed and angle of direction.

The method *behavior (self)*, which alters the pedestrian's state and compels him to move from one location to another until he reaches the exit (the objective), is defined as follows. The pedestrian may go into a state of blocking, or halt traveling, if he encounters physical obstacles in every direction.

To receive decision-making information for the pedestrian self and determine each pedestrian's movement, we call the previously mentioned function *get_distance_perceptions(self)*. The goal-seeking and obstacle-avoiding behaviors are combined in the technique behavior (self) to acquire new locations.

These new roles are possible candidates for the pedestrian (decision-maker) transfer. A formula that specifies the preference between the two weighting parameters, alpha and beta, and offers a moderate weighting between the distance and direction angle from the target is used to determine the best new position.

The pedestrian self, or decision maker, first assesses if he is close to someone who has already achieved the objective. He enters an exit state if that's the case.

Otherwise, he will utilize the procedure *goal_seeking_behavior(self, distance_perceptions)* to check if all obstacles, boundaries, and pedestrians are far away from him in order to decide whether to pursue the path for chasing the goal.

One form of global conduct known as "goal-seeking behavior" is the propensity of the decision-maker to consistently move closer to his objective, independent of the circumstances surrounding him. It is defined by the goal angle γg, which can be in the *LargeNeg, SmallNeg, Zero, SmallPos,* and *LargePos* classes, and the goal distance *dg*, which can be in the *Near* or *Far* classes.

By fostering a worldwide goal-seeking behavior, pedestrians are motivated to approach the goal. The decision maker reduces speed and quickly turns in the direction of the goal without missing it as the pedestrian approaches the exit without facing it. Conversely, when facing the target, he moves swiftly and freely in its direction.

Using this approach, the decision-maker first looks at the sectors, which stand for remote interior barriers, blocked pedestrians, and exterior obstacles (boundaries). He adds the choice to the list of choices related to "gsb" (goal-seeking behavior) if any of these circumstances hold true. This allows him to go swiftly in the direction that the expression *abs(self.get_angle(self.position,sp)-self.get_angle (self.position, goal_position))* indicates.

Before adding the same movement direction angle and halting, the decision-maker checks to see if there is any kind of obstacle in the way. If not, he adds as fast as the movement speed.

The method *obstacle_avoiding_behavior(self, distance_perceptions)* defines frontal obstacles, which must be avoided by invoking the obstacle-avoiding behavior if the decision maker cannot find a sector position (the list *new_positions* is empty) to seek the goal, indicating that he is surrounded by obstacles from all directions defined by the sectors of his visual field.

He employs this technique to ascertain whether each barrier is of a particular type because they are all near pedestrians. To the list of *"pab"* (pedestrian (obstacle)-avoiding behavior), the code's decision maker appends the option to follow the direction, which is established by the following statement. the absolute value of the angle between the sector point from which the decision maker has observed someone approaching him and the goal, as well as the difference between the angle between the choice maker and the goal when traveling slowly.

If the type is traveling in the other direction, the resolution to add continues in the same direction as before but ceases until later simulation stages.

The decision-maker is in a blocking condition (see the behavior approach) and is unable to evacuate (i.e., have access to the exit) if there are no kind-related obstacles, only internal or external barriers remain.

Obstacle-avoiding behavior is the propensity of a pedestrian to shift directions gently and smoothly as opposed to abruptly. Therefore, if the pedestrian-obstacle distance is the same in each sector, the code is made so that pedestrians will typically move in the same direction.

To help with the interpretation of the simulation results, the description of this crowd evacuation model offers a way to change parameters like the number of pedestrians, the size of the space, the number of visual field sectors, the weighting parameters, the number of obstacles, the design of each obstacle, and so on. The complexity of the model may rise with certain parameter settings, requiring more time and space to simulate.

3.5 Crowd simulation approach BCM general description

The primary goal of the Behavioral Clustering Method (BCM) is to simulate and predict crowd behaviors during evacuation scenarios by identifying and clustering individuals into behaviorally similar subgroups. This clustering approach allows the model to account for the diverse decision-making processes and movement patterns exhibited by different individuals in a crowd.

Unlike traditional methods that treat the crowd as a homogeneous entity, BCM integrates behavioral diversity, capturing nuances such as leadership tendencies, social attachment, and risk aversion. By doing so, BCM provides more accurate and actionable insights, enabling planners and decision-makers to design tailored evacuation strategies that enhance both efficiency and safety.

Crowd heterogeneity plays a crucial role in evacuation scenarios, as individuals respond to emergencies in diverse ways influenced by psychological, physical, and social factors. For instance, some individuals may panic, moving erratically and creating bottlenecks that impede overall crowd movement. In contrast, others may remain calm, acting as natural leaders who guide groups toward exits.

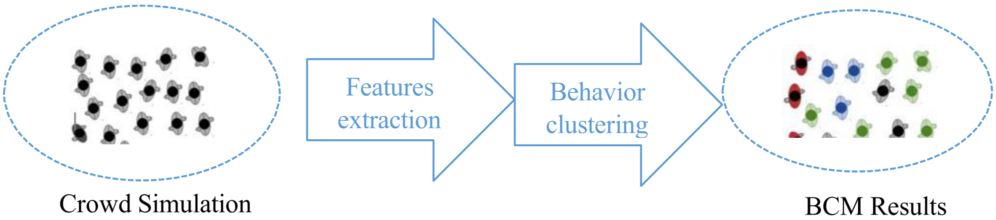


Figure 7: Behavioral clustering method (BCM) workflow for crowd simulation in evacuation scenarios

This diversity in behavior introduces complex, emergent dynamics that cannot be adequately captured by traditional evacuation models treating the crowd as a uniform entity. Behavioral clustering addresses this challenge by grouping individuals with similar traits and responses, enabling more specialized and realistic modeling of crowd interactions. By considering heterogeneity, the method enhances the predictive accuracy of simulations and supports the development of strategies to mitigate risks and improve evacuation efficiency.

