

# Performance Analysis and Failure Mitigation Strategies for a Resilient Dynamic Evacuation Guidance System

Akira Tsurushima<sup>a</sup>

Intelligent Systems Laboratory, SECOM CO., LTD., Japan

**Keywords:** Distributed Problem Solving, Multi-Agent Simulation, System Failure Analysis.

**Abstract:** Resilience is a critical factor in dynamic evacuation guidance systems, which must remain functional in harsh environments. However, most evacuation studies have seldom addressed system resilience. This study proposes a distributed dynamic evacuation guidance system that sustains functionality even when some components are damaged during evacuation, thereby enhancing the overall reliability and redundancy of the system by avoiding single points of failure. We evaluated the system performance through asynchronous multi-agent simulations to assess its effectiveness in maintaining guidance during a spreading fire that compromised its components. The experiments revealed that the proposed system with failed components performed comparably to a fully operational system when failures occurred in response to the fire severity. The adverse effects of random component failure were mitigated using two strategies: spatial interpolation and persistent guidance, resulting in a performance comparable to that of a failure-free system.

## 1 INTRODUCTION

Disaster evacuation occurs in harsh environments where most computer systems are prone to failure. Ensuring system resilience is crucial for evacuation support systems because advanced functions become ineffective if the system ceases to operate. Numerous studies have explored dynamic evacuation guidance systems that assist people in exiting buildings during disasters (Galea et al., 2014; Galea et al., 2017; Lin et al., 2017). These systems use sensors and signage to alter evacuation routes and display directions based on the sensor data. However, many of these studies assume perfect system functionality in harsh environments and often overlook the importance of system resilience. In this study, resilience is based on the concept that a system can maintain its overall functionality, even if some agents are lost.

Distributed systems, in which independent autonomous agents communicate through peer-to-peer networks, offer promising solutions for enhancing resilience. Unlike centralized systems with a single point of failure, distributed systems maintain their functionality even if agents fail or communication is interrupted. The independence of the agents allows the remaining network to continue functioning and

to coordinate to solve problems. However, the localized information associated with each agent, communication uncertainties, and coordination complexities present challenges. Whether such systems can achieve acceptable evacuation guidance performance remains uncertain, particularly when agents lose their functionality.

In this study, we used a dynamic evacuation guidance system incorporating the distributed *Broadcast* and *UpdateSign* algorithms proposed by (Tsurushima, 2024b) to conduct evacuation simulations under fire conditions with system components malfunctioning because of disasters. Two failure scenarios were investigated: component failure according to fire severity and random component failure. In the former scenario, the system with failed components performed nearly as well as the fully operational system, whereas in the latter scenario, the performance was inferior. To mitigate the adverse effects of system failures, two strategies have been proposed: imputing missing data from failed components using spatial interpolation and employing persistent guidance, where failed components continue to display the last guidance direction before malfunctioning. Both strategies demonstrated positive results in terms of reducing the adverse effects of system failures on evacuation guidance. The effects of time delays and drifts on the performance of the distributed system were also studied.

<sup>a</sup>  <https://orcid.org/0000-0003-2711-297X>

The remainder of this paper is organized as follows. Section 2 reviews related work, and Section 3 outlines the problems discussed in this paper. The distributed algorithms employed in this method are detailed in Section 4, and Section 5 describes the asynchronous simulation framework. Section 6 discusses the experiments and results of the failure simulations. Finally, the discussion and conclusions are presented in the subsequent sections.

