

AGENT-BASED EVACUATION IN PASSENGER SHIPS USING A GOAL-DRIVEN DECISION-MAKING MODEL

Baocheng Ni

College of Ship Building, Harbin Engineering University University, China

Zhen Li

College of Mechanical and Electrical Engineering, Harbin Engineering University, China

Xiang Li

Dalian Neusoft University of Information, China

ABSTRACT

A new agent-based model is proposed to support designers in assessing the evacuation capabilities of passenger ships and in improving ship safety. It comprises models for goal-driven decision-making, path planning, and movement. The goal-driven decision-making model determines an agent's target by decomposing abstract goals into subgoals. The path-planning model plans the shortest path from the agent's current position to its target. The movement model is a combination of social-force and steering models to control the agent in moving along its path. The utility of the proposed model is verified using 11 tests for passenger ships proposed by the Maritime Safety Committee of the International Maritime Organization.

Keywords: ship evacuation analysis; agent-based model; goal-driven decision-making model; social force model; steering behavior

INTRODUCTION

Recent developments in shipbuilding for cruise liners have enabled a single passenger ship to carry several thousand people. Prompted by a series of disasters, the safety of large passenger ships has become a matter for urgent investigation worldwide. Because of such concerns, the Maritime Safety Committee (MSC) of the International Maritime Organization (IMO) now requires evacuation simulation to be performed during the design process. This mandatory requirement is stated clearly in the IMO's Guidelines for Evacuation Analysis for New and Existing Passenger Ships [1]. These guidelines offer a simplified method for evacuation analysis that calculates the evacuation time by means of relatively simple expressions, and an advanced method that uses computer-based simulation to evaluate the evacuation time. The latter form of analysis is not yet mandatory but it tends to be required in practice. Hence, we focus on advanced evacuation analysis in this study.

A variety of modeling methods have been used to study crowd evacuation in various situations. One of the most popular discrete models for evacuation is the cellular automaton [2–4]. Cellular automata are widely used in architectural design [5–7]. In this model, each pedestrian is located on a cell and moves to a neighboring one depending on the status of adjacent cells and a predefined set of rules. Cellular automata update their cell states at discrete time intervals. One advantage of this approach is its computational efficiency: time, space, and states are all discrete. Recently, an advanced evacuation-simulation model for passenger ships was developed based on cellular automata and its usefulness was proven [8, 9]. However, because of the discrete nature of cellular automata, they cannot be used to compute accurate evacuation times for pedestrians.

In contrast to cellular automata, social-force models [10, 11] are continuous ones that assume that individuals are subject to physical and social forces. These forces are determined by solving Newton's equations of motion, which can be extended to include various types of behaviors such as overtaking

[12], counterflow collision avoidance [13], and information exchange [14]. However, the need for complicated expressions (which are relatively hard to calibrate) to calculate these forces imposes a heavy computational burden.

Developments in artificial intelligence have increased the popularity of agent-based models [15–17] in which pedestrians are represented as autonomous agents with cognitive and learning abilities. Such models can simulate heterogeneous pedestrians by allowing them to have their own individual behavior and properties. Generally speaking, most agent-based models use predefined rules to govern the behavior of agents. Modeling such rules is a highly sophisticated task and the level of each agent’s autonomy is limited.

In this work, we use a goal-driven decision-making model instead of a rule-driven one to decide which plan to follow. Goal-driven models allow agents to achieve hierarchical goals that mirror aspects of human deliberation in planning. They also offer significant flexibility to add other goals at later stages. Here, we combine a social-force model with a steering behavior model to design a movement model that keeps the computational complexity within acceptable limits. Our model is capable of simulating thousands of agents in real time with an Intel CPU. Experiments conducted in this study verified the utility of the proposed model for evacuation simulation.

MODEL FRAMEWORK

We developed an agent-based framework that contains models for goal-driven decision-making, crowd movement, and path planning. An agent-based model is a microscale one that uses simple dynamically interacting agents to recreate complex phenomena [15]. The goal of such modeling is to provide explanatory insights into the collective behavior of agents which individually are obeying simple behavioral rules. This modeling approach has been used extensively in biology, ecology, and social science.

The autonomous agents in our evacuation model represent passengers on a ship. Each agent comprises three components: perception, decision-making, and locomotion (see Fig. 1).

The perception component mimics human vision, hearing, and other senses. The agent perceives its environment through sensors and updates its knowledge in relation to walls, doors,

exits, and other agents within a certain radius. The decision-making component decomposes abstract goals into subgoals to create a concrete plan that contains a series of targets. The locomotion component represents the mechanical aspects of an agent’s movement. Each agent is governed by a crowd-movement model that facilitates navigation toward a target.

MODEL IMPLEMENTATION

ENVIRONMENT REPRESENTATION

The environment is represented in the form of a graph $G(V, E)$ that contains all the locations in the environment that agents may visit and all the connections (edges) between those points. The graph is defined as a set of nodes N linked to a set of edges E . Each edge of the graph contains the cost of moving from one node to another. The cost is the distance between the two nodes that the edge connects. The graph $G(V, E)$ is a digraph (directed graph) in which the connections are directional. We use this digraph to reflect the terrain gradient. Each edge has additional information (a flag) that is used to indicate the type of terrain (uphill or downhill). It is quicker and more efficient for an agent to travel downhill than uphill. Fig. 2 shows an example of the graph $G(V, E)$ and its generation. We first set a single node called the seed node in the environment and then apply a breadth-first search to obtain all nodes of the graph. These are obtained by expanding nodes and edges outward from the seed in eight directions unless any of these are unavailable because of obstructions.

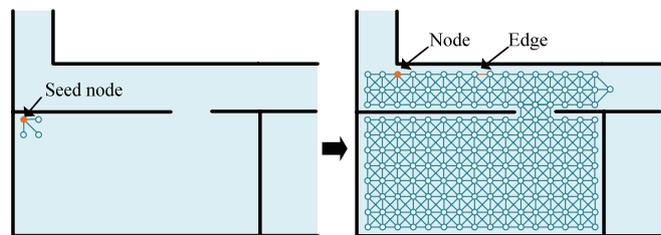


Fig. 2. Constructing the graph of the environment.

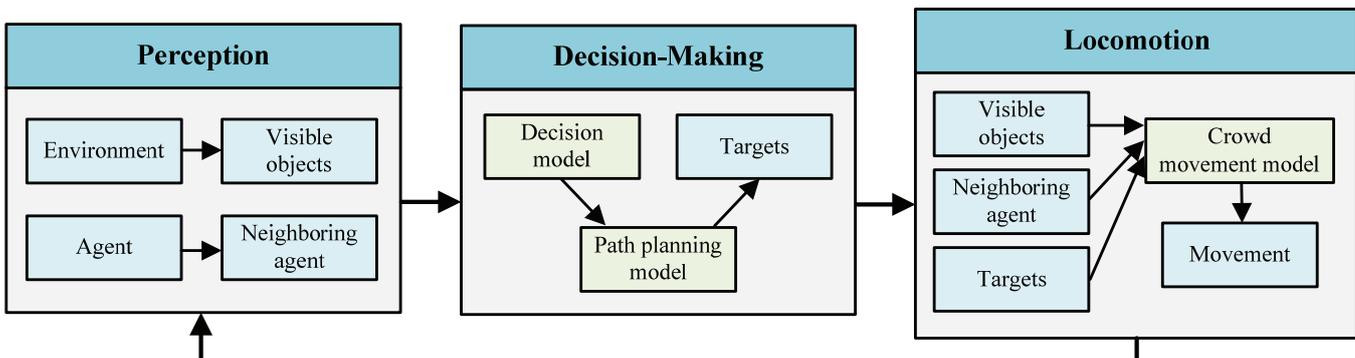


Fig. 1. Components of an agent in the evacuation model.

