Integrated Simulation Approach for Dynamic Distributed Evacuation Guidance Under Fire Spread and Rare but Catastrophic Events

Akira Tsurushima[a]
Intelligent Systems Laboratory, SECOM CO., LTD., Japan

Keywords: Distributed Problem Solving, Multi-Agent Simulation, Average Value at Risk, Crowd Evacuation, Dynamic Evacuation Guidance.

Abstract: The dynamics of smoke, fire, and toxic gases inside a building is complex and difficult to predict. Models have been used to analyze and develop efficient evacuation protocols for fire-spread evacuation situations. In this study, an integrated model that includes fire spread, evacuation agent, and evacuation guidance signage models, is developed as an efficient control mechanism for a dynamically distributed evacuation guidance system. This mechanism is based on Tsurushima's distributed algorithm, which does not assume any central control and only requires local information in providing efficient evacuation guidance, thereby minimizing total evacuation time while directing evacuees away from hazards. The parameters of the mechanism were calibrated to mitigate the occurrence of low-probability catastrophic events, which is crucial in the development of evacuation guidance protocols.

1 INTRODUCTION

A cable car accident occurred on November 11, 2000, in Kaprun, Austria, which resulted in 155 victims. In this case, the heater placed at the lower end of the cable car broke, and the car stopped in the middle of the tunnel because of brake system failure. Many people tried to escape the fire and headed toward the top of the tunnel. All these people died as the toxic gases generated by the fire had moved upward because of the chimney effect. However, a German ex-firefighter knew about the chimney effect and advised some people to escape downward. All 12 passengers who escaped downward, following his advice, survived the fire (Fraser-Mitchell and Charters, 2005; Ferscha and Zia, 2010; Fridolf, 2010).

In this accident, the decision to escape upward or downward was a decisive factor that made the difference between life and death. However, choosing the correct route is difficult because it depends on the location and movement of the fire and toxic gases, which can be very complex and difficult to predict. Effectively guiding crowds to evacuate a building during a fire while considering the spread of fire, smoke, and gas, is critical but challenging.

Numerous studies have been conducted on dynamic evacuation guidance systems to navigate crowds to efficiently evacuate buildings (Mirahadi and McCabe, 2020; Zhao et al., 2022). These systems typically consist of sensors and digital signage. The sensors collect local safety information, such as temperature and smoke or toxic gas, while the digital signage indicates the travel direction at specific locations within the building. Most of these systems generate efficient evacuation routes in the building using optimization algorithms that can maximize or minimize certain performance indices, such as minimizing total evacuation distance or total evacuation time. However, from a fire evacuation perspective, minimizing the travel distance or evacuation time is insufficient because evacuees may be exposed to the smoke or toxic gases generated by the fire while traveling to the exit, which may endanger their life or health. Thus, it is critical that evacuation guidance systems guide evacuees away from danger and minimize total evacuation time (Mirahadi and McCabe, 2020).

The results of a fire can vary unexpectedly, depending on the conditions under which the fire started, such as the location of its origin, structure and materials of the building's interior. Simulations are useful tools for analyzing the variety of outcomes caused by fires; however, a simulation analysis with a handful of scenarios is insufficient to cover the wide variety of possibilities, which can lead to unexpected and

[a] https://orcid.org/0000-0003-2711-297X

Tsurushima, A.
Integrated Simulation Approach for Dynamic Distributed Evacuation Guidance Under Fire Spread and Rare but Catastrophic Events.
DOI: 10.5220/0012416100003636

In *Proceedings of the 16th International Conference on Agents and Artificial Intelligence (ICAART 2024) - Volume 1*, pages 105-116
ISBN: 978-989-758-680-4; ISSN: 2184-433X

catastrophic outcomes not considered in these scenarios. Thus, a systematic method for analyzing rare but catastrophic events (RBCE) is critical in designing effective disaster evacuation protocols.

In this study, we propose a dynamic evacuation guidance method that efficiently guides evacuees to exits while considering exposure to smoke or toxic gases that dynamically spread during evacuation. This method is based on Tsurushima's distributed evacuation guidance algorithm that is resilient to disaster environments because it has no single point of failure, such as a centralized server, and requires only local information (Tsurushima, 2024). The contributions of this study are as follows:

1. The simulations on fire spread, agent-based evacuation, and distributed dynamic evacuation guidance are integrated into a single coherent platform;
2. Tsurushima's distributed algorithm is applied to fire evacuation guidance;
3. A parametric calibration method is introduced into the algorithm to avoid the occurrence of RBCE.

To address the problem of minimizing evacuation time while ensuring that evacuees avoid harmful exposure, this study uses two objective functions and assumes that all components function as intended.

2 RELATED WORK

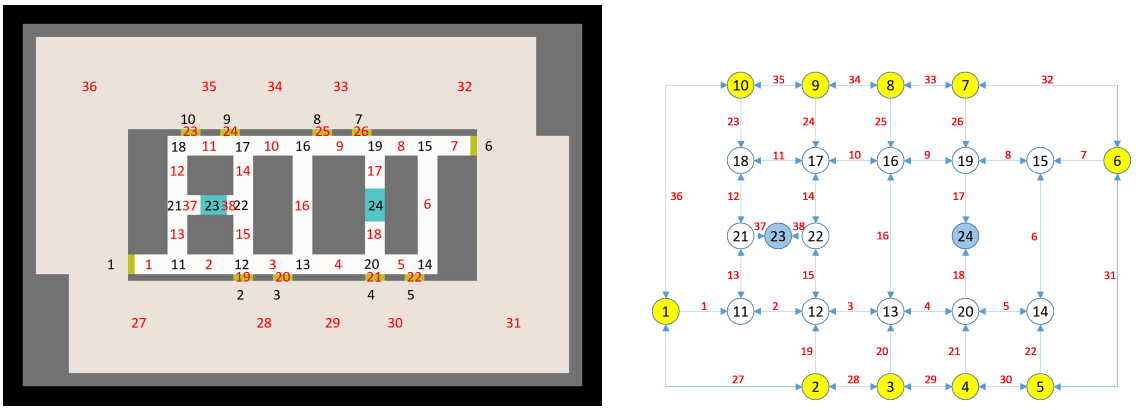
The dynamics of fire and smoke spreading in a building are complex and difficult to predict. Computational fluid dynamics and large eddy simulation models developed to reproduce these complex phenomena are based on the computationally expensive Navier-Stokes equations. These models, combined with evacuation simulation models, can accurately reproduce fire dynamics and smoke in a building, which is useful for analyzing the safety of evacuees in case of fire. Using this combination, Jasztal et al. evaluated the Warsaw Modlin Airport to determine whether evacuees could safely exit the building within the allotted time (Jasztal et al., 2022). Liao et al. evaluated the safety of the underground space of Guangzhou International Financial City and suggested improvements for safety (Liao et al., 2023). Coskun et al. assessed the health of evacuees from the Sakarya University building using fractional effective dose (Coşkun et al., 2022). These studies used the PyroSim software for fire-spread simulation and PathFinder for evacuation simulation. Other researchers have used a combination of fire spread and evacuation simulations for more general purposes. Zheng et al. based their simulations on the floor field model (FFM), to model human evacuation behavior under fire (Zheng et al., 2017). Lee et al. also used this combination to model evacuation decisions based on fire spread (Lee et al., 2018).