The Behavioral Clustering Method (BCM) follows a systematic workflow designed to simulate and analyze crowd behaviors during evacuation scenarios with high accuracy. This workflow consists of the following key steps:

1. Data is gathered from crowd simulations, encompassing attributes such as movement trajectories, reaction times, and proximity to others. Preprocessing techniques like noise reduction and normalization ensure the data is clean and suitable for analysis.
2. Critical features that define an agent's behavior are extracted from the prepared data using advanced feature extraction techniques. These features include movement speed, decision-making tendencies, social attachment levels, and influence on nearby agents. By employing methods such as statistical analysis, time-series processing, and trajectory analysis, the raw data is transformed into meaningful attributes that accurately represent individual behaviors. These extracted features form the foundation for clustering individuals into behaviorally similar subgroups, enabling a detailed and realistic understanding of crowd dynamics.
3. The BCM employs artificial intelligence techniques, such as the K-means clustering algorithm, to group agents based on the similarity of their extracted behavioral traits. K-means partitions the dataset into distinct clusters by minimizing the within-cluster variance, ensuring that agents in the same cluster exhibit closely related behaviors. This approach leverages the efficiency and scalability of K-means to handle large datasets, making it suitable for complex evacuation scenarios. By accurately segmenting the crowd into meaningful groups, the method enables a detailed analysis of emergent patterns and interactions during simulations.

4. Once clusters are formed, the simulation assigns each group specific decision-making rules and movement models. These clusters interact within the simulation environment, allowing researchers to observe emergent crowd patterns, bottlenecks, and evacuation dynamics.

in the figure 9 we have summarized the Behavioral Clustering Method (BCM) workflow for crowd simulation in evacuation scenarios. The flow consists of key stages listed above.

3.5.1 Input data and features

The (BCM) relies on rich, high-dimensional input data to capture the nuances of individual behaviors during crowd evacuation scenarios. This data is collected over discrete time intervals within simulation environments. Each time step records critical pedestrian movements and decision-making actions, including their chosen paths, interactions, and movement patterns. These observations form the foundation for statistical analysis and feature extraction, which are essential for unsupervised learning.

The data generation and collection process is at the core of the Behavioral Clustering Method (BCM), ensuring that the input data accurately represents the complex and diverse behaviors exhibited by individuals during evacuation scenarios. This section delves into the methodology for capturing, processing, and refining the data, focusing on its relevance to understanding crowd dynamics.

During the simulation, pedestrian behavior is meticulously observed and recorded at regular intervals, resulting in a dynamic and comprehensive dataset. This dataset captures both individual and collective movement patterns within the simulated environment, which may include corridors, open spaces, and bottleneck areas. The primary attributes monitored include trajectories, action patterns, and path information.

Trajectories describe the sequential positions of pedestrians as they progress toward their goals. These include spatial coordinates (xt,yt)(x_t, y_t)(xt,yt), which mark each pedestrian's position at a given time step ttt, and temporal changes, offering insights into trends in movement, deviations, and congestion dynamics. Additionally, trajectories reveal interactions with static or dynamic obstacles, shedding light on navigational strategies.

Action patterns provide a detailed account of decision-making processes. These patterns involve directional changes, reflecting how pedestrians adjust their paths; speed transitions, identifying variations in speed such as acceleration or deceleration; and stopping events, which may signify hesitation or congestion. Collectively, these attributes highlight the complexity of pedestrian behavior.

Path information quantifies the total distance traveled by each individual, serving as a measure of navigational efficiency and movement complexity. Straight paths often indicate direct movement toward a goal, while deviated paths suggest detours or hesitation influenced by environmental factors or social dynamics.

Feature extraction plays a crucial role in the Behavioral Clustering Method (BCM), transforming raw simulation data into structured representations of pedestrian behavior. By encapsulating decision-making patterns, movement characteristics, and interactions, these features enable detailed clustering and analysis. Through systematic statistical and trajectory analysis, the BCM effectively captures the diversity and intricacy of crowd behavior, supporting robust clustering and providing actionable insights.

3.5.2 Feature extraction serves several critical objectives

Feature extraction is pivotal in the Behavioral Clustering Method (BCM) for reducing data complexity and enabling actionable insights into pedestrian behavior. By summarizing raw data into meaningful attributes, this process achieves dimensionality reduction, simplifying datasets while preserving critical behavioral information. Extracted features serve as a bridge between abstract pedestrian actions and quantifiable metrics, enabling detailed comparisons and analysis. Furthermore, the structured features facilitate clustering readiness, providing a robust foundation for unsupervised learning algorithms to effectively group individuals into behaviorally similar subgroups.

