Information Sciences 436–437 (2018) 247–267

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Crowd evacuation simulation approach based on navigation knowledge and two-layer control mechanism

Hong Liu[a,b], Baoxi Liu[a,b], Hao Zhang[a,b], Liang Li[a,b], Xin Qin[a,b], Guijuan Zhang[a,b,*]

[a] School of Information Science and Engineering, Shandong Normal University, Jinan, China
[b] Shandong Provincial Key Laboratory for Novel Distributed Computer Software Technology, Jinan, China

ARTICLE INFO

Article history:
Received 10 November 2016
Revised 6 January 2018
Accepted 13 January 2018
Available online 30 January 2018

ABSTRACT

This paper presents a crowd evacuation simulation approach that is based on navigation knowledge and two-layer control mechanism. In this approach, using the multi-population cultural algorithm framework, the control mechanism of the crowd evacuation simulation is divided into two parts, namely, the belief and population spaces. The population space is divided into groups (sub-populations), and a leader is selected in each group according to a fitness value. The belief space comprises multiple agents and a knowledge base. Each navigation agent corresponds to a group leader. A navigation agent obtains a leader's position through the acceptance function and later passes the information to the knowledge base. On the basis of the position, the obstacles, and the congestion situation provided by the navigation agent, the knowledge base management agent dynamically plans the path and provides the navigation agent the next position along the path. The navigation agent later passes the information to the leader through the affection function. The individuals in the group follow the leader through the social force model in moving to the location provided by the navigation agent. The entire process is repeated until the exit is reached. The path information that successfully reached the exit is recorded, and the knowledge base is updated. This method establishes the relationship between the population and the navigation agent with knowledge and transforms a blindly moving crowd into a guided evacuation as the mass evacuation simulation problem is decomposed into a sub-problem of moving blocks. This approach effectively solves the problem of microscopic models because each individual calculates the path and resolves the slow speed problem. The simulation results illustrate the effectiveness of this method.

1. Introduction

In past decades, pedestrian evacuation has become an important social issue. Increasing attention has been provided for the safety of people's lives during evacuations. In the morning and evening rush hours, the venues or facilities of major festivals of large gatherings, sporting events, or other cultural and public activities often have dense crowds. When the crowd intensity in the facilities is high, the pedestrian flow becomes mixed, and a small disturbance could cause crowd destabilization. If there were not timely and effective control, the behavior of a crowd could easily lead to trampling accidents.

* Corresponding author at: School of Information Science and Engineering, Shandong Normal University, No. 88, East Wenhua Road, Jinan, China.
E-mail address: zhangguijuan@sdnu.edu.cn (G. Zhang).

https://doi.org/10.1016/j.ins.2018.01.023

Crowds have a wide variety of characteristics; thus, they exhibit different behaviors in public areas during an emergency. Modeling the behavioral variety during an emergency evacuation and the management of crowds is a significant challenge.

Currently, experiments and simulation modeling are the primary methods for studying evacuations. Many evacuation models have been developed by researchers, such as the social force model [14,15], cellular automata model [21], fluid dynamics model [13], fuzzy logic-based model [49], and agent-based model [42].

These models have been extensively used for crowd evacuation simulations. However, given that these models are from the perspective of an individual, planning a path for each individual is necessary, and the amount of computation is excessively large, which thereby makes the evacuation simulation of large-scale crowds difficult.

Crowd evacuation from large and complex building spaces is usually slowed down by insufficient knowledge of the internal connectivity of the rooms inside the buildings. In such circumstances, individuals can be unaware of all suitable paths for evacuation. Building occupants usually decide to use familiar exits, which are often the building entrances [10], and they tend to ignore emergency exits or paths that are not normally used for circulation. Therefore, navigation with knowledge during pre-planning can be especially useful for reducing possible damage, training responders, raising public awareness, and managing crowds in emergencies.

This paper proposes the navigation strategy with knowledge and the simulation method based on a two-layer control mechanism. The primary contributions of this paper are as follows:

(1) The multi-population cultural algorithm and social force model are combined, and the interaction between belief and population spaces is established using the multi-population cultural algorithm framework. The belief space comprises multiple agents and a knowledge base, which contains the shortest paths to exits and the congestion degree of each exit compared with others. In this manner, the relationship between the population and the navigation agents with knowledge is established, which thereby effectively guides the crowd evacuation.
(2) The simulation problem of large-scale crowd evacuation is divided into a group of sub-problems. Each group moves toward exits only through the guidance of the navigation agent and the leader, while group members follow the leader. Given that navigation agents do not access the knowledge base simultaneously, the extremely slow calculation speed problem for each individual planning path is solved.
(3) Evacuation by grouping the population and using guidance by navigation agents can effectively improve the utilization rate of the evacuation channels in public places and the security of people during emergencies. This method is suitable for designing evacuation plans and evacuation management systems.

The remainder of this paper is organized as follows. Section 2 introduces the related literature on cultural algorithms, agent-based models and social force model. In Section 3, the proposed two-layer control mechanism framework is elaborated. Section 4 defines agents and knowledge in the belief space in detail. Section 5 presents the improved social force model for crowd evacuation simulations in the population space. Section 6 illustrates the simulation of experiments to show the efficiency of the proposed method. The study's conclusions and several future research recommendations are presented in Section 7.

2. Related research

2.1. Cultural algorithm

The cultural algorithm (CA) was initially presented by Reynolds [39] in 1994. This model was inspired by human sociology and was developed to model the evolving cultural components of an evolutionary computational system over time because it accumulates experience. CA simulates social and cultural changes. The algorithm is divided into two parts: population and belief spaces. These two spaces evolve respectively and communicate with each other through specific protocols. The algorithm realizes the extraction and management of the knowledge of the population space through the belief space and guides the evolution direction of the population. This double-layered structure and evolutionary mechanism have been successfully applied in many fields of research. However, CAs still suffer from immature convergence. One of the most common strategies to overcome this limitation is to incorporate multiple populations. Digalakis and Margaritis proposed a Multi-Population Culture Algorithm (MPCA) [5] (in the following, we will use MPCA to represent Multi-Population Culture Algorithm) and applied it to complex combinatorial optimization problems; it is proven that different evolutionary behaviors among populations as well as information exchange and collaboration in the populations are divided into a number of sub-populations, and each sub-population is directed by a local CA. The local CAs communicate with each other by migrating their extracted knowledge [30].

Hlynka and Kobti summarized the features of MPCA [16]. In MPCA, different populations evolve independently and at the same time, while their information sharing and delivery affect each other. This circumstance can allow competitive, co-operative, or independent forms of CA to be run in parallel for the same problem domain. MPCA has attracted the attention of many researchers. Raeesi and Kobti [37] proposed a heterogeneous architecture for MPCA. In this architecture, one belief space is shared among all local CAs. Local CAs do not communicate with each other directly. In this architecture, a shared belief space is considered to record the best parameters. Local CAs send their best partial solutions to the belief space every generation. The shared belief space is responsible for keeping track of the best parameters sent by local CAs. This structure helps to fuse the knowledge that comes from local CAs in the belief space and obtain the globally optimal solution.

Consequently, MPCA can provide an explicit mechanism for global knowledge sharing and a useful algorithm framework that is used to combine different intelligent algorithms, especially for addressing multi-population problems.

Nguyen and Yao [34] proposed a hybrid framework that consists of CAs and iterated local search, in which they used a shared knowledge space that was responsible for integrating the knowledge from multi-populations. Knowledge migration in this context was used to guide new directions with a lower communication cost. Ali et al. [1] used a hybrid CA framework with a trajectory-based search to solve the global numerical optimization problem. Mohammad and Kobti introduced a hybrid approach to extract useful knowledge from a genetic algorithm in the population space for the solution of job shop scheduling problems [31].

Hlynka and Kobti [17] presented a method for knowledge transfer in MPCA through agent migration. This method did not extract knowledge from each sub-population to update the global trust space itself, but instead exchanged with each other in such a way that each sub-population and exotic individuals were introduced to one another. In this way, the transfer agent knows the solution and communicates with other agents. This approach is beneficial to obtaining the globally optimal solution.

The work presented in [12] investigated the problem of hybrid coordination in multi-agent networks with hierarchical leaders. In the case of movable main leaders with hierarchical delays, the condition provided to the multi-agent network allowed deriving a desired group behavior. Guo et al. proposed a multi-population cooperative cultural algorithm that combined a cultural algorithm with a competition co-evolution model [9].

Zadeh and Kobti proposed an MPCA to solve community detection in social networks [48]. In their algorithm, the individuals of each sub-population were selected according to a fitness function, and the selected individuals updated their belief space. Knowledge was extracted from the belief space to guide the search direction.

As an ideal framework, MPCA has been successfully combined with many models and algorithms to solve many problems. However, the literatures that take MPCA as a framework and combine multi-agent model and social force model to solve crowd evacuation problems are still limited. Thus, this topic is worthy of exploration.

2.2. Agent-based modeling

Agent-based modeling is an approach to modeling systems that are composed of individual, autonomous, interacting "agents." This model is based on the representation of global behavior from the rules provided to individuals, which enable them to view the macro-level consequences of micro-level interactions. Different sets of individuals correspond to various behaviors and attributes that are relative to the environment. We can envision how individuals will interact with the environment and with other individuals in specific periods [29]. Each agent individually assesses the situation and makes an evacuation decision based on a set of rules. Agent-based modeling has been demonstrated to provide insights that are unavailable from other methods and to capture the dynamics of natural and human systems [4].

Shi et al. [40] adopted an agent-based model to analyze the exit progress in a large indoor stadium. This model combined rule reasoning with numerical calculations and crowd pedestrian flow phenomena. Wagner and Agrawal [44] presented an agent-based crowd evacuation simulation system to simulate crowd evacuation in fire disaster cases and studied the evacuation performance under multiple disaster scenarios. Each of these agents was a separate individual and showed a discrete state, not forming a group.

In reference [35], Pan et al. introduced a multi-agent based framework for simulating human and social behavior in emergency evacuation processes. In addition to completing the basic steering behavior, the system presented a series of interesting guidance behaviors, including random walks (to determine the evacuation before walking in the virtual scene); seeking behavior (to determine the target point); and negotiation behavior (this behavior control allows different agents to exchange information and reach an agreement between them).

Iizuka and Iizuka presented a disaster evacuation assistance system based on multi-agent cooperation [18]. Each pedestrian in the scene was viewed as an independent agent. Through the judgment and prediction of the evacuation scene, the system provided evacuation guidance for pedestrians in the scene via ad-hoc by mobile devices, which included the time of evacuation and the route of evacuation. The system required only mobile devices as the platform without any central server. The results of the paper's experiment showed that the guidance from this system could decease the evacuation time.

In the master's thesis of Ha [11], an agent-based modeling of emergency building evacuation was introduced. This model considered each pedestrian as a particle without structure whose motion was governed by Newton's equations. This work proposed the use of the social-force model to study the evacuation of buildings with complicated floor plans to explore how the complexity of the building design affects the overall evacuation process.

Fu et al. proposed a combination of cellular automata and multi-agent technology to conduct a simulation study of crowd behaviors [8]. Every cell in cellular space was considered as one agent, and its status was encapsulated with a cell as the type and state property of the agent. This approach designed a perceptual model and decision model of the agents and tested the validity of the models.

Liu et al. [27] proposed a dynamic route decision model based on multi-agents by considering group evacuation. Nagarajan et al. [33] suggested an agent-based simulation model in a hypothetical community to study the influence of behavior on warning dissemination. Similarly, Dawson et al. [4] employed a dynamic agent-based model to manage flood incidents. Agent-based modeling focuses on the behavior of evacuees and incorporates dynamic travel costs when deciding the speed of travel and the locations of the evacuees.