The aforementioned studies used fire spread and evacuation simulations for analysis purposes. Mirahadi et al. used this combination to develop a decision support system—EvacuSafe—that can provide safe evacuation. EvacuSafe simulates several fire scenarios to evaluate various evacuation strategies and selects the best evacuation strategy when an emergency is identified. This strategy is then sent to the active dynamic signage system, to provide efficient evacuation guidance to evacuees (Mirahadi and McCabe, 2020).

In this study, fire-spread and evacuation agent models were used to simulate crowd evacuation under fire conditions. Our evacuation agents follow only the evacuation signs and never intentionally flee from the fire. Dynamic signs are incorporated into the model to advise evacuees of specific evacuation directions. An efficient control mechanism is investigated for evacuation signs to safely guide evacuees to the exits, using an integrated simulation model that includes fire-spread, evacuation agent, and evacuation signs models.

3 PROBLEM

Fig. 1 (A) illustrates the evacuation environment (Tsurushima, 2024) filled with cells $(x,y) \in C$ identified by the coordinates $(x,y) \in \mathbb{Z}_1 \times \mathbb{Z}_2$ where $\mathbb{Z}_1 = -40,\ldots,40$, $\mathbb{Z}_2 = -25,\ldots,25$ represent the positions in the environment used in this study. The environment is divided into two parts: the central core (dark gray square region) and the surrounding space (light brown region) outside the central core, which is typical of high-rise buildings. The central core consists of two exits 23 and 24 (blue), aisles (white), and corners (white with numbers). Doors connecting the surrounding space and aisles in the central core are shown in yellow in the figure, indicating evacuees in the surrounding space to move to the aisles and exits. Evacuation signs that direct evacuees at each location are assigned to the doors $(1,\ldots,10)$, corners $(11,\ldots,22)$, and exits (23 and 24). The aisles and surrounding space are divided into 38 areas $C = \bigcup_{i=1}^{38} C_i$ and represented by the red numbers: 1 to 18 and 27 to 36, respectively. The former represents the space between two corners. Note that 19 to 26 represent small areas occupied by doors, and 37 and 38 represent small areas between the two corners (21 and 22)



(A) (B)

Figure 1: (A) Example floor plan. (B) Graphical representation of the floor plan.

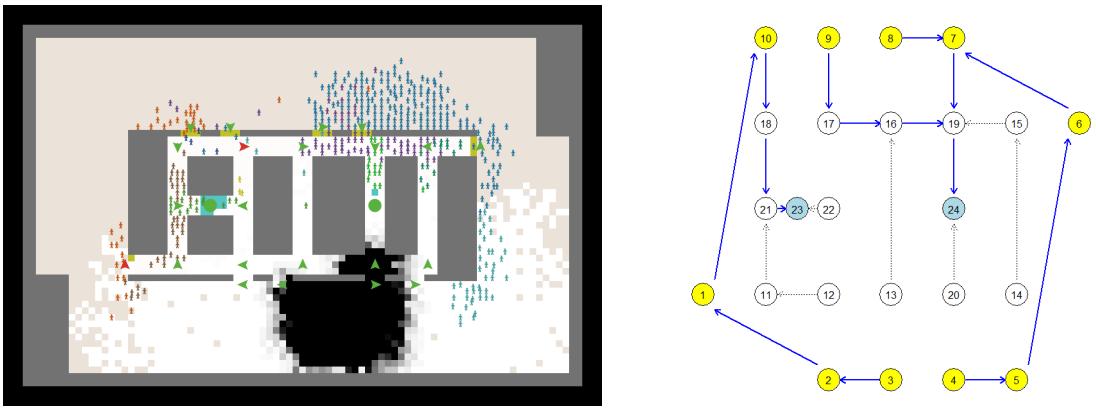
and exit 23. Although not clearly illustrated, these areas have sensors to identify the level of danger from heat, smoke, or toxic gases produced by the fire. The red numbers represent the values of the sensors associated with these areas.

Fig. 1 (B) presents universal evacuation graph $G = (V, E, W)$ of Fig. 1 (A) (Tsurushima, 2024). The nodes ($v_i \in V$) of the universal evacuation graph represents doors ($v_1, \ldots, v_{10}$), corners ($v_{11}, \ldots, v_{22}$), two exits ($V_g = \{v_{23}, v_{24}\}$), and edges ($\{v_i, v_j\} = e_k \in E$) represents the areas C_k between the two adjacent nodes. $I_C : \{1, \ldots, |V|\}^2 \rightarrow \{1, \ldots, 38\}$ maps the indices $\{i, j\}$ to the index k, e.g., $I_C(1,2) = 27$. Directed edges also represent the possible directions that an evacuee at the node can take in the next trip. Another interpretation of the universal evacuation graph is that a node represents the evacuation sign at the location, and an edge represents the value of the sensor corresponding to the area, as indicated by the red number. The direction of an edge, in this interpretation, is that of the adjacent nodes that the evacuation sign advises to evacuees at the location as the next direction. For example, at door 1, an evacuee has three choices: move to door 2, door 10, or corner 11; thus, the evacuation sign on door 1 displays either 2, 10, or 11. In universal evacuation graph, an edge weight ($w_{(i,j)} \in W$) represents the hazard level that the corresponding sensor reports as its value $w_{(i,j)} \in \mathbb{R}+$.

Initially, 800 agents $A = \{a_1, \ldots, a_{800}\}$ are randomly assigned to the cells in the surrounding space, with one cell accommodating at most one agent. One of the cells in the surrounding space is also randomly selected as the fire starting point. The fire propagates to adjacent cells with some probability and spreads to the other cells as the simulation progresses. Agents representing the evacuees begin to evacuate the surrounding spaces through the doors, aisles, and corners of the exits according to the directions displayed on the evacuation signs.

Fig. 2 (A) illustrates an instance from a sample simulation. The small green arrowheads on the doors and corners illustrate the evacuation signs, and the direction of an arrowhead depicts the guiding direction displayed on each sign. The two green circles illustrate the evacuation signs on the exits, indicating the positions of the exits. Black (or dark gray) regions in the lower part of the surrounding space represent the area occupied by the fire, and the white regions around the fire represent the area of penetrating smoke. Fig. 2 (B) illustrates the evacuation graph $\dot{G} = (V, \dot{E}, \dot{W})$ (Tsurushima, 2024), which represents the evacuation routes suggested by the evacuation signs, corresponding to (A). The evacuation graph consists of two trees with the roots denoting exits. The blue arrows in (B) depict the route from the doors to the exits. The evacuation graph in (B) has no loops, which also indicates that an efficient evacuation graph $\ddot{G} = (V, \ddot{E}, \ddot{W})$ (Tsurushima, 2024). $\delta : V \rightarrow V,\ i \in V,\ (i, \delta(i)) \in \dot{E}$ denotes the direction (node) displayed on evacuation sign i, and $\Delta : V \rightarrow 2^V,\ j \in \Delta(i),\ (i, j) \in E$ denotes the possible directions that evacuation sign i can display; $\Delta(i)$ denotes the set of nodes adjacent to i.

The problem lies in finding an efficient control mechanism for evacuation following the suggestions in the signs for the best direction for evacuees to escape at each location. The control mechanisms were evaluated based on total evacuation time and total amount of hazard to which the evacuees were exposed during the evacuation.