## 2 RELATED WORK

Numerous studies have investigated public safety, crowd dynamics, and evacuation processes (Haghani, 2020a; Haghani, 2020b). Various crowd models have been proposed (Helbing et al., 2000; Lovreglio et al., 2016) to demonstrate the effectiveness of multi-agent simulations in evacuation studies. Research on dynamic evacuation guidance using signage systems has garnered significant attention (Galea et al., 2014; Galea et al., 2017), followed by studies employing multi-agent simulation models (Lin et al., 2017). Technologies that aim to enhance the efficiency of dynamic evacuation guidance include shortest-path finding using the Dijkstra algorithm (Baidal et al., 2020), and the Bellman–Ford algorithm (Zu and Dai, 2017). To reduce computational costs, a decomposed approach that divides the original problem into subproblems was investigated (Lujak et al., 2017; Nguyen et al., 2022). Systems that use smartphones or PDA devices for evacuation guidance have also been explored (Lujak et al., 2017; Zu and Dai, 2017; Kawahara et al., 2023; Zhao et al., 2017); however, these distributed approaches do not address system resilience. Distributed systems are a promising approach for solving optimization problems (Rust et al., 2020). Zhao et al. were the first to discuss distributed approaches to system resilience in evacuation guidance systems (Zhao et al., 2022). However, their work lacked technical details and did not analyze the system failures during evacuation.

To the best of our knowledge, this is the first study to perform dynamic evacuation guidance simulations that incorporate component failure during guidance and to analyze its impact on system performance.

## 3 PROBLEM

### 3.1 Environment

Let  $t = 0, \dots, T_{max}$  denote simulation time. An example of a fire-spread evacuation simulation within

a central core floor (Tsurushima, 2024a) is shown in Fig. 1(a). The floor comprises cells  $(x, y)$  where  $-40 \leq x \leq 40$  and  $-25 \leq y \leq 25$ . Two exits, labeled 23 and 24 (blue), are located in the central core (dark gray square region), along with the aisles and corners (labeled 11–22 in black). The doors connecting the aisles to the surrounding area (light brown region) are marked in yellow and numbered 1–10 in black, allowing evacuees to move from the surrounding spaces to the aisles and eventually to an exit. The surrounding space and aisles in the central core are divided into 38 spaces  $C_1, \dots, C_{38}$  (indicated by the red numbers in (A)). These spaces correspond to the edges of the universal evacuation graph (Fig. 1(B)). Note that the red numbers 19–26 represent the small spaces occupied by doors. Each space  $C_i$  has representative coordinates  $x(C_i), y(C_i)$ , and the distance between the two spaces  $C_i$  and  $C_j$  is defined as  $d_{ij} = \sqrt{(x(C_i) - x(C_j))^2 + (y(C_i) - y(C_j))^2}$ . The black cells at the top right of the surrounding space indicate the area occupied by fire, and the light-gray cells around the black cells indicate that they were filled with smoke. Fire in the environment was simulated using a simple but reasonable fire-spread model (Tsurushima, 2024a). The fire hazard level of cell  $(x, y)$  at time  $t$  is represented by:  $h_{(x,y)}^t \in \mathbb{R}$ , which propagates to the neighboring cells.  $\zeta(x, y)$  with probability  $p_1$  by

$$h_{(x,y)}^t = (1 - p_2)h_{(x,y)}^{t-1} + p_2 \max\{h_{(x,y) \in \zeta(x,y)}^t\}, \quad (1)$$

where  $p_2$  is the ratio of these two terms. We assumed that  $p_1 = 0.07$  and  $p_2 = 0.2$ . In (A), cells with  $h_{(x,y)} \geq 100$  are shown in black, and  $100 > h_{(x,y)} \geq 0$  are shown in gray to white, depending on the value. A cell  $(x, y)$  was randomly selected from the surrounding space, and  $h_{(x,y)}^0$  was set to  $10^5$  at time  $t = 0$ .

### 3.2 Evacuation Agent

At time  $t = 0$ , 800 evacuation agents  $A^0 = \{a_1, \dots, a_{800}\}$  were randomly distributed in the surrounding space. When the simulation began, each agent attempted to exit the surrounding space through the doors, aisles, and exits. If agents behaved intelligently, this would overshadow the true effectiveness of the system. Therefore, unlike humans, evacuation agents lack intelligence and cannot independently avoid hazards. They operate based on a simple decision model that allows random selection in decision-making situations. The only decision rule that evacuation agents follow, aside from making random choices, is to adhere to the guidance provided by a signage agent; however, this is contingent on the signage agent being within their field of view. The agents