Agent-based models usually require more computation because they can model the heterogeneity of pedestrian behavior. The complexity of the model depends on the set of rules that govern the movement of each agent. The tradeoff of this flexibility is the high cost of running the model. The development of agent-based models still faces many challenges. Including the abovementioned literature, the existing research on the multi-agent model of crowd evacuation simulation has laid the foundation for the research in this paper.

2.3. Social force model

Crowd motion in the evacuation process is a very complex physical process. Its dynamic adjustments are directly or indirectly restricted by many factors. The interactions among the crowd and the psychological states of the individuals are important factors that influence a crowd's motion.

The social force model (SFM) was developed by Helbing and Molnar [15] and Helbing et al. [14]. This model solves Newton's equation to determine the position of each pedestrian by considering exclusive interactions, friction forces, dissipation, and fluctuations. In this model, pedestrians have a desired velocity in the direction of their destination, and their acceleration (deceleration) is the result of different forces. Individuals experience forces toward the direction of their target destination and exclusive forces from obstacles (e.g., walls) and other pedestrians. Time and space are modeled in a continuous manner. The model reproduces well-known, self-organizing crowd phenomena, such as lane formation in bidirectional flows and oscillatory effects at bottlenecks.

This model was applied and generalized further to simulate other scenarios, such as densely populated crowds [36], pedestrian evolution [24], and escape panic. Given that the model successfully simulates the dynamic characteristics of pedestrians, it has attracted the attention of many scholars who have attempted to improve the model [7,19,43,45].

Ji and Gao proposed a leader–follower model for a crowd evacuation simulation [19]. A crowd included several groups, and each group had a leader and several followers. Leaders were responsible for finding the evacuation path for their followers. The objective of their simulation was to determine the effect of different numbers of leaders on the efficiency of evacuation. Vihas et al. proposed a cellular automaton model for crowd movement simulation by embedding the follow-the-leader technique as its fundamental driving mechanism [43].

Crowds are rarely composed exclusively of unrelated and independently moving individuals. Rather, crowds often comprise many small social groups based on friendship or kinship [32]. The SFM was modified by Fisher [7], who introduced the "small group" phenomenon in the population. Under normal circumstances, both model and experience show that pedestrian crowds self-organize. No one orders people to act in a certain way. Pedestrian crowds, who come from multiple directions and interact only with others nearby, self-organize to pass through bottlenecks in the most efficient manner [41]. Even if the individuals in a crowd are not socially connected, the emergency experience may lead to a shared identity, which can connect them in a similar way to social groups [6]. Based on eyewitness reports, members of social groups, especially families, display strong affiliative behavior and stay close to one another even under severe environmental threats [23].

The leader–follower and self-organization related research above established the foundation for the formation of the groups in our evacuation simulation. During an evacuation, evacuation leaders who are familiar with the internal layout of the building can recognize the escape route to the exits and guide evacuees while significantly reducing the casualties. These actual evacuation experiences suggest that leadership could have a positive effect on crowd evacuation [3,28].

In our approach, the leader relies on the information provided by the navigation agent in the process of moving toward the exit and dynamically adjusts the route when faced with congestion and obstacles according to the knowledge interaction between the groups. The navigation agent provides the leader with relevant building information. This information enables the leader to identify the escape route toward the exits and guide evacuees, which is the main difference between the proposed method and the existing leader–follower model.

3. Framework of the two-layer control mechanism

3.1. Description of the crowd evacuation problem

It is a rather common phenomenon that pedestrians are unfamiliar with the area where they are and insensible of the positions of exits, especially when they are too frightened to remain clearheaded in emergencies [46]. At present, the role of a guide who rescues them from emergencies is very important for frightened people. With the improving awareness of safety, many inspection staff members who can act as guides in an escape are arranged to be within a certain range of the large-scale building to be responsible for safety inspection.

Because the evacuation process is dynamic, rescue workers understand how timely the congestion is at the exits and the obstacles and choose a reasonable balance between distance and congestion to lead the pedestrians as soon as possible to evacuate the danger zone, which is a very important problem.

The multi-population cultural algorithm provides a framework for knowledge-based evacuation navigation. The framework combines multi-agent technology and the social force model to provide support for dynamic knowledge acquisition and the instruction evacuation process.

3.2. Formal description of the scheme of the two-layer control mechanism

The scheme of the two-layer control mechanism has the ability to incorporate heterogeneous and diverse knowledge sources into its structures. As such, they are ideal frameworks within to support hybrid amalgams of knowledge sources and population components.

Inspired by the framework of MPCA, the two-layer control mechanism framework has two parts, namely, the belief and population spaces.

Definition 3.1 (Belief space)**.** The belief space consists of multiple agents and a knowledge base. The knowledge base contains five types of knowledge: situational, normative, topographical, domain and historical knowledge.

The definitions and detailed descriptions of the agents are covered in Section 4.1.

Definition 3.2 (Situational knowledge (SK))**.** Situational knowledge consists of a set of cases that help the individuals move toward exits. In our evacuation scene, situational knowledge is the current location of each leader and the crowd flow information.

Definition 3.3 (Normative knowledge (NK))**.** This knowledge source stores performance standards and guidelines that can lead the individuals into more promising regions. Normative knowledge is responsible for keeping individuals in or moving to better regions by saving the acceptable behavior of promising individuals and their ranges.

Definition 3.4 (Topographical knowledge (TK))**.** Topographical knowledge is the dynamic path planning for execution by the D* Lite algorithm according to the location of each leader and the congestion of the obstacles and the exits.

Definition 3.5 (Domain knowledge (DK))**.** Domain knowledge is characterized by the domain ranges of all parameters and the best examples from the population along with any constraints on their relationships.

Definition 3.6 (Historical knowledge)**.** Historical knowledge is the path record and running time after each evacuation.

For a detailed description of the knowledge, see Section 4.2.

Definition 3.7 (Population space)**.** The population space consists of evacuated individuals. These individuals are divided into groups (sub-populations) according to their physical locations.

Each group selects a leader based on the highest degree of fitness. Each leader corresponds to a navigation agent in the belief space. The leader has access to path planning knowledge in real time through the interaction with the navigation agent and leads the individuals and the group toward the exit.

The communication protocol consists of an acceptance function and an influence function. Each leader and corresponding navigation agent interacts by the accepted function and the influence function.

Definition 3.8 (Acceptance function)**.** Overall, the acceptance function is in charge of determining how much new experiential information from the population space will be provided to update the knowledge sources in the belief space. Here, we use the function Accepting (CurrentX, CurrentY, CurrentZ) to pass the current position of the leader to the navigation agent.

Definition 3.9 (Influence function)**.** Overall, the influence function determines how many individuals in the population will use a knowledge source to make changes. Here, we use function Influence (NextX, NextY, NextZ) to pass the leader's next location from the navigation agent to the leader.

The parameters in the function Accepting (CurrentX, CurrentY, CurrentZ) and the function Influence (NextX, NextY, NextZ) represent the 3D coordinate information of the current and the next positions, respectively. X and Y are the planar grid positions, and Z is the location of the floor.

3.3. Knowledge-based crowd evacuation process

After a formal description is given, the use of the two-layer control mechanism based on MPCA to simulate the knowledge-based crowd evacuation process will be introduced step by step in this section.

The scheme of the two-layer control mechanism based on MPCA is shown in Fig. 1.

Knowledge-based crowd evacuation simulation process

Step 1: Initialization involves the parameters of the evacuated scene, and a counter is set up at each exit and a critical barrier, for a congestion calculation. The number of individuals, the locations of the individuals in the evacuation scene, and the link table of the relationships between the individuals are initialized.

Step 2: Individuals are grouped according to their relationships and their distances from the exits. A leader is selected from among the individuals in each group based on his/her fitness value.

Step 3: Based on the MPCA framework, the relevance between the belief and population spaces is established. The number of navigation agents in the belief space is determined according to the number of groups in the population space and the one-on-one relationship between the navigation agent and the leader.

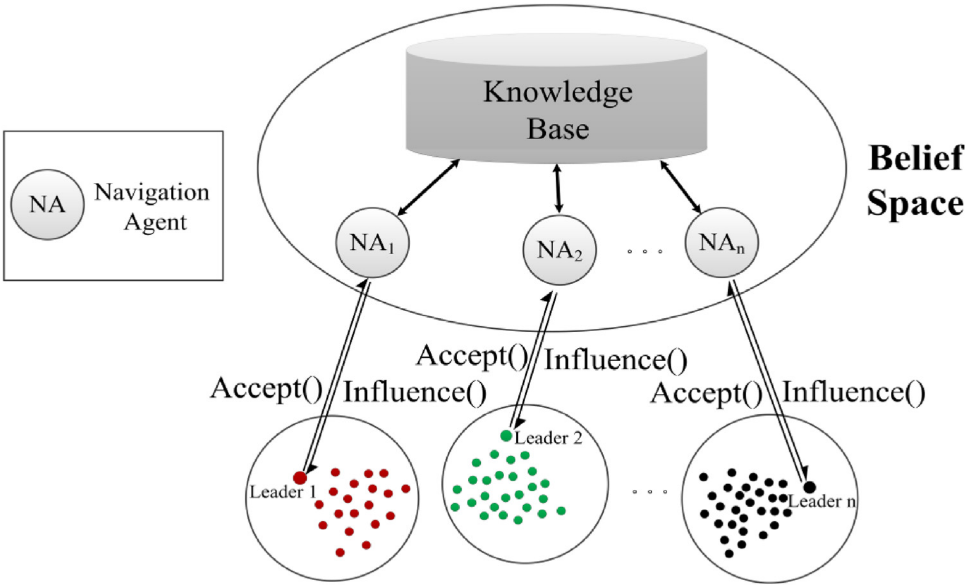


Fig. 1. Scheme of the two-layer control mechanism based on MPCA.

Step 4: The leader passes the current position on to the navigation agent by the function Accepting (CurrentX, CurrentY, CurrentZ). Then, the navigation agent interacts with the knowledge base, executes dynamic path planning algorithm D* Lite, and computes the shortest path from the current position to the exit. Subsequently, the next position of the leader is selected, and the information is passed to the leader by the function Influence (NextX, NextY, NextZ).
Step 5: The group proceeds to the next location provided by the navigation agent as the goal, and the group members execute the modified SFM to follow the leader. After the movement, the new position of the leader is returned to the navigation agent.
Step 6: If the group leader arrives at a safe exit area, then he/she stops at the gate until all of the individuals in the group are evacuated. The route, time, and number of individuals of this evacuation are recorded and sent to the knowledge base by the navigation agent. Then, the knowledge is updated.
Step 7: All of the groups are evacuated, and the evacuation process is completed.

In the definitions of 3.8, 3.9 and step 4 in the knowledge-based crowd evacuation simulation process, we define and use a three-dimensional model. The current location of the leader is (CurrentX, CurrentY, CurrentZ), and the next location of the leader is (NextX, NextY, NextZ). The purpose of the definitions is to enable our method to be used in high-rise buildings. However, in the simulation experiments in this paper, the evacuation of high-rise buildings has not been shown to date, only the experiment conducted on the first floor of the teaching building. In this case, CurrentZ and NextZ use a constant of 1.

4. Agent and knowledge

4.1. Agent

The majority of agents in the belief space are navigation agents. A navigation agent is a domain-dependent and semi-autonomous agent, whose behavior is driven by the achievement of goals and is very responsible. Navigation agents possess the following capabilities:

- Can guide the leader and check the constraints during the evacuation process;
- Can maintain and interpret knowledge related to oneself, other agents, and the belief space.

Definition 4.1 (Navigation agent)**.** A navigation agent can be defined by a six tuple ⟨Aid, Input, Communication, Output, Goal, Trigger⟩.

Aid is the identifier of a navigation agent, such as NA_1, NA_2, …, and NA_n.
Input refers to the input interface component, in which the position information of the leader is obtained from the function Accept (CurrentX, CurrentY, CurrentZ) and is then passed on to the knowledge base by the communication component. Input also derives the count information obtained from the counter, which is passed on to the knowledge base by the communication component.

