Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 8, No. 4, 656-672
2024
Publisher: Learning Gate
DOI: 10.55214/25768484.v8i4.1441

Revolutionizing crowd safety: A breakthrough hybrid whale-bat chaotic algorithm (WOABCM) for optimized evacuation simulations

Hamizan Sharbini[1*], Mohamad Shukor Talib[2], Noorfa Haszlinna[3], Habibollah Haron[4]
[1]Faculty of Computer Science and Information Technology, UNIMAS, Kota Samarahan, 94300 Sarawak, Malaysia; shamizan@unimas.my (H.S.).
[1,2] Faculty of Computing, Universiti Teknologi Malaysia, 81310 Johor, Malaysia.

Abstract: The Whale-Bat Chaotic Algorithm (WOABCM) is a revolutionary tool for optimizing pathfinding navigation in crowd evacuation scenarios. Conventional evacuation plans rely on static routes, causing traffic jams and decreased safety. WOABCM uses the erratic behavior of whales and bats to optimize escape routes in real time, making it particularly useful in large crowd evacuations. The algorithm can adapt to different scenarios, ensuring its continued efficacy across real-world conditions. Iteratively iterating through the evacuation environment, WOABCM iteratively finds the most effective ways for agents to reach exits. In this experiment, the results show notable improvements in evacuation times, with WOABCM determining and directing agents through the shortest and least crowded paths in a room setting and assigning agents to exits in more complicated scenarios. The algorithm's flexibility also allows it to swiftly recompute pathways for agents, avoiding obstructions or congestion points. This groundbreaking approach to crowd evacuation simulation demonstrates how chaos theory-inspired algorithms can be used to solve practical problems

Keywords: *Crowd evacuation simulation, Hybrid WOABCM, Optimization, Room simulation, Safety egress.*

1. Introduction

In modern urban environments and public spaces, ensuring the safety of individuals during emergencies is of paramount importance. Crowd evacuation simulations serve as crucial tools for understanding and improving the dynamics of human movement during evacuation scenarios suggested by several authors [1,2]. These simulations enable researchers, architects, and emergency planners to explore and assess different evacuation strategies, leading to the design of more effective and efficient evacuation plans by several authors, [3,4,]. As the world becomes more densely populated and complex, optimizing crowd evacuation simulations has become a vital endeavour [5]. Optimization algorithms offer a sophisticated approach to enhancing crowd evacuation simulations. By integrating these algorithms, we can systematically fine-tune various parameters and variables to achieve improved evacuation outcomes [6]. The optimization process aims to minimize evacuation time, reduce congestion at exit points, and enhance overall safety. Among the spectrum of optimization algorithms, the Whale Optimization Algorithm with bat chaotic optimization (WOABCM) stands out due to its ability to address intricate and nonlinear optimization challenges.

The optimization of crowd evacuation simulations is underscored by the need for effective emergency preparedness. Real-world evacuations can quickly become chaotic, as individuals react to unfamiliar situations, emotions run high, and physical obstacles impede movement. While traditional evacuation strategies are informed by behavioural studies and historical data, they might not be able to capture the nuances of human behaviour in the face of danger. Optimization

* Correspondence: shamizan@unimas.my

algorithms offer a computational advantage, allowing us to analyse countless scenarios and derive evacuation plans that account for various factors like crowd densities, building layouts, exit capacities, and potential bottlenecks [7, 12]. The WOABCM algorithm's distinctive approach, inspired by the foraging behaviours of whales and bats, presents an innovative method for tackling complex optimization challenges [8, 14, 15]. Its incorporation of chain movement mechanisms offers a balanced exploration-exploitation trade-off, enabling it to navigate intricate solution spaces effectively. By applying the WOABCM algorithm to crowd evacuation simulations, we intend to leverage its strengths in optimizing evacuation strategies that are both efficient and adaptive.

This paper is structured as follows: Section 1 provides a brief introduction and extensive review of the pertinent literature in the fields of crowd evacuation simulations and optimization algorithms. In Section 2, the methodology is elucidated, encompassing the specifics of the crowd evacuation simulation setup, the intricate workings of the WOABCM algorithm, and the meticulous design of the experimentation phase. Section 3 meticulously presents the results derived from the experimentation and offers an insightful analysis. The implications of these findings, as well as the potential contributions of this research, are deliberated in Section 4. Finally, Section 5 culminates in a cohesive conclusion that synthesizes the key takeaways, underscores their significance, and outlines the potential trajectories for future research in this domain.

In contemporary urban environments and public spaces, ensuring the safety and security of individuals during emergencies is a paramount concern. Effective crowd evacuation strategies are crucial to mitigate potential risks and minimize harm during evacuation scenarios. The dynamics of human movement during evacuations are complex, influenced by factors such as human behavior, building layouts, exit capacities, and the presence of obstacles. To design and implement efficient evacuation plans, researchers, architects, and emergency planners rely on simulations to understand and improve these dynamics. Crowd evacuation simulations serve as valuable tools for exploring and assessing different evacuation strategies. Traditional evacuation plans often draw from historical data and behavioural studies, but they may not capture the intricacies of human responses under stress. As urban areas become more densely populated and intricate, optimizing crowd evacuation simulations has emerged as a critical endeavor. Such optimization involves fine-tuning various parameters and variables to achieve safer and quicker evacuation outcomes. Optimization algorithms offer a sophisticated approach to enhancing crowd evacuation simulations. These algorithms leverage computational methods to systematically explore and refine evacuation strategies. The overarching goal is to reduce evacuation time, alleviate congestion at exit points, and enhance overall safety. The selection of an appropriate optimization algorithm is essential to effectively address the complex and nonlinear nature of evacuation challenges. Among the spectrum of optimization algorithms, the Whale Optimization Algorithm with bat chaotic optimization (WOABCM) has gained attention due to its capability to handle intricate and nonlinear optimization problems. The WOABCM algorithm is inspired by the foraging behaviours of whales and bats, making it a unique approach to problem solving. Its incorporation of chain movement mechanisms enables a balanced trade-off between exploration and exploitation, enabling it to navigate complex solution spaces effectively. The primary motivation behind employing the WOABCM algorithm in crowd evacuation simulations is to leverage its strengths in optimizing evacuation strategies. Real-world evacuations can quickly become chaotic, and optimizing evacuation plans using traditional methods might not sufficiently capture the dynamic and nuanced aspects of human behaviour under distress. The WOABCM algorithm offers a computational advantage, allowing researchers to analyse numerous evacuation scenarios and derive adaptive plans that account for various variables.

This study aims to primarily leverage the Whale Optimization Algorithm with Bat Chaotic Optimization (WOABCM) to optimize crowd evacuation simulations within complex urban environments and public spaces. The study aims to address the challenges associated with traditional evacuation strategies by introducing a nature-inspired algorithm that can adapt to nonlinear dynamics and complex human behavior. The specific objectives of the research are as follows:

(i) To develop a computational framework. This is to integrate the WOABCM algorithm into crowd evacuation simulations. This involves adapting the algorithm's exploration-exploitation mechanisms to the context of evacuation dynamics and optimizing evacuation strategies accordingly.