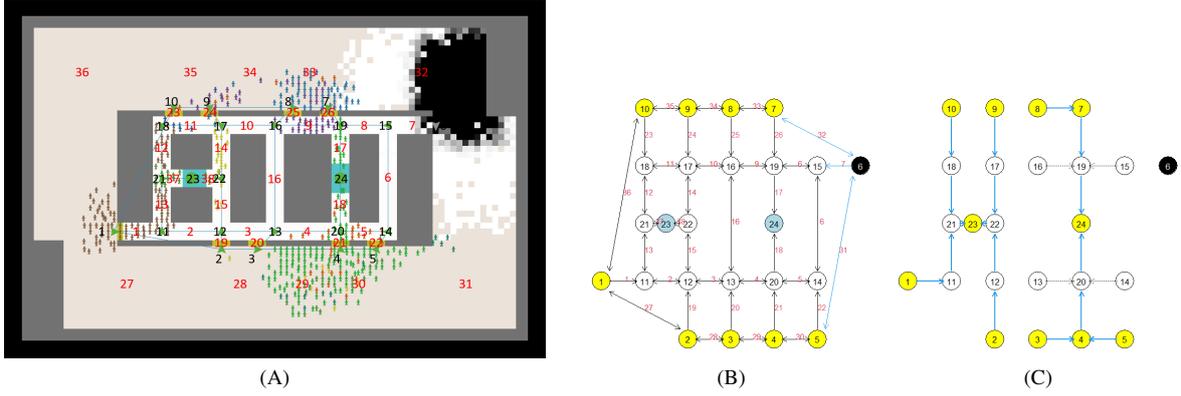


Figure 1: (A) Simulation screen, (B) Universal evacuation graph, (C) Evacuation graph.

choose doors if they are located in the surrounding space and corners, or exits if they are in the central core.

The hazard contamination of an evacuation agent  $a$  is denoted by  $\theta_a$  if the agent is exposed to cells with  $h > 0$ . This value accumulates over the simulation period according to  $\theta_a = \sum_{t=0}^{T_{max}} \min\{100, h^t\}$ , where  $T_{max}$  is the time at which all evacuation agents complete the evacuation or when  $(h_{(x(v_{23}), y(v_{23}))} > 100) \wedge (h_{(x(v_{24}), y(v_{24}))} > 100)$ . Here,  $x(v)$  and  $y(v)$  denote the  $x$ - and  $y$ -coordinates of node  $v$ , indicating an incomplete evacuation when these conditions are met.

### 3.3 Signage Agent

Signage agents directing evacuation agents to evacuation routes are indicated by green arrowheads located at doors or corners, as shown in Fig. 1 (A). They are positioned either on doors (1–10), corners (11–22), or (23, 24). Figure 1(B) shows universal evacuation graph  $G = (V, E, W)$ , a graphical representation in (A) (Tsurushima, 2024b). Nodes ( $v_i \in V$ ) represent signage agents at doors ( $v_1, \dots, v_{10}$ ), corners ( $v_{11}, \dots, v_{22}$ ), or exits ( $V_g = \{v_{23}, v_{24}\}$ ); the edges ( $((v_i, v_j) = e_k \in E)$  represent possible evacuation guidance directions at each location. Furthermore,  $w_k \in W, w_k \in \mathbb{R}$  denote the associated edge weights on  $e_k$ , representing fire severity in  $C_k$ , which is critical for determining a safe evacuation route. A universal evacuation graph can be considered as a representation of an evacuation guidance system, where the nodes represent signage agents, and directed edges  $(i, j)$  represent possible guidance directions for node  $i$  ( $\Delta: V \rightarrow 2^V, j \in \Delta(i), (i, j) \in E$ ) at their corresponding locations. The edges also represent the system connections of the signage agents, where the signage agents communicate with each other through these edges. The black nodes indicate faulty components and the blue edges indicate failures caused by fire.

Thus, Node 6 is faulty and unable to communicate with adjacent Nodes 5, 7, or 15;  $\varphi: V \cup E \rightarrow \{0, 1\}$  indicates whether the nodes or edges function  $\varphi(v) = 1$  if signage agent  $v$  functions and  $\varphi(v) = 0$  otherwise. We assume  $\varphi(\{i, j\}) = 0$  if either  $\varphi(i) = 0$  or  $\varphi(j) = 0$  is true.