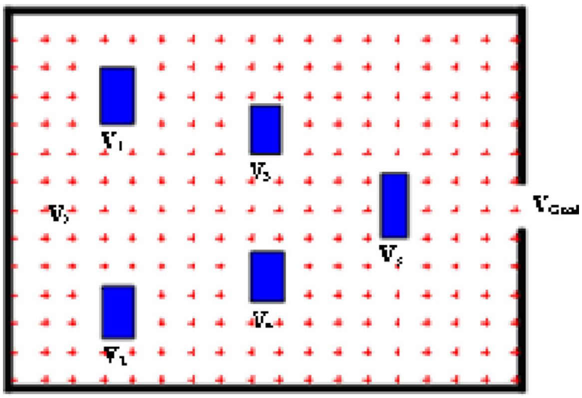


Fig. 2. Evacuation scenario with five obstacles and one exit.

Communication is the communication component that receives messages and passes the information from and to the knowledge base.

Output is the output interface component that returns the next position to the leader by the function Influence(NextX, NextY, NextZ).

Goal refers to the selected evacuation exit and comprises several constraints.

Trigger is the activity triggering component that consists of event-condition-action rules.

Furthermore, Trigger detects an event, determines whether crowd evacuation is necessary, and then activates the agent to perform the appropriate action [25].

Definition 4.2 (Knowledge base management agent)**.** A knowledge base management agent can be defined by a seven tuple: ⟨Aid, Input, Bulletin_Board, Transformer, Knowledge _Update, Output, Trigger⟩.

The current location of each leader and the crowd flow information through the input are passed to the knowledge management agent.

Bulletin_Board records the current position of the leader that corresponds to the navigation agent and the crowd flow information in real time [26].

Transformer is a knowledge-based transform component that executes the D* lite algorithm in the knowledge base to calculate the next position of the leader according to the current information provided by the Bulletin_Board.

The next position of the leader is passed to the navigation agent by the Output component.

After the evacuation is completed, the Knowledge_Update component updates the historical knowledge in the knowledge base.

Trigger is the activity triggering component that consists of event-condition-action rules. This component detects situations during an event that satisfy the condition for evacuation and then activates the agent to perform the appropriate action.

4.2. Navigation knowledge

The belief space contains five types of knowledge: situational, normative, topographical, domain and historical.

Situational, normative, topographical and domain knowledge can be used to obtain personal and global best positions. Individuals move according to the obtained personal and global best positions. Historical knowledge is the path record and running time after each evacuation and is adopted in the update operator to update the historical knowledge.

The goal of dynamic path planning is to help evacuees find optimal routes to reach safe exits. Therefore, the path selection criteria aim to ensure that evacuees avoid passing through congested zones and minimize the evacuation time. The D* Lite algorithm was proposed by Koening and Likhachev [22] in 2002 at the AAAI/IAAI conference. Our method is inspired by the D* Lite algorithm and takes the weighted distance plus crowd flow as a cost value to guide the dynamic path planning. This arrangement fully considers the crowded effect on the evacuation speed and is conducive for rapid evacuation and reducing the congestion.

Next, we introduce the dynamic path planning algorithm in detail.

Let the node be the obstacle in the scene, the exit be the target node, the current position of the individual be the starting node, the connection between two nodes be an edge, and the distance between nodes be the cost value. Thus, we can obtain the weighted directed graph G = (V, E, C), where V is the set of nodes, E is the set of edges, and C is the set of cost values. Fig. 5 shows an evacuation scenario with five obstacles and one exit. Fig. 3 presents the weighted directed graph based on the evacuation scene in Fig. 2.

Crowd congestion occurs mainly when obstacles are encountered and at exits. Thus, counters are established at the key obstacles and at the exits to reflect the congestion situation in real time by testing the flow of individuals. Accidents or disasters in the scene, such as someone falling, automatically become obstacles, and the cost of reaching the obstacles is defined as infinity for the subsequent individuals to escape from these obstacles.

During the process when individuals follow the leader to the exit, the exit remains the same, and the position of the leader is constantly changing. Therefore, the algorithm uses a reverse search that traverses backward from the exit.

The D* Lite algorithm maintains the minimum cost path between the nodes and the exit in the OPEN table, thereby saving two important values: (1) the minimum cost $g(i)$ from node v_i to the exit, and (2) the estimated value of $h(i)$ (see Eq. (4.1)).

$$h(i) = \begin{cases} 0 & If\ v_i\ is\ an\ exit \\ \min_{j\in Next(i)}\{g(j) + Cost(i,j)\} & Otherwise \end{cases}, \tag{4.1}$$

where $Next(i)$ is the subsequent node set of node v_i, and $Cost(i, j)$ is the cost from node v_i to node v_j.

$$Cost(i,j) = w_1 \times d(i,j) + w_2 \times f(i,j), \tag{4.2}$$

where w_1 and w_2 are the weight values, and $w_1 + w_2 = 1$. Without loss of generality, let $w_1=0.5$ and $w_2=0.5$. Here, $d(i, j)$ is the distance from node v_i to node v_j, and $f(i, j)$ is the congestion degree of node j.

$$f(i,j) = \frac{\text{the current value of the population flow through node } j}{\text{the average value of the population flow through node } j}. \tag{4.3}$$

When $f(i, j) > 2$ or when a disaster occurs near an obstacle j, let $Cost(i, j) = \infty$. The follow-up individuals will give up the obstacle that corresponds to node j and select the other path until $Cost(i, j)$ returns to normal.

Assuming that node v_i and node v_j are neighbors, if $h(i)=g(i)$, then node v_j is called a continuous node. Otherwise, node v_j is a non-continuous node. If the cost value of the path is changed, then all nodes on the path remain continuous. Then, after the final planning of the path and the path is still optimal, the cost value of the relevant sections is unaffected. Otherwise, the path should be modified.

Modified D* Lite algorithm

Step 1: Initialize the g and h values of the exit node: let the exit node be v_{goal}, $g(v_{goal}) = \infty$, $h(v_{goal}) = 0$; OPEN table is set to empty. The g and h values of v_{goal} are different; therefore, v_{goal} is a non-continuous node and is inserted into the OPEN table.
Step 2: Search for an optimal path as an initial path and recommend it to the leader. Group movement follows the leader along the initial path.
Step 3: If the leader arrives at the exit, then proceed to Step 7; otherwise, proceed to Step 4;
Step 4: The leader moves to the next position and updates it as the new starting position of the leader;
Step 5: The counter for population flow statistics is used to determine whether a change occurs in the road conditions and the nodes on the optimal path are non-continuous. If a non-continuous node is found, then proceed to Step 6; otherwise, proceed to Step 3.
Step 6: Update the cost value of the edges and place the non-continuous nodes into the OPEN table. Search nodes in the OPEN until the state of the v_{goal} is also changed to continuous, and then, the optimal path planning is obtained. Afterward, return to Step 3.
Step 7: The leader arrives at the exit, and the evacuation is completed.

4.3. Knowledge update

Rule 1: IF the evacuation path is the first run, THEN this path is added to the knowledge base.
Rule 2: IF the evacuation runs a path coming from the database, THEN the frequency of this path will be plus one.

The execution time and the size of the population being evacuated are recorded in the knowledge base together with this path and frequency. Afterward, the frequency, average time and evacuation number of paths in the database are comprehensively evaluated and sorted to reduce the dynamic programming adjustment.

4.4. Selection of leader

In this paper, each group in the population space will have a leader to guide the evacuation of the entire group. The leader in the group knows the most about where the exits and evacuation routes are located. The leader should have the following characteristics:

(1) Knowledge: each leader should be the individual who is most familiar with the situation and who understands the current position of the escape path in the scene;
(2) Location: when the scene is visible, the individual near the exit should be a priority;
(3) Evacuation process: in the evacuation process, the leader should walk in front of the group to play the leading role.

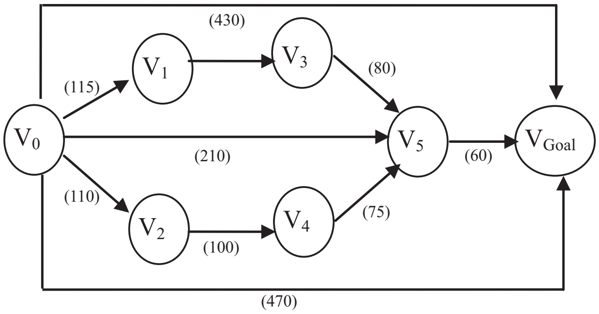


Fig. 3. Weighted directed graph based on the evacuation scene in Fig. 2.

Definition 4.1 (Fitness function)**.** The fitness function is used to select the leader for each group in the evacuation crowd. The individual with the highest fitness value is selected as a leader in the group.

According to the preceding basic principles, the fitness function $fit(x_{ij})$ is determined by Eq. (4.4):

$$\mathrm{fit}(x_{ij}) = w_1 \times \max_{x_{ij}\in Group\,i}\{k(x_{ij})\} + w_2 \times \min_{x_{ij}\in Group\,i}\{d(x_{ij})\}, \tag{4.4}$$

where w_1 and w_2 are the weight values, and $w_1 + w_2 = 1$. Without loss of generality, let $w_1 = 0.5$ and $w_2 = 0.5$. $k(x_{ij})$ is the degree of familiarity of scene x_{ij}, $d(x_{ij})$ is the distance from the nearest exit of x_{ij}, and x_{ij} is the individual j in group i.

If there are multiple individuals with the highest fitness value in the same group, x_{ij} with the highest $k(x_{ij})$ value is chosen to be the leader.

5. Improved social force model

According to the characteristics of collective behavior, Helbing established the social force model (SFM) based on Newtonian mechanics [15]. The social force refers to the force that one individual obtains from the environment (including humans and objects), whereas the physical force is the force that is directly applied to the individual. Based on the different motivations of pedestrians and impacts from the environment, SFM has four forces: (1) the driving force, (2) the interaction force between human beings, (3) the interaction force between individuals and obstacles, and (4) the disturbing force.

The resultant force of the three forces impacts pedestrians and contributes to the acceleration. The internal driving force guides the individual to move toward the target. However, before body contact, the exclusive force becomes involved in preventing individuals from colliding with one another. The interaction force between people and obstacles prevents individuals from colliding with obstacles. This stage can be interpreted by Newton's second law. The expression is shown in Eq. (5.1):

$$m_i \frac{d\vec{v}_i(t)}{dt} = \vec{f}_i^0 + \sum_{j(\neq i)} \vec{f}_{ij} + \sum_{w} \vec{f}_{iw} + \vec{\xi}(t), \tag{5.1}$$

where m_i is the mass of pedestrian i, and $v_i^{\rightarrow}(t)$ is the actual walking velocity. Eq. (5.1) shows that the motion of pedestrian i is affected by four types of forces, which include the pedestrian's driving force $f_i^{0\rightarrow}$, the interaction force between pedestrian i and the other pedestrians $\sum_{j(\neq i)} f_{ij}^{\rightarrow}$, the interaction force between pedestrian i and obstacles $\sum_w f_{iw}^{\rightarrow}$, and the disturbing force $\xi(t)^{\rightarrow}$. The position of pedestrian i changes under the interactions of the four forces. In this equation, m_i is the mass, and $v_i^{\rightarrow}$ is the actual velocity of pedestrian i. Fig. 4 shows the direction of each force in Eq. (5.1).

$$\vec{f}_i^0 = m_i \frac{v_i^0(t)\,\vec{e}_i^0(t) - \vec{v}_i(t)}{\tau_i}. \tag{5.2}$$

Eq. (5.2) describes the driving force $\vec{f}_i^0$ of pedestrian i. In the moving process, pedestrian i constantly adjusts his actual velocity $\vec{v}_i(t)$ and aims to move toward the destination at a desired speed $v_i^0(t)$. In this equation, τ_i is the characteristic time of pedestrian i, and $\vec{e}_i^0(t)$ is the direction from pedestrian i to the destination. The forces that correspond to Eq. (5.2) are shown in Fig. 3, where

$$\vec{v}_i(t) = \frac{d\vec{r}_i}{dt}. \tag{5.3}$$

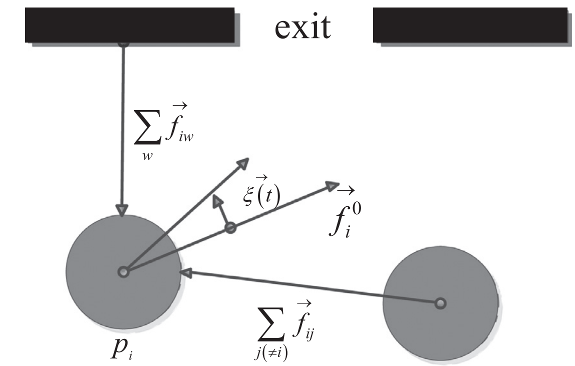


Fig. 4. Direction of each force in Eq. (5.1).