PATH-PLANNING MODEL

The path-planning model identifies the shortest path P_{ij} leading from the current position of agent i to exit j . The shortest path P_i heading toward the nearest exit is calculated as

$$P_i = \min(P_{i1}, P_{i2}, \dots) \quad (1)$$

while the shortest path P_{ij} is calculated as

$$P_{ij} = f^*(G(V, E), \mathbf{p}_i, \mathbf{e}_j) \quad (2)$$

where \mathbf{p}_i is the position of agent i , \mathbf{e}_j is the j^{th} exit position, and function f^* is the well-known A* search algorithm [14]. If there is more than one exit, the path-planning model compares the cost of all paths and selects the one with the minimum cost as the shortest path leading toward the nearest exit. The A* algorithm constructs a tree of possible paths starting from the initial node, expanding the tree one step at a time until one of its paths ends at the target node. Moreover, the A* algorithm selects the next node for expansion by identifying the node with the lowest evaluation value. The evaluation value of node n is calculated as

$$f(n) = g(n) + h(n) \quad (3)$$

where $g(n)$ is the real path cost from the start node to node n and $h(n)$ is the estimated cost of the cheapest path from node n to the target. We chose the straight-line distance between node and the target node as our estimated cost function.

Fig. 3 shows a simple graph that includes all edge costs and straight-line distances of nodes leading to node L. We use this simple graph to illustrate how the A* algorithm finds the shortest path from node C to node L. The search tree of the A* algorithm and the relevant values of f are shown in Fig. 4. In the figure, blue nodes represent nodes that have been expanded, while green nodes represent nodes selected for expansion. In the initial state, shown in Fig. 4(a), node C is expanded, and then the values of f are computed for all adjacent nodes of C based on the costs and straight-line distances shown in Fig. 3.

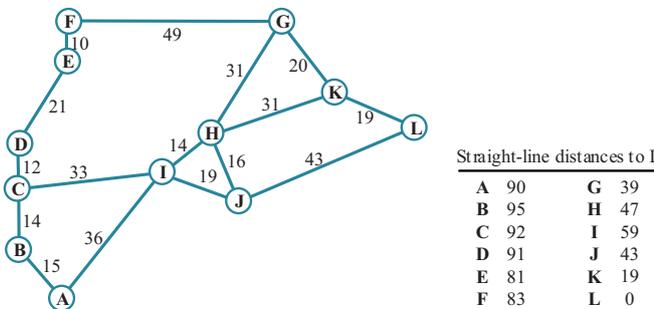


Fig. 3. A simple graph with the cost of each edge and straight-line distances to node L shown.

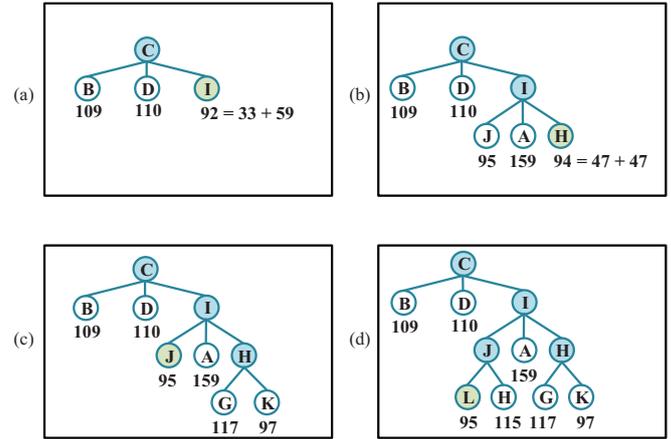


Fig. 4. Stages in the A* algorithm search for node L: (a) after expanding node C, (b) after expanding node I, (c) after expanding node H, and (d) after expanding node J.

After expanding node C, we select node I for expansion given that its evaluation value ($f = 92$) is the lowest among the leaf nodes B, D, and I of the tree. At the next step, shown in Fig. 4(b), we select node H (with $f = 94$ being the lowest value) for expansion from leaf nodes B, D, J, A, and H. This selection and expansion process is repeated until the target node L is selected for expansion, as shown in Fig. 4(d). At this point, the length of the shortest path is given by the algorithm. After this algorithm runs, the target node L will point to its parent (node J), and this parent node will point to its parent node. This process continues until a node's parent node is the start node C; in this example, node I is the node with the parent node C. The given path from leaf node L to root node C (L→J→I→C) is then identified as the shortest path from node L to node C. Finally, we reverse the above path to get the actual shortest path (C→I→J→L) from node C to L.

AGENT REPRESENTATION

Each passenger is modeled as an autonomous agent who is interacting with the dynamic environment and other agents. Agents have their own individual properties and behavioral models. The cross-sectional shape of each agent is treated as a combination of three circles (see Fig. 5) and is analogous to the representation in the Simulx program [19] and in the FDS+Evac simulation software [20]. The body dimensions of the agents are listed in Tab. 1 and are the same as those in the Simulx program.

Tab. 1. Body dimensions of different types of agents

Body type	R' (m)	R'/R'	R^h/R'
Child	0.210 ± 0.015	0.3333	0.5714
Male	0.270 ± 0.020	0.3704	0.5926
Female	0.240 ± 0.020	0.3750	0.5833
Elderly	0.250 ± 0.020	0.3600	0.6000

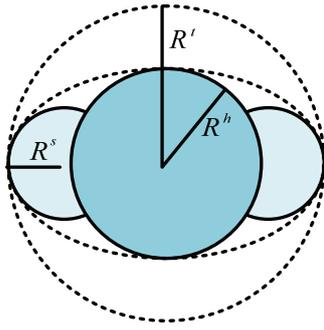


Fig. 5. Cross-sectional shape of an agent approximated by three circles.

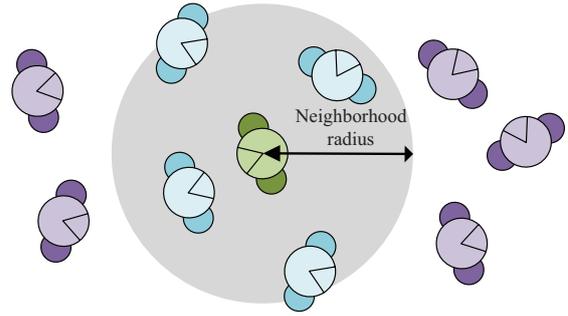


Fig. 6. Neighborhood of an agent.