Figure 1(C) shows  $\dot{G} = (V, \dot{E}, \dot{W})$  (Tsurushima, 2024b) where  $\dot{E} \subset E, \dot{W} \subset W$  represents the evacuation routes suggested by the signage agents. Each edge indicates the suggested direction  $\delta(i)$  ( $\delta: V \rightarrow V, j \in \Delta(i), (i, j) \in \dot{E}$ ) that a signage agent  $i$  displays for evacuees to follow for safe evacuation at the corresponding location. Here,  $\Delta(i)$  denotes the adjacent nodes of  $i$  and  $\Omega(i)$  represents the neighboring edges. For instance,  $\Delta(1) = \{2, 10, 11\}$  and  $\Omega(1) = \{1, 27, 36\}$ . If a node's signage agent fails, it ceases to indicate the evacuation direction, resulting in random behavior by evacuation agents.

Evacuation guidance performance was evaluated based on the total evacuation time ( $T_{max}$ ) and mean hazard contamination ( $\bar{\theta}$ ). The decision making of a signage agent follows two distributed algorithms: *Broadcast* and *UpdateSign*, which are detailed in the next section.

## 4 ALGORITHM

We employed the distributed broadcast and *UpdateSign* algorithms proposed in (Tsurushima, 2024b; Tsurushima, 2024a) to solve the dynamic evacuation guidance problem presented in the previous section. To ensure the asynchronous execution of signage agents within the simulation, these algorithms were adapted to use a message-passing system (lines 10 and 11 in Algorithms 1 and 2, respectively) instead of the function calls in their original versions. Furthermore, a data imputation mechanism was introduced in Algorithm 2 (lines 3–5) to enhance the re-

Algorithm 1: Broadcast at node  $i$ .

---

**Local variable:**  $\hat{E}, \hat{W}^C, \hat{T}$

**1 Procedure**

Broadcast ( $m_b = \langle o, j, t, \mathcal{L}(o) \rangle$ ):

2    $\langle \delta(o), \Omega(o), \Upsilon^C(o) \rangle = \mathcal{L}(o)$ ;

3   **if**  $\varphi(i) = 1$  &  $\hat{T}[o] < t$  **then**

4      $\hat{T}[o] \leftarrow t$ ;

5      $\hat{E}[o] \leftarrow \delta(o)$ ;

6     **foreach**  $k \in 1, \dots, |\Omega(o)|$  **do**

7        $\hat{W}^C[\Omega(o)[k]] \leftarrow \Upsilon^C(o)[k]$ ;

8     **end**

9     **foreach**  $a \in \Delta(i) \setminus \{j\}$  **do**

10       send  $m_b = \langle o, i, t, \mathcal{L}(o) \rangle$  to node  $a$ ;

11     **end**

12   **end**

13 **end**

---

Algorithm 2: UpdateSign at node  $i$ .

---

**Local variable:**  $\hat{E}, \hat{W}^C, \delta(i), \hat{T}$

**1 Procedure** UpdateSign:

2   **if**  $\varphi(i) = 1$  **then**

3     **if** Failure node found in  $\hat{T}$  **then**

4        $\hat{W}^C \leftarrow \text{SpatialInterpolation}$ ;

5     **end**

6      $l_{ig}^* \leftarrow \text{Search}(i, \hat{E}, \hat{W}^C)$  ;

7      $(i, \text{next}) \leftarrow l_{ig}^*[1]$ ;

8      $\hat{E}[i] \leftarrow (i, \text{next})$ ;

9      $\delta(i) \leftarrow \hat{E}[i]$ ;

10    **if**  $\hat{E}$  include a cycle **then**

11      send message  $m_u$  to node  $\text{next}$ ;

12    **end**

13   **end**

14 **end**

---

silence of the guidance system. The nodes in Fig. 1 (B) represent signage agents that function as computational processes capable of executing both *Broadcast* and *UpdateSign*. To provide efficient evacuation guidance, both algorithms were executed in two ways: 1. at regular intervals at each node, or 2. triggered by messages from the other nodes.