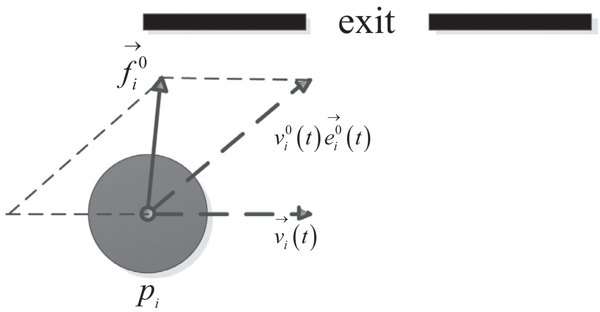


Fig. 5. Forces that correspond to Eq. (5.2).

The detailed description for the SFM parameters can be viewed in reference [15]. In this paper, the leader links the navigation agent and the group, and the social force model is improved as

$$m_i \frac{d\vec{v}_i(t)}{dt} = \vec{f}_{il}^0 + \sum_{j(\neq i)} \vec{f}_{ij} + \sum_w \vec{f}_{iw} + \vec{\xi}(t). \tag{5.4}$$

The acceleration equation is similar to that in the basic SFM, whereas $\vec{f}_{il}^0$ is the bond between the individuals and the leader within the group. This circumstance is expressed by Eq. (5.5):

$$\vec{f}_{il}^0 = m_i \frac{v_{il}^0(t)\,\vec{e}_{il}^0 - \vec{v}_i(t)}{\tau_i}. \tag{5.5}$$

The difference between Eqs. (5.2) and (5.5) is that $\vec{e}_i^0(t)$ in Eq. (5.2) is the direction that points from individual i to the destination, whereas $\vec{e}_{il}^0$ points to the position of the leader in Eq. (5.5), as shown in Fig. 6.

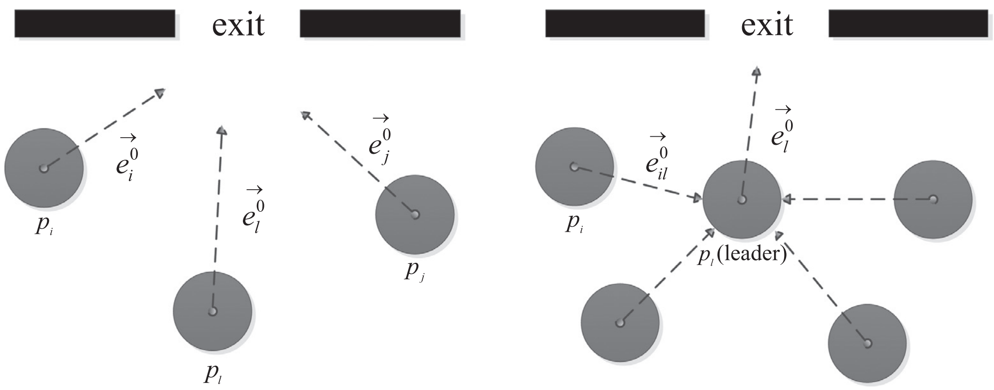
While the group is moving, the leader directs the group and acts as the authority at the gate. When the leader moves to the exit, he/she should stay in the exit and continue to lead the group until all of the individuals are evacuated.

6. Simulation experiments

We design the experiments according to the three contributions of this paper (see Section 1).

(1) The experiments in Section 6.1 show that our model can guide crowd evacuation effectively by combining the MPCA and social force model. The effectiveness can be reflected by the evacuation time in two different scenarios: a scenario without obstacles and a scenario with obstacles.
(2) The experiments in Section 6.2 show that the computation time for the crowd navigation is reduced since only the group leader is required to access the knowledge base, and the group members follow the leader directly.
(3) The experiments in Section 6.3 show that our model can improve the utilization of the evacuation routes/exits in public places. The knowledge of the group leader allows people to avoid the crowded places in the route selection. As a result, the evacuation time is saved, while the security of people during emergencies is guaranteed.

The simulation results are demonstrated on our crowd evacuation simulation platform intuitively in Section 6.4.



(a) $\vec{e}_i^0(t)$ in Eq.(5.2) is the direction pointing from the individual i to the exit

(b) $\vec{e}_i^0(t)$ in Eq.(5.5) is the direction pointing from the individual i to the leader

Fig. 6. Difference between Eqs. (5.2) and (5.5).

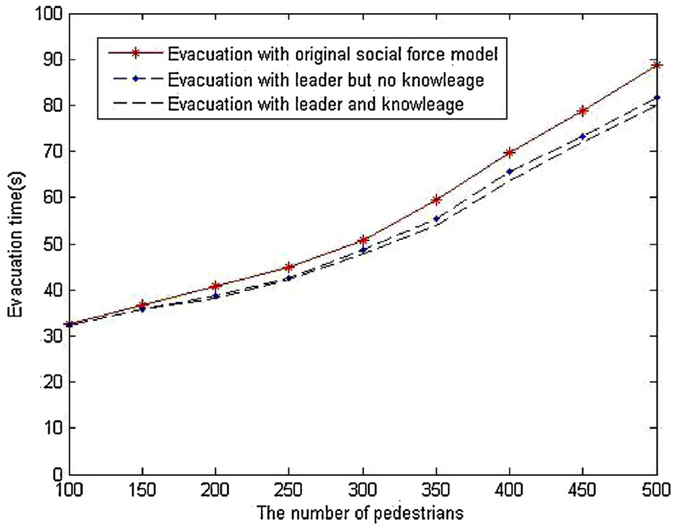


Fig. 7. Evacuation times of the three models in the scenario without obstacles: two exits are used.

6.1. Evacuation time of our model

We use the evacuation time to evaluate the effectiveness of our model in guiding the crowd evacuation. Our method saves the evacuation time because the knowledge base and navigation agents guide the crowd to select paths to the exit that are shorter and that have a lower degree of congestion.

In this section, we compare the results of three models in two different scenarios to verify the role of the knowledge base and navigation agents. The three models are the following: the original SFM, the social model with the leader but with no knowledge, and our model (i.e., the social model with the leader that has knowledge). We performed 50 experiments to obtain the average evacuation time for each case.

(1) Results in a scenario without obstacles

In this example, we test the evacuation time in an empty room without obstacles. The population size is varied from 100 to 500. We obtained two groups of experimental results, as shown in Figs. 7 and 8, where two exits of the room are used in Fig. 7 and four exits are used in Fig. 8. The two figures show that when the number of individuals is lower, no congestion occurs, and the evacuation time gap is not obvious. As the number of individuals increases, the advantage of our model with the leader with knowledge gradually becomes evident. With an increase in the population size, the advantage becomes obvious in the rooms with multiple exits.

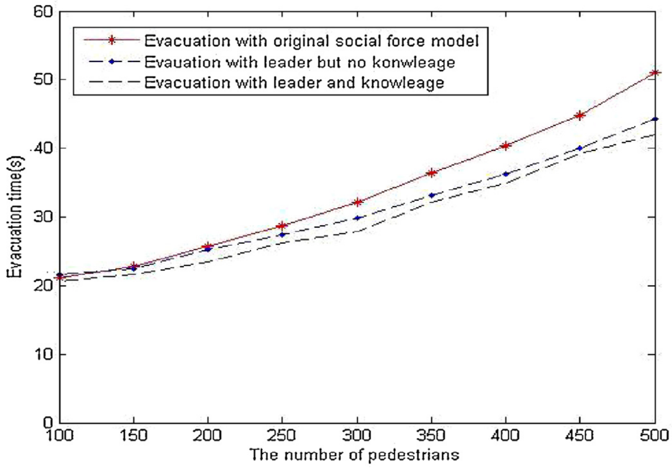


Fig. 8. Evacuation times of the three models in the scenario without obstacles: four exits are used.

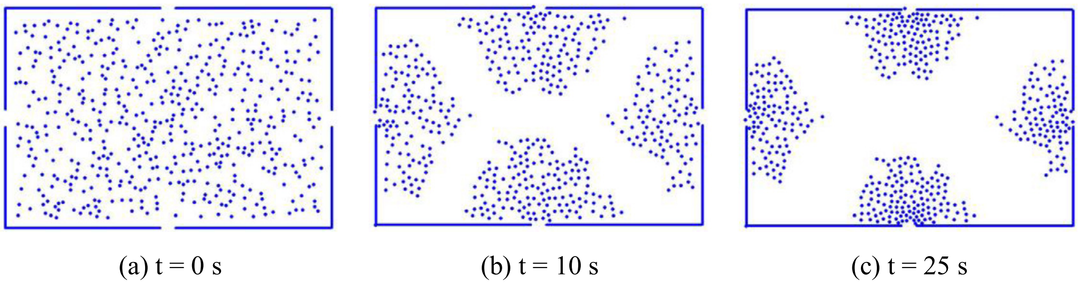


Fig. 9. Evacuation simulation of the original social force model.

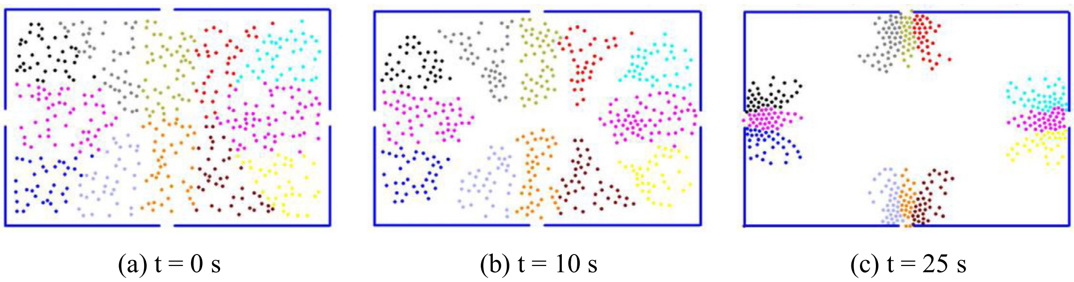


Fig. 10. Evacuation simulation of our approach. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

To reveal why our method can take less evacuation time intuitively, we also give a sequence of crowd evacuation simulation results, shown in Figs. 9 and 10. We used Matlab as the simulation platform in these two examples, where the population size is set to 500. In Fig. 9, the original SFM is used during the crowd evacuation simulation, while Fig. 10 is the result of our approach. The colors in Fig. 10 indicate the grouping results. Fig. 9 shows that individual congestion and exit become a bottleneck that affects the speed of evacuation. Fig. 10 shows that our model can relieve the bottleneck problem of the arch at the exit and effectively improve the efficiency of the evacuation. The above experimental results show that our model can reduce the evacuation time, especially when the size of the population is large in a scenario. With an increase in the exits, the effect becomes more apparent. It benefits from the knowledge base and navigation agents to guide the path finding process significantly.