CROWD-MOVEMENT MODEL

This section describes a crowd-movement model that is a combination of models for social force [10, 11] and steering behavior [21, 22]. Analogous to the social-force model, this model uses the laws of mechanics to calculate the agents' movement during the simulation. Furthermore, the model uses a collection of steering behaviors to represent the interaction of agents. The model assumes that an agent's behavior is affected by a combination of sociopsychological and physical forces. Each agent i of mass m_i prefers to move with a certain desired speed $v_i^0 = |v_i^0|$ toward a target specified by the decision-making model. Furthermore, agent i likes to change its current velocity v_i over a characteristic time τ_i , and consequently the agent accelerates toward its desired speed. Simultaneously, a steering force f_i^{St} acts upon the agent so that it can avoid collisions with its neighbors or cooperate with others, and a repulsive force f_i^w that makes the agent keep a certain distance from any wall. The change of velocity of an agent in time t can be calculated by the acceleration equation

$$m_i \frac{dv_i}{dt} = m_i \frac{v_i^0(t) - v_i(t)}{\tau_i} + f_i^{St} + f_i^w \quad (4)$$

while the change of position $p_i(t)$ of agent i is given by the velocity $v_i(t) = dp_i/dt$. The first term on the right-hand side of Eq. (4) describes the motive force acting upon the agent that accelerates it toward its desired velocity that is given by

$$v_i^0(t) = v_i^{max} \frac{(p_i^0(t) - p_i(t))}{\|p_i^0(t) - p_i(t)\|} \quad (5)$$

where v_i^{max} is the maximum walking speed of agent and the target position $p_i^0(t)$ of agent i is determined by the decision-making model.

The steering force f_i^{St} acting upon agent comprises three parts:

$$f_i^{St} = k_s f_i^{Se} + k_c f_i^{Co} + k_a f_i^{Al} \quad (6)$$

where f_i^{Se} , f_i^{Co} , and f_i^{Al} are the behavioral forces of separation, cohesion, and alignment, respectively. Coefficients k_s , k_c , and k_a represent the strength of each respective force.

When calculating the steering force, an agent considers only agents within a circular radius r (referred to as the neighborhood radius). In Fig. 6, the gray circle represents the neighborhood of the green agent. All the blue agents are considered to be the green one's neighbors whereas the purple agents are not. In mathematical terms, B_i is used to denote the set of agents within the neighborhood radius of agent:

$$B_i = \{j \mid j \in A, j \neq i, \|p_i - p_j\| < r\} \quad (7)$$

where p_i and p_j denote the position vectors of agents i and j , respectively, and A denotes the set of all agents.

All passengers try to avoid colliding with their neighbors. This tendency is known as separation behavior, and it creates a steering force that moves an agent away from its neighbors (see Fig. 7). The separation force can be calculated as

$$f_i^{Se} = \sum_{j \in B_i} \frac{1}{\|p_i - p_j\|} \cdot \frac{(p_i - p_j)}{\|p_i - p_j\|} = \sum_{j \in B_i} \frac{n_{ij}}{\|p_i - p_j\|} \quad (8)$$

where the vector $n_{ij} = (p_i - p_j)/\|p_i - p_j\|$ is the unit vector in the direction from agent i to agent j .

Each passenger tends to stay close to the local group formed by his or her neighbors during an evacuation. This tendency is referred to as cohesion behavior, and it produces a force that steers an agent toward the center (centroid) of the geometrical shape formed by neighbors (see Fig. 8). This cohesion force can be determined from

$$f_i^{Co} = \sum_{j \in B_i} \frac{p_j}{|B_i|} - p_i \quad (9)$$

where $|B_i|$ is the number of members of set.

Passengers tend to match the direction and speed of their neighbors. This tendency is referred to as alignment behavior, which causes passengers to follow their neighbors (see Fig. 9). The alignment force is calculated by averaging the velocity of the neighbors:

$$f_i^{Al} = \sum_{j \in B_i} \frac{v_j}{|B_i|} - v_i \quad (10)$$

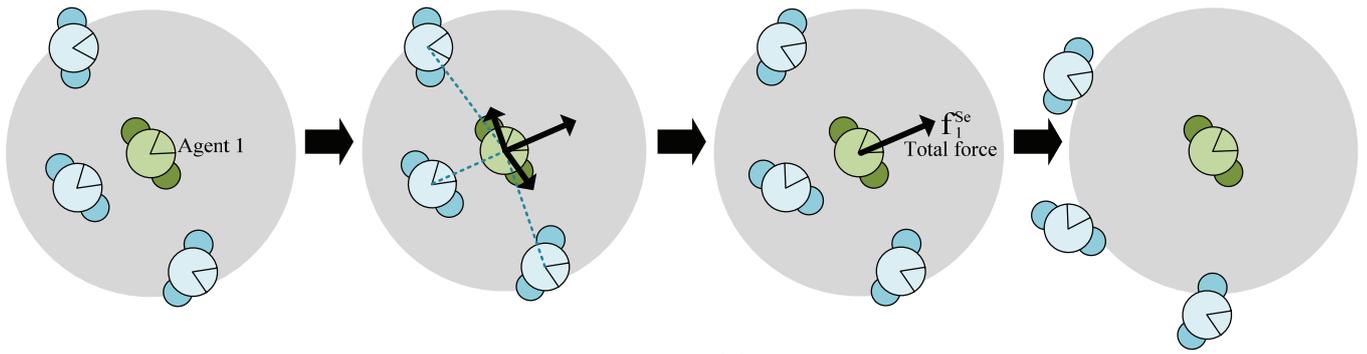


Fig. 7. Separation behavior.

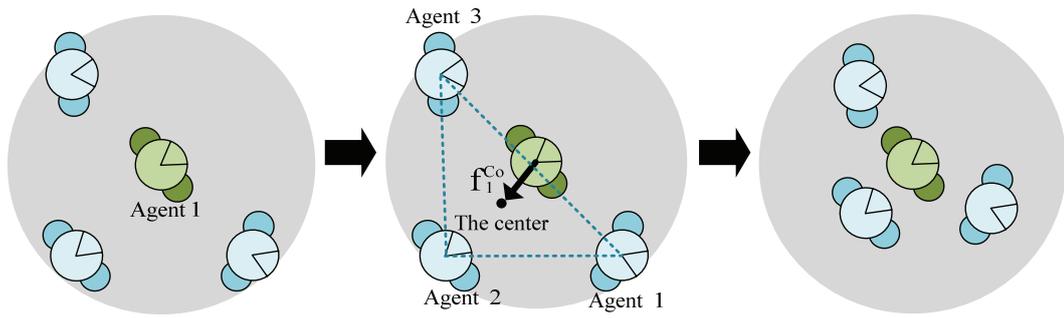


Fig. 8. Cohesion behavior.