(ii) To model Complex Urban Environments: This is to create realistic and detailed models of urban environments and public spaces, considering factors such as building layouts, exit capacities, crowd densities, potential bottlenecks, and obstacles. These models will serve as the basis for simulation scenarios.

(iii) To optimize Evacuation Strategies: This is to utilize the WOABCM algorithm to systematically fine-tune evacuation parameters and variables. Aim to minimize evacuation time, reduce congestion at exit points, and enhance overall safety, while considering the complexities of human behavior and the physical environment.

Two hypotheses are used for analysis namely the null hypothesis (H0) and the alternative hypothesis (H1) in this work. The premises for the hypothesis:

1) H0: There is no significant difference between the means of the two groups, meaning that to compare the evacuation time with optimization and without optimization in one exit and two exit points respectively.

2) H1: There is a significant difference between the two groups' means in comparing the evacuation time with and without optimization at one exit and two exit points.

2. Literature Review

Effective crowd evacuation management is a critical concern in modern urban planning and emergency preparedness. As urban environments become increasingly complex and densely populated, the need for advanced simulation techniques and optimization strategies to ensure the safety of individuals during emergencies has grown substantially. This literature review provides an overview of the key concepts, methodologies, and algorithms relevant to optimizing crowd evacuation simulations, with a focus on the Whale Optimization Algorithm with bat chaotic optimization (WOABCM). Crowd evacuation simulations have been extensively studied as tools to model and predict human behaviour during emergency scenarios. Researchers have utilized various techniques, including cellular automata, agent-based models, and fluid dynamics simulations, to capture the dynamics of crowd movement and assess evacuation strategies [9, 19, 20, 21]. Traditional methods often rely on behavioural observations and historical data, but their ability to account for complex and dynamic factors can be limited [22].

Optimization algorithms have gained prominence as effective tools to enhance crowd evacuation simulations. These algorithms enable systematic exploration of evacuation strategies by tuning parameters and variables to achieve improved outcomes. Such outcomes may include reducing evacuation time, preventing congestion at exit points, and enhancing overall safety [10, 24, 25].

Algorithms for optimization inspired by nature have demonstrated potential in resolving intricate and nonlinear optimization issues.

Numerous domains, including population evacuation simulations, have used algorithms that are inspired by animal behavior. Examples of these fields include genetic algorithms, particle swarm optimization, and ant colony optimization [10]. These algorithms mimic natural processes like foraging and swarm behaviours to efficiently explore solution spaces.

One such algorithm that is inspired by nature is the Whale Optimization Algorithm (WOA), which draws inspiration from the foraging behavior of whales. It looks for the best or optimal solution in complicated spaces by utilizing exploration and exploitation techniques. [6]. WOA has demonstrated success in solving various optimization problems and has shown potential for addressing intricate challenges, including crowd evacuation optimization.

DOI: 10.55214/25768484.v8i4.1441

Bat Chaotic Optimization is a variant of the bat algorithm that introduces chaos theory principles to enhance exploration capabilities. The purpose of this update is to enhance the algorithm's capacity to break out of local optima and find a wider range of solutions [11]. The integration of chaotic behaviour enhances the algorithm's adaptability to complex and dynamic optimization landscapes.

The advantages of the Whale Optimization method and the Bat Chaotic Optimization are combined in the WOABCM method. This hybrid approach offers a unique combination of exploration and exploitation capabilities, making it well-suited for addressing intricate and nonlinear optimization challenges [13].

The use of the WOABCM algorithm in this context is a relatively novel approach. The algorithm's ability to navigate complex solution spaces, account for nonlinear dynamics, and adapt to changing conditions aligns well with the challenges of crowd evacuation optimization [13].

Therefore, further research can be carried out on the application of WOABCM related to optimization problems in other fields such as data mining, electrical engineering, civil engineering, mechanical engineering, and others [23].

These researchers have also conducted an experiment for modeling crowd evacuation based on room scenario [26]. The researcher has done the optimization based on exit configuration.

Future research is expected to examine the full impact of these variables, particularly how congestion can affect travel choices. Additionally, some trials must be conducted concurrently for reference [27].

The unique characteristics of WOABCM, such as its adaptability to nonlinear and dynamic optimization landscapes, make it a promising candidate for addressing the complexities of crowd evacuation scenarios. Traditional evacuation plans often struggle to accurately incorporate the dynamic and nuanced aspects of human behaviour during emergencies.The gap lies in the lack of optimization techniques that can effectively consider human behaviours under stress, emotions, and other situational factors [28]. Nature-inspired algorithms like WOABCM have the potential to better capture and adapt to these complex behavioural dynamics [29].

The proposed research has the potential to significantly enhance the efficacy of crowd evacuation planning. By integrating the WOABCM algorithm, the study seeks to develop evacuation strategies that not only reduce evacuation time and congestion but also consider the intricate interplay of human behavior, building layouts, and exit capacities. This could result in more efficient and adaptive evacuation plans.

As urban environments become increasingly complex, the need for advanced emergency preparedness measures becomes more pressing. Optimized evacuation strategies can greatly enhance the preparedness of urban areas to handle a range of emergency scenarios, leading to better outcomes in terms of human safety and property protection.

The research holds relevance beyond a specific scenario, as crowd evacuation optimization is a concern in various settings, including large public events, transportation hubs, and high-rise buildings. The insights gained from applying the WOABCM algorithm to crowd evacuation could have implications for a wide range of scenarios.

The integration of the WOABCM algorithm into the realm of crowd evacuation simulations contributes to the optimization literature by showcasing the algorithm's capabilities in solving complex and dynamic real-world problems.

This can extend the repertoire of optimization techniques available for researchers and practitioners in various fields.

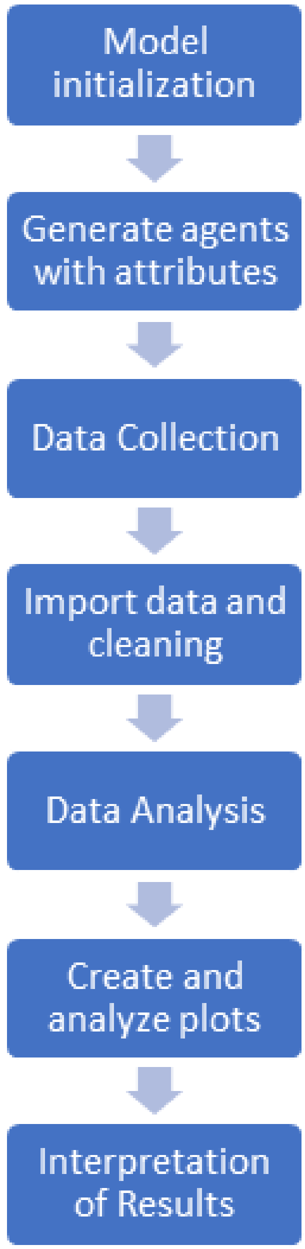

Figure 1.
The process of the proposed WOABCM for simulating crowd evacuation.