The *Broadcast* procedure, which is a simple flooding algorithm that distributes the local information of a node throughout graph  $G$ , is presented in Algorithm 1. In this procedure,  $o$  is the initiator node of the *Broadcast*,  $j$  is the message-sending node activating this procedure,  $t$  is the time of initiation of  $o$ , and  $\mathcal{L}(o)$  is the local information of  $o$ , which is distributed to

the other nodes; that is,  $\mathcal{L}(o) = \langle \delta(o), \Omega(o), \Upsilon^C(o) \rangle$ , where  $\delta(o)$  denotes the guidance direction of  $o$ ,  $\Omega(o)$  the neighboring edges of  $o$ , and  $\Upsilon^C(o)$  the edge weight  $w^C$  associated with  $\Omega(o)$ .

The weights  $w_i^C$  used to derive efficient evacuation guidance were calculated as follows:

$$w_i^C(t) = \alpha(t)l_\xi(w_i^B(t)) + (1 - \alpha(t))w_i^A(t), \quad (2)$$

where  $l_\xi = \begin{cases} x & \text{if } x \geq \xi \\ 0 & \text{otherwise,} \end{cases}$  and  $\alpha(t) = \beta \alpha(t-1)$ .

Here,  $w_i^A$  and  $w_i^B$  are defined as

$$w_i^A = \frac{1}{|C_i|^2} \sum_{(x,y) \in C_i} h_{(x,y)}, \quad (3)$$

and

$$w_i^B = w_i^A(t) + \gamma \max\{\zeta(e_i)/|C_i|\}, \quad (4)$$

where  $\zeta : E \rightarrow 2^{W^C(t-1)}$ , which refers to previous weights of the neighboring edges of  $e_i$ , and  $\gamma$  denote the scaling factor.

Each node  $i$  has its own local variables:  $\hat{E}$ ,  $\hat{W}^C$ ,  $\hat{T}$ , and  $\delta(i)$ . where  $|\hat{E}| = |\hat{T}|$ , and  $\hat{W}^C = \{w_1^C, \dots, w_{38}^C\}$ . These variables were used to manage local information during evacuation guidance.  $\hat{T}$  records the initiation time of each node.

The broadcast procedure is triggered when a message  $m_b = \langle o, j, t, \mathcal{L}(o) \rangle$  is received or when it is initiated by sending  $m_b = \langle i, i, t, \mathcal{L}(i) \rangle$  to itself after a specified interval. Each time  $m_b$  is initiated, the previous weights  $\hat{W}^C$  are stored in the local memory to calculate  $w_i^B$  in Equation 4. To initiate  $m_b$ ,  $\Upsilon^C$  is calculated using the node  $i$ 's sensors and previous weights, as shown in Equations 2–4.

*UpdateSign* is presented in Algorithm 2. *UpdateSign* searches for the minimum weight path to one of the exits, based on the local information provided by *Broadcast*, using a breadth-first search (line 6). This process determines the guidance direction  $\delta(i)$ . If a cycle is found on the path to the exit, *UpdateSign* sends a message  $m_u$  to the *next* node to break the cycle.

When a node receives message  $m_u$ , the *UpdateSign* procedure is executed. Node  $i$  also initiates message  $m_u$  after a certain delay to begin the *UpdateSign* process. The parameters were optimized using multi-objective black-box optimization (Tsurushima, 2024a), yielding estimates  $\alpha(0) = 0.223$ ,  $\beta = 0.024$ ,  $\xi = 1.26e-5$ , and  $\gamma = 0.020$ .

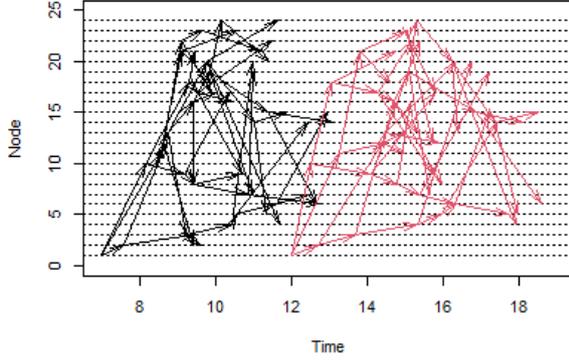


Figure 2: Message delivery of the *Broadcast* procedure between nodes initiated by node 1. The x-axis represents the simulation time, and the y-axis indicates the node IDs involved in the communication process.