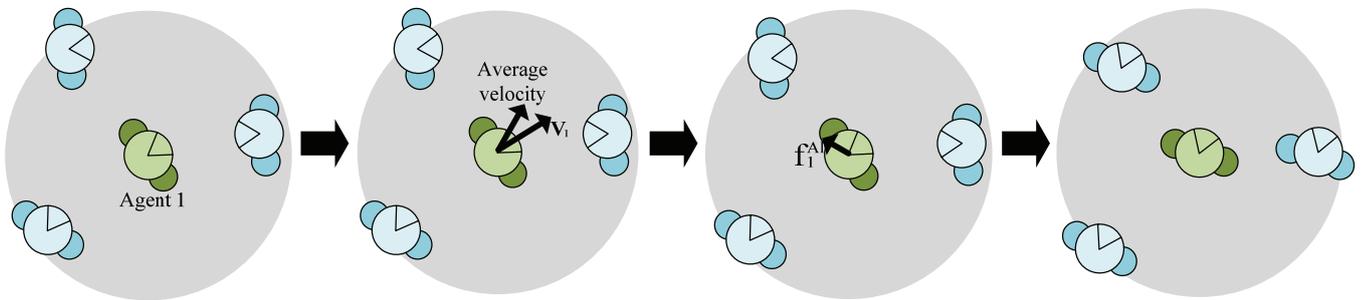


Fig. 9. Alignment behavior.

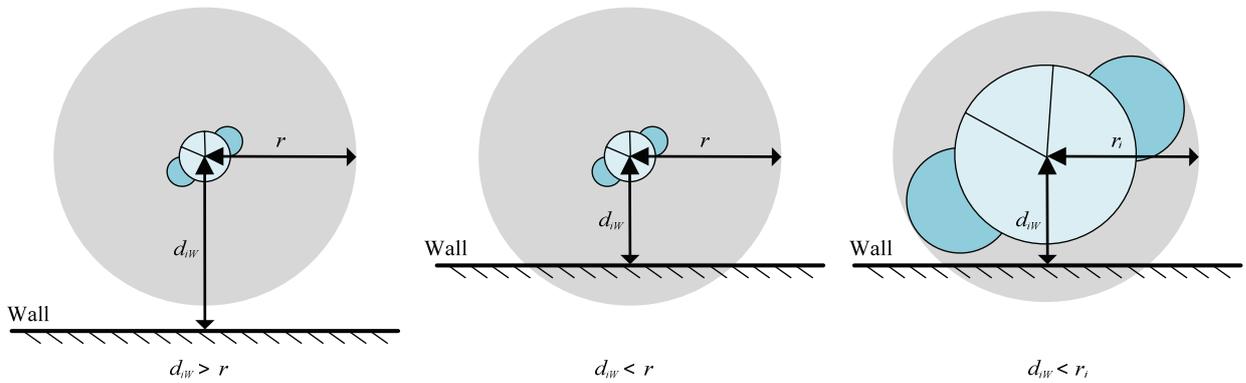


Fig. 10. Three typical distances between an agent and a wall.

The repulsive wall force \mathbf{f}_i^w that keeps each agent a certain distance from any wall is given by

$$\mathbf{f}_i^w = k_w \{ \sum_W [g(d_{iW} - r) + k_r g(d_{iW} - r_i)] \mathbf{n}_{iW} \} \quad (11)$$

and

$$g(x) = \begin{cases} 0 & x > 0 \\ |x| & x < 0 \end{cases} \quad (12)$$

where k_w is the strength of the repulsive force, d_{iW} is the distance between agent i and wall j , r_i is the radius of agent i , r is the agent's neighborhood radius (see Fig. 10), the force constant k_r denotes the radial elastic force strength, and \mathbf{n}_{iW} denotes the direction perpendicular to the wall.

Eq. (4)–(12) describe the displacement of the agents. The rotation of an agent is treated analogously

$$\frac{d^2 \varphi_i(t)}{dt^2} = \frac{[\omega_i^0(t) - \omega_i(t)]}{\tau_i} \quad (13)$$

where $\varphi_i(t)$ is the agent's azimuthal angle, $\omega_i(t) = d\varphi_i(t)/dt$ is the angular velocity, and $\omega_i^0(t)$ is the desired angular velocity. Time τ_i is the characteristic one over which the agent alters its current angular velocity to the desired one given by

$$\omega_i^0(t) = \frac{1}{\pi} \arccos \left[\frac{(\mathbf{h}_i^0(t) \cdot \mathbf{h}_i(t))}{\|\mathbf{h}_i^0(t)\| \|\mathbf{h}_i(t)\|} \right] f(\mathbf{h}_i^0(t), \mathbf{h}_i(t)) \omega_i^{max} \quad (14)$$

where $\mathbf{h}_i^0(t)$ is the actual facing direction of agent i , $\mathbf{h}_i^0(t) = \mathbf{p}_i^0(t) - \mathbf{p}_i(t)$ is the desired facing direction, and ω_i^{max} is the maximum desired angular velocity of agent i . Note that the angle between $\mathbf{h}_i^0(t)$ and $\mathbf{h}_i(t)$ lies in the interval $(0, \pi)$. The function $f(\mathbf{x}, \mathbf{y})$ is equal to 1 if the vector \mathbf{x} is clockwise of the vector \mathbf{y} , and -1 otherwise. In summary, Eq. (13)–(14) describe the rotation of each agent.

There are various parameters in the agent-movement model. Some are related to the physical traits and demographics of humans, such as m_i , v_i^{max} , ω_i^{max} , and r_i . Others are relevant to the nature of the movement model itself. Some of these parameters are estimated from simulations (see Tab. 2) and others are taken from the literature [8, 11, 23] by considering each passenger's characteristics (e.g.; age, gender, etc.). We chose the same value for each parameter for all passengers in order to minimize the number of parameters for the sake of robustness and adjustment, while acknowledging that parameter values differ somewhat among passengers in reality.

Tab. 2. Default values used in crowd-movement model.

Parameter	Default value	Unit	Description
k_s	3.0	dimensionless	strength of separation force
k_c	0.5	dimensionless	strength of cohesion force
k_a	0.2	dimensionless	strength of alignment force
k_w	0.8	dimensionless	strength of wall repulsive force
k_r	1.0	dimensionless	strength of radial elastic force
d	0.3	m	neighborhood radius

GOAL-DRIVEN DECISION-MAKING MODEL

The goal-driven decision-making model plays the same role as that of the human brain in choosing goals and deciding which plan to follow. Humans prefer to select higher-level abstract goals based on their desires and then decompose them recursively into a plan of action that can be executed without deliberation [24]. For example, a passenger might decide to *evacuate*. This is an abstract goal that cannot be executed unless it is decomposed into smaller ones such as *leave cabin*, *move to assembly station*, and *enter lifeboat*. Each of these goals is abstract and needs to be decomposed further. For instance, the goal *leave cabin* can be broke down into smaller ones such as *leave bed*, *move to door*, *open door*, and *move outside*. A goal is typically not decomposed until it is executed, which is the same as in the human though process.

In some ways, goal-driven decision-making mirrors aspects of human deliberation in its planning. At each update, an agent evaluates its environmental state and chooses its most desired high-level goal. The agent then decomposes this into subgoals and satisfies each one in turn.

Five goals are predefined in our model: (i) *evacuate*, (ii) *plan path*, (iii) *follow path*, (iv) *move to node*, and (v) *enter assembly station*. Goal (iv) is a prime one that cannot be decomposed, whereas the others are composite goals comprising several subgoals. When an evacuation starts, an individual agent feels a strong desire to evacuate and so pursues the goal *evacuate*. However, this goal is too abstract and needs to be decomposed. To achieve it, the agent must plan a path to the assembly station, follow that path, and enter the assembly station. The goal *evacuate* consists of the subgoals *plan path*, *follow path*, and *enter assembly station*. The goal *plan path* is satisfied by requesting the path-planning model to plan a path to the assembly station, after which it is deleted from the goal list. The agent then pursues the goal *follow path* that can be decomposed into a series of prime *move to node* goals, each of which contains a target node that is located on the path to the assembly station and that is the target position $\mathbf{p}_i^0(t)$ needed in Eq. (5). The process of decomposing and satisfying goals continues until they have all been executed.