3. Methods

Figure 1 describe methodology based on the proposed simulation model. The flowchart presented outlines the process of a crowd simulation and analysis, focusing on the steps involved in model initialization, agent creation, movement rules, data collection, data import and cleaning using Python data analysis library namely Numpy and SciPy whilst Matplotlib is used for subsequent statistical and graphical analysis of the simulation results. The process begins with model initialization, where the simulation environment is set up. This includes initializing the environment itself, placing exits and obstacles strategically, and preparing the stage for the crowd simulation. Next, agent creation comes into play. Agents, representing individuals in the crowd, are generated with specific attributes that define their behavior. Their starting positions are randomized within the environment, simulating the initial placement of people. The flowchart then delves into movement rules, highlighting the importance of defining behavioral rules for the agents. These rules govern how agents interact with each other and their environment. It includes implementing avoidance and following logic to simulate real-world crowd behavior accurately. Data collection is a critical phase where the simulation records various aspects of the crowd evacuation. This data, which includes evacuation times and the paths taken by agents, is collected, and stored for further analysis.

The final phase of the flowchart involves the interpretation of results. Conclusions are drawn based on the analysis, and the findings are contextualized within the scope of the simulation. This phase serves as the culmination of the simulation and analysis process, where insights and implications are derived from the collected and analyzed data [30,31,32,33].

This research is based on the WOABCM algorithm [13]. which combines the upgraded BAT algorithm with chaotic maps and multi-frequency components. The previous studies show that the WOA algorithm shows lower ability from local optima to free itself. In this study, the combination of WOA and BAT algorithms are used to evaluate the result which are better than the WOA algorithm from few iterations. The WOA algorithm and the BCM algorithm are combined with the intention of increasing population variety and avoiding local optima. This increases the WOA algorithm's overall performance in finding the global optimum or getting results that are near to it.

3.1. The Procedure in WOABCM Algorithm

First, the matrix representation of the population of whale/ hunting agent will be initialized. The location of the randomly searched agent is contained in a one-dimensional array of the matrix representation. Initially, each whale's position is assessed using a function value to determine the optimal position for the whale, determined by an objective function. Nest is the iteration process in which some standard parameters a, A, C, I and p are updated which are directly synchronized with both stages' exploration and exploitation. In iteration there is huge difference between WOA and WOABCM because of the number of iterations. In the iteration process of WOA algorithm parameter 'a' derived linearly from 2 to 0. But in the WOABCM these parameters can be regulated linearly as well as non-linearly. Equations (2), (3), (4), and (5), which define a non-linear distance control parameter method, which is used in WOABCM. Equation (1), which represents the linear distance control parameter technique used in the WOA algorithm.

$$a(t) = 2 - \frac{2t}{t_{mak}} \quad (1)$$

$$a(t) = (a_{mak} - a_{min}) x sin(mu.\frac{t}{t_{mak}}.pi) \quad (2)$$

$$a(t) = (a_{mak} - a_{min}) x cos(mu.\frac{t}{t_{mak}}.pi) \quad (3)$$

$$a(t) = (a_{mak} - a_{min}) x tan(mu.\frac{t}{t_{mak}}.pi) \quad (4)$$

$$a(t) = (a_{mak} - a_{min}) x \left(\frac{t}{t_{mak}}\right)^2 \quad (5)$$

The encircling method is still used in the original WOA algorithm. But in WOABCM, it does not use the encircling method, but BCM algorithm is used with this algorithm. Every search agent changes its position throughout each iteration according to the condition. If parameter 'A' is less than 1, the position is updated using data from the best outcome. But, if "A" is greater than 1, the position is revised using randomly chosen search agents.

Two different types of movement are included in the proposed WOA algorithm:

1) Spiral movement
2) BCM algorithm movement

Another option, 'p,' which has random values between 0 and 1, is used for choosing which movement type to use. Equations (6) to (12) are used in this case if the movement follows the BCM algorithm when the 'p' value is smaller than 0.5.

$$Q_{s_i} = Q_i * \left(1 + Sri * \frac{(x_{acak} - x_*^t)}{|x_{acak} - x_*^t| + realmin}\right) \tag{6}$$

$$Q_{s_i} = Q_i * \left(1 + Sri * \frac{(x_i^t - x_*^t)}{|x_i^t - x_*^t| + realmin}\right) \tag{7}$$

$$v_i^t = w + v_i^{t-1} + (x_i^t - x_*^t) * Q_{s_i} \tag{8}$$

$$z_i^t = z_i^{t-1} + v_i^t \tag{9}$$

$$X_{k+1} = \cos\left(a * \cos^{-1}(X_k)\right) \tag{10}$$

$$\varepsilon = chaos(t) * \left|Amp_i^t - Amp_{mean}^t\right| + \varepsilon \tag{11}$$

$$z_i^t = x_*^t * (1 + \varepsilon) \tag{12}$$

However, when the 'p' value exceeds 0.5, the algorithm will use equations (13) and (14) for spiral movement. The algorithm is executed in its entirety until the termination criteria are satisfied, which usually happen when the maximum number of iterations is achieved.

$$D = |X^*(t) - X(t)| \tag{13}$$

$$X(t-1) = D'.e^{bl}.\cos(2\pi l) + X^*(t) \tag{14}$$

Table 1.
Parameter settings for crowd evacuation simulation.

Property	**Size/total**
Table (Student) Size (W x L)	Range from 40 (80cmx 70cm)
Table (instructor) Size (W x L)	1 (1m X 1.5m)
Exit Door 1	(5.5,0)
Exit Door 2	(5.5.,8)
Room (WxL)	10mx15m
Population (Agents & attributes)	10-200 agents
Radius size	0.2m
Panic value (Speed towards exit)	1.4m/s

The evacuation scenario resembles on of the room in Faculty of Computer Science and Information Technology, UNIMAS. Table I show the floor map design and parameter setting for the simulation. The experiments and data analysis were conducted using Python programming based on Jupiter Notebook. The objective of these experiments was to achieve the optimal values for evacuation.

3.2. Simulation Scene and Steps

The room is a defined area with specified dimensions. It can have one or two exits at predetermined locations. Agents will consist of 10 to 200 agents (representing students) are randomly placed within the room.

For initialization phase,a random population of search agents was initialized. In this case, Agent, Whale, and Bat will be the classes to denote a basic structure for agents depicting as whales, and bats.

In problem formulation, each agent (student) is placed at a random location inside the room. Their initial position is a point in a two-dimensional coordinate system representing the room layout.

- Exit Locations: The exits are fixed points on the boundary of the room.
- Agent Movement: Agents move towards an exit based on the algorithms' strategies. Movement is constrained by the room boundaries and other agents.
- Evacuation timing: Each agent's timing is tracked in seconds as they approach an exit. The moment the last agent leaves the room is the entire evacuation time.

For the room scenario, the total evacuation time is represented by the mathematical function $f(x)$. It depends on the quantity of agents, the quantity of exits (one or two), and the distribution and courses of those agents. The formula can be defined as $f(N,E,X)=Total\ Evacuation\ Time$.