## 5 ASYNCHRONOUS SIMULATION

By incorporating the *Broadcast* and *UpdateSign* algorithms, the nodes in the graph communicate through edges representing the communication channels, as shown in Fig. 1 (B).

These algorithms are structured to operate asynchronously, accommodating potential malfunctions of nodes (signage agents) or communication edges. In asynchronous systems, there is no unified global time for synchronizing node operations. Message transfer can experience delays of unknown duration, and the execution interval of each procedure may vary unpredictably with no guaranteed upper execution time limit. To model these conditions, we assume the following speed hierarchy for the system processes:  $s_p, s_c, s_w$ , and  $s_f$ , which represent the speeds of the internal process execution, inter-node communication, evacuee movement, and fire spread, respectively. The relationship  $s_p \succ s_c \succ s_w \succ s_f$  indicates that each process is faster than the one that succeed.

In *UpdateSign*, the update of the guiding direction  $\delta(i)$  precedes the dispatching of cycle-breaking messages 11 in Algorithm 2). Considering  $s_p \succ s_c \succ s_w$ , this is reasonable because breaking the cycle on the evacuation routes can be achieved more quickly than when the evacuees reach the next signage agents.

For the integrated simulation model, fire-spread, evacuation agent, and evacuation guidance system models were developed (Tsurushima, 2024a) to investigate effective control mechanisms for the problems stated in Section 3. The model was implemented using the multiagent simulation platform NetLogo 6.0.2 (Wilensky, 1999), which does not provide an asynchronously distributed simulation mechanism. We

developed a message-passing framework that emulates an asynchronous distributed simulation, in addition to NetLogo 6.0.2.

Let  $I_b^*$  and  $I_u^*$  be the intervals between the initiation of *Broadcast* and *UpdateSign*, respectively. They are initiated with intervals of  $I_b = I_b^* + \epsilon_b$  and  $I_u = I_u^* + \epsilon_u$  owing to drifts. Additionally, let  $z = \epsilon_z$  denote the delay for a message sent between two nodes. Random variables  $\epsilon_b \sim U(0, \epsilon_b^*)$ ,  $\epsilon_u \sim U(0, \epsilon_u^*)$ , and  $\epsilon_z \sim U(0, \epsilon_z^*)$  are introduced to emulate the drifts for each interval. Consequently, *Broadcast* and *UpdateSign* occur after  $I_b$  and  $I_u$  time units from previous executions. The destination node receives messages  $z$  time units after they are sent.

An agenda queue  $Q$  is introduced to schedule the messages sent by *Broadcast* and *UpdateSign* in lines 10 and 11 of Algorithms 1 and 2. Both types of messages are posted to  $Q$  along with the execution timestamp  $\tau = t + z$ , where  $t$  denotes the current time if it is an initiation, or the timestamp of the previous message if it is a message reception. The messages are sorted by  $\tau$  in  $Q$ , and messages with  $\tau$  earlier than the current time are selected from the top and sequentially executed. Notably, the execution of a message can generate other messages stored in  $Q$  that are scheduled for execution.

Figure 2 illustrates how *Broadcast* messages are delivered between nodes when the first two messages are sent by Node 1, assuming  $I_b^* = 5, \epsilon_b^* = 3$  and  $\epsilon_z^* = 2$ . The arrow represents messages originating from the root node to the target nodes, with  $2|E| - |V| + 1$  messages sent per broadcast. The second message was sent by Node 1 before the first message was received by the other nodes. In this study, the processing times for both *Broadcast* and *UpdateSign* within a process are assumed to be negligible, with  $I_b^* = 5$  and  $I_u^* = 10$ .

### 5.1 Analysis

Distributed dynamic evacuation guidance is provided by *UpdateSign* running on individual signs. The performance of *UpdateSign* depends on the information provided by the *Broadcast*. Evacuation guidance is effective when all the local information  $\hat{G}$  provided by *Broadcast* is consistent for each sign. Ensuring consistency in the local information on every sign is crucial.