Table 1: Description of the features used for BMC

Feature Name	Description	Behavioral Insight
x	The pedestrian's initial x-coordinate position.	Indicates the starting horizontal position, essential for understanding spatial context and trajectory.
y	The pedestrian's initial y-coordinate position.	Indicates the starting vertical position, complementing the spatial representation.
Initial Angle	The angle formed between the pedestrian's initial orientation and the direct line to the goal.	Reflects the initial heading alignment and potential for detours or straight paths.
Initial Distance	The Euclidean distance between the pedestrian's starting position and the goal at the beginning.	Represents the initial effort required to reach the goal, influencing urgency and path choice.
no_zero	The frequency with which the pedestrian chooses a zero-degree angle, indicating direct movement.	Highlights preference for straightforward paths, signifying decisiveness and low environmental influence.
no_small_pos	The frequency of selecting small positive angles, reflecting slight rightward adjustments.	Suggests subtle course corrections to the right, often in response to environmental or social stimuli.
no_small_neg	The frequency of selecting small negative angles, reflecting slight leftward adjustments.	Indicates similar slight adjustments but to the left.
no_large_pos	The frequency of selecting large positive angles, indicating significant rightward turns.	Captures substantial changes in direction to the right, often signaling avoidance or reorientation.
no_large_neg	The frequency of selecting large negative angles, indicating significant leftward turns.	Mirrors large directional shifts to the left, possibly reflecting obstacles or group influences.
speed_zero	The number of times the pedestrian stops (speed reduced to zero).	Represents hesitation, rest, or congestion-related behavior.
speed_slow	The number of times the pedestrian transitions to a slow pace (walking slowly).	Indicates cautious movement, often in response to dense areas or social interactions.
speed_fast	The number of times the pedestrian accelerates to a fast pace (running).	Reflects urgency, goal-oriented movement, or response to emergencies.

The extracted features are categorized based on their insights into pedestrian dynamics. Position-based features include the pedestrian's initial x and y coordinates, the initial angle formed between the pedestrian's orientation and the goal, and the initial distance to the goal. These attributes define spatial context and starting conditions, influencing movement decisions. Angular preferences measure the frequency of specific directional choices, such as zero-degree (direct movement), small adjustments, or large turns to either side, which reflect environmental influences or obstacles. Speed dynamics capture the pedestrian's adaptability, urgency, and decision-making under changing conditions through metrics like stops, slow paces, and rapid movements.

Before extracting features, the raw simulation data undergoes rigorous preprocessing to ensure accuracy and reliability. Noise reduction techniques, including moving average filters and spline interpolation, address sudden position changes and trajectory artifacts while maintaining realistic movement curves. Normalization brings all features to a uniform scale, preventing numerical dominance by attributes with larger ranges. For instance, the Min-Max normalization rescales data values to the range [0,1] using the formula (01):

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \qquad (01)$$

Outlier detection and removal are critical for maintaining dataset integrity. Statistical thresholding identifies anomalies based on deviations from the mean, while density-based clustering algorithms like DBSCAN isolate abnormal behaviors, such as excessive directional changes or uncharacteristically high speeds.

3.5.3 Model and Techniques

After preprocessing and feature extraction, the Behavioral Clustering Method (BCM) proceeds to identify and segment individuals into behaviorally similar subgroups. This phase involves determining the optimal number of clusters and applying clustering techniques to group pedestrians based on their extracted behavioral features. Below, we outline the methods used and the workflow for achieving accurate clustering.

3.5.3.1 Optimal number of clusters

Determining the appropriate number of clusters is a critical step in the clustering process, as it directly affects the granularity and interpretability of the results. We relied on technique of Silhouette Analysis: Silhouette Analysis measures the cohesion and separation of clusters by calculating the mean silhouette coefficient for each cluster. This coefficient ranges from -1 (poor clustering) to +1 (highly distinct clusters). Using this approach, we found an optimal cluster number where the silhouette score peaked, indicating well-separated and meaningful clusters. This technique proved suitable for our data as it accounts for the diversity and subtle differences in pedestrian behaviors.

3.5.4 Clustering technique: K-means clustering

After identifying the optimal number of clusters using silhouette analysis, we applied K-means clustering to group pedestrians into behaviorally similar subgroups. K-means is a centroid-based algorithm that iteratively partitions the dataset by minimizing the within-cluster variance. Its key advantages include:

- **Efficiency**: K-means is computationally efficient and scalable, making it suitable for large datasets generated during crowd simulations.
- **Simplicity**: The algorithm is straightforward to implement and interpret, providing clear group assignments for each pedestrian.

However, K-means assumes spherical clusters with relatively uniform sizes. To address potential limitations, we ensured careful feature scaling and preprocessing during earlier stages to align with these assumptions.

3.5.5 Workflow for behavioral clustering

The complete workflow for clustering in BCM involves the following steps:

1. **Data Preparation and feature extraction**: Preprocessed simulation data is transformed into structured features that capture individual behaviors, such as movement tendencies, decision-making patterns, and interaction metrics.
2. **Optimal cluster selection**:
   - Apply the Elbow Method to analyze within-cluster variance across different cluster counts.
   - Use Silhouette Analysis to evaluate cluster cohesion and separation, selecting the cluster number with the highest silhouette score.
3. **Clustering with K-means**:
   - Initialize K-means with the selected number of clusters.
   - Randomly assign initial cluster centroids and iteratively update them by minimizing the within-cluster variance.
   - Assign each pedestrian to the nearest cluster based on behavioral feature similarity.