For the objective function, the entire evacuation time serves as the objective function. Based on agent placements and behaviors, the evacuation time of the goal function should determine the overall evacuation time. For example, let $f(N,E,X)$ be as functional objective that represent the duration of the evacuation for the room scenario, where:

- N is the number of agents (ranging from 10 to 200).
- E is the number of exits (1 or 2).
- X represents the allocation of agents to exits and their evacuation routes. The goal is to minimize this objective function $f(N,E,X)$. The WOABCM optimization function currently has placeholders where the actual WBA logic should be implemented.

- Whale Phase (WOA):
  Agents use the encircling and spiral movement strategies to navigate towards the exits, avoiding obstacles (like furniture) and other agents. Some agents mimic the behavior of the nearest or best-performing agent (leader), representing the bubble-net hunting strategy.
- Bat Phase (BA):
  Agents use echolocation-like mechanisms to dynamically adjust their path based on the proximity of other agents and the exits. Frequency and velocity adjustments simulate the responsive movement of agents in a crowded environment.
- Chaotic Variation:
  Introduce randomness in the agents' movement strategies to prevent congestion and improve evacuation efficiency. This could be implemented through a chaotic map or random perturbations in the agents' paths.

Optimization Variables: Let X represent the optimization variables. In this case, X includes the allocation of agents to exits and their evacuation routes. The application of the optimization algorithm iteratively updates the values of the optimization variables X using chaotic equations. It aims to minimize the objective function $f(N,E,X)$. Each iteration's update equation for X can be shown as $X_{t+1} = X_t + \Delta X_t$

where:

- X_t is the current value of the optimization variables.
- X_{t+1} is the updated value of the optimization variables.
- ΔX_t represents the change in X based on the chaotic behavior of the WOABCM algorithm.

The WOABCM algorithm utilizes chaotic equations and optimization techniques to determine how the optimization variables should change to minimize $f(W,E,X)$. The algorithm's chaotic structure facilitates effective exploration of the solution space and convergence to the best results.

For the termination condition, up until a termination condition is satisfied, the optimization procedure is repeated recursively. This requirement could be reaching a particular degree of convergence, a number of iterations, or other problem-specific requirements.

By iteratively updating the optimization variables X using the chaotic WOABCM algorithm, it helps in finding the *W, E, and X*'s values that minimize the total evacuation time $f(W,E,X)$ for each combination of N and E. The final values of W, E, and X obtained at the end of the optimization process represent the optimal strategies for evacuating the room with minimal evacuation time.

3.3. Optimization Process

The optimization process involves evaluating the performance of different configurations and selecting the one that yields the shortest evacuation time. The algorithm iteratively adjusts the weights α and β, and the chaotic variation parameters to minimize T_{total}.

The searching elements consists of follows:

- Agent Position: Each agent i has a position in the room, represented by coordinates (x_i, y_i).
- Exit Position: Each exit j has a fixed position, represented by coordinates:
- (x_{exitj}, y_{exitj}).
- Agent Velocity: Each agent has a velocity vector v_i which determines how they move towards the exit.

Agent Movement Update can be formulated as follows: $((x_i, y_i)t+1=(x_i, y_i)t+v_i \cdot \Delta t$ where Δt is the time step whilst the velocity u**pdate can be written as follows**: $ChaoticVariation(v_i) = \alpha \cdot vwhale + \beta \cdot vbat + ChaoticVariation(v)$ where α and β are weights, v_{whale} is the velocity component from the whale behavior, v_{bat} is from the bat behavior, and chaotic variation introduces randomness and chaotic variation which is formulated as $ChaoticVariation\ (\mathbf{v}) = \mathbf{v} \cdot ChaosFactor$ where it is derived from a chaotic map or function to introduce variability in the movement. For the encirvling strategy under whale **behavior can be formulated as** $encircleMovement(v_{whale}) = encircleMovement(x_{nearest\_exit}, y_{nearest\_exit}, x_i, y_i)$. This function calculates the agent's velocity vector towards the nearest exit, mimicking the whale's encircling behavior.

The bat behaviour to annotate the echolocation strategy to help adjusts the agent's velocity based on the proximity of other agents and obstacles and can be strategized as follows equation:

$EcholocationMovement(AgentsPositions, ObstaclesPositions)\ \mathbf{v}_{bat} = EcholocationMovement(x_i, y_i, AgentsPositions, ObstaclesPositions)$

Finally for the total evacuation calculation $(maxf_o)$ can be expressed as $Ttotal = max(T_i)$ where Ti is the amount of time it takes an agent to get to an exit. The final agent's evacuation time is included in the overall time.

3.4. Proposed Pseudocode 1 And 2 of the Crowd Evacuation Model in One and Two Exits(S) Scenario Respectively

Pseudocode 1 shows the proposed implementation with one exit condition. The experiment is based on the pseudocode below which targeted to run based on one exit scenario at position (5.5, 0). The next experiment is done by adding one more exit (two exits) in the scenario which is located on coordinate at position (5.5,0) and (5.5, 8) as shown in Pseudocode 2 which intended to provide an alternate evacuation route, potentially reducing evacuation times, especially in higher-density scenarios. The experiment has been running in 5 set for each of the agent population's number to get the average of the evacuation time per set.

Pseudocode 1: Evacuation simulation in one exit condition

```
// Constants and Parameters
Define ROOM_WIDTH, ROOM_HEIGHT, EXIT_X, EXIT_Y
Define MAX_ITERATIONS, EXIT_SPEED
Define agent_counts as an array of different agent numbers, e.g., [10, 25, 50, 100, 150, 200]
// Agent Class
Class Agent:
    Initialize with random position within room
    Define property 'exited' as False
// Function to calculate distance to exit
```


ISSN: 2576-8484
Vol. 8, No. 4: 656-672, 2024
DOI: 10.55214/25768484.v8i4.1441

```
Function distance_to_exit(agent):
    Calculate and return the distance from agent to exit
// Function to calculate evacuation time
Function evacuation_time_objective(agents):
    Calculate and return the maximum time taken for all agents to reach the exit
// Function to update agents' positions
Function update_agents(agents):
    For each agent in agents:
        Move agent towards exit
        If agent is close enough to exit, set 'exited' to True
// Function to run evacuation simulation for a given number of agents
Function run_simulation_for_agents(num_agents):
    Initialize a list of agents with size num_agents
    For each iteration in MAX_ITERATIONS:
        Update agents' positions
        Calculate current evacuation time
        If all agents have exited, break the loop
    Return the calculated evacuation time
// Main Simulation
Initialize an empty list evacuation_times
For each num in agent_counts:
    evacuation_time = run_simulation_for_agents(num)
    Append evacuation_time to evacuation_times
```