Fig. 11 shows a flow diagram of an agent pursuing certain goals involved in this model and their related responses. The rectangles denote the different goals and the rhombuses represent the judgment conditions. Arrows indicate the direction of action flow. An agent with the prime goal *move to node* needs to evaluate the state of that goal. Note that sometimes the agent may encounter congestion and cannot move to the node in time, in which case the goal is not satisfied and the agent needs to plan a potentially different path to the assembly station. If the goal is satisfied and the node of the goal is the final one on the path, the agent will pursue the goal enter assembly station. Otherwise, it will execute another move to node goal with the next node in the path.

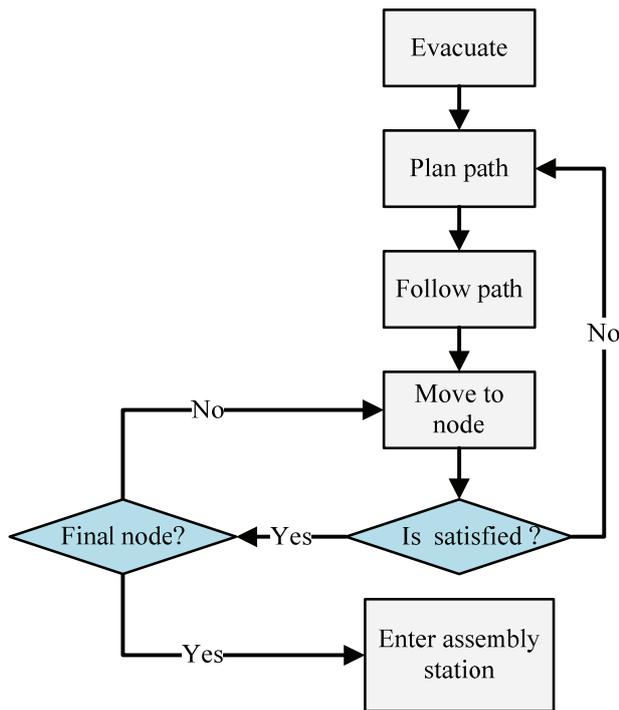


Fig. 11. Diagram of agent pursuing goals.

VALIDATION OF THE EVACUATION MODEL AGAINST IMO TESTS

We have verified the model successfully against 11 tests given in the appendix to the IMO Guidelines [1] for validating/verifying evacuation simulation tools. Tests 1–7 are component ones that check whether the major subcomponents of the model perform as intended. Tests 8–11 involve qualitative verification to ensure that the model is able to produce realistic behavior. The 11 tests proposed by the IMO are listed in Tab. 3. It was not feasible to detail all the tests in this paper, so only tests 4, 6, 9, and 10 are described.

Tab. 3. Tests proposed by IMO for validating/verifying evacuation simulation tools.

Test	Description	Category
Test 1	Maintaining set walking speed in corridor	component testing
Test 2	Maintaining set walking speed up staircase	component testing
Test 3	Maintaining set walking speed down staircase	component testing
Test 4	Exit flow rate	component testing
Test 5	Response time	component testing
Test 6	Rounding corners	component testing
Test 7	Assignment of population demographic parameters	component testing
Test 8	Counterflow: two rooms connected via a corridor	qualitative verification
Test 9	Exit flow: crowd dissipation from a large public room	qualitative verification
Test 10	Exit route allocation	qualitative verification
Test 11	Staircase	qualitative verification

IMO TEST 4: EXIT FLOW RATE

One hundred passengers are distributed uniformly in a room measuring 5 m × 8 m with a 1-m-wide exit located centrally on the 5-m wall (see Fig. 12). The maximum passenger walking speeds are distributed uniformly over 0.97–1.62 m/s. The expectation is for the exit flow rate over the entire period to be ≤1.33 passengers/s/m. The exit flow rate is the number of passengers escaping past a point in the escape route per unit time per unit width of the route involved. Fig. 13 shows the instantaneous and average flow rates of passengers leaving the room for IMO test 4. Note that the average flow rate is 0.91 passengers/s/m and the instantaneous one is <1.33 passengers/s/m during the simulation. Hence, the results are consistent with the expectation.

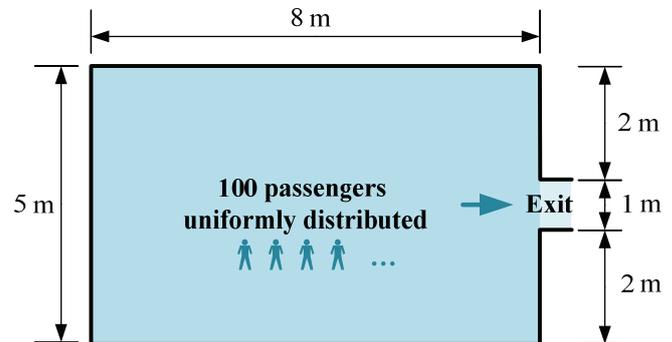


Fig. 12. Arrangement for IMO test 4.

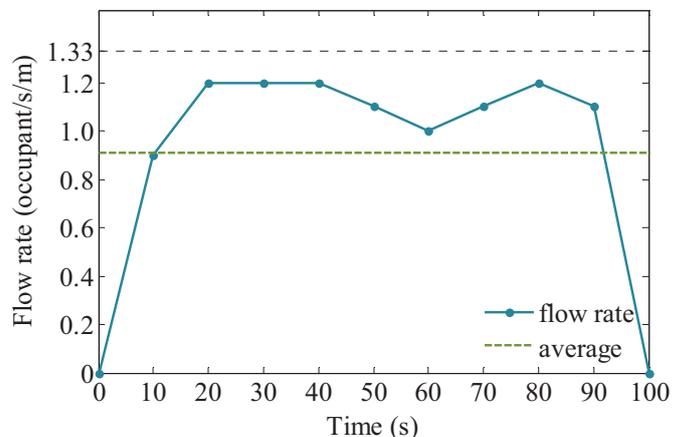


Fig. 13. Instantaneous and average flows rate of passengers leaving the room (IMO test 4).

IMO TEST 6: ROUNDING CORNERS

IMO test 6 involves 20 passengers walking toward a corner in a 2-m-wide corridor. The passengers are distributed uniformly in an area of 2 m × 4 m. The arrangement of the test case is shown in Fig. 14. The expectation is that the passengers will successfully navigate around the corner without penetrating the walls. Fig. 15 shows the passenger trajectories in blue. The simulation results agree well with the

expectation of the test. These movement trajectories verify that all the passengers successfully navigated around the corner.