Let  $\hat{E}_i$  and  $\hat{W}_i^C$  denote the local information vectors for the evacuation routes and edge weights at signage agent  $i$ ;  $\hat{E}_i[k]$  and  $\hat{W}_i^C[k]$  represent the  $k$ th element of vectors  $\hat{E}_i$  and  $\hat{W}_i^C$ , respectively. Consider the cosine similarity between  $\hat{E}_i$  and  $\hat{E}_j$ , denoted by

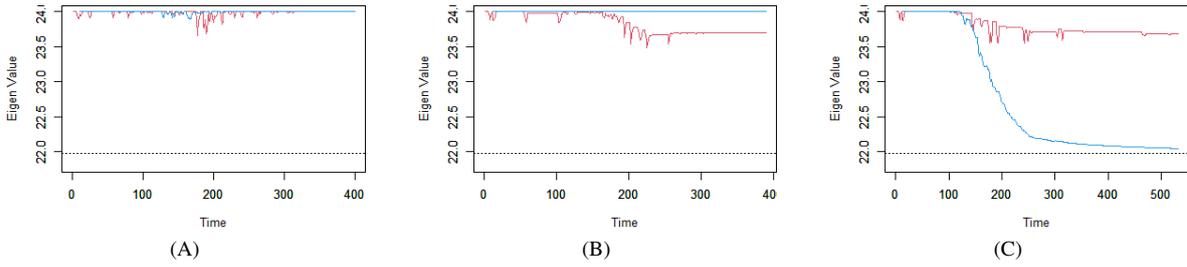


Figure 3: Temporal change in the maximum eigenvalue of the similarity matrix of local images with no failure (A), Scenario 1 (B), and Scenario 2 (C).

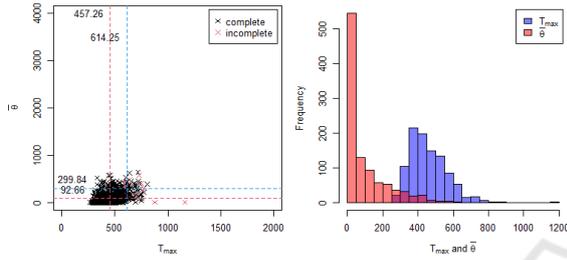


Figure 4: Baseline with  $\epsilon_b^* = 3$ ,  $\epsilon_u^* = 3$ , and  $\epsilon_z^* = 2$ .

$r_{ij}^E$ , and between  $\widehat{W}_i^C$  and  $\widehat{W}_j^C$ , denoted as  $r_{ij}^W$ . We have  $|N| \times |N|$  similarity matrices  $R^E$  for  $r_{ij}^E$  and  $R^W$  for  $r_{ij}^W$ . Subsequently, the maximum eigenvalues  $\lambda^E$  and  $\lambda^W$  are obtained for  $R^E$ ,  $R^W$ . Specifically,  $\lambda^E = 24$  when all  $\widehat{E}_i$ s are identical, and the same is true for  $\lambda^W$  because  $r_{ij} = 1$  if the vectors  $i, j$  are identical. The minimum value of  $\lambda^E$  is unknown because  $\widehat{E}$  can only assume certain values. For example,  $\widehat{E}[1] \in \{2, 10, 11\}$ . As a baseline, the minimum value of  $\lambda^E$  was estimated to be 21.98 in 1000 Monte Carlo simulations for randomly generated vectors  $\widehat{E}$ .

Assuming that all components of the system are functional throughout the simulation, the temporal evolutions of  $\lambda^E$  and  $\lambda^W$  are presented in red and blue, respectively, in Fig. 3 (A), indicating that both values were approximately 24 during the simulation.