(2) Results in a scenario with obstacles

We also perform several experiments in a scenario with obstacles. In this example, the population size varies from 50 to 400. The evacuation times of the three models in the scenario with only one exit (Fig. 2) is shown in Fig. 11 (Table 1). The social model with leaders and knowledge in a scene with obstacles performs better than the scene without obstacles. The

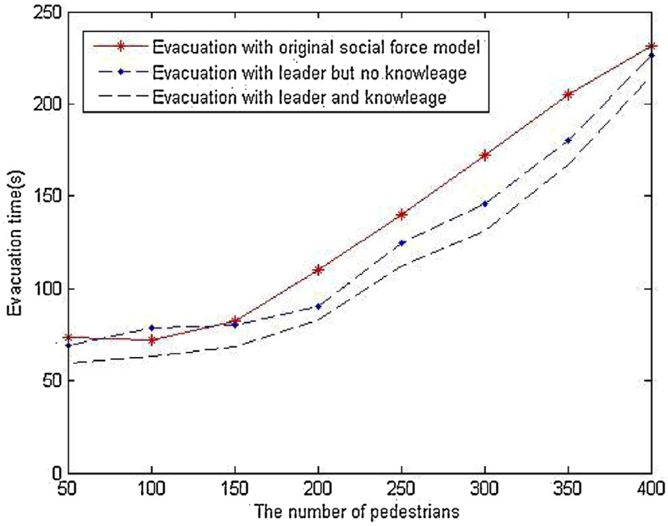
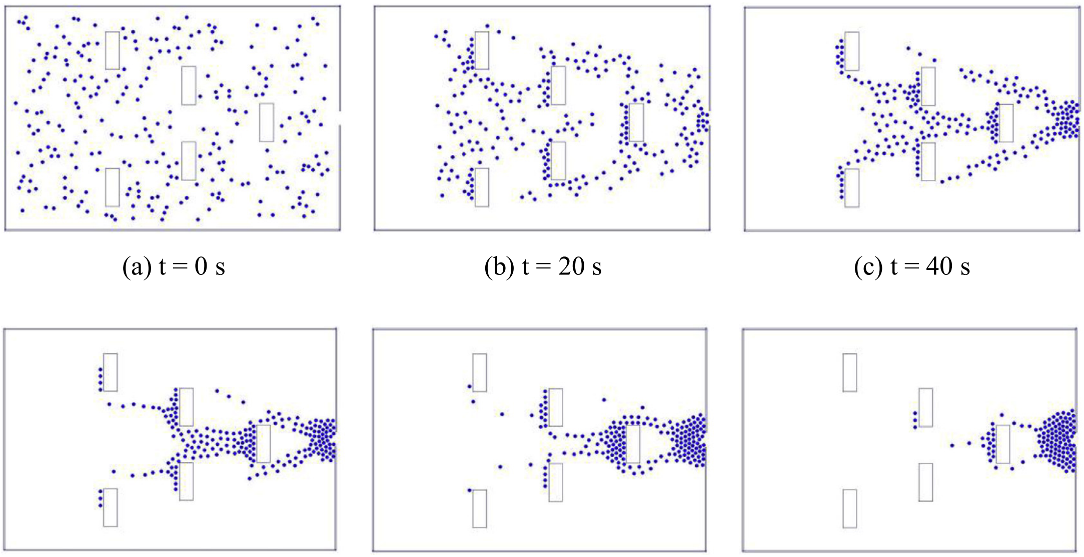


Fig. 11. Evacuation times of the three models in the scene with obstacles.

Table 1
Evacuation time (s) of the three models.

Number of individuals	50	100	150	200	250	300	350	400
Evacuation with original social force model	73.19	71.72	82.03	109.75	139.88	172.18	204.77	231.00
Evacuation with leader but no knowledge	68.82	78.43	80.35	90.42	124.55	145.93	180.28	226.11
Evacuation with leader and knowledge	59.33	63.27	68.23	83.21	112.22	130.99	167.00	215.25



(a) t = 0 s (b) t = 20 s (c) t = 40 s

(d) t = 60 s (e) t = 90 s (f) t = 120 s

Fig. 12. Evacuation simulation of the original social force model.

results also reflect that our method take less evacuation time because the group leader has knowledge about the shorter path.

Figs. 12 and 13 show the effects of the evacuation simulation. In these two examples, the population size is 250, and only one exit is used. In Fig. 12, the original SFM is used to plan the crowd evacuation routes, while our model is used to select the evacuation routes in Fig. 13.

Figs. 12(a) and (a) is the initial states of the random distribution of individuals, and the other states are the intermediate states of the evacuation. The figures show that when $t = 20$ s, the difference between the two methods is not large. However,

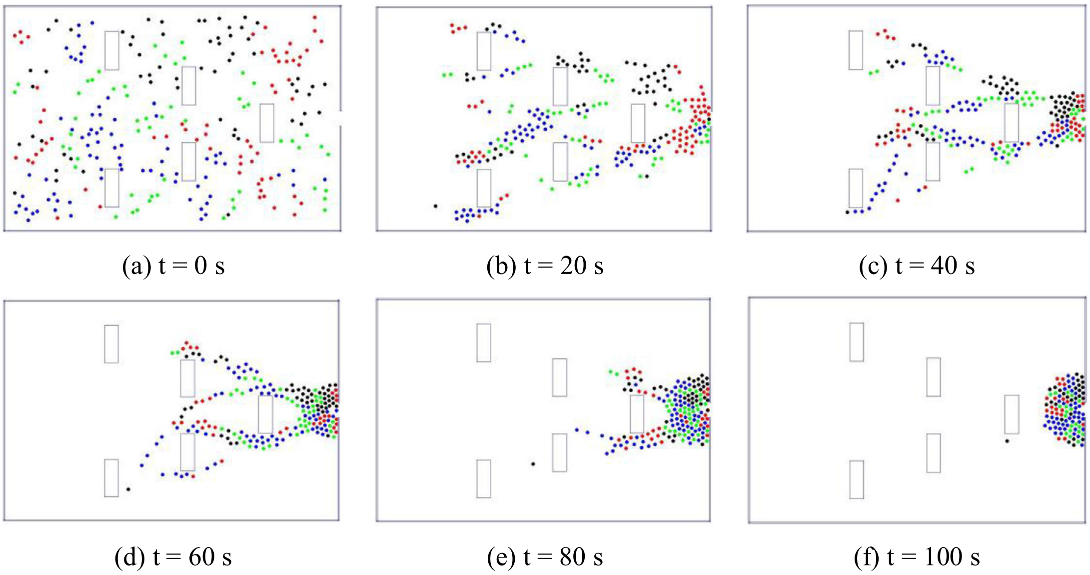


Fig. 13. Evacuation simulation with the leader and knowledge. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 2
Comparison of the average computational time (s) in a scenario without obstacles.

	Number of groups	Population size							
		25	50	75	100	125	150	175	200
Method of [11,18,35,44]	–	148.720	239.408	297.742	438.252	622.551	741.893	1015.410	1387.440
Method of [8]	1	158.140	257.458	311.340	457.451	654.240	796.452	1101.789	1501.420
	2	154.127	241.418	304.157	448.412	634.574	774.879	1057.912	1410.740
	4	152.471	240.740	291.2447	409.451	594.475	717.154	941.254	1187.420
	8	153.458	230.841	274.158	384.120	547.240	697.874	887.419	1012.450
Our model	1	149.102	237.092	298.422	421.470	611.645	739.328	995.983	1303.830
	2	137.893	228.119	289.420	397.910	573.070	710.552	959.903	1109.330
	4	128.248	224.461	268.710	377.898	557.451	653.569	771.886	907.465
	8	124.189	221.371	262.392	363.964	522.275	651.714	739.328	834.278

given the use of the original SFM, the evacuation of the individuals during the activity encountered obstacles, and they wandered for a while to bypass the obstacles. With increasing time, the gap between the two methods widens.

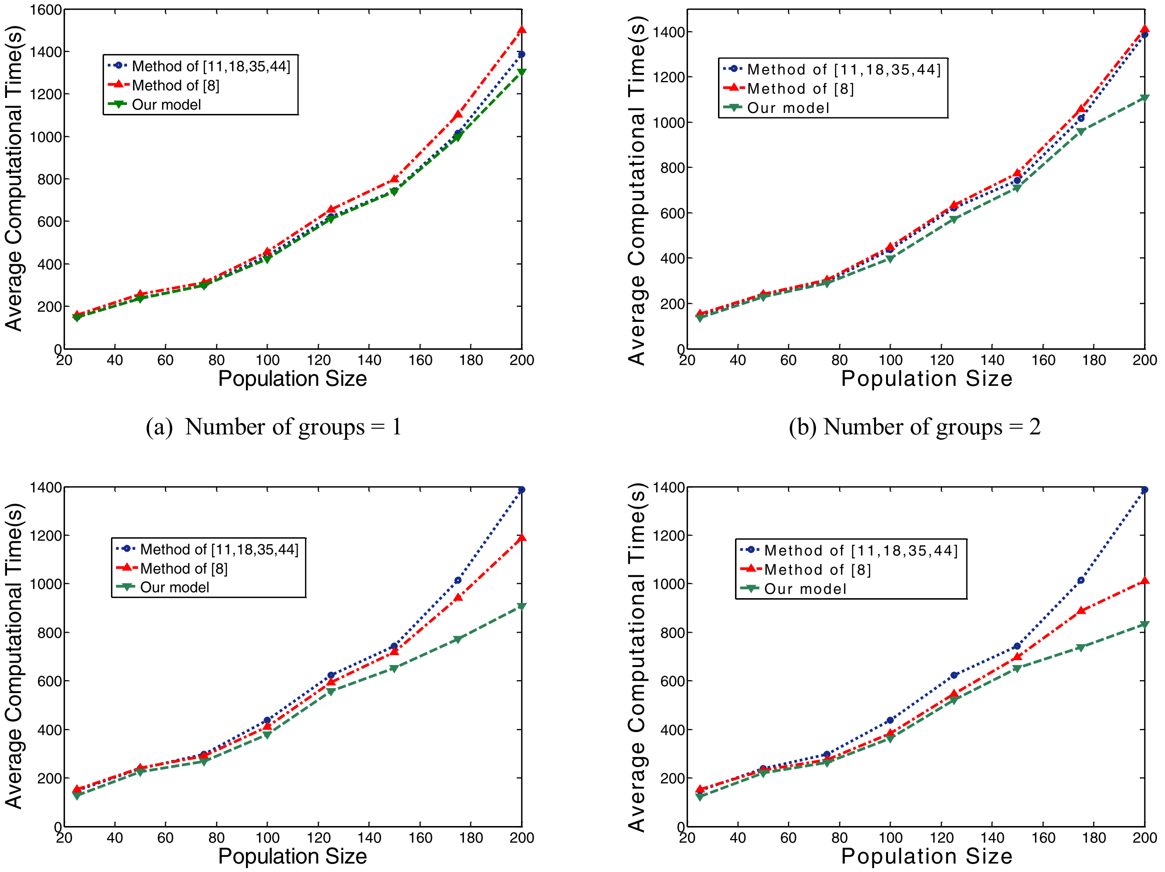
The colors in Fig. 13 indicate the grouping. Using our method in Figs. 13(c) and (d), the individuals in a group follow the leaders through the navigation agent to provide the navigation knowledge to bypass obstacles. The implementation of the Lite D* algorithm makes the distribution of all groups even, and no large amounts of congestion are observed at any of the obstacles. The entire evacuation process by the leaders through navigation agents interacts with the knowledge base to obtain the evacuation path information, and the evacuation is completed according to the path information.

The experiments in the two scenarios show that our method is effective when guiding crowd evacuations.