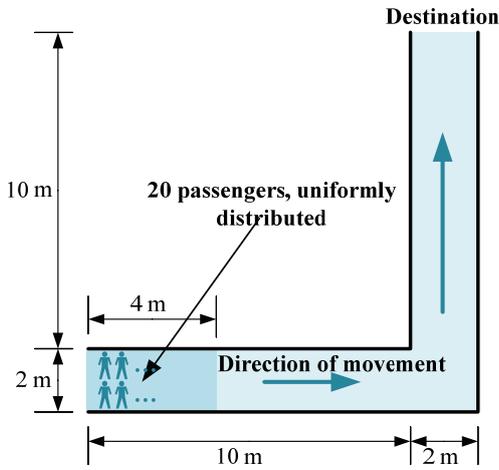


Fig. 14. Arrangement for IMO test 6.

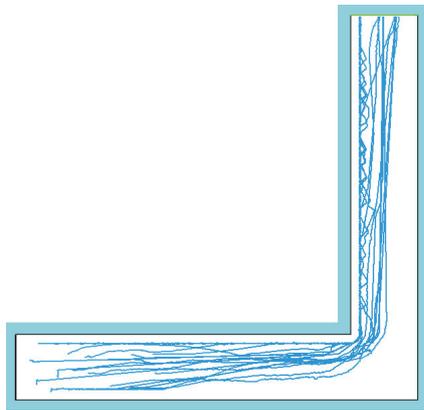


Fig. 15. Simulated trajectories of 20 passengers (IMO test 6).

IMO TEST 9: EXIT FLOW

One thousand male passengers aged 30–50 are located in a public room (30 m × 20 m) with four exits; each passenger leaves via his nearest exit. The passengers are distributed uniformly in the center of the room, 2 m from each wall. As mentioned in the IMO guidelines, their maximum unhindered walking speeds on flat terrain are distributed uniformly over 0.97–1.62 m/s. The arrangement of IMO test 9 is shown in Fig. 16. The test simulates two scenarios: all four exits are available; only two of them are available. The expectation is that the two-exit evacuation time is approximately double that of the four-exit one.

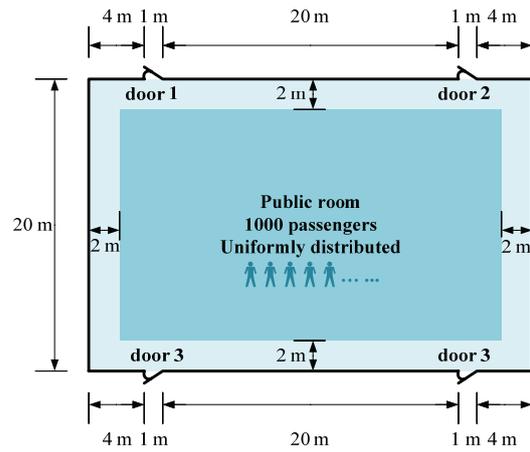


Fig. 16. Arrangement for IMO test 9.

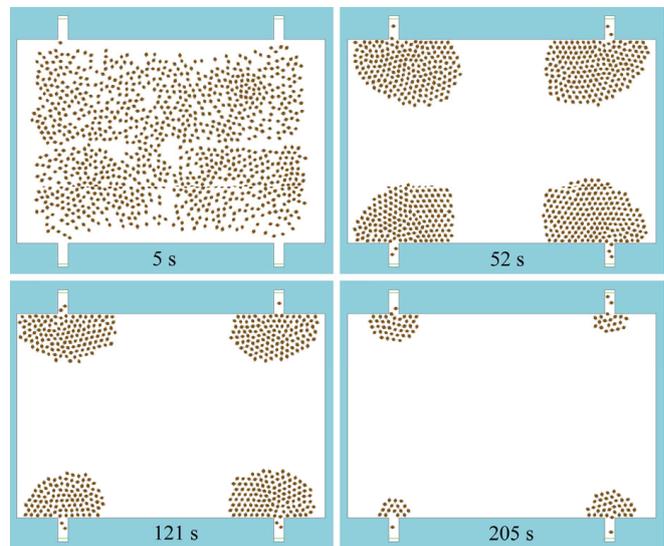


Fig. 17. Snapshots of the four-exit simulation (IMO test 9).

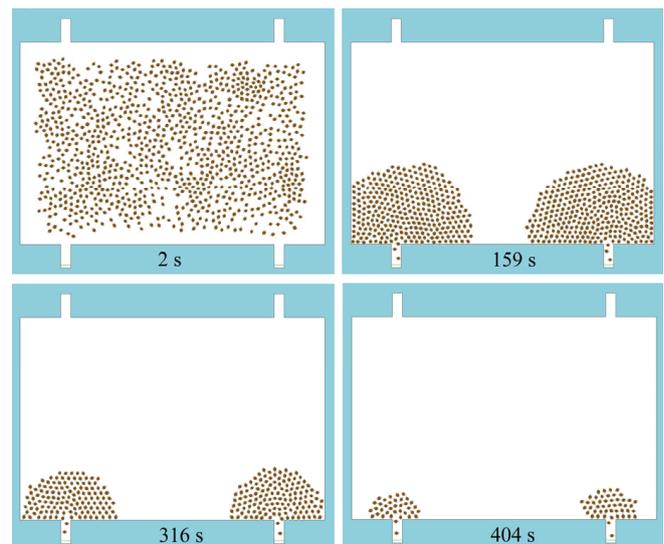


Fig. 18. Snapshots of the two-exit simulation (IMO test 9).

Each test scenario was simulated 10 times. Snapshots from the simulations are shown in Fig. 17 and 18. Fig. 19 and 20 show the maximum, minimum, and mean cumulative percentages of evacuated passengers as functions of time for the four- and two-exit IMO test 9, respectively. The smoothness of each mean curve implies that 10 simulation runs are adequate for evaluating the evacuation times. Fig. 21 shows individual and mean evacuation times for the 10 simulation runs for each scenario. Note that the evacuation times fluctuate within a certain range because of the randomness of the evacuation process.

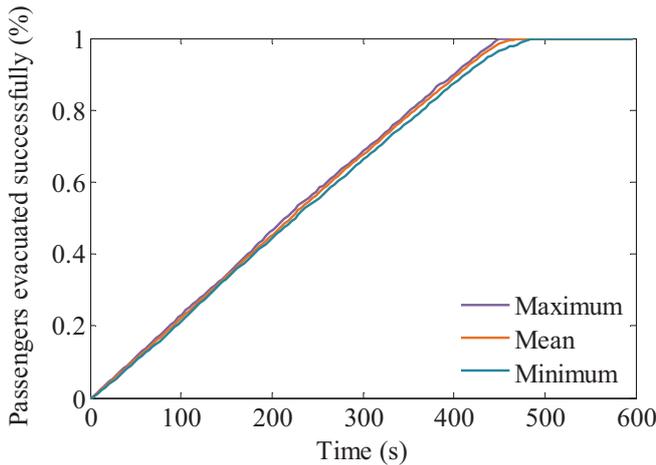


Fig. 19. Cumulative percentage of evacuated passengers (4 exits; IMO Test 9).