This structured approach ensures that the clustering process effectively segments the crowd into behaviorally relevant subgroups, forming the foundation for realistic and actionable crowd behavior simulations. The integration of Silhouette Analysis and K-means clustering ensures robustness and scalability, accommodating the complexity of pedestrian behaviors in dynamic evacuation scenarios.

4. Validation and Simulation

The approach we have outlined provides a structured workflow for simulating and validating pedestrian movement during a crowd evacuation. Here is a breakdown of the implementation of the main functions and their roles. Firstly, the function *generate_initial_state()*, this is for initializes random positions for pedestrians in the simulation space. Means assigning an initial status of zero (moving) to each pedestrian figure 10(a). secondly, the function *move_pedestrians()*, this is for updates pedestrian positions incrementally toward the goal, while avoiding collisions by considering obstacles in the path. Checks for overlaps between pedestrians' new positions and obstacles and prevents invalid movements (figure 8). Thirdly, the function *generate_animation_data()*, Logs positions and statuses of pedestrians at each simulation step. And so it provides us input data for visualization (figure 8). At the end, the function *animate()*, converts recorded simulation data into an animation. We relied on some colors to represents pedestrians, obstacles, and the goal with visual markers: (green points: Pedestrians, red points: Obstacles, dashed green line: Goal) (figure 8).

(a) Initial state (b) Final state

Figure 8: Animation with Graphical Information Format file

The output for our simulation produces a GIF file named *"pedestrian_simulation.gif"* that visualizes pedestrian movement and final states. The animation confirms logical pedestrian behavior, such as avoiding obstacles and stopping upon reaching the goal.

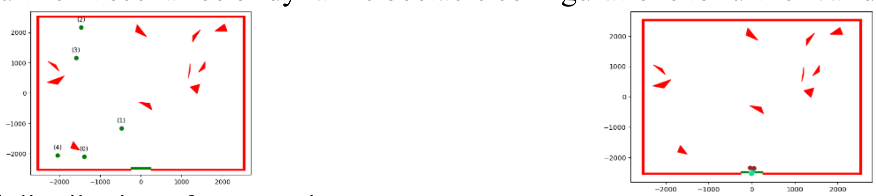
(a) Initial animation (b) Final animation

Figure 9: Simulation of the crowd evacuation model

The performance evaluation of the proposed crowd evacuation model described in Section 3 focuses on assessing scalability and the trade-offs between scalability and pedestrian randomization in the simulation environment. This evaluation is based on metrics such as effectiveness, efficiency, and total evacuation time to analyze the overall performance of the behavioral clustering model (BCM).

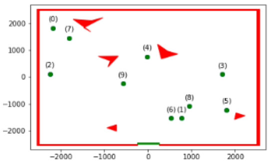
To evaluate the model, experiments were conducted in a simulated square hall with varying crowd sizes and obstacle configurations. The hall included 10 internal obstacles of varying shapes and a single exit (goal) with a fixed width. The pedestrian distribution and obstacle locations were randomized for each experiment.

The BCM model demonstrates high scalability and effectiveness, maintaining smooth evacuation performance even with increasing crowd sizes and densities. The results align with prior research, affirming the model's robustness and reliability for crowd evacuation analysis. Future experiments may explore multi-exit scenarios or dynamic obstacle configurations to further validate the model.

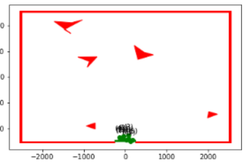


(a1) Initial distribution of an experiment with 5 pedestrians

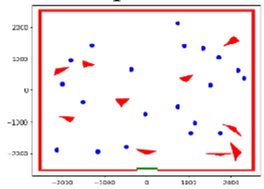
(b1) Final state of an experiment with 5 pedestrians (t=0.3s)



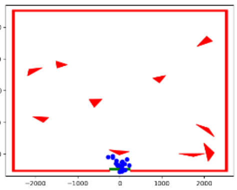
(a2) Initial distribution of an experiment with 10 pedestrians



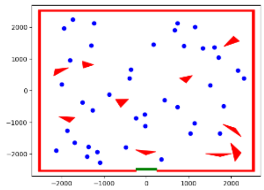
(b2) Final state of an experiment with 10 pedestrians (t=0.83s)



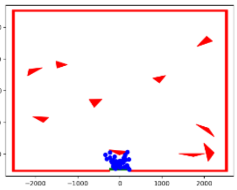
(a3) Initial distribution of an experiment with 20 pedestrians



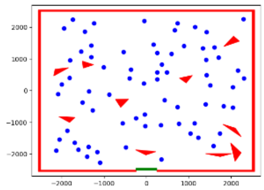
(b3) Final state of an experiment with 20 pedestrians (t=2.21s)



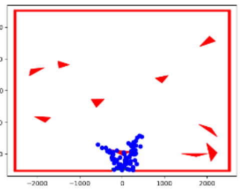
(a4) Initial distribution of an experiment with 40 pedestrians



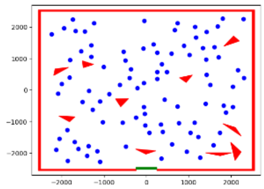
(b4) Final state of an experiment with 40 pedestrians (t=4.91s)



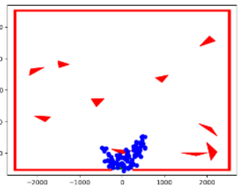
(a5) Initial distribution of an experiment with 70 pedestrians