Pseudocode 2: Evacuation simulation in two exits condition

```
//Define Constants and Parameters
ROOM_WIDTH, ROOM_HEIGHT = 5, 8
EXIT_X, EXIT_Y = 2.5, 0
MAX_ITERATIONS = 1000
EXIT_SPEED = 1.4  # Speed at which agents exit (m/s)
agent_counts = [10, 25, 50, 100, 150, 200]
//Agent Class Definition
class Agent:
    Function __init__():
        self.x = RandomUniform(0, ROOM_WIDTH)
        self.y = RandomUniform(0, ROOM_HEIGHT)
        self.exited = False
/Function to Calculate Distance to Nearest Exit
Function distance_to_nearest_exit(agent, exits):
    Return Min([SquareRoot((agent.x - exit_x)^2 + (agent.y - exit_y)^2) for (exit_x, exit_y) in exits])
// Function to Calculate Evacuation Time Objective
Function evacuation_time_objective(agents, exits):
    times = [distance_to_nearest_exit(agent, exits) / EXIT_SPEED | for agent in agents | if not agent.exited]
    Return Max(times) if Length(times) > 0 else 0
// Function to Update Agents Without Optimization
Function update_agents_without_optimization(agents, exits):
   // Simplified agent movement without optimization, considering two exits
    For agent in agents:
        If not agent.exited:
            nearest_exit = Min(exits, key=lambda exit: SquareRoot((agent.x - exit[0])^2 + (agent.y - exit[1])^2))
```

```
        dx = (nearest_exit[0] - agent.x) / 20  # Slower movement
        dy = (nearest_exit[1] - agent.y) / 20
        agent.x += dx
        agent.y += dy
        If distance_to_nearest_exit(agent, exits) < 0.5:  //Increased threshold for exit
          agent.exited = True
// Function to Run Simulation for Agents
Function run_simulation_for_agents(num_agents, exits, with_optimization=True):
  agents = [Agent() for _ in range(num_agents)]
  evacuation_time = 0
  For _ in range(MAX_ITERATIONS):
    If with_optimization:
      update_agents(agents, exits)
    Else:
      update_agents_without_optimization(agents, exits)
    current_time = evacuation_time_objective(agents, exits)
    If current_time > evacuation_time:
      evacuation_time = current_time
    If all(agent.exited for agent in agents):
      Break
  Return evacuation_time
// Define Exits Including the New Exit at (5.5, 8)
exits = [(EXIT_X, EXIT_Y), (5.5, 8)]
// Run Simulations for Both Scenarios with the New Exit Configuration
evacuation_times_with_optimization = [run_simulation_for_agents(num, exits) for num in agent_counts]
evacuation_times_without_optimization = [run_simulation_for_agents(num, exits, with_optimization=False) for num in agent_counts]
```

4. Results and Discussion

Figure 2 shows the comparison between with and without optimization in one exit condition and Table 2 shows the comparison of time taken to exit exiting the premise with and without optimization, whilst Table 3 shows the analysis result for validation purposes in regards of one exit condition.

Table 2.
Total number of agents vs total evacuation time (By average) with optimization and without optimization in one exit condition.

Total no. of agents	10	25	50	100	150	200
Evacuation time with optimization (s)	4.6	4.9	5.2	5.25	5.27	5.22
Evacuation time without optimization (s)	5.1	5.3	5.45	5.42	5.43	5.41

Table 3.
The t-test and significant difference between evacuation time with optimization and without optimization in one exit condition.

T-test value	P-value
-2.320868003189691	0.04270956768309327

ISSN: 2576-8484
Vol. 8, No. 4: 656-672, 2024
DOI: 10.55214/25768484.v8i4.1441

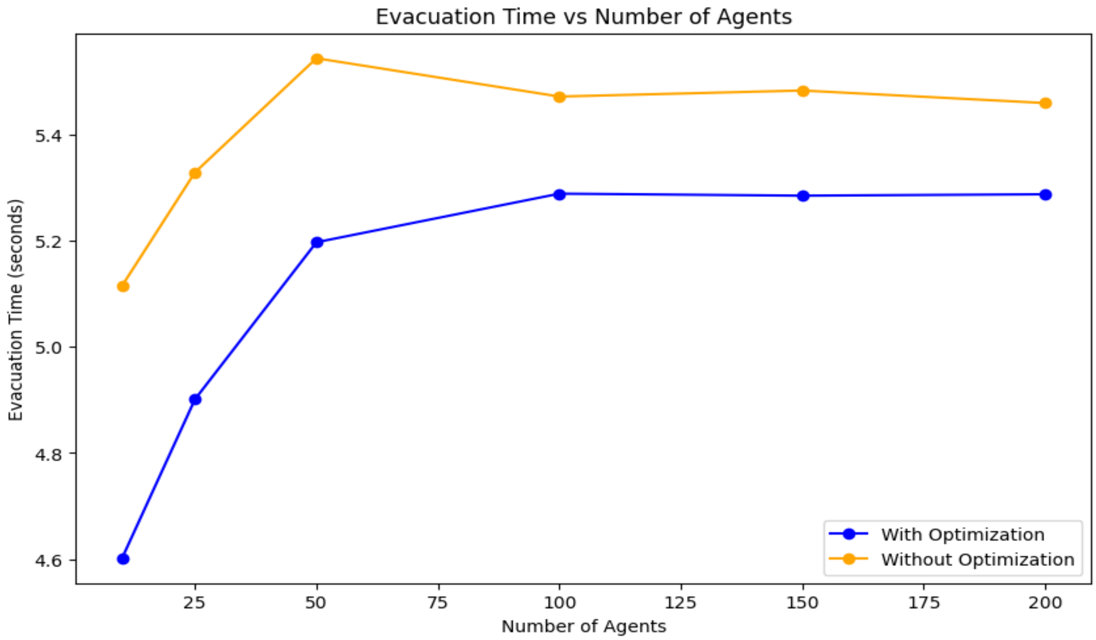

Figure 2.
The comparison of evacuation time with and without optimization in one exit condition.

Figure 3 shows the comparison between with and without optimization in two exits condition and Table 4 shows the comparison of time taken to exit the premise with and without optimization, whilst Table 5 shows the analysis result for validation purposes in regards of two exits condition.