6.2. Computation time of our model

In this section, we show the computation efficiency in crowd navigation. We compare our model with traditional agent-based methods [8,11,18,35,44]. The existing agent-based crowd simulation methods fall into two categories. One is a method without groups [11,18,35,44], and the other is a method with groups, but there is no leader. However, the underlying models of these methods for computing the motion of a crowd are different. For example, SFM is used in [11], and a cellular automata model is used in [8]. To make this comparison possible, we implement the main idea of these methods on the original SFM.

The computation time in the scenario without obstacles is collected in Table 2 (Fig. 14). The results show that our method consistently outperforms the other two types of methods. Consider all of the cases in Table 2. The average computational time of our method improves by 18.522% compared to the method of [11,18,35,44] and improves by 9.869% compared to the method of [8]. When the group number is 8 and the population size is 200, the competitive advantage becomes most pro-



(a) Number of groups = 1 (b) Number of groups = 2

(c) Number of groups = 4 (d) Number of groups = 8

Fig. 14. Comparison of average computational time in the scenario without obstacles.

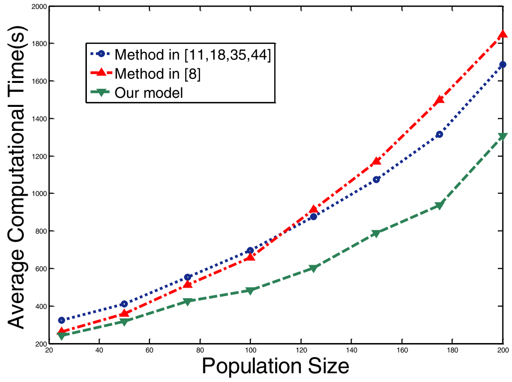
Table 3
Comparison of the average computational time (s) in the scenario with obstacles.

	Number of groups	Population size							
		50	100	150	200	250	300	350	400
Method of [11,18,35,44]	–	324.150	411.510	554.120	695.250	875.243	1075.216	1315.410	1687.440
Method of [8]	4	261.154	357.152	512.268	658.841	914.524	1171.212	1498.544	1848.650
	8	258.180	325.041	497.548	594.912	841.142	978.740	1284.178	1724.647
Our model	4	241.874	317.529	425.140	483.415	603.786	789.913	935.748	1305.790
	8	243.889	324.748	412.750	446.875	564.846	711.687	898.433	1247.650

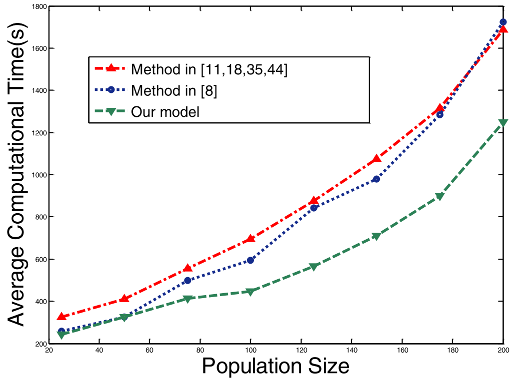
nounced. The improvement reaches 39.869% compared to the method of [11,18,35,44] and 17.598% compared to the method of [8].

Similar results are also obtained in Table 3 (Fig. 15). In this table, the average timing statistics of the three methods in the scenario with obstacles are given. The results show that the efficiencies of the two types of methods [8,11,18,35,44] are approximately the same in this scenario. Our method also outperforms the other two methods. The average efficiency of our method improves by 27.819% compared to the method in [11,18,35,44] and improves by 22.559% compared to the method in [8].

The computation time of our model is reduced because only the group leader is required to obtain the knowledge from the knowledge base, whereas the other individuals follow the leader directly. In contrast, the methods of [8,11,18,35,44] should compute the navigation path for each individual, which costs a large amount of time. Observe that the group method of [8] also outperforms the method of [11,18,35,44]. The motion consistency inside a group reduces the potential cost when computing the collision avoidance. However, the method of [8] is still inferior to ours because the knowledge of the leader is not considered.

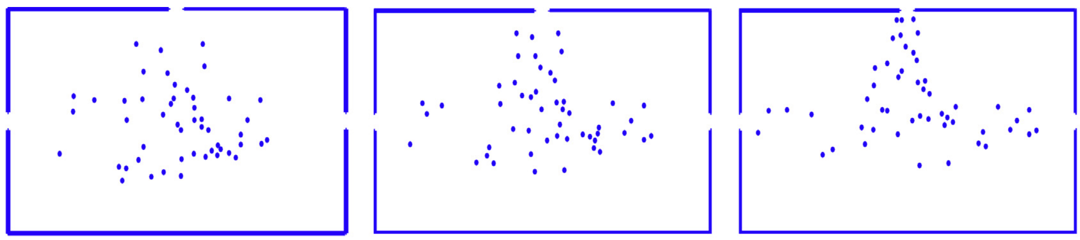


(a) Number of groups = 4

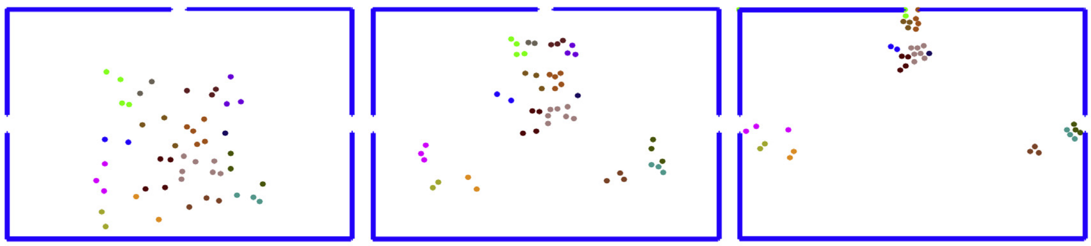


(b) Number of groups = 8

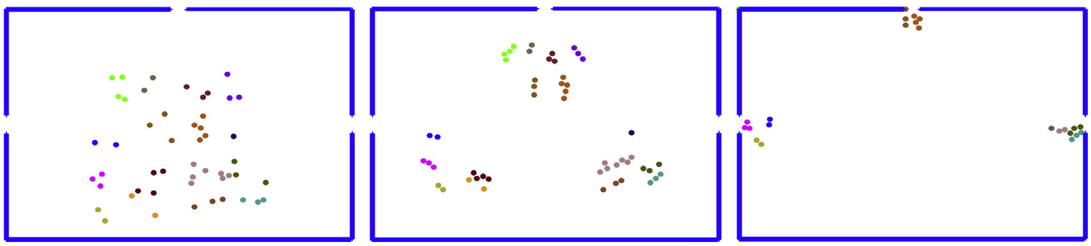
Fig. 15. Comparison of the average computational time in the scenario with obstacles.



(a) t=5 s (b) t=10 s (c) t=20 s



(d) t=5 s (e) t=10 s (f) t=20 s



(h) t=5 s (i) t=10 s (j) t=20 s

Fig. 16. Evacuation simulation with three different methods. The population size is 50. (a)–(c) are the results from methods in [11,18,35,44], (d)–(f) are the results from the method in [8], and (h)–(j) are the results from our model.

6.3. Utilization of the evacuation routes/exits

In this study, our goal is to show the effectiveness of our method in utilizing the evacuation routes/exits. To accomplish this goal, we simulate crowd evacuation in an empty room with 3 exits. The three agent-based models (i.e., agent-based model without groups [11,18,35,44], agent-based model with groups but there is no leader [8], and our model) are used in

Table 4
Comparison of the average evacuation time (s) in the scenario with three exits.

Methods	Population size							
	50	100	150	200	250	300	350	400
Method of [11,18,35,44]	62.760	65.373	76.610	102.617	132.593	160.823	184.890	232.190
Method of [8]	28.872	33.671	37.886	44.319	48.172	53.104	63.417	82.254
Our model	21.578	23.639	26.170	30.732	35.320	41.648	46.430	58.971

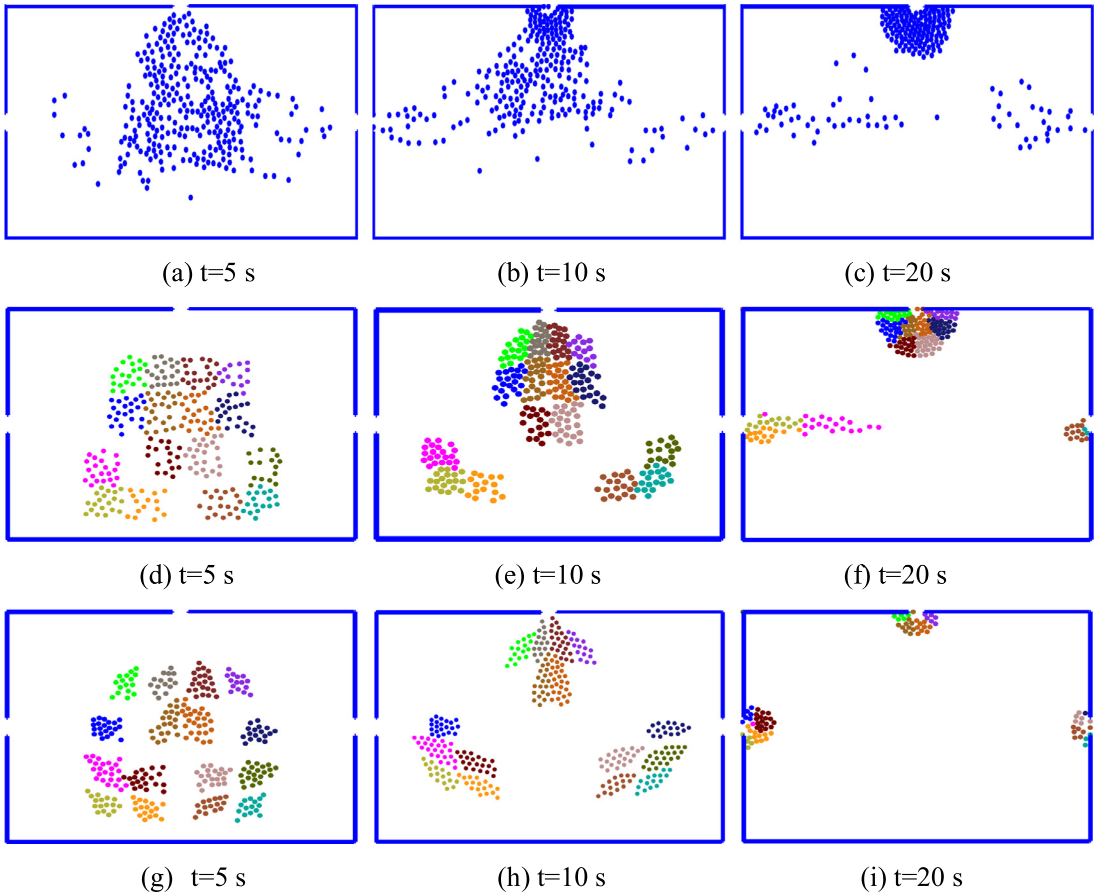
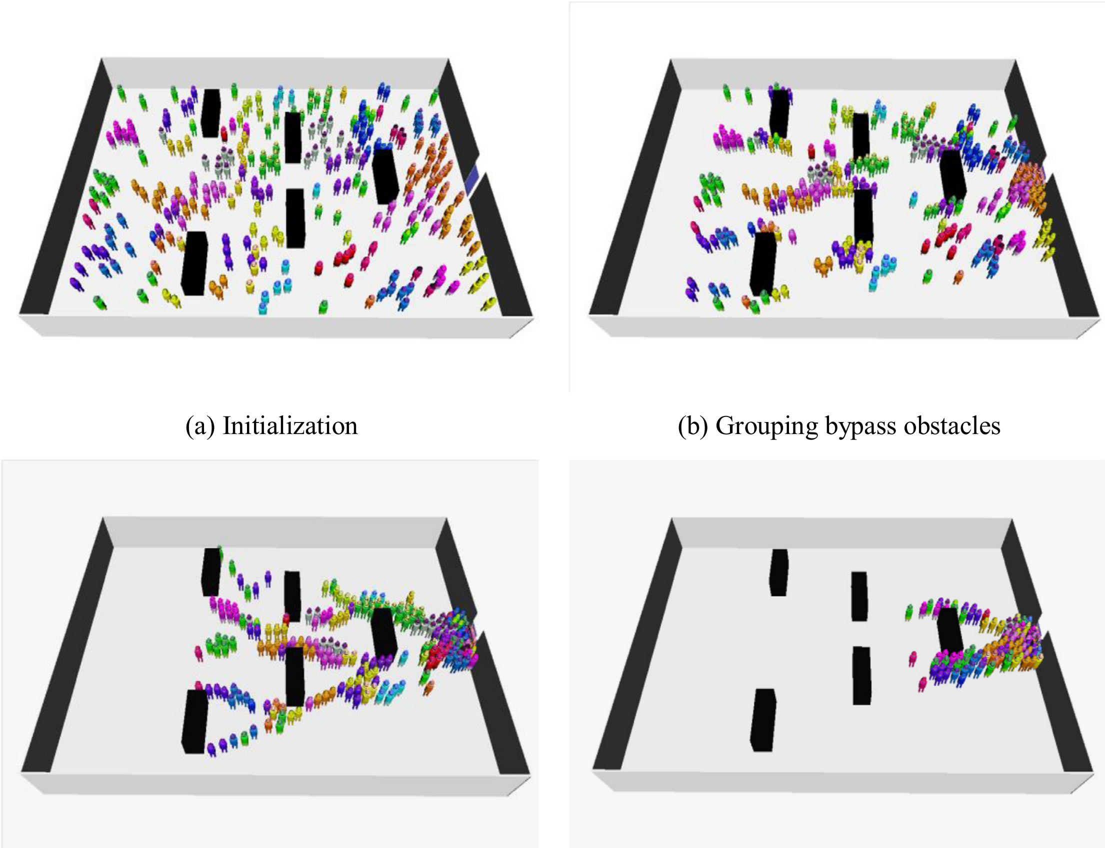