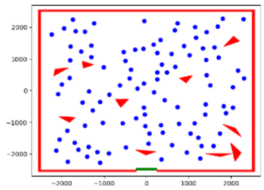
(b5) Final state of an experiment with 70 pedestrians (t=10.72s)



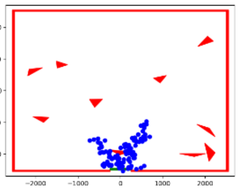
(a6) Initial distribution of an experiment with 90 pedestrians



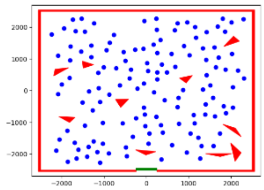
(b6) Final state of an experiment with 90 pedestrians (t=16.45s)



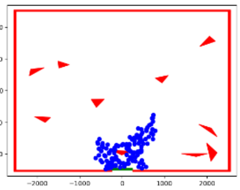
(a7) Initial distribution of an experiment with 100 pedestrians



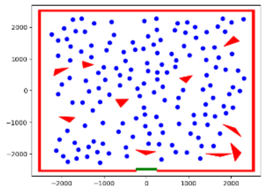
(b7) Final state of an experiment with 100 pedestrians (t=21.25s)



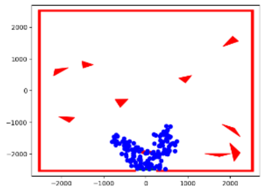
(a8) Initial distribution of an experiment with 130 pedestrians



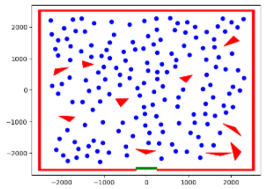
(b8) Final state of an experiment with 130 pedestrians (t=30.00s)



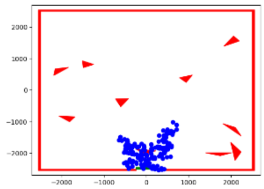
(a9) Initial distribution of an experiment with 150 pedestrians



(b9) Final state of an experiment with 150 pedestrians (t=39.95s)



(a10) Initial distribution of an experiment with 160 pedestrians



(b10) Final state of an experiment with 160 pedestrians (t=40.52s)

Figure 10: Crowd simulation with different situations number of obstacles and number of pedestrians

The table 1 evaluates pedestrian evacuation performance across 10 experiments illustrating in Figure 12 with varying densities and numbers of pedestrians.

Table 2: Experimental results

Experiment	N. of Pedestrians	Density	N. of Movements	N. of Reached Goal	Number of Clusters	Silhouette Score	WCSS	T(seconds)	Effectiveness	Efficiency
e = 1	5	3.00%	60	5	2	0.475	1.294	0.3	100.00%	82.78%
e = 2	10	6.00%	156	10	4	0.565	1.565	0.83	100.00%	87.84%
e = 3	20	13.00%	366	20	4	0.342	5.263	2.21	100.00%	78.35%
e = 4	40	25.00%	629	39	5	0.410	9.482	4.91	97.50%	77.92%
e = 5	70	44.00%	1062	69	5	0.459	10.324	10.72	98.57%	78.61%
e = 6	90	56.99%	1435	89	4	0.427	14.504	16.45	98.89%	74.39%
e = 7	100	63.00%	1739	98	4	0.402	17.664	21.25	98.00%	68.82%
e = 8	130	82.00%	2079	127	4	0.420	19.980	30	97.69%	71.94%
e = 9	150	94.00%	2504	142	5	0.408	18.051	39.95	94.67%	67.95%
e = 10	160	100.00%	2400	149	4	0.400	28.532	40.52	93.71%	70.97%

The analysis confirms the achievement of the main goal with the arrival of almost all pedestrians reaching the goal (exit), validating the model's effectiveness across varying densities. This behavior aligns with findings from prior studies [32][33].

Evacuation time represents the total duration taken for all pedestrians to reach the exit in each experimental scenario. Table 2 outlines the results for different densities and pedestrian counts. Below are key findings:

- Low-density scenarios (experiments 1–3) show shorter evacuation times (0.3s to 2.21s) due to reduced congestion and fewer movement constraints.
- Medium-density scenarios (experiments 4–7) show moderate evacuation times (4.91s to 21.25s) due to increased congestion.
- High-density scenarios (experiments 8–10) result in longer evacuation times (30s to 40.52s), highlighting the challenges of high-density environments.

We note that the effectiveness values remain consistently high, near or above 94%, indicating a robust model with slight decline in effectiveness as density increases, this is to due congestion and obstacles.

Regarding efficiency, it peaked at 87.84% for Experiment 2, then gradually decreases as pedestrian numbers and density rise. Higher densities lead to more movement constraints, reducing efficiency.

Simulation time scales with density and pedestrian count, peaking at 40.52 seconds in Experiment 10 (higher densities require more computation and prolong evacuation).

The number of clusters stabilizes at 4-5 for most experiments. Silhouette Scores range from 0.34 to 0.56, indicating moderate cluster separation quality. As for the WCSS (Within-Cluster Sum of Squares) increases with the number of pedestrians and density, suggesting more variation within clusters due to higher complexity (figure 11).