## 6 EXPERIMENT AND RESULT

Assuming that all system components function as expected during fire-spread evacuations, 1000 evacuation guidance simulations were performed as a baseline for the analysis. The left-hand sides of Fig. 4 presents the results, where the x-axis represents  $T_{max}$  and the y-axis represents  $\bar{\theta}$ . Black  $\times$  denotes complete evacuation, implying that all evacuees fled, whereas red  $\times$  denotes incomplete evacuation, signifying that the fire occupied both exits before all evacuees escaped. The red dashed lines indicate the mean,

and the blue dashed lines represent  $AVaR_{0.2}$ , which denotes the mean of the worst 20 % of the samples (Tsurushima, 2024a). In this analysis,  $\epsilon_b^* = 3$ ,  $\epsilon_u^* = 3$ , and  $\epsilon_z^* = 2$  were assumed. A summary is presented in row A of Table 2. The right-hand chart in the figure shows the histograms of  $T_{max}$  and  $\bar{\theta}$ ; the former is illustrated in blue, and the latter in red. The non-Gaussian nature of the distributions, particularly  $\bar{\theta}$ , complicates the analysis.

The effects of system failure on the evacuation performance of the distributed dynamic evacuation guidance system were evaluated through simulations covering two random scenarios: system component failure according to fire severity (Scenario 1) and random failure (Scenario 2).

*Scenario 1* In this scenario, the signage agents of the evacuation guidance system fail based on the fire hazard levels ( $h'_{(x,y)}$ ) of the cells in which they were installed. The failure probability of  $i$  is  $h'_{(x,y)} \times 10^{-2}$  and changes  $\varphi(v_i) = 1$  to zero. Edge  $\{i, j\}$  fails based on  $i$  or  $j$ . In this scenario, the system attempts to move the evacuees away from the fire where the signage agents are most likely to fail. If the system works well, signage agent failures will have little effect on evacuation performance, because evacuees will have left the area in which signage agents are likely to fail. In addition, the evacuation guidance system is expected to be adversely affected by process-time drift and communication delays inherent in asynchronous systems. Therefore, the effects of system failures and delays on evacuation guidance performance should be assessed.

Figure 3 (B) shows the temporal evolution of  $\lambda^E$  and  $\lambda^W$  in Scenario 1. Compared to Fig. 3 (A),  $\lambda^E$  decreases but remains at approximately 23.5, which is still above the baseline of 21.98. This suggests that the variance among the local images in the signage agents may not be substantial.

*Scenario 2* In real buildings, variations in the aisle and connection patterns between signage agents can lead to random failures if the connection between two adjacent signage agents is located within a fire-affected

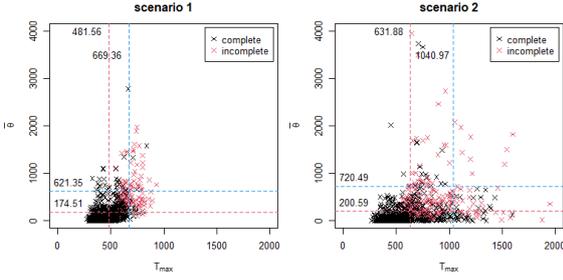
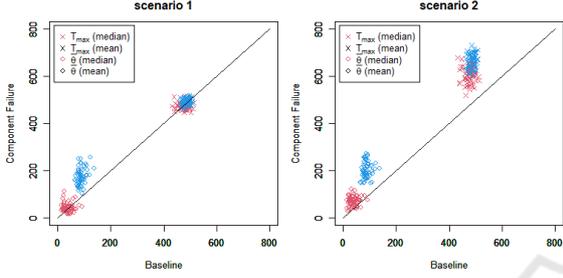

 Figure 5: Dynamic guidance with  $\epsilon_b^* = 3$ ,  $\epsilon_u^* = 3$ , &  $\epsilon_z^* = 2$ .


Figure 6: Dynamic guidance with and without failure for 64 drift cases.

area. In this scenario, signage agents fail randomly irrespective of the fire hazard  $h'_{(x,y)}$ . The failure probability  $p_f$  remains constant throughout the simulation, and each signage agent fails with this probability at every simulation time  $t$ . Because the specific sign that will cause failure is unknown, the impact of system failure is expected to be significant; a sign that is critical to several evacuees may fail at a crucial moment. The connectivity of the graph is likely compromised because of random failures. By contrast, in Scenario 1, the failure nodes were typically clustered together because they were highly related to the areas occupied by the fire, which preserved high connectivity in the graph. The loss of graph connectivity can significantly affect the performance of distributed algorithms because messages cannot be