Fig. 17. Evacuation simulation with three different methods. The population size is 300. (a)–(c) are results from method of [11,18,35,44], (d)–(f) are results from the method of [8], and (h)–(j) are results from our model.

this experiment. Observe that less evacuation time is required if an evacuation plan can achieve a much higher utilization of the evacuation routes/exits. Therefore, we use the evacuation time to evaluate the utilization of the evacuation routes/exits.

Table 4 shows the comparison results. It shows that the evacuation time increases with the population size. When the population size is 400, the evacuation time of the method [11,18,35,44] is 232.190 s. When introducing the notion of a group in the crowd as in method [8], the evacuation time is reduced to 82.254 s when the population size is 400. This finding arises because the motion consistency inside a group reduces the potential collision avoidance. In addition, a group decides to reach one exit as a whole, and the crowding of some exits in the scenario is reduced. However, the method of [8] is still not promising because there is not any leader with knowledge in the group who helps the individuals to select the optimal evacuation route/exits. Our model improves this situation significantly by adding knowledge and leaders to the groups. The optimal route/exits selected by the population can avoid congestion in some exits, and thus, the evacuation time is reduced to 58.971 s in this case.

Figs. 16 and 17 show the simulation results at different times for crowd evacuation with different population sizes. The results demonstrate that our model can improve the utilization of the evacuation routes/exits. In the agent-based method without groups [11,18,35,44], severe congestion appears. When the population size increases, the congestion is more obvious. Considering Fig. 17(c), the left and right exits are relatively idle, and there is still a large number of people crowded in the upper exit. The method of [8] can improve this situation to a certain extent because a decision is made for each group, rather



(a) Initialization (b) Grouping bypass obstacles

(c) Grouping bypass obstacles (d) Close to exit

Fig. 18. Evacuation simulation with leaders and knowledge in a scene with one exit.

than for each individual, as in the method in [11,18,35,44]. However, the utilization rates of exits on the two sides are still not promising. As shown in Fig. 17(e), most of the groups move to the upper exit, and only a small number of groups selects the exits on the two sides. The reason is that although the leader can help to guide the navigation process, the crowd cannot bypass the congested exits because of a lack of dynamic knowledge. Using the knowledge base and navigation agents, our approach can guide the population to choose a better path/exit and reduce serious congestion. As a result, the evacuation time is reduced significantly.

6.4. Simulation

We adopt the Microsoft.Net Framework 4.5, Microsoft XNA Game Studio 4.0, Microsoft Visual Studio 2012, OpenSceneGraph-3.2.1, and MonoGame 3.2 to develop a crowd evacuation animation platform and illustrate the performance efficiency of the proposed simulation model and algorithm for crowd evacuation. The population size is 300, the individual radius is $r = 0.25\,\mathrm{m}$, the quality of the individual is $m = 80\,\mathrm{kg}$, and the expected rate of the individual is $v = 0.8\,\mathrm{m/s}$. The parameters of the SFM are set as follows: $A_i = 2 \times 10^3$ N, $B_i = 0.08\,\mathrm{m}$, $\tau_i = 0.5\,\mathrm{s}$, $v_i^0 = 1.5\,\mathrm{m\,s^{-1}}$, $k = 1.2 \times 10^5\,\mathrm{kg\,s^{-2}}$, and $\kappa = 2.4 \times 10^5\mathrm{kg\,m^{-1}s^{-1}}$. The desired velocity v_i^0 can reach more than 5 m s^{-1}. We perform the computer crowd evacuation simulation from a 300 m × 250 m plane area in the scene with obstacles and one exit (Fig. 2), as shown in Fig. 18.

We use the first floor of the teaching building no. 3 in Shandong Normal University as the scene, and we conduct the simulation experiments for the original SFM and our model, as shown in Figs. 19 and 20.

The figures show that when $t = 6\,\mathrm{s}$ and $t = 12\,\mathrm{s}$, the two models of the evacuation effect difference are not significantly different. However, when $t = 24\,\mathrm{s}$ and the knowledge of the original SFM is insufficient, the upper exit is extremely crowded, and the lower exit is idle. Thus, certain individuals still select the upper exit, which results in continuous jamming of the upper exit. Using our method, the following individuals turn to the vacant lower exit, which can effectively utilize the free exit and improve the evacuation efficiency.

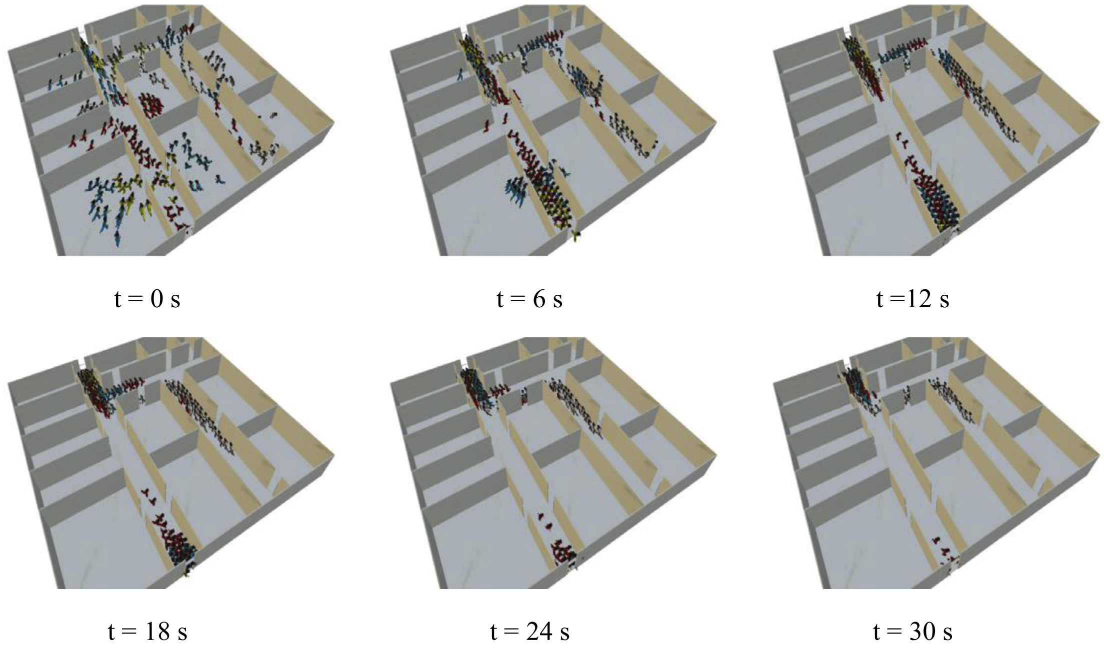


Fig. 19. Evacuation simulation of the original SFM in the building.

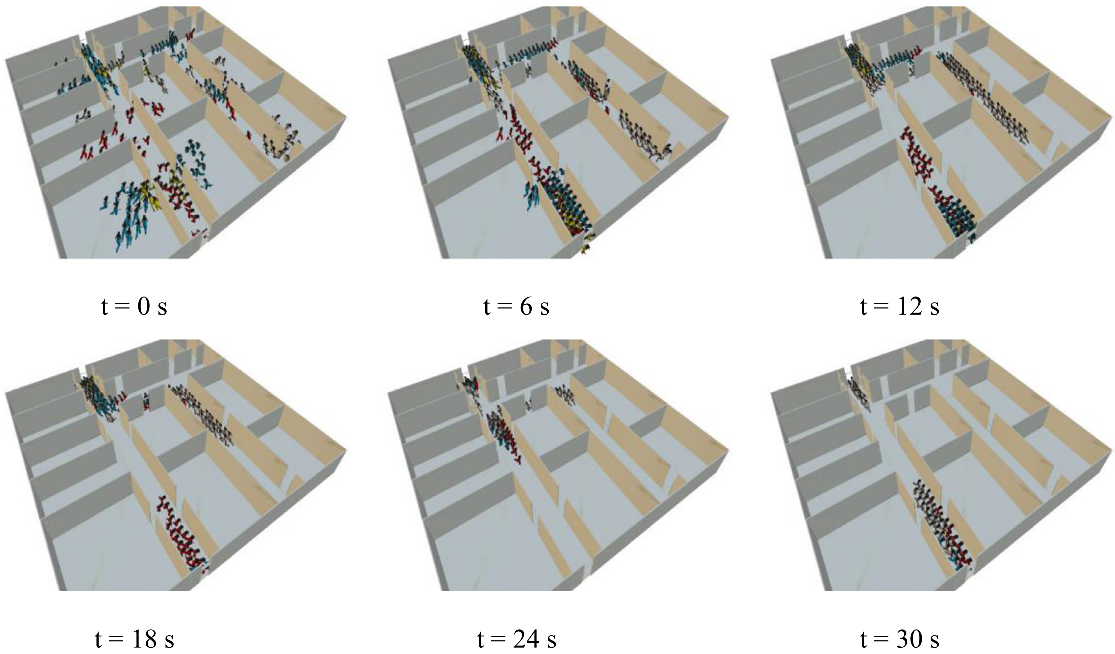


Fig. 20. Evacuation simulation with leaders and knowledge in the building.

7. Conclusions

This study proposes a knowledge-based crowd evacuation simulation method with a two- layer control mechanism. The CA and SFM are combined, and the interaction between the belief and the population spaces is established using the CA framework. Evacuation by grouping the population is guided by the navigation agents who can effectively improve the utilization rate of the channels in public places and the security of the pedestrians during a crisis. The performance of the approach presented in this paper is evaluated using a simulation system.