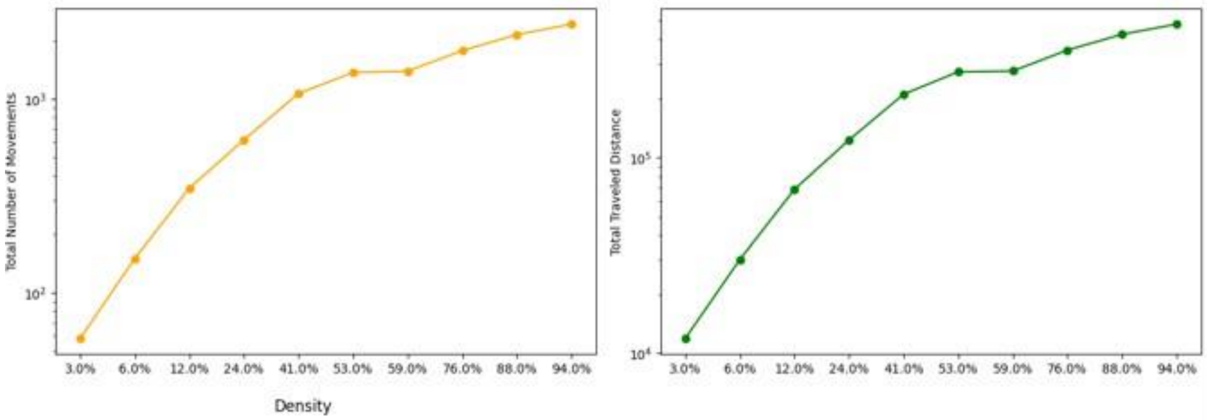


Figure 11: Density relationships of average speed (left), WCSS (right)

Our model performs well at low to medium densities but faces challenges in high-density scenarios. We observe in experiments (e = 1-3) a low density, high effectiveness (100%), minimal time required, and high efficiency due to less congestion and fewer obstacles. in the experiments (e = 4-7) a medium density, slight dips in effectiveness (97.5%-98.89%), and efficiency starts declining as density increases (figure 12). But when the density is high (e = 8-10) we observe noticeable reductions in both effectiveness (93.71%-97.69%) and efficiency (67.95%-71.94%), and movement complexity and cluster overlaps become more pronounced. As a result of our analysis, we conclude that our model performs well at low to medium densities but faces challenges in high-density scenarios. These results indicate that the model performs well across all experiments, efficiently evacuating crowds with high effectiveness, also our model work to group individuals into clusters based on similar behaviors or movement patterns. This clustering provides insights into crowd dynamics, which can inform crowd management strategies, evacuation planning, or analysis of pedestrian behavior in various settings (figure 11).

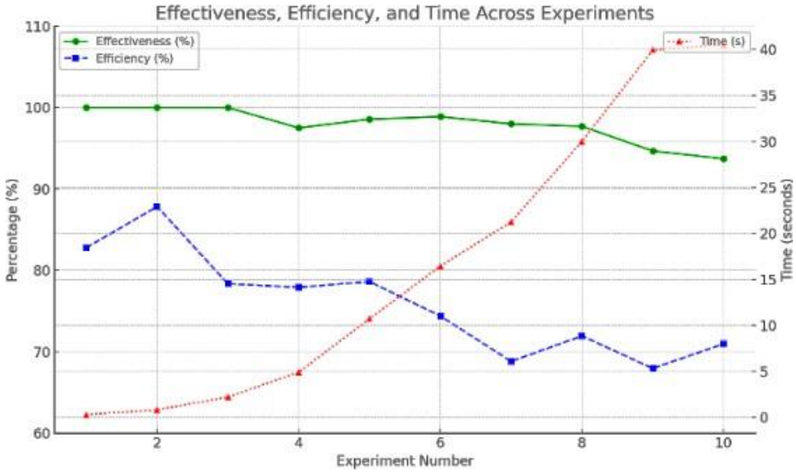


Figure 12: Performance evaluation

They are also unable to capture behavioral nuances like leadership, social attachment, or panic responses. longer evacuation times are consumed in high-density scenarios due to oversimplified movement models.

Accounts for heterogeneity by clustering agents with similar behaviors. Contrary to what has been stated, simulations BCM includes realistic dynamics such as leadership and hesitation, improving predictive accuracy. BCM reduces bottlenecks by grouping agents based on movement and decision-making tendencies (table 3).

Table 3 : Quantitative comparison

Metric	Traditional Homogeneous Models	BCM	Improvement (%)
Evacuation Time (High-Density)	50s–60s	40.52s	20–30% faster
Effectiveness (%)	~85–90%	93.71–98.89%	5–10% higher
Congestion Levels	Severe in all density scenarios	Moderate to low in low/medium	Reduced significantly
Silhouette Score	N/A (not behavior-based)	0.34–0.56	Provides clustering insights
Efficiency (%)	~60–65%	67.95–87.84%	10–20% higher

The BCM consistently performs better than traditional methods across all metrics, particularly in high-density scenarios. Behavioral clustering provides granular insights into pedestrian dynamics, enabling targeted strategies for bottleneck reduction. While BCM performs well, further optimization could address high-density challenges (e.g., multi-exit scenarios or real-time dynamic clustering).