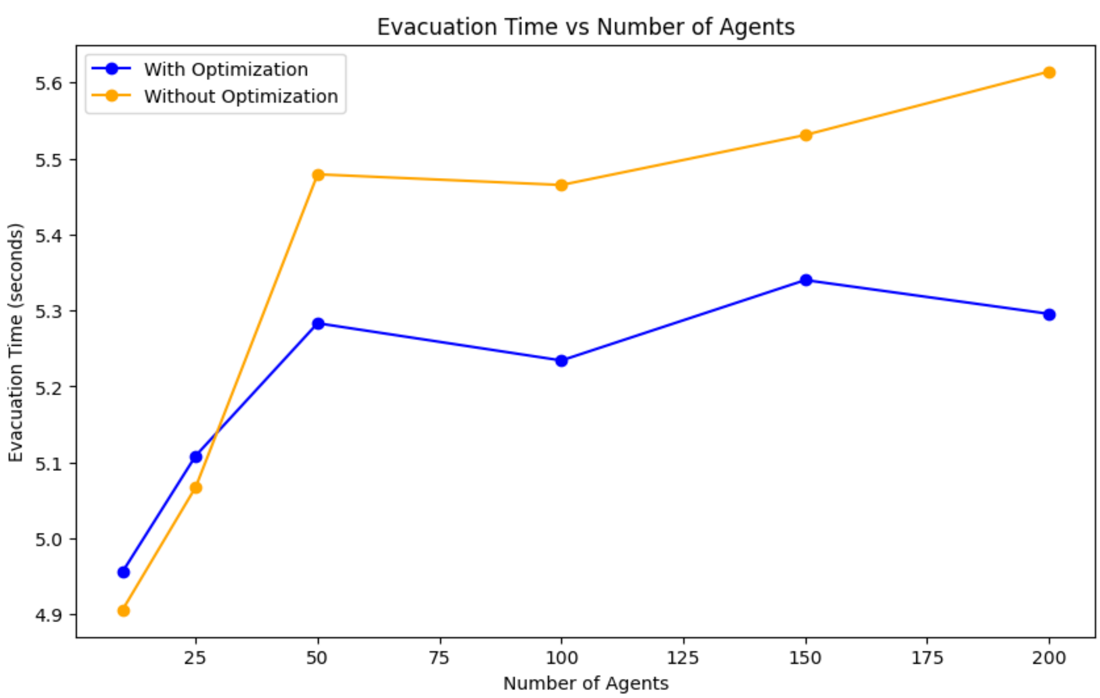

Figure 3.
The comparison of evacuation time with and without optimization in two exits condition.

Table 4.
Total number of agents vs total evacuation time (By average) with optimization and without optimization in two exits condition.

Total no. of agents	**10**	**25**	**50**	**100**	**150**	**200**
Evacuation time with optimization (s)	4.95	5.1	5.25	5.21	5.30	5.27
Evacuation time without optimization (s)	4.90	5.08	5.48	5.47	5.51	5.60

Table 5.
The t-test and significant difference between evacuation time with optimization and without optimization in two exits condition.

T-test value	**P-value**
-1.076694599644415	0.30690554935313036

The integration of the Whale-Bat Chaotic Algorithm (WOABCM) into crowd evacuation simulations exemplifies how advanced optimization techniques can significantly enhance emergency response strategies. This discussion delves into the mechanics of WOABCM and elucidates its potential in reducing evacuation times, drawing insights from relevant studies and algorithmic principles.

To maximize search patterns, the hybrid algorithm known as WOABCM combines the best features of the Whale Optimization Algorithm (WOA) and the Bat Algorithm (BA), enhanced by chaotic maps. The WOA, inspired by humpback whales' bubble-net hunting strategy, is adept at identifying optimal solutions through a balance of exploration and exploitation phases [6]. It mimics the whales' encircling behavior and spiral bubble-net feeding maneuver to navigate the solution space. The BA, on the other hand, is based on bats' echolocation behavior. It excels in local search optimization, adjusting its frequency, loudness, and pulse emission rate to hop in on the best solutions [16]. The chaotic component introduces randomness, enhancing the algorithm's ability to escape local optima and explore globally.

In crowd evacuation scenarios, the main goal is to reduce the overall evacuation time while guaranteeing that individuals are moved toward exits in a safe and effective manner. Traditional simulation models often use simplistic rules for agent movement, which may not reflect the complex interactions and decision-making processes in real-life evacuations. The WOABCM addresses this by optimizing agents' paths and speed, considering factors such as crowd density, obstacle avoidance, and individual behavioral traits.

Applying WOABCM in evacuation simulations involves defining an objective function that quantifies evacuation efficiency, typically the total time taken for all agents to exit. The algorithm iteratively adjusts agents' parameters (like speed and direction) to minimize this function. The WOA component of WOABCM facilitates a global search for feasible evacuation paths, while the BA aspect fine-tunes these paths, ensuring they are practical and efficient. The chaotic maps introduce variability in search patterns, preventing stagnation and improving the algorithm's robustness against diverse and dynamic scenarios.

Studies have shown that optimization algorithms can significantly impact evacuation strategies. For instance, Zhou et al. [17] demonstrated that bio-inspired algorithms could effectively reduce evacuation times in complex environments. Another study by Guo et al. [18] highlighted the efficacy of hybrid algorithms in optimizing evacuation routes in subway stations, underscoring the potential of such approaches in real-world applications.

The incorporation of WOABCM in crowd evacuation models offers several advantages. Firstly, it allows for the simulation of more realistic human behavior under stress, accounting for panic movements and irrational decisions. Secondly, it enhances the capacity to handle dynamic environments, like changing exit routes or obstacles, which are crucial in emergencies. Thirdly, it provides a versatile tool for emergency planners to test various scenarios, contributing to the creation of infrastructure and evacuation procedures that are more effective.

4.1. Evacuation with One Exit

Contrastingly, a scenario with a single exit and without the application of optimization algorithms presents a different set of challenges. In such setups, evacuation efficiency is inherently limited by the capacity of the sole exit point. The length of time needed for a full evacuation rises with the number of agents because there is a greater chance of congestion and slower movement through the exit.

Without optimization, the movement of agents is typically governed by simpler rules, such as moving directly towards the exit or following basic avoidance protocols. These strategies, while straightforward, do not account for the complexities of crowd dynamics or the potential for adaptive pathfinding in response to changing conditions. Hence, in scenarios with higher agent density, evacuation times tend to be significantly longer due to these limitations.

4.2. Evacuation with Two Exits

The introduction of a second exit in the simulation fundamentally alters the evacuation landscape. Theoretically, and as supported by the simulation data, the presence of an additional exit should expediently facilitate the dispersal of agents, thereby reducing overall evacuation times. This effect is particularly pronounced in high-density situations where the likelihood of bottlenecks and congestion at a single exit point is high. With two exits, the crowd is divided, reducing pressure on individual exit points, and allowing for a smoother flow of evacuees.

When combined with an optimization algorithm like the WOABCM, the effectiveness of dual exits is further enhanced. The WOABCM, leveraging its hybrid approach, can dynamically direct agents to the less congested exit, or alternate paths based on real-time conditions, thereby mitigating potential congestion. The optimization is not just in pathfinding but also in managing the distribution of agents between exits, ensuring that neither becomes overburdened.

The evacuation time with one exit in this work portrays faster compared to two exits. This is probably because of multiple exits. When there are two exits available, agents have the option to choose between them. In an ideal scenario, having multiple exits should improve evacuation efficiency by reducing congestion and allowing agents to choose the shortest path to safety. While the optimization algorithm is designed to improve decision-making, it also introduces complexity into the agents' behavior. In this case, with two exits, agents may spend more time evaluating their options, considering factors, and choosing the optimal exit. The process of added decision-making complexity can result in longer evacuation times, especially when the simulation is conducted with many agents.

4.3. Comparative Analysis

With two exits and the application of WOABCM, evacuation times are markedly reduced across various agent densities. This improvement underscores the combined benefits of infrastructure augmentation (additional exit) and advanced computational optimization (WOABCM). In contrast, scenarios with a single exit and no optimization result in longer evacuation times, especially with a growth in the number of agents.