(A) (B)

Figure 2: (A) Sample simulation. (B) The evacuation graph corresponding to (A).

4 TSURUSHIMA'S ALGORITHM

Consider the universal evacuation graph given in Fig. 1(B). Each node in the graph represents a evacuation sign, which is also a distributed computational process in which the system can communicate with its neighboring nodes. Incorporating Tsurushima's distributed algorithm allows the generation of the optimal evacuation direction at the corresponding location. In Tsurushima's algorithm, each node can independently calculate the optimal direction at that node and automatically coordinate all evacuation routes to form efficient evacuation graph $\ddot{G} = (V, \ddot{E}, \ddot{W})$, eliminating harmful loops that result in evacuee congestion (Theorem 4 in (Tsurushima, 2024)). The algorithm comprises two procedures: *Broadcast* distributes the newest local information throughout the graph to form a local image of the global graph $\widehat{G} = (V, \widehat{E}, \widehat{W})$ at each node, and *UpdateSign* computes the optimal direction at that node based on $\widehat{G}$.

Tsurushima's algorithm is a distributed optimization algorithm that minimizes the total path weight (TPW) on an universal evacuation graph. Any efficient evacuation graph can be represented as a set of trees. For example, $\ddot{G} = \{\mathcal{T}_1, \ldots, \mathcal{T}_{|v_g|}\}$, where $\mathcal{T}$ represents a tree with an element in V_g as the root (Lemma 1 in (Tsurushima, 2024)). TPW of $\ddot{G}$ is defined as $TPW(\ddot{G}) = \sum_j TPW(\mathcal{T}_j)$ and TPW of $\mathcal{T}$ is defined as $TPW(\mathcal{T}) = \sum_{i \in V \setminus \{g\}} weight(\mathcal{T}(i,g))$, where $g \in V_g$, and $weight(\mathcal{T}(i,j))$ is the total path weight from i to j for $\mathcal{T}$. Applying Tsurushima's algorithm repeatedly and independently at each node, the system automatically generates a series of efficient evacuation graphs that finally converge to an efficient evacuation graph with minimum TPW, which is called the minimum efficient evacuation graph G^*, unless E and W remain unchanged (Theorem 5 in (Tsurushima, 2024)).

Algorithm 1: Broadcast (Tsurushima, 2024).

Local variable: $\widehat{E}, \widehat{W}, \widehat{T}$
1 **Function** `Broadcast`$(o,\ i,\ \delta(o),\ t,\ \Omega_W,\ \Upsilon_W)$**:**
2 **if** $\widehat{T}[o] < t$ **then**
3 $\widehat{T}[o] \leftarrow t$;
4 $\widehat{E}[o] \leftarrow \delta(o)$;
5 **foreach** $j \in 1, \ldots, |\Omega_W|$ **do**
6 $\widehat{W}[\Omega_W[j]] \leftarrow \Upsilon_W[j]$
7 **end**
8 **foreach** $a \in \Delta(i)$ **do**
9 call `Broadcast`$(o, a, \delta(o), t, \Omega_W, \Upsilon_W)$ on node a
10 **end**
11 **end**
12 **end**

As shown in Algorithm 1, *Broadcast* updates the local infomation $\widehat{E}$ and $\widehat{W}$ as new information arrives from other nodes and distributes this information to adjacent nodes. *UpdateSign* searches for the minimum weight path to evacuation exits using breadth-first search, to determine the evacuation direction, as shown in Algorithm 2. If the new direction forms a cycle, the process is delegated to the next node to break the cycle. For details, see (Tsurushima, 2024).

Algorithm 2: UpdateSign (Tsurushima, 2024).

Local variable: $\widehat{E}, \widehat{W}, \delta(i)$
Function `UpdateSign(`i`)`**:**
  $l^*_{ig} \leftarrow$ `Search(`i, $\widehat{W}$`)`;
  $(i,\ next) \leftarrow l^*_{ig}[1]$;
  $\widehat{E}[i] \leftarrow (i, next)$;
  if $\widehat{E}$ *include a cycle* **then**
    call `UpdateSign(`*next*`)` on *next*;
    await $\widehat{E} \leftarrow$ `UpdateSign(`*next*`)` from *next*
  end
  $\delta(i) \leftarrow \widehat{E}[i]$;
  return $\widehat{E}$
end

5 MODEL

To investigate efficient control mechanisms for evacuation signs, we developed three models: the fire spread, evacuation agent, and evacuation sign models. We combined these three models to form a single coherent simulation model to represent our problem. The simulation runs for a simulation time within the range of $t = 1, \ldots, T_{max}$, the time at which the last agent evacuated from the environment.

5.1 Fire Spread Model

Many advanced fire spread models have been developed considering geometric shapes of the building, materials, density, thermal conditions of indoor objects, ventilation conditions, and the locations of fire extinguishers as inputs, to study the dynamics of fire spreading. Most models calculate the complex dynamics of fires using computational fluid dynamics and large-eddy simulations based on the Navier-Stokes equations. These models can produce accurate results at high computational costs that are sometimes unaffordable in machine learning or black-box simulations. In this study, we developed a simple pseudo-fire-spread model to avoid high computational cost. The model produces a somewhat imprecise but reasonably well-simulated fire-spread dynamics in buildings with low computational cost. The model simply propagates the value $h^t_{(x,y)} \in \mathbb{R}$ for a cell (x,y), which represents the fire hazard level of that cell at time t to neighboring cells $\zeta(x,y)$. With probability p_1, the value $h^t_{(x,y)}$ is updated as follows:

$$h^t_{(x,y)} = (1-p_2)\ h^{t-1}_{(x,y)} + p_2\ max\{h^t_{(v,w)\in\zeta(x,y)}\},$$

where $\zeta(x,y)$ maps (x,y) to a set of adjacent cells. In Fig. 2 (A), cells with $h_{(x,y)} \geq 100$ are illustrated in black, with $100 > h_{(x,y)} > 0$ shown in gray to white, depending on the values.

A fire origin (x,y) was randomly selected from the cells in the surrounding space at the beginning of the simulation ($t = 1$), and $h^0_{(x,y)}$ was set to 100000, $p_1 = 0.07$, and $p_2 = 0.2$.

5.2 Evacuation Agent Model

To evaluate the effects of the evacuation signs, a simple agent model was employed to represent an evacuee. Unlike humans, an agent is not intelligent and makes only random decisions and cannot avoid hazardous situations by itself. Following a evacuation sign within its field of view, is the only thing that an agent can do besides a random choice. We assumed a narrow field of view for our agents so that they would fit the narrow aisles inside the central core, such as 20° within a distance of 10 cells, according to the tunnel vision hypothesis (Tsurushima, 2021; Tsurushima, 2022c). The size and shape of an agent's field of view is only considered when the agent sees an evacuation sign. The field of view can be disregarded to narrow down the number of alternatives in random selection.

An agent has a short-term goal, which is a door if the agent is in the surrounding space or a corner or exit if it is in the central core. The agent is always facing and trying to approach this short-term goal. The short-term goal is updated to a new goal under the following three scenarios: when the short-term goal is reached, an evacuation sign is in its field of view, or the agent is physically pushed into a position where the short-term goal is invisible to the agent. The short-term goal is updated to $\delta(i)$ if evacuation sign i is in its field of view, or otherwise randomly selected from adjacent corners or visible doors. When the simulation starts, the short-term goal is randomly set to a door that is physically visible from the agent's position without any obstacles.