The following conclusions can be drawn from the simulation results:

(1) When the population density is low, the impact of the number of exits and the models of the evacuation speed is relatively small. With the increase in the population density, the influence of the number of exits and the models on the evacuation speed increase.
(2) In the scenarios with obstacles and multiple exits, our knowledge-driven model has obvious advantages over the original SFM and the leader–follower model.
(3) In the case of the same circumstances, a reasonable management strategy can improve the evacuation speed, reduce the congestion at exits and obstacles, and stabilize the evacuation process.

According to existing research and our simulation results, a reasonable building layout can accelerate the evacuation process [20,47]. Simultaneously, rational management strategies and knowledgeable guidance that are supported by computer tools are helpful in expediting evacuation processes [2,38]. The smooth and fast evacuation of a pedestrian group is dependent on guidance services provided by the evacuation management system.

The aim of this research is to develop an evacuation management system. An evacuation management system includes a knowledge management server, mobile wireless navigation agent, monitoring counters located in the obstacles and exits, and system management software. We envision a crowd evacuation that is self-organized into multiple groups according to their own position and social relations. Each group selects a leader based on his/her authority and a distance from the exit. The selected leader holds a wireless navigation device, which communicates with the navigation agent on the server through the mobile network. The navigation agent sends the leader's location information to the knowledge base management agent. The knowledge base management agent in the server executes the algorithm through the current position of the leader, the monitoring information, and the congestion conditions to determine the arrival time of the group at the exit and conducts dynamic control by sending the information to the navigation agent and then pass it to the leader. In this way, the system can reduce the congestion caused by the exits and obstacles and form a smooth evacuation process to improve the evacuation speed.

The study of this paper is just a beginning to achieving the above goal. The effectiveness of the proposed system is investigated by means of a simulation method. Therefore, it would be better to physically create evacuation devices in order to provide a series of real-world experiments. We are going to develop a system which uses client-server architecture. The client is a .app software installed on the mobile phone, using the GPS positioning of the mobile phone, communicating through a mobile operator's network, sending location information to the server, and accepting information from server on evacuation route and evacuation time for selecting a reasonable evacuation route. Furthermore, we intend to use navigation robots as leaders and extend the system's application to robotic-guided crowd evacuation in high-risk mine environments.

Our simulation is a relatively simple scene. The modeling and simulation of complex environments in buildings must be further explored for the better methods to be achieved. Using a machine learning method to extract and update the path information and using the swarm intelligence algorithm to optimize the evacuation paths would be highly valuable in conducting further studies in the future.

Acknowledgments

This research is supported by the National Natural Science Foundation of China (61472232, 61572299, 61373149 and 61402270) and by the project of Taishan scholarship.

References

[1] M.Z. Ali, N.H. Awad, P.N. Suganthan, R.M. Duwairi, R.G. Reynolds, A novel hybrid cultural algorithms framework with trajectory-based search for global numerical optimization, Inf. Sci. 334 (2016) 219–249.
[2] G. Bernardini, M. Azzolini, M. D'orazio, Intelligent evacuation guidance systems for improving fire safety of Italian-style historical theatres without altering their architectural characteristics, J. Cult. Heritage 22 (2016) 1006–1018.
[3] S. Cao, W. Song, W. Lv, Modeling pedestrian evacuation with guiders based on a multi-grid model, Phys. Lett. A 380 (2016) 540–547.
[4] R.J. Dawson, R. Peppe, M. Wang, An agent-based model for risk-based flood incident management, Nat. Hazards 59 (1) (2011) 167–189.
[5] J.G. Digalakis, K.G. Margaritis, A multipopulation cultural algorithm for the electrical generator scheduling problem, Math. Comput. Simul. 60 (2002) 293–301.
[6] J. Drury, C. Cocking, S. Reicher, Everyone for themselves? A comparative study of crowd solidarity among emergency survivors, Brit. J. Social Psychol. 48 (2009) 487–506.
[7] L. Fisher, The Perfect Swarm: the Science of Complexity in Everyday Life, Basic Books, 2009.
[8] Y.W. Fu, J.H. Liang, Q.P. Liu, X.Q. Hu, Crowd simulation for evacuation behaviors based on multi-agent system and cellular automaton, in: International Conference on Virtual Reality and Visualization, IEEE, 2015, pp. 103–109.
[9] Y.N. Guo, D. Liu, J. Cheng, Multi-population cooperative cultural algorithms, in: International Conference on Intelligent Computing (ICIC): Bio-Inspired Computing and Applications, 2011, pp. 199–206.
[10] S. Gwynne, E.R. Galea, P.J. Lawrence, L. Filippidis, Modeling occupant interaction with fire conditions using the building EXODUS evacuation model, Fire Saf. J. 36 (2001) 327–357.
[11] V.Q. Ha, Agent-Based Modeling of Emergency Building Evacuation Master's thesis, 2012.
[12] D.X. He, G.H. Xu, Z.H. Guan, M. Chi, D.F. Zheng, Hybrid coordination of multi-agent networks with hierarchical leaders? Commun. Nonlinear Sci. Numer. Simul. 27 (1) (2015) 110–119.
[13] D. Helbing, A fluid-dynamic model for the movement of pedestrians, Complex Syst. 6 (1992) 391–415.
[14] D. Helbing, I. Farkas, T. Vicsek, Simulating dynamical features of escape panic, Nature 407 (2000) 487–490.
[15] D. Helbing, P. Molnar, Social force model for pedestrian dynamics, Phys. Rev. E 51 (1995) 4282–4286.

[16] A.W. Hlynka, Using Heritage in Multi-Population Evolutionary Algorithms, University of Winds Scholarship at U Winds or Electronic Theses and Dissertations, 2015.
[17] A.W. Hlynka, Z. Kobti, Knowledge sharing through agent migration with multi-population cultural algorithm, in: Proceedings of the Twenty-Sixth International Florida Artificial Intelligence Research Society Conference, 2013, pp. 345–350.
[18] Y. Iizuka, K. Iizuka, Disaster evacuation assistance system based on multi-agent cooperation, in: Hawaii International Conference on System Sciences, IEEE, 2015, pp. 173–181.
[19] Q. Ji, C. Gao, Simulating crowd evacuation with a leader-follower model, Int. J. Comput. Sci. Eng. Syst. 1 (4) (2007) 249–252.
[20] J. Kang, I.J. Jeong, J.B. Kwun, Optimal facility-final exit assignment algorithm for building complex evacuation, Comput. Ind. Eng. 85 (2015) 169–176.
[21] A. Kirchner, A. Schadschneider, Simulation of evacuation processes using a bionics inspired cellular automaton model for pedestrian dynamics, Physica A 312 (2002) 260–276.
[22] S. Koenig, M. Likhachev, D* Lite, in: Proceedings of AAAI/IAAI, 2002, pp. 476–483.
[23] G. Köster, M. Seitz, F. Treml, D. Hartmann, W. Klein, On modeling the influence of group formations in a crowd, Contemp. Soc. Sci. 6 (2011) 397–414.
[24] T.I. Lakoba, D. Kaup, N.M. Finkelstein, Modifications of the Helbing–Molnar–Farkas–Vicsek social force model for pedestrian evolution, Simulation 81 (5) (2005) 339–352.
[25] H. Liu, J.H. Frazer, Supporting evolution in a multi-agent cooperative design environment, Adv. Eng. Softw. 33 (6) (2002) 319–328.
[26] H. Liu, Z.K. Lin, A cooperative design approach supporting dynamic task assignation, J. Softw. (in Chinese). 12 (12) (2001) 1830–1836.
[27] Y. Liu, N. Okada, Y. Takeuchi, Dynamic route decision model-based multi-agent evacuation simulation-case study of Nagata Ward, Kobe, J. Nat. Disaster. Sci. 28 (2) (2008) 91–98.
[28] Y. Ma, R.K.K. Yuen, E.W.M. Lee, Effective leadership for crowd evacuation, Physica A 450 (2016) 333–341.
[29] A.G. Milla, F. Borges, R. Suppi, E. Luque, Crowd evacuations SaaS: an ABM approach, Proc. Comput. Sci. 51 (2015) 473–482.
[30] R.R.N. Mohammad, Z. Kobti, A multiagent system to solve JSSP using a multi-population cultural algorithm, in: Canadian Conference on Advances in Artificial Intelligence, 2012, pp. 362–367.
[31] R.R.N. Mohammad, Z. Kobti, A knowledge-migration-based multi-population cultural algorithm to solve job shop scheduling, in: The 25th Florida Artificial Intelligence Research Society Conference, 2012, pp. 68–73.
[32] M. Moussaïd, N. Perozo, S. Garnier, D. Helbing, G. Theraulaz, The walking behavior of pedestrian social groups and its impact on crowd dynamics, PLoS One 5 (4) (2010) e.0010047.
[33] M. Nagarajan, D. Shaw, P. Albores, Disseminating a warning message to evacuate: a simulation study of the behaviour of neighbours, Eur. J. Oper. Res. 220 (1) (2012) 810–819.
[34] T.T. Nguyen, X. Yao, An experimental study of hybridizing cultural algorithms and local search, Int. J. Neural Syst. 18 (2008) 1–18.
[35] X. Pan, C.S. Han, K. Dauber, K.H. Law, A multi-agent based framework for the simulation of human and social behaviors during emergency evacuations, Ai Soc. 22 (2) (2007) 113–132.
[36] N. Pelechano, J. Allbeck, N. Badler, Controlling individual agents in high-density crowd simulation, in: Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, Eurographics Association, SanDiego, California, 2007, pp. 99–108.
[37] M.R. Raeesi N., Z. Kobti, Heterogeneous multi-population cultural algorithm, in: 2013 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2013, pp. 292–299.
[38] H.C. Ren, L.H. Sun, X.Z. Gao, Influences of intelligent evacuation guidance system on crowd evacuation in building fire, Autom. Constr. 41 (2014) 78–82.
[39] R.G. Reynolds, An introduction to cultural algorithms, in: Proc. Third Annu. Conf. Evol. Program, 1994, pp. 131–139.
[40] J. Shi, A. Ren, C. Chen, Agent-based evacuation model of large public buildings under fire conditions, Autom. Constr. 18 (3) (2009) 338–347.
[41] J.D. Sime, Affiliative behaviour during escape to building exits, J. Environ. Psychol. 3 (1983) 21–41.
[42] L. Tan, M. Hu, H. Lin, Agent-based simulation of building evacuation: combining human behavior with predictable spatial accessibility in a fire emergency, Inf. Sci. 295 (2015) 53–66.
[43] C. Vihas, I.G. Georgoudas, G.C. Sirakoulis, Cellular automata incorporating follow-the-leader principles to model crowd dynamics, J. Cell. Autom. 8 (5–6) (2013) 333–346.
[44] N. Wagner, V. Agrawal, An agent-based simulation system for concert venue crowd evacuation modeling in the presence of a fire disaster, Expert Syst. Appl. 41 (6) (2014) 2807–2815.
[45] M. Xu, Y. Wu, P. Lv, H. Jiang, M. Luo, Y. Ye, miSFM: on combination of mutual information and social force model towards simulating crowd evacuation, Neurocomputing 168 (30) (2015) 529–537.
[46] X. Yang, H. Dong, Q. Wang, Y. Chen, X. Hu, Guided crowd dynamics via modified social force model, Physica A 411 (2014) 63–73.
[47] C.M. Yuan, C. Li, G. Li, P.H. Zhang, Safety evacuation in building engineering design by using building exodus, Syst. Eng. Proc. 5 (2012) 87–92.
[48] P.M. Zadeh, Z. Kobti, A multi-population cultural algorithm for community detection in social networks, Proc. Comput. Sci. 52 (2015) 342–349.
[49] M. Zhou, H. Dong, F.Y. Wang, Q. Wang, X. Yang, Modeling and simulation of pedestrian dynamical behavior based on a fuzzy logic approach, Inf. Sci. 360 (2016) 112–130.