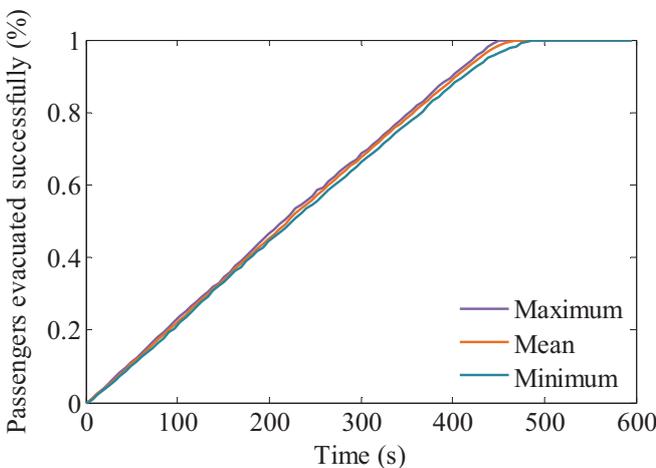


Fig. 20. Cumulative percentage of evacuated passengers (2 exits; IMO Test 9).

Tab. 4 compares the simulation results with those from other models. The evacuation times predicted by the present model are 463.6 s for two exits and 237.9 s for four exits. The ratio of the two evacuation times is 1.95, meaning that the simulation meets the test expectation. It should be noted that the present model gives evacuation times that are 11% less than those given by Pathfinder 2016 (Society of Fire Protection Engineers (SFPE)) [25] but 35% more than those given by SIMPEV [9]. Furthermore, there are insufficient reliable experimental data to evaluate which model is better in predicting the evacuation times.

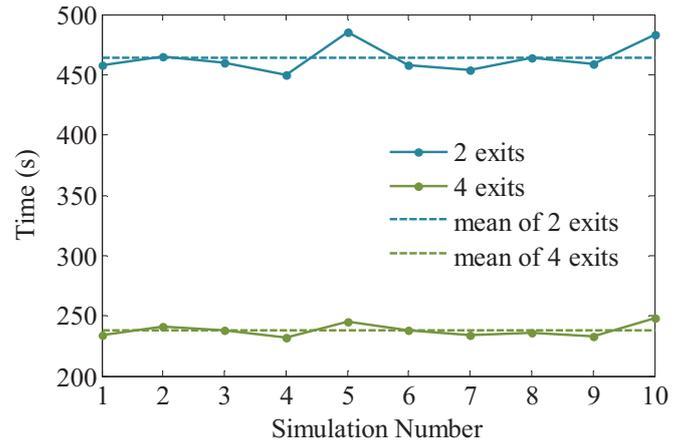


Fig. 21. Evacuation times for each simulation run (IMO test 9).

Tab. 4. Evacuation times and ratios predicted by different models (IMO test 9).

	Evacuation time for 4 exits (s)	Evacuation time for 2 exits (s)	Ratio of the evacuation times
Pathfinder 2016 (Steering) [25]	211	429	2.03
Pathfinder 2016 (Steering+SFPE)[25]	295	584.3	1.98
Pathfinder 2016 (SFPE) [25]	270.2	545	2.02
FDS+EVAC [20]	240.8	420.0	1.74
SIMPEV [9]	155	299	1.92
The present model	237.9	463.6	1.95

IMO TEST 10: EXIT ROUTE ALLOCATION

As show in Fig. 22, a total of 23 passengers are distributed in a cabin corridor section with two exits. The passengers are male, aged 30–50, and their maximum unhindered walking speeds on flat terrain are distributed uniformly over 0.97–1.62 m/s. The passengers in cabins 5, 6, 11, and 12 are assigned to leave via the secondary exit, while those in the other cabins are allocated the main exit. The expectation is that each passenger leaves by his assigned exit.

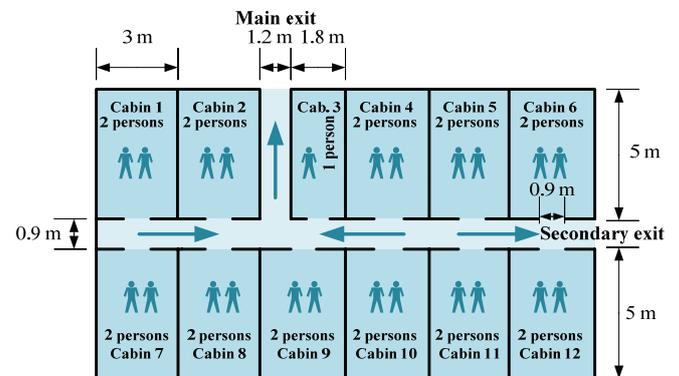


Fig. 22. Arrangement for IMO test 10.

The test case was simulated 10 times. Snapshots from the simulation are shown in Fig. 23. Fig. 24 shows the paths taken by the passengers in one simulation. The trails of the four passengers intended to use the secondary exit are shown in red; all other passenger trails are shown in blue. The simulation results agree well with the expectation of the test.

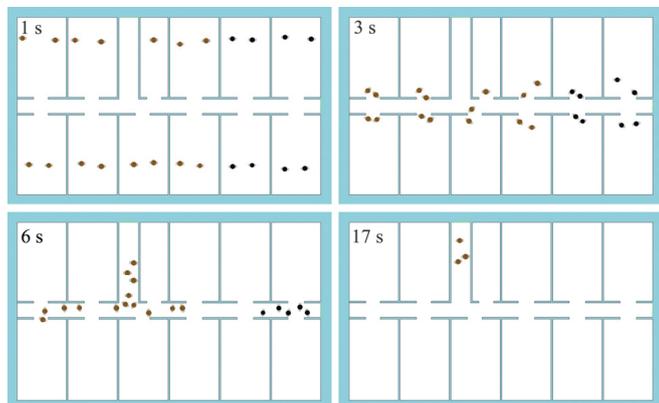


Fig. 23. Snapshots of the simulation (IMO test 10).

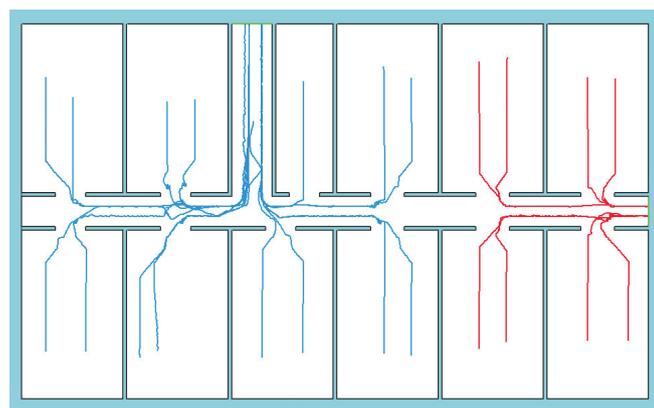


Fig. 24. Simulated trajectories of 23 passengers (IMO test 10).

COMPARISON OF CALCULATION TIMES

All the simulations were carried out on a computer with an Intel 2.5 GHz CPU and 4 GB RAM. Our model is efficient enough to simulate 1,000 agents at >30 frames/s. The calculation times were compared with the SIMPEV model [8] by considering each passenger's characteristics (e.g.; age, gender, etc.). Tab. 5 lists the calculation times for IMO tests 8 and 9 for each model. The calculation times for the proposed model are significantly less than those for the SIMPEV one. However, it is difficult to say which model is more efficient because the computer specifications used for the SIMPEV calculations were not given.