After the short-term goal is set, the agent moves by one cell toward the short-term goal at each simulation step unless no other agent occupies the cell. If the cell is occupied, the agent randomly selects an adjacent unoccupied cell and moves to that cell, which is equivalent to the agent being physically pushed by the crowd. If there are no adjacent unoccupied cells, the agent remains in its original position.

Hazard contamination is assumed if an agent is exposed to cells with $h > 0$. Contamination of each agent accumulates over the simulation time, whereby the accumulated contamination of agent a is calcu-

lated as: $\theta_a = \sum_{t=0}^{T_{max}} min\{100,\ h^t\}$, representing the total hazard to which the agent is exposed during the evacuation; $\bar{\theta} = \frac{1}{|A|} \sum_{a \in A} \theta_a$ denotes the mean contamination of all the agents.

5.3 Evacuation Sign Model

The role of the evacuation signs in the simulation is to guide the evacuees to achieve an efficient evacuation, which has two objectives: minimizing the total evacuation time (*min* T_{max}) and mean hazard contamination (*min* $\bar{\theta}$).

After starting the simulation, an evacuation sign incorporating Tsurushima's algorithm starts to execute *Broadcast* and *UpdateSign* repeatedly at different times to achieve the two objectives, resulting in a series of efficient evacuation graphs that lead the agents to the exits. Theorem 5 in (Tsurushima, 2024) ensures that the minimum efficient evacuation graph is achieved if W is static; however, in real evacuation situations, this is unlikely to occur because of the rapid changes in W. Tsurushima used simulations to demonstrate that the algorithm can effectively guide evacuees to exits by dynamically changing W.

Each evacuation sign i knows universal evacuation graph $G = (V, E, \widehat{W}(t))$ and its neighboring nodes $\Delta(i)$, with the current weights of universal evacuation graph $\widehat{W}(t)$ representing the local image of W that is delivered and updated by *Broadcast* . Among other local variables, $\widehat{E}$ denotes the local image of the evacuation graph; $\widehat{T}$ denotes the array containing the receiving time of the last *Broadcast* at all the nodes; and $\delta(i)$ is the current guiding direction of i. Suppose that the evacuation sign 1 in Fig. 2 (A) is where the spread of *Broadcast* originates at time 160, executing

```
Broadcast(1, 1, 10, 160, [2, 10, 11],
          [0.2, 0.01, 0.19]),
```

where 0.2, 0.01, and 0.19 are the edge weights between node 1 and nodes 2, 10, and 11, respectively; *UpdateSign* may be executed at different times,

```
UpdateSign(1).
```

Tsurushima's algorithm was used to minimize the total evacuation time by reducing the congestion between each node and the exits (Tsurushima, 2024). However, guiding evacuees to avoid hazards is difficult for the algorithm, that is, *min* $\bar{\theta}$, because the hazard to be avoided is usually not located between an evacuation sign and an exit, but the evacuation sign is located between the hazard to be avoided and the exit; this is difficult to handle. The algorithm, particularly at the beginning of the simulation, provides the shortest path from each node to the exit because *Search* in line 2 of *UpdateSign* searches for the shortest route using a breadth-first search, as almost all the elements in $\widehat{W}(t)$ are zero. The weights of the edges included in the routes are zero, and the high-weight value on the hazardous edge is disregarded and excluded from the evacuation routes, which is not sufficient for evacuees to avoid hazards, as they need to *move away* from the hazard. The high weights on the hazardous edges do not affect the results. In the following section, we propose a method to address this problem, which is particularly relevant to fire scenarios.

6 METHOD

In this section, we present a method that adapts Tsurushima's algorithm for fire spread evacuations. This method depends on the way the node weights W are computed without modifying the original algorithm. The easiest method of adapting Tsurushima's algorithm to our problem would be to use the mean hazard level of area i as the weight, e.g., $w_i(t) = \frac{1}{|C_i|} \sum_{(x,y) \in C_i} h^t_{(x,y)}$. However, this approach does not suit our purpose because the evacuation signs cannot lead evacuees away from the fire. Many evacuees move into the fire to follow the least-cost path during evacuation.

Fig. 3 illustrates the results of the three simulations, each with 1000 runs for different Ws. The x-axis represents T_{max}, and the y-axis represents $\bar{\theta}$. Fig. 3 (A) presents the results for $w^A_i(t) \in W^A(t)$ such that $w^A_i(t) = \frac{1}{|C_i|^2} \sum_{(x,y) \in C_i} h^t_{(x,y)}$. We divide the mean of $h^t_{(x,y)}$ by $|C_i|$ because a large area is preferable to a small area in providing a safe evacuation space for evacuees. The figure shows a wide range of outcomes for both T_{max} and $\bar{\theta}$. In particular, the values of $\bar{\theta}$, the mean total contamination of the hazard, vary from 0 to 655.5, which can be a serious problem in real-world situations. The large variance in $\bar{\theta}$ stems from the fire start location. If the location is close to the outer edges of the surrounding space, then evacuees can evacuate safely and quickly; however, if it is near the doors or the outer edges of the central core, many evacuees are trapped in the fire early in the simulation.

Fig. 3(B) shows the results of 1000 simulations with $w^B_i(t) \in W^B(t)$ such that $w^B_i(t) = w^A_i(t) + \gamma\ max\{\zeta(w_i)\}/|C_i|$, where $\zeta : W^B(t) \rightarrow 2^{W^B(t-1)}$ which maps the weight of the edge i to the previous weights of the neighboring edges of i, and γ is a scaling factor. For example, in Fig. 1(B), if $i = 1$, $\zeta(w_1)$ denotes a set of weights for the edges $2, 13, 27$, and 36 at time $t-1$. The first term represents the current value obtained from the sensor, whereas the latter

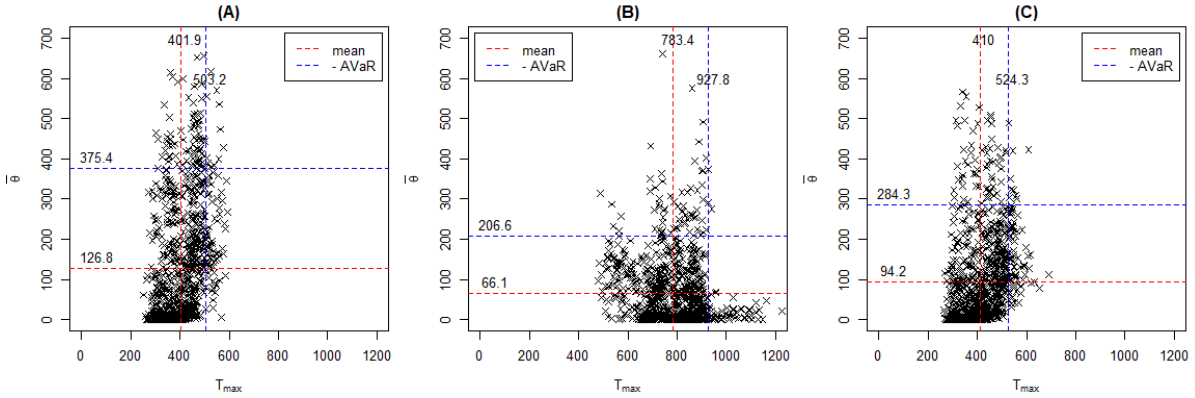


Figure 3: 1000 simulation results with W^A, W^B, and W^C, in (A), (B), and (C), respectively.

denotes the value held in memory inside the evacuation sign. The latter term propagates the weight of an edge to its neighboring edges with some delay, indicating that the neighboring areas of high-hazard areas also have certain risks that should be avoided during evacuations. This term prevents evacuees from being exposed to high-risk cells in the early phases of the simulation.