The table provides a comprehensive evaluation of the Behavioral Clustering Method (BCM) in various crowd evacuation scenarios, focusing on key performance metrics. Evacuation time quantifies the duration required for pedestrians to exit the simulation environment, with shorter times indicating more efficient evacuation processes. Effectiveness represents the percentage of pedestrians who successfully reach the goal, consistently high across experiments, showing the model's reliability even in challenging scenarios. Congestion levels are indirectly reflected in the efficiency and movement patterns, highlighting how crowd density and obstacles impact flow dynamics. The Silhouette Score measures the quality of behavioral clustering, with moderate values indicating reasonable differentiation between behavioral groups. Efficiency combines multiple aspects, such as path optimization and speed, revealing the system's ability to maintain smooth movement while minimizing delays. As density increases, both effectiveness and efficiency slightly decline due to congestion and overlapping behaviors, demonstrating the model's strengths and limitations under varying conditions. This detailed analysis validates BCM's robust performance while identifying areas for improvement in high-density scenarios.

5. Conclusion and Perspectives

We presented in this paper a novel method, called Behavioral Clustering Method (BCM), to improve crowd evacuation strategies by analyzing pedestrian behaviors during emergencies. It uses synthetic data generated from simulations to identify groups of individuals based on shared behavioral patterns. The method provides valuable insights into crowd dynamics. The validation of our model through Python-based animations demonstrates its capacity to accurately predict evacuation behaviors, thereby enhancing the effectiveness of evacuation planning.

The results obtained indicate that understanding the nuances of pedestrian movement and decision-making can significantly reduce congestion and improve safety during emergency evacuations. Further, this paper emphasizes the importance of integrating behavioral analysis into crowd management systems, as it provides a framework for developing more adaptive and responsive evacuation protocols.

For future work, more research could look into ways of gathering real-time data so BCM's predictions get even better. Also, adjusting the model to think about different environmental conditions and changing sizes of crowds will make it useful in many cases. Future studies might also check into psychological influences on group behavior for a deeper comprehension of what happens during evacuations.

Finally, we can say that BCM represents a significant step forward in crowd evacuation analysis, with the potential to inform urban planning and emergency response strategies effectively. By continuing to refine this method and exploring new approaches for research, we can contribute to safer and more efficient public spaces in times of crisis.

References

[1] *Shin, Y., Moon, I., "Robust building evacuation planning in a dynamic network flow model under collapsible nodes and arcs. Socio-Economic Planning Sciences", 86: 101455. (2023). https://doi.org/10.1016/j.seps.2022.101455.*

[2] *Adrian, J., Seyfried, A., Sieben, A., "Crowds in front of bottlenecks at entrances from the perspective of physics and social psychology". Journal of the Royal Society Interface, 17(165): 20190871. (2020). https://doi.org/10.1098/rsif.2019.0871.*

[3] *Bakar, N.A.A., Majid, M.A., Ismail, K.A., "An overview of crowd evacuation simulation. Advanced Science Letters", 23(11): 11428-11431. (2017). https://doi.org/10.1166/asl.2017.10298.*

[4] *Zheng, L., Peng, X., Wang, L., Sun, D., "Simulation of pedestrian evacuation considering emergency spread and pedestrian panic. Physica A: Statistical Mechanics and its Applications", 522: 167-181. (2019). https://doi.org/10.1016/j.physa.2019.01.128.*

[5] *Zhang, J., Zhu, J., Dang, P., Wu, J., Zhou, Y., Li, W., Fu, L., Guo, Y., You, J., "An improved social force model (ISFM)-based crowd evacuation simulation method in virtual reality with a subway fire as a case study". International Journal of Digital Earth, 16(1): 1186-1204. (2023). https://doi.org/10.1080/17538947.2023.2197261.*

[6] *Zhang, D., Li, W., Gong, J., Zhang, G., Liu, J., Huang, L., Liu, H., Ma, H., "Deep reinforcement learning and 3D physical environments applied to crowd evacuation in congested scenarios". International Journal of Digital Earth, 16(1): 691-714. (2023). https://doi.org/10.1080/17538947.2023.2182376.*

[7] *Zhou, M., Dong, H., Wang, F.Y., Wang, Q., Yang, X., "Modeling and simulation of pedestrian dynamical behavior based on a fuzzy logic approach". Information Sciences, 360: 112-130. (2016). https://doi.org/10.1016/j.ins.2016.04.018.*

[8] *Zhou, M., Dong, H., Wen, D., Yao, X., Sun, X., "Modeling of crowd evacuation with assailants via a fuzzy logic approach". IEEE Transactions on Intelligent Transportation Systems, 17(9): 2395-2407. (2016). https://doi.org/10.1109/TITS.2016.2521783.*

[9] *Baker, C., "Artificial Intelligence: Learning Automation Skills with Python (2 books in 1: Artificial Intelligence a Modern Approach & Artificial Intelligence Business Applications)". Independently published. (2019).*

[10] *Teso, S., Alkan, Ö., Stammer, W., Daly, E., "Leveraging explanations in interactive machine learning: An overview. Frontiers in Artificial Intelligence", 6: 1066049. (2023). https://doi.org/10.3389/frai.2023.1066049.*

[11] *Helbing, D., Molnar, P., "Social force model for pedestrian dynamics". Physical Review E, 51(5): 4282. (1995). https://doi.org/10.1103/PhysRevE.51.4282.*

[12] *Reynolds, C. W., "Flocks, herds, and schools: A distributed behavioral model, ACM SIGGRAPH Computer Graphics", Volume 21, Issue 4 Pages 25 – 34, (1987). https://doi.org/10.1145/37402.37406.*

[13] *Lee, J.-G., Han, J., & Li, X., "Trajectory clustering: A partition-and-group framework", (2008). https://dl.acm.org/doi/10.1145/1247480.1247546.*