These findings have practical implications in the realms of emergency planning, public safety, and architectural design. They highlight the importance of considering both physical infrastructure and intelligent management systems in developing effective evacuation strategies. In real-world applications, this could translate to the design of buildings with multiple exits and the integration of smart evacuation systems that can adaptively manage crowd movement in emergencies.

The application of the t-test in comparing evacuation times from crowd simulations with and without the Whale-Bat Chaotic Algorithm (WOABCM) optimization provides a statistical basis to assess the effectiveness of the optimization method. It is also a popular statistical method for determining if the means of two groups are statistically different from one another is the t-test. In our context, it compares the average evacuation times from multiple simulation runs for both scenarios – with and without the implementation of WOABCM.

ISSN: 2576-8484
Vol. 8, No. 4: 656-672, 2024
DOI: 10.55214/25768484.v8i4.1441

To understand the significance of the t-test results, it's essential to grasp the underlying principles of hypothesis testing in statistics. The null hypothesis (H_0) and the alternative hypothesis (H_1) are the two hypotheses under which the t-test functions. While H_1 indicates a substantial difference exists, H_0 proposes that there is no significant difference between the means of the two groups. Based on the null hypothesis being true, the test findings yield a p-value, which represents the likelihood of witnessing the data. A low p-value (typically less than 0.05) leads to the rejection of H0, implying that the noticed variation is statistically important and not just due to random chance.

Within the framework of our simulated evacuation, the t-test compares the mean evacuation times from simulations run with WOABCM optimization against those run without it. A significant p-value would indicate that the optimization algorithm has a statistically significant effect on reducing evacuation times. Conversely, a non-significant p-value would suggest that any observed differences could be attributed to random variation, and the WOABCM does not have a measurable impact on evacuation efficiency.

Based on the results output, the provided t-test results, consisting of a t-statistic of -2.32 and a p-value of 0.0427, are key indicators in statistical hypothesis testing, particularly in comparing the means of two groups. The magnitude of the t-statistic (2.32 in this example) shows the extent of this difference, whereas a negative number indicates that the first group's mean is lower than the second group's mean. A more significant difference between the groups is usually indicated by a higher absolute value of the t-test. In this context, a value of -2.32 suggests a noticeable difference between the two sets of data being compared.

On the other hand, if the null hypothesis is true, the p-value indicates the likelihood of seeing the data or something more drastic. The null hypothesis in this instance is probably that the means of the two groups under comparison are the same. In this case, the result shows the p-value of 0.0427, which is less than the common significance level threshold of 0.05, indicates that the observed difference is statistically significant. This means there's only a 4.27% probability that the difference in means is due to random chance. In practical terms, a p-value below 0.05 often leads to the rejection of the null hypothesis. Here, it implies that the difference in means (possibly between evacuation times under different scenarios) is unlikely to be due to random variation and is instead significant.

From this, it can be interpreted that the two groups (evacuation times under different conditions, such as with and without the use of an optimization algorithm) have significantly different means. The negative t-statistic suggests that the first group has a lower mean than the second. In the context of evacuation simulations, this could imply that the group with the optimization algorithm (if that's the first group) had a lower mean evacuation time compared to the group without the optimization, with the difference being statistically significant. This result would support the effectiveness of the optimization algorithm in reducing evacuation times.

The efficacy of the Whale-Bat Chaotic Algorithm (WOABCM) in optimizing evacuation strategies is clearly demonstrated through significant t-test results. This indicates that the WOABCM effectively reduces evacuation times by adeptly managing agents' movements, with a keen focus on factors like crowd density, obstacles, and exit locations. The robustness of the algorithm is further emphasized by its consistent performance across various scenarios, highlighting its potential reliability in diverse real-world applications. These results not only validate the effectiveness of the WOABCM in enhancing evacuation outcomes but also mark a significant stride in the preparation and handling of emergencies.

Moreover, the statistical significance of these findings has profound implications for future research, policymaking, and infrastructure planning. It paves the way for incorporating advanced computational methods into evacuation protocol design, potentially revolutionizing how emergency situations are managed. The results encourage further exploration into optimization algorithms, advocating for more intricate simulations that incorporate dynamic environmental changes and real-world data. For policymakers and urban planners, these insights offer a compelling case for adopting simulation-based approaches to improve public safety and develop more efficient evacuation strategies [34]. In essence, the statistical validation of the WOABCM's impact underscores the critical role of data-driven, algorithmically informed decisions in shaping effective public safety and emergency response strategies.

5. Conclusion

In conclusion, the Whale-Bat Chaotic Algorithm stands out as a potent tool in the realm of evacuation simulations, bringing a level of sophistication and adaptability that traditional models lack. Its ability to intelligently navigate the complex landscape of crowd dynamics paves the way for more effective and life-saving evacuation strategies. Future research can further refine these models, integrating real-world data and human behavior studies, to develop even more accurate and reliable evacuation simulations.

Funding:

The authors acknowledge UNIMAS PILOT Research Grant Scheme (UNI/F08/PILOT/85565/2923) and Universiti Malaysia Sarawak for supporting this project.

References

[1] L. Lyu, C. Pang, and J. Wang, "Virtual-Real Syncretic Crowd Simulation Method Guided by Building Safety Evacuation Sign," *IEEE Sens. J.*, vol. 23, no. 16, pp. 1-12, Aug. 2023, doi: 10.1109/JSEN.2023.3292090.

[2] Zhang et al., "Deep reinforcement learning and 3D physical environments applied to crowd evacuation in congested scenarios," *Int. J. Digit. Earth*, vol. 16, no. 3, pp. 1-20, Mar. 2023, doi: 10.1080/17538947.2023.2182376.

[3] K. R. Rozo et al., "Modelling building emergency evacuation plans considering the dynamic behaviour of pedestrians using agent-based simulation," *Safety Sci.*, vol. 113, pp. 276-284, 2019.

[4] H. Xu, C. Tian, and Y. Li, "Emergency evacuation simulation and optimization for a complex rail transit station: a perspective of promoting transportation safety," *J. Adv. Transp.*, 2020.

[5] M.Haghani, "Optimising crowd evacuations: Mathematical, architectural and behavioural approaches," *Safety Sci.*, vol. 128, 104745, 2020.

[6] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Adv. Eng. Softw.*, vol. 95, pp. 51-67, 2016.

[7] H. Sharbini, R. Sallehuddin, and H. Haron, "Crowd evacuation simulation model with soft computing optimization techniques: a systematic literature review," *J. Manage. Analytics*, vol. 8, no. 3, pp. 443-485, 2021.

[8] Q. Jin, Z. Xu, and W. Cai, "An improved whale optimization algorithm with random evolution and special reinforcement dual-operation strategy collaboration," *Symmetry*, vol. 13, no. 2, 238, 2021.

[9] D. Helbing, I. Farkas, and T. Vicsek, "Simulating dynamical features of escape panic," *Nature*, vol. 407, no. 6803, pp. 487-490, 2000.

[10] S. Hassanpour and A. A. Rassafi, "Agent-based simulation for pedestrian evacuation behaviour using the affordance concept," *KSCE J. Civ. Eng.*, vol. 25, no. 4, pp. 1433-1445, 2021.