The vertical and horizontal red dashed lines represent the mean values of T_{max} and $\bar{\theta}$, respectively. Fig. 3 (B) shows that the large variance of $\bar{\theta}$ in (A) is suppressed by substituting the large variance of T_{max} because of the heavy congestion of evacuees caused by the guidance; the mean value of $\bar{\theta}$ in (A) decreases from 126.8 to 66.1 in (B) with the increase in T_{max} from 401.9 to 783.4, which leads to another problem, a long evacuation time.

$W^A(t)$ and $W^B(t)$ can be considered two extremes of $W(t)$; the former is suitable for T_{max} but not for $\bar{\theta}$, whereas the latter is suitable for $\bar{\theta}$ but not for T_{max}. In practice, a compromise between the two extremes is sometimes preferable, to obtain a moderate solution. Fig. 3 (C) shows the results of 1000 simulations with $w_i^C(t) \in W^C(t)$ compromising the two extreme weights $W^A(t)$ and $W^B(t)$ such that

$$w_i^C(t) = \lambda(t)\, l_\xi(w_i^B(t)) + (1-\lambda(t))W_i^A(t),$$

where

$$l_\xi(x) = \begin{cases} x & \text{if } x \geq \xi \\ 0 & \text{otherwise,} \end{cases}$$

and

$$\lambda(t) = \beta\, \lambda(t-1).$$

l_ξ is a function that truncates a variable less than ξ to zero, and λ is a dynamic scaling factor that coordinates W^A and W^B, which are varied over the course of the simulation.

Fig. 3(C) shows that the variances in both T_{max} and $\bar{\theta}$ are suppressed against W^B and W^A, respectively. The means of T_{max} and $\bar{\theta}$ are 94.2 and 410.0, which are smaller than the 126.8 of W^A and 783.4 of W^B, indicating that a reasonable compromise has been reached.

In this section, we use the notation $W(t)$ for simplicity although it is a local image $\widehat{W(t)}$ within a evacuation sign and introduce two additional local variables $\lambda(t)$ and $\widehat{W(t-1)}$ to implement W^C for each evacuation sign.

The parameters used in Fig. 3 (C) are as follows: $\lambda(0) = 0.9$, $\beta = 0.6$, $\xi = 0.5$, and $\gamma = 0.8$.

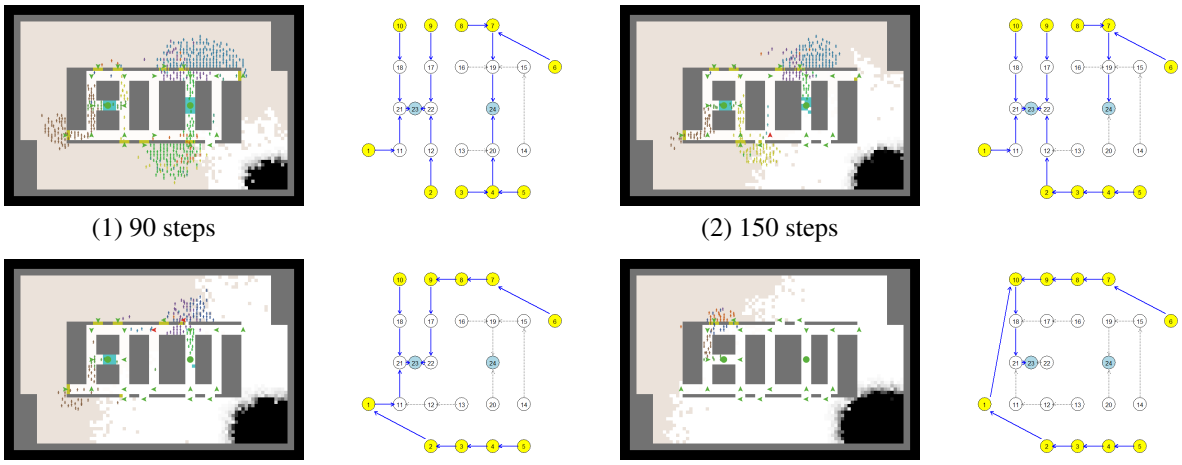
6.1 Case Study

A simulation example using the aforementioned parameters is presented in Fig. 4. The simulation screens for Steps 90, 150, 200, and 260 are presented in (1), (2), (3), and (4), respectively. The corresponding evacuation graph representing the displayed direction of each evacuation sign is shown on the right side of each screen. The fire originated at (36, -25), which is close to the lower right corner in Step 1.

In Step 90, the fire area (black region in the lower right of Fig. 4 (1)) expands, and the agents are evacuated according to the evacuation signs; evacuation graph (1) illustrates that both exits 23 and 24 are used for swift evacuation; doors 1, 2, 9, and 19 are used to access exit 23; doors 4 and 7 are used to access exit 24.

In step 150, the smoke penetrates the right half of the lower area (the white area in Fig. 4 (2)), preventing the use of door 4. The evacuation graph in Fig. 4 (2) shows that door 4 is no longer used to access exit 24, and the evacuation sign at door 4 indicates that evacuees are required to move to the left. Access to exit 24 is still available through door 7.

In step 200, the fire and smoke spread into the wider area, interfering with the use of door 7, and exit 24 is no longer used for evacuation. The evacuation sign at door 2 points to the left to direct evacuees to



(1) 90 steps (2) 150 steps

(3) 200 steps (4) 260 steps

Figure 4: Simulation screens with 90, 150, 200, and 260 simulation steps. Corresponding evacuation graphs are shown to the right of each screen.

door 1. Access to exit 23 is still available through doors 1, 9, and 10.

In step 260, most of the surrounding space is filled with smoke. Access to Exit 23, currently the only exit used for evacuation is available only through door 10, which is located on the side opposite to where the fire originated.

This example illustrates that many doors are used for swift evacuation during the early stages of the simulation. Subsequently, based on the dynamics of fire and smoke evacuation graph rapidly changes its topology, suppressing congestions at certain doors. The few evacuees remaining in the final stage are guided to the safest door opposite to where the fire originated. T_{max} and $\bar{\theta}$ in this example are 337 and 0.5, respectively.