Tab. 5. Comparison of calculation times.

Test	Calculation time for 1 s evacuation simulation	
	SIMPEV model	Our model
IMO test 8 (200 passengers)	0.31 s	0.15 s
IMO test 9 (1,000 passengers)	2.31 s	0.68 s

CONCLUSIONS

This paper presented an agent-based passenger-ship evacuation model that was able to simulate passengers whose walking speeds and body dimensions were different. In the proposed model, the underlying movement model for agents to avoid colliding with walls and to interact with other agents was a combination of social- and steering-force models. The goal-driven decision-making model decomposed an abstract goal into subgoals and satisfied each one in turn. This process mirrored aspects of human deliberation in planning and was flexible in relation to adding further goals. The proposed model was efficient enough to simulate thousands of agents in real time with an Intel CPU. Furthermore, the simulations met all the requirements of the 11 tests proposed by the IMO.

We presented an evacuation model for a passenger ship in order to help designers to improve ship safety in the early design process. However, passenger behavior and goals are much more complex in an actual evacuation. For example, passengers tend to search for missing kinsfolk, find a life jacket, or choose exits with which they are familiar. Furthermore, the 11 tests offered by the IMO are ideal ones in simple compartments. Even though our model satisfied all the test cases, the estimated evacuation times could be different from those in a real evacuation situation. Given these facts, more effort is needed to validate the model and make it more applicable.

ACKNOWLEDGEMENTS

The study is supported by the Natural Science Foundation of China (No. 61004008) and National Science and Technology Major Project of the Ministry of Science and Technology of China (Z12SJENA0014).

REFERENCES

1. IMO: Guidelines for evacuation analysis for new and existing passenger ships. International Maritime Organization, pp. 1–46, 2007.
2. Wolfram, S.: Statistical mechanics of cellular automata. *Reviews of Modern Physics*, 55(3), pp. 601–644, 1983.
3. Burstedde, C., Klauck, K., Schadschneider, A. and Zittartz, J.: Simulation of pedestrian dynamics using a two-dimensional cellular automaton. *Physica A: Statistical*

- Mechanics and its Applications, 295(3–4), pp. 507–525, 2001.
4. Guo, R. Y.: New insights into discretization effects in cellular automata models for pedestrian evacuation. *Physica A: Statistical Mechanics and its Applications*, 400, pp. 1–11, 2014.
 5. Guo, R. Y., Huang, H. J. and Wong, S. C.: Route choice in pedestrian evacuation under conditions of good and zero visibility: Experimental and simulation results. *Transportation Research Part B: Methodological*, 46(6), pp. 669–686, 2012.
 6. Guo, R.-Y., Huang, H.-J. and Wong, S. C.: A potential field approach to the modeling of route choice in pedestrian evacuation. *Journal of Statistical Mechanics: Theory and Experiment*, 2013(2), p. P02010, 2013.
 7. Tang, T. Q., Chen, L., Guo, R. Y. and Shang, H. Y.: An evacuation model accounting for elementary students' individual properties. *Physica A: Statistical Mechanics and its Applications*, 440, pp. 49–56, 2015.
 8. Ha, S., Ku, N. K., Roh, M. Il and Lee, K. Y.: Cell-based evacuation simulation considering human behavior in a passenger ship. *Ocean Engineering*, 53, pp. 138–152, 2012.
 9. Park, K. P., Ham, S. H. and Ha, S.: Validation of advanced evacuation analysis on passenger ships using experimental scenario and data of full-scale evacuation. *Computers in Industry*, 71, pp. 103–115, 2015.
 10. Helbing, D. and Molnár, P.: Social force model for pedestrian dynamics. *Physical Review E*, 51(5), pp. 4282–4286, 1995.
 11. Helbing, D., Farkas, I. and Vicsek, T.: Simulating dynamical features of escape panic. *Nature*, 407(6803), pp. 487–490, 2000.
 12. Yuen, J. K. K. and Lee, E. W. M.: The effect of overtaking behavior on unidirectional pedestrian flow. *Safety Science*, 50(8), pp. 1704–1714, 2012.
 13. Heliövaara, S., Korhonen, T., Hostikka, S. and Ehtamo, H.: Counterflow model for agent-based simulation of crowd dynamics. *Building and Environment*, 48(1), pp. 89–100, 2012.
 14. Xu, M., Wu, Y., Lv, P., Jiang, H., Luo, M. and Ye, Y.: MiSFM: On combination of Mutual Information and Social Force Model towards simulating crowd evacuation. *Neurocomputing*, 168, pp. 529–537, 2015.
 15. Bonabeau, E.: Agent-based modeling: methods and techniques for simulating human systems. *Proceedings of the National Academy of Sciences*, 99(suppl. 3), pp. 7280–7287, 2002.
 16. Tan, L., Hu, M. and Lin, H.: Agent-based simulation of building evacuation: Combining human behavior with predictable spatial accessibility in a fire emergency. *Information Sciences*, 295, pp. 53–66, 2015.
 17. Tang, F. and Ren, A.: GIS-based 3D evacuation simulation for indoor fire. *Building and Environment*, 49(1), pp. 193–202, 2012.
 18. Hart, P. E., Nilsson, N. J. and Raphael, B.: Correction to 'A Formal Basis for the Heuristic Determination of Minimum Cost Paths'. *ACM SIGART Bulletin*, 37(37), pp. 28–29, 1972.
 19. Thompson, P. S. and Marchant, E. W.: Testing and Application of the Computer Model 'SIMULEX'. *Fire Safety Journal*, 24(2), pp. 149–166, 1995.
 20. Korhonen, T. and Hostikka, S.: Fire Dynamics Simulator with Evacuation: FDS+Evac, Technical Reference and User's Guide (FDS 5.5.0, Evac 2.2.1). VTT Technical Research Centre of Finland, 2010.
 21. Reynolds, C. W.: Flocks, herds and schools: A distributed behavioral model. *ACM SIGGRAPH Computer Graphics*, 21(4), pp. 25–34, 1987.
 22. Hartman, C. and Beneš, B.: Autonomous boids. 17(3–4), pp. 199–206, 2006.
 23. Langston, P. A., Masling, R. and Asmar, B. N.: Crowd dynamics discrete element multi-circle model. *Safety Science*, 44(5), pp. 395–417, 2006.
 24. Russell, S. J. and Norvig, P.: *Artificial Intelligence: A Modern Approach*. Third Edit, Prentice Hall, Upper Saddle River, 2010.
 25. *Thunderhead Engineering: Pathfinder: Verification and Validation*. Pathfinder, Thunderhead Engineering, Manhattan, 2016.

CONTACT WITH THE AUTHORS

Baocheng Ni

e-mail: tjm002@qq.com

College of Ship Building
Harbin Engineering University
15001 Harbin

CHINA

Zhen Li

e-mail: 845260698@qq.com

College of Mechanical and Electrical Engineering
No.145 Nantong Street, Nangang District
15001 Harbin

CHINA

Xiang Li

e-mail: 758470971@qq.com

Dalian Neusoft University of Information
No.8 Software Park Road
116023 Dalian

CHINA