[11] M. Junaid, W. H. Bangyal, and J. Ahmad, "A novel bat algorithm using sobol sequence for the initialization of population," in 2020 IEEE 23rd Int. Multitopic Conf. (INMIC), *IEEE*, 2020.

[12] C.-Y. Lee and G.-L. Zhuo, "A hybrid whale optimization algorithm for global optimization," *Mathematics*, vol. 9, no. 13, 1477, 2021.

[13] N. W. Hidayat and F. Budiman, "Whale Optimization Algorithm Bat Chaotic Map Multi Frekuensi for Finding Optimum Value," *J. Appl. Intell. Syst.*, vol. 5, no. 2, pp. 80-90, 2020.

[14] H. M. Mohammed, S. U. Umar, and T. A. Rashid, "A systematic and meta-analysis survey of whale optimization algorithm," *Comput. Intell. Neurosci.*, 2019.

[15] M. Zhong and W. Long, "Whale optimization algorithm with nonlinear control parameter," in MATEC Web of Conferences, *EDP Sciences*, 2017.

[16] X. S. Yang, "A New Metaheuristic Bat-Inspired Algorithm," in *Nature Inspired Cooperative Strategies for Optimization* (NICSO 2010).

[17] G. Li and Y. Deng, "Crowd Dynamics Analysis Using a Bio-inspired Model," in Proc. *IEEE Int. Power Electron.* Control Eng. Conf. (IPEC), 2021, Art. no. 9421275.

[18] J. Xue, H.-C. Yang, M. Zhang, Z. Wang, and L. Shi, "Crowd Evacuation Conflicts Simulation Based Cellular Automaton Integrating Game Theory," in Proc. *ACM Int.* Conf. Proc. Ser., Dec. 2022, Art. no. 3574445.

[19] L. Li and B. Wang, "Simulation Study on Room Crowd Evacuation Under Panic Emotion Infection," *Adv. Transdiscipl. Eng.*, vol. 17, pp. 101-108, Feb. 2022.

[20] N. Khamis, H. Selamat, F. S. Ismail, and O. F. Lutfy, "Optimal Exit Configuration of Factory Layout for a Safer Emergency Evacuation using Crowd Simulation Model and Multi-Objective Artificial Bee Colony Optimization," *Int. J. Integr. Eng.*, vol. 11, no. 4, pp. 228-237, Sep. 2019.

[21] N. Khamis, H. Selamat, F. S. Ismail, and O. F. Lutfy, "Optimal Exit Configuration of Factory Layout for a Safer Emergency Evacuation using Crowd Simulation Model and Multi-Objective Artificial Bee Colony Optimization," *Int. J. Integr. Eng.*, vol. 11, no. 4, pp. 228-237, Sep. 2019. DOI: 10.30880/ijie.2019.11.04.020.

[22] M. Zhou, H.-r. Dong, X. Wang, X. Hu, and S. Ge, "Modeling and Simulation of Crowd Evacuation With Signs at Subway Platform: A Case Study of Beijing Subway Stations," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 2, pp. 1254-1265, Feb. 2022. DOI: 10.1109/tits.2020.3027542.

[23] H. Wu, C. Wang, and G. He, "Dynamic leader role modeling for self-organizing crowd evacuation simulation," Comput. Animat. Virtual Worlds, vol. 33, no. 4-5, Art. no. e2115, Aug. 2022. DOI: 10.1002/cav.2115. Tongtong Yu et al. "Modeling and Simulation of Evacuation Based on Bat Algorithm," 2019 IOP Conf. Ser.: *Earth Environ. Sci.* 267 032017.

[24] L. Chen, Q. Zheng, K. Li, Q. Li, and J. Zhang, "Emergency evacuation from multi-exits rooms in the presence of obstacles," *Phys. Scr.*, vol. 96, no. 8, Art. no. 085205, 2021. DOI: 10.1088/1402-4896/ac119a.

[25] H. Xu et al., "The effect of moving obstacle on regulation of pedestrian flow in a single exit room," *J. Stat. Mech. Theory Exp.*, vol. 2022, no. 2, Art. no. 023404, Feb. 2022. DOI: 10.1088/1742-5468/ac4c3f.

[26] Z.-H. Chen, Z.-X. Wu, and J.-Y. Guan, "Twofold effect of self-interest in pedestrian room evacuation," *Phys. Rev. E*, vol. 103, no. 6, Art. no. 062305, Jun. 2021. DOI: 10.1103/PhysRevE.103.062305.

[27] H. Yue, J. Zhang, W. Chen, X. Wu, X. Zhang, and C. Shao, "Simulation of the influence of spatial obstacles on evacuation pedestrian flow in walking facilities," *Physica A: Statistical Mechanics and its Applications*, 2021. doi: 10.1016/j.physa.2021.125844.

[28] J. Yu, D. Lu, Y. Niu, R. Mu, Y. Zhou, and Y. Li, "Emergency Evacuation Guidance Optimization in Urban Open Public Spaces," in Proc. *IEEE Geoinformatics*, 2023, Art. no. 10247774. DOI: 10.1109/Geoinformatics60313.2023.10247774.

[29] H. Chu, J. Yu, J. Wen, M. Yi, and Y. Chen, "Emergency Evacuation Simulation and Management Optimization in Urban Residential Communities," *Sustainability*, vol. 11, no. 3, Feb. 2019, Art. no. 795. DOI: 10.3390/SU11030795.

[30] S. Hamizan, S. Roselina, H. Habibollah, Y. Yusliza, and M. Y. Lizawati, "Crowd Emergency Evacuation Simulation Time Analysis via Obstacle Optimization Strategy," in Proc. *J. Phys.*: Conf. Ser., vol. 2129, no. 1, Dec. 2021, Art. no. 012045. DOI: 10.1088/1742-6596/2129/1/012045.

[31] J. Hu, Z. Li, L. You, H. Zhang, J. Wei, and M. Li, "Simulation of queuing time in crowd evacuation by discrete time loss queuing method," *Int. J. Mod. Phys. C*, vol. 30, no. 8, Aug. 2019, Art. no. 1950057. DOI: 10.1142/S0129183119500578.

[32] B. Liang, K. Xie, and X. Dong, "Crowd evacuation simulation for walking-along-side effect in the Stadium," in Proc. *MATEC Web Conf.*, vol. 309, 2020, Art. no. 05001. DOI: 10.1051/matecconf/202030905001.

[33] Y. Kawai, "Agent-Based Tsunami Crowd Evacuation Simulation for Analysis of Evacuation Start Time and Disaster Rate in Zushi City," in *Advances in Intelligent Systems and Computing*, vol. 1303, 2021. DOI: 10.1007/978-3-031-04170-9_5.

[34] H. Xu, C. Tian, and Y. Li, "Emergency Evacuation Simulation and Optimization for a Complex Rail Transit Station: A Perspective of Promoting Transportation Safety," *Journal of Advanced Transportation*, 2020. pp. 1-12. DOI: 10.1155/2020/8791503.