[14] *Li, X., Han, D., Guo, R., & Mao, J., "Pedestrian evacuation modeling and simulation on metro platforms considering panic impacts", (2016). https://doi.org/10.1016/j.sbspro.2014.07.209.*

[15] *Yu, Q., Ma, Y., & Wang, P., "Reinforcement learning for heterogeneous crowd simulation". (2021).*

[16] *Moussaid, M., Helbing, D., & Theraulaz, G., "How simple rules determine pedestrian behavior and crowd disasters". Proceedings of the National Academy of Sciences, 108(17), 6884–6888, (2011).*

[17] *Bouzat, S., Kuperman, M.N., "Game theory in models of pedestrian room evacuation". Physical Review E, 89(3): 032806. (2014). https://doi.org/10.1103/PhysRevE.89.032806*

[18] *Johansson, A., Helbing, D., & Shukla, P. K., "Specification of the social force pedestrian model by evolutionary adjustment to video tracking data. Advances in Complex Systems", 10(2), 271–288, (2007).*

[19] *Frantzich, H., Nilsson, D., "Evacuation experiments in a smoke filled tunnel". In 3rd International Symposium on Human Behaviour in Fire, Interscience Communications Ltd, United Kingdom, pp. 229-238. (2004).*

[20] *Kobes, M., Helsloot, I., De Vries, B., Post, J.G., Oberijé, N., Groenewegen, K., "Way finding during fire evacuation; An analysis of unannounced fire drills in a hotel at night". Building and Environment, 45(3): 537-548. (2010). https://doi.org/10.1016/j.buildenv.2009.07.004.*

[21] *Ma, J., Lo, S.M., Song, W.G., "Cellular automaton modeling approach for optimum ultra high-rise building evacuation design". Fire Safety Journal, 54: 57-66. (2012). https://doi.org/10.1016/j.firesaf.2012.07.008.*

[22] *Song, Y., Gong, J., Li, Y., Cui, T., Fang, L., Cao, W., "Crowd evacuation simulation for bioterrorism in micro-spatial environments based on virtual geographic environments". Safety Science, 53: 105-113. (2013). https://doi.org/10.1016/j.ssci.2012.08.011*

[23] *Nasir, M., Lim, C.P., Nahavandi, S., Creighton, D., "A genetic fuzzy system to model pedestrian walking path in a built environment". Simulation Modelling Practice and Theory, 45: 18-34. (2014). https://doi.org/10.1016/j.simpat.2014.03.002.*

[24] *Dell'Orco, M., Marinelli, M., Ottomanelli, M., "Simulation of crowd dynamics in panic situations using a fuzzy logic-based behavioural model". In: de Sousa, J., Rossi, R. (eds) Computer-based Modelling and Optimization in Transportation. Advances in Intelligent Systems and Computing, Springer, Cham, 262. (2014). https://doi.org/10.1007/978-3-319-04630-3_18.*

[25] *Bahamid, A., Mohd Ibrahim, A., "A review on crowd analysis of evacuation and abnormality detection based on machine learning systems". Neural Computing and Applications, 34(24): 21641-21655. (2022). https://doi.org/10.1007/s00521-022-07758-5.*

[26] *Lee, J., Won, J., Lee, J., "Crowd simulation by deep reinforcement learning". In Proceedings of the 11th ACM SIGGRAPH Conference on Motion, Interaction and Games, pp. 1-7. (2018). https://doi.org/10.1145/3274247.3274510.*

[27] *Wang, K., Shi, X., Goh, A.P.X., Qian, S., "A machine learning based study on pedestrian movement dynamics under emergency evacuation". Fire Safety Journal, 106: 163-176. (2019). https://doi.org/10.1016/j.firesaf.2019.04.008.*

[28] *Yao, Z., Zhang, G., Lu, D., Liu, H., "Data-driven crowd evacuation: A reinforcement learning method". Neurocomputing, 366: 314-327. (2019). https://doi.org/10.1016/j.neucom.2019.08.021.*

[29] *Li, X., Liang, Y., Zhao, M., Wang, C., Bai, H., Jiang, Y., "Simulation of evacuating crowd based on deep learning and social force model". IEEE Access, 7: 155361-155371. (2019). https://doi.org/10.1109/ACCESS.2019.2949106.*

[30] *Raschka, S., Patterson, J., Nolet, C., "Machine learning in python: Main developments and technology trends in data science, machine learning, and artificial intelligence". Information, 11(4): 193. (2020). https://doi.org/10.3390/info11040193.*

[31] *Zidane, Y.J.T., Olsson, N.O., "Defining project efficiency, effectiveness and efficacy". International Journal of Managing Projects in Business, 10(3): 621-641. (2017). https://doi.org/10.1108/IJMPB-10-2016-0085.*

[32] *Helbing, D., Johansson, A., Al-Abideen, H.Z., "Dynamics of crowd disasters: An empirical study". Physical Review E, 75(4): 046109. (2007). https://doi.org/10.1103/PhysRevE.75.046109.*

[33] *Helbing, D., Farkas, I., Vicsek, T., "Simulating dynamical features of escape panic". Nature, 407(6803): 487-490. (2000). https://doi.org/10.1038/3503502.*

[34] *M Chatra and M Bourahla, "Agent-Based Simulation of Crowd Evacuation Through Complex Spaces", Ingénierie des Systèmes d'Information, 2024, Vol 29, Issue 1, p83, https://doi.org/10.18280/isi.290110*