7 RARE BUT CATASTROPHIC EVENTS

In Section 6, we discussed the variance of the results of the fire spread evacuation simulations. This variance will lead to serious consequences in real-world situations. For example, in Fig. 3(A), the worst value of $\bar{\theta}$ indicates the mean hazardous contamination level of 800 evacuees as 655.5. The fire origin in this case is $(6,-16)$, located between doors 3 and 4 and close to the central core, as shown in Fig. 1(A),which can significantly expose evacuees to hazard. If we consider the set of contamination levels for evacuees exposed to hazardous cells to be $\Theta^+ = \{\theta_a | \theta_a \geq 1.0\}$, we have $|\Theta^+| = 227$, *mean* $\Theta^+ = 2310.1$, and *max* $\Theta^+ = 7859.1$; 277 of the 800 evacuees are severely damaged by the fire, which would be unacceptable in many cases. Although such cases are rare, a method for dealing with such RBCE is required. The W^C in the previous section can be used to coordinate the two extremes W^A and W^B and suppress the variances of the two objectives T_{max} and $\bar{\theta}$; however, the results still vary depending on four parameters: $\lambda(0)$, β, ξ, and γ. For example, Fig. 5(D) shows the results of the simulation for $\lambda(0) = 1.0$, $\beta = 0.9$, $\xi = 0.00001$, and $\gamma = 0.8$, and (E) presents the results for $\lambda(0) = 0.1$, $\beta = 0.01$, $\xi = 0.1$, and $\gamma = 0.2$; the former has a large variance in $\bar{\theta}$ and the latter in T_{max}, which are considered as two extreme cases of different parameters for W^C. These parameters depend on the size and shape of the target evacuation environment, and choosing good parameters is difficult. Thus, techniques are required for the systematic calibration of these parameters.

Tsurushima introduced a method that coordinates two objective functions, while reducing the risks—unfavourable extreme outcomes—in these objectives (Tsurushima, 2022a; Tsurushima, 2022b). This method is based on a mean risk model which uses the average value at risk (AVaR), as the risk measure, and has been applied to solve the visual evacuation signage assignment problem by generating a set of Pareto optimal solutions that coordinate the two objective functions and suppress the risk measures. In this section, we discuss Tsurushima's method for calibrating the four parameters of W^C, leading to an efficient control mechanism for evacuation signs to facilitate safe evacuation guidance.

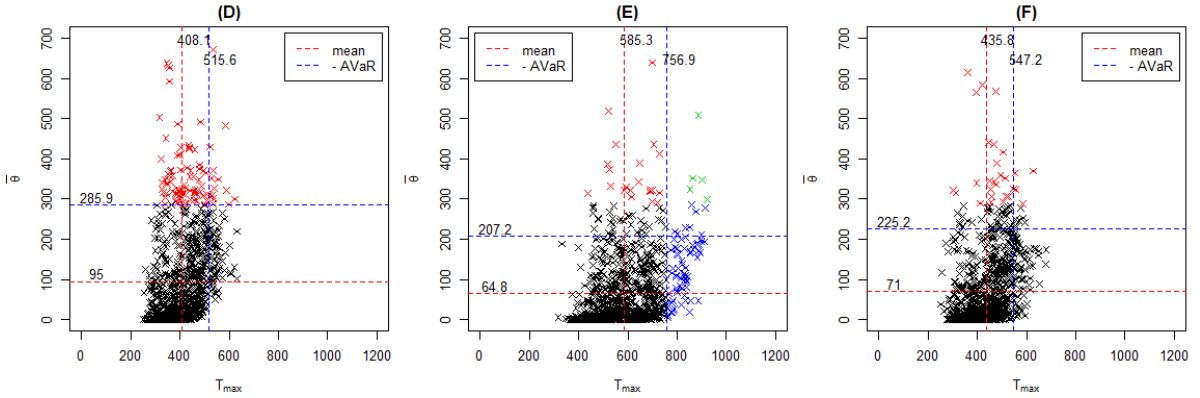


Figure 5: Results with different W^Cs. (D) and (E) are the results with arbitrarily chosen parameters, and (F) displays the results obtained by Tsurushima's method.

7.1 Average Value at Risk

The AVaR, which is a coherent and widely used risk measure in economics and finance, is consistent with the maximum expected utility principle and second-order stochastic dominance (Gutjahr and Pichler, 2016). The AVaR of a random variable X at level α $(0 < \alpha \leq 1)$ is defined based on value at risk (VaR), which is the α quantile of the random variable X:

$$AVaR_\alpha(X) = \frac{1}{\alpha}\int_0^\alpha VaR_p(X)dp \quad (1)$$

$$= E[X|X \leq VaR_\alpha(X)], \quad (2)$$

where

$$VaR_\alpha(X) = F_X^{-1}(\alpha). \quad (3)$$

Intuitively, $AVaR_\alpha(X)$ can be considered the expected value of the realizations of the random variable X that are worse than the α-quantile of X.

Suppose we have n sorted samples denoted by $r_{(1)} \leq r_{(2)} \leq, \ldots, \leq r_{(n)}$. The AVaR of the above samples at level α can be calculated as,

$$AVaR_\alpha(r) = \quad (4)$$

$$\frac{1}{\alpha}\left(\frac{1}{n}\sum_{k=1}^{\lceil n\alpha\rceil-1} r_{(k)} + \left(\alpha - \frac{\lceil n\alpha\rceil - 1}{n}\right) r_{(\lceil n\alpha\rceil)}\right),$$

where $\lceil x\rceil$ denotes the smallest integer larger than x (Rachec et al., 2008).

In finance, AVaR is typically defined as above because a higher return is always better than a lower return. However, in our problem, a smaller outcome is preferable to a larger outcome because we want to minimize the objective functions T_{max} and $\bar{\theta}$. Therefore, in this study, we maximized the AVaR of the negative variable, that is, $max\ AVaR(-x)$, and use the negative AVaR, $-AVaR(-x)$, to indicate negative values as positive. In Figs. 3 and 5, the vertical and horizontal blue dashed lines indicate the $-AVaR$ values of T_{max} and $\bar{\theta}$, respectively.

7.2 Multi-Objective Model

To determine the optimal parameters of the algorithm for the environment given in Fig. 1 (A), a black-box optimiaztion with the model presented in Section 5 is performed. The parameters are evaluated based on the results obtained from n simulations conducted to solve the following multi-objective problem:

$$min\ f_1 = mean\{T_{max}^1, \ldots, T_{max}^n\} \quad (5)$$

$$min\ f_2 = mean\{\bar{\theta}^1, \ldots, \bar{\theta}^n\} \quad (6)$$

$$max\ f_3 = AVaR_{0.2}\{-T_{max}^1, \ldots, -T_{max}^n\} \quad (7)$$

$$max\ f_4 = AVaR_{0.2}\{-\bar{\theta}^1, \ldots, -\bar{\theta}^n\} \quad (8)$$

$$s.t.\ 0 < \lambda(0) \leq 1,\ 0 < \beta \leq 1,$$
$$0 < \xi \leq 1,\ 0 < \gamma \leq 1. \quad (9)$$

The first two objectives minimize the means of T_{max} and $\bar{\theta}$, whereas the last two objectives suppress the occurrence of RBCE by maximizing the AVaR of $-T_{max}$ and $-\bar{\theta}$ to a level of 0.2. This problem is difficult to solve as it involves complex tradeoffs among the four objective functions. The NSGA-II algorithm (Deb et al., 2002), a widely used multi-objective evolutionary algorithm, was used to obtain a set of Pareto optimal solutions (Gutjahr and Pichler, 2016) for the multi-objective problem in Tsurushima's method.

To address this problem, a significant number of simulations are performed, which is computationally expensive. Thus, a two-phase approach was introduced to reduce the computational costs (Tsurushima, 2022a; Tsurushima, 2022b). In the first phase, the problem is solved using NSGA-II with a relatively

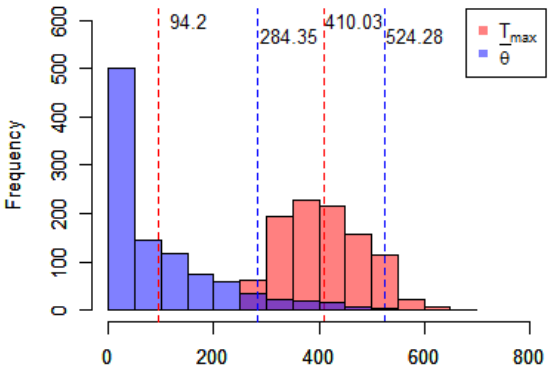


Figure 6: Histograms of T_{max} and $\bar{\theta}$ of Fig. 3 (C).

small n to reduce the computational cost, and Pareto optimal solutions of manageable sizes are explored. In the second phase, validation simulations with large n are performed for each solution, to examine the true Pareto efficiency. Subsequently, a small number of Pareto optimal solutions are selected by evaluating expected value and AVaR Pareto optimal (EAPO) efficiency with respect to both expected values and AVaRs, which is the criterion for selecting the appropriate size of the solutions from among the Pareto optimal solutions generated by NSGA-II. The EAPO solutions obtained are then presented to the decision maker to choose the most preferable solution.

8 EXPERIMENT

In this section, we describe the experiment conducted to explore the Pareto optimal solutions of the parameter set in a given environment (Fig. 1) using Tsurushima's method. The histograms of T_{max} and $\bar{\theta}$ in Fig. 3(C) are presented in Fig. 6, where the former is depicted in red and the latter in blue. The distributions of these two are different; the latter, in particular, has a heavy-tailed distribution that is difficult to minimize because representative values may vary significantly between small and large sample sizes. The Pareto optimal solutions obtained may be inaccurate because Tsurushima's method performs NSGA-II on small samples, which may lead to imprecise optimal solutions. Minimizing AVaR, that is, $max\ AVaR(-x)$, is even more difficult because it is the expected value of rare events, which may vary significantly with sample size. To address this issue, we conducted preliminary studies to obtain some useful information prior to the experiment.

1. Instead of solving the problem stated in Section 5 directly, we took two objective functions (5) and (6) from the problem and formulated a bi-objective problem because in this problem, correlations between the expected value and AVaR are greater than 0.95 for both T_{max} and $\bar{\theta}$ according to random parameter simulations.
2. To reduce the search space, we introduced a constraint on the parameter ξ as $0.0001 \leq \xi \leq 0.01$ because the value of ξ, within a certain range, significantly affects the results according to some observations and random parameter simulations. We assumed $\lambda(0) = 0.5$ and $n = 48$, which may not be sufficient to guarantee the utmost accuracy in results, thereby sacrificing accuracy at the expense of computational cost. However, the assumptions appear reasonable because based on our extensive experimentation, a significantly better solution could rarely be achieved.

The four objective problems in Section 7.2 are relaxed as follows:

$$min\ f_1 = mean\{T^1_{max}, \ldots, T^n_{max}\} \tag{10}$$

$$min\ f_2 = mean\{\bar{\theta}^1, \ldots, \bar{\theta}^n\} \tag{11}$$

$$s.t.\ \lambda(0) = 0.5,\ 0 < \beta \leq 1,\quad 0.0001 < \xi \leq 0.01,\ 0 < \gamma \leq 1. \tag{12}$$

The NSGA-II algorithm, with 40 populations, 40 generations, and $n = 48$ was applied to solve the aforementioned bi-objective black box optimization problem. 1000 validation simulations were performed for each of the Pareto optimal solutions generated by NSGA-II to verify the true Pareto efficiency and EAPO solutions were selected. We used NetLogo 6.0.2 (Wilensky, 1999) for the multi-agent simulations and R x64.3.5.1 to implement the optimization algorithm and analyze the results. NetLogo was chosen because of its good connectivity with the R language. The libraries used included nsga2R, parallel, and RNetLogo. The simulations and optimizations were executed on a machine with an *Intel Core i7 – 6700* CPU.

9 RESULTS AND DISCUSSION

Seven Pareto optimal solutions were generated using NSGA-II, and three EAPO solutions were selected after 1000 validation simulations. Fig. 7 shows the seven Pareto optimal solutions generated by NSGA-II (depicted as $\times$) and the results of validation simulations associated with the Pareto optima. The expected value and AVaR of the validation simulation results are depicted by red and blue circles, respectively. Solid circles indicate the EAPO solutions. The results are summarized in Table 1. The results of

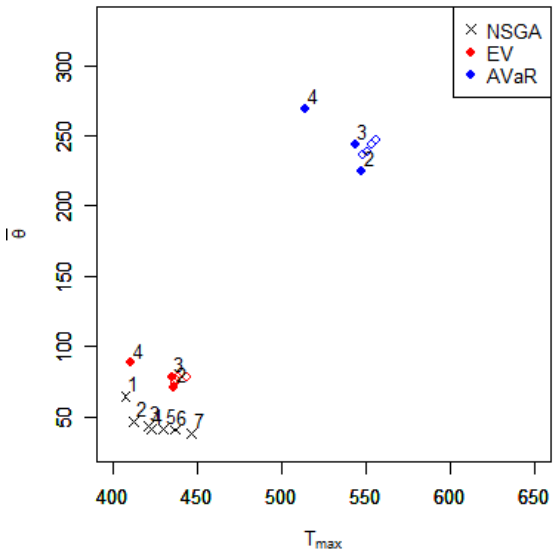


Figure 7: Pareto optimal solutions by NSGA-II and corresponding solutions by validation simulations.

Table 1: Three EAPO solutions. f_1, f_2, f_3, and f_4 denote $mean\{T_{max}\}$, $mean\{\bar{\theta}\}$, $-AVaR_{0.2}\{-T_{max}\}$, and $-AVaR_{0.2}\{-\bar{\theta}\}$, respectively.

	β	ξ	γ	f_1	f_2	f_3	f_4
2	0.15	0.0098	0.98	435.8	71.0	547.2	225.2
3	0.15	0.0098	0.93	434.7	77.8	543.4	243.4
4	0.68	0.0080	0.02	410.2	88.2	514.3	269.1

1000 simulations for solution 2 (blue row in the table) are shown in Fig. 5 (F). Let T_{max} of (D), (E), and (F) in Fig. 5 be T^D_{max}, T^E_{max}, T^F_{max}, respectively, and similarly $\bar{\theta}^D$, $\bar{\theta}^E$, $\bar{\theta}^F$. In this figure, the solutions for $T_{max} > -AVaR\{-T^E_{max}\}$, $\bar{\theta} > -AVaR\{-\bar{\theta}^D\}$, and $T_{max} > -AVaR\{-T^E_{max}\} \wedge \bar{\theta} > -AVaR\{-\bar{\theta}^D\}$ are indicated in red, blue, and green, respectively. In Solution 2 ((F) in Fig. 5), $|\{\bar{\theta} \mid \bar{\theta}^F > -AVaR\{-\bar{\theta}^D\}\}| = 29$, which is reduced from $|\{\bar{\theta} \mid \bar{\theta}^D > -AVaR\{-\bar{\theta}^D\}\}| = 89$, $\{T_{max} \mid T^F_{max} > -AVaR\{T^E_{max}\}\} = \phi$.

In Fig. 5, comparing (F) with (D) and (E), a reasonably good parameter set was found, allowing for an efficient control mechanism for the distributed evacuation guidance system; however, Fig. 7 indicates that there are certain limitations to our approach. The seven Pareto optimal solutions in the figure obtained by NSGA-II with small sample size (× in the figure) and the expected values of the corresonding validation simulations (red circles in the figure) are not close to each other. Moreover, the latter are densely packed. These facts suggest that the Pareto frontier of the multi-objective problem has not been sufficiently explored. The solutions obtained are not true Pareto optima, in that, better solutions may have been overlooked.

A limitation of our approach is the accuracy of the results because of limited computational resources. The dynamics of the fire spread in our simulations may have been inaccurate because we used a pseudo-fire-spread model rather than a model based on the computationally expensive Navier-Stokes equations. This may have caused our pseudo-fire-spread model to produce unrealistic results. However, this model can produce a substantial number of fire-spread scenarios, which may be sufficient for exploring an efficient control mechanism because the goal in this study was not to provide a precise analysis but to develop a mechanism for distributed guidance systems.

In Tsurushima's method, we explored Pareto optimal solutions using NSGA-II with a small number of samples, which resulted in a set of solutions that may not be truly Pareto optimum because of noisy objective functions. The method we used may not be the best way to solve the problem of the four objectives presented in Section 7.2. Other multi-objective evolutionary algorithms, such as NSGA-III, MOEA/D, may be applicable instead of NSGA-II, which perhaps is not appropriate for many objective problems. Multi-objective black-box optimization techniques such as multi-objective Bayesian optimization may be applicable to our problem; however, the non-Gaussian error distribution in the objective function would make this problem still difficult to solve. Therefore, new techniques are required to address these issues.

Although a few studies (Zhao et al., 2022; Tsurushima, 2024) were conducted, there is a crucial need for a distributed evacuation guidance algorithm that does not assume that all components work as intended because the assumption of a perfectly functioning system is unrealistic in real disaster evacuation scenarios. Furthermore, evacuation studies have focused only on average cases or a few scenarios, disregarding low-probability catastrophic cases despite their importance. In spite of limitations, our study provides a first step toward a solution addressing these problems.

10 CONCLUSION

An effective control mechanism based on Tsurushima's algorithm was developed for a distributed evacuation guidance signage system during fire spread. Additionally, Tsurushima's parameter calibration method was applied to the system to minimize the occurrence of low-probability catastrophic events. Experimental results demonstrated the proficiency of the system in crowd control and its capa-

bility to suppress RBCE. This study takes the initial steps towards resolving two crucial yet often overlooked challenges in evacuation guidance research: resilience and RBCE. It is imperative that any computer system designed to operate under disaster conditions, such as urban traffic control, emergency communication, or emergency fire suppression system, effectively tackle both of these issues. The presented approach offers valuable insights for addressing these challenges in such systems.

ACKNOWLEDGMENT

The author would like to thank Mr. Kei Marukawa for his assistance and helpful discussions. We would like to thank Editage (www.editage.jp) for English language editing.

REFERENCES

Coşkun, G., Demir, U., and Soyhan, H. (2022). Investigation of the smoke ventilation and evacuation strategies to decrease smoke poisoning risk by coupling fire and evacuation simulations. *Journal of Applied Fluid Mechanics*, 15:659–671.

Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197.

Ferscha, A. and Zia, K. (2010). Lifebelt: Crowd evacuation based on vibro-tactile guidance. *IEEE Pervasive Computing*, 9(4):33–42.

Fraser-Mitchell, J. and Charters, D. (2005). Human behaviour in tunnel fire incidents. *Fire Safety Science*, 8:543–554.

Fridolf, K. (2010). Fire evacuation in undergaound transportation systems: a review of accidents and research. Technical report, The Department of Fire Safety Engineering and Systems Safety, Lund University. Report 3151.

Gutjahr, W. J. and Pichler, A. (2016). Stochastic multi-objective optimization: a survey on non-scalarizing methods. *Annals of Operations Research*, 236(2):475–499.

Jasztal, M., Omen, c., Kowalski, M., and Jaskolowski, W. (2022). Numerical simulation of the airport evacuation process under fire conditions. *Advances in Science and Technology Research Journal*, 16:249–261.

Lee, J., Lee, M., and Jun, C. (2018). Fire evacuation simulation considering the movement of pedestrian according to fire spread. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-4/W9:273–281.

Liao, L., Li, H., Li, P., Bao, X., Hong, C., Wang, D., Xie, X., Fan, J., and Wu, P. (2023). Underground evacuation and smoke flow simulation in Guangzhou International Financial City during fire. *Fire*, 6:266.

Mirahadi, F. and McCabe, B. (2020). EvacuSafe: Building evacuation strategy selection using route risk index. *Journal of Computing in Civil Engineering*, 34(2):04019051.

Rachec, S. T., Stoyanov, S. V., and Fabozzi, F. J. (2008). *Advanced stochastic Models, Risk Assessment, and Portfolio Optimization: The Ideal Risk, Uncertainty, and Performance Measures*. Wiley.

Tsurushima, A. (2021). Reproducing evacuation behaviors of evacuees during the Great East Japan Earthquake using the evacuation decision model with realistic settings. In *Proceedings of the 13th International Conference on Agents and Artificial Intelligence - Volume 1: ICAART,*, pages 17–27.

Tsurushima, A. (2022a). Efficient visual sign assignment for crowd evacuation guidance considering risks and multiple objectives. In Rocha, A. P., Steels, L., and van den Herik, J., editors, *Agents and Artificial Intelligence*, pages 3–26, Cham. Springer International Publishing.

Tsurushima, A. (2022b). Multi-objective risk analysis for crowd evacuation guidance using multiple visual signs. In *Proceedings of the 14th International Conference on Agents and Artificial Intelligence - Volume 1: ICAART,*, pages 71–82. INSTICC, SciTePress.

Tsurushima, A. (2022c). Tunnel vision hypothesis: Cognitive factor affecting crowd evacuation decisions. *SN Computer Science*, 3(332).

Tsurushima, A. (2024). Simulation analysis of evacuation guidance using dynamic distributed signage. In *Proceedings of the 16th International Conference on Agents and Artificial Intelligence (ICAART 2024)*.

Wilensky, U. (1999). Netlogo. Center for Connected Learning and Computer-Based Modeling, Northwesten University, Evanston, IL.

Zhao, H., Schwabe, A., Schläfli, F., Thrash, T., Aguilar, L., Dubey, R., Karjalainen, J., Hölscher, C., Helbing, D., and Schinazi, V. (2022). Fire evacuation supported by centralized and decentralized visual guidance systems. *Safety Science*, 145:105451.

Zheng, Y., Jia, B., Li, X.-G., and Jiang, R. (2017). Evacuation dynamics considering pedestrians' movement behavior change with fire and smoke spreading. *Safety Science*, 92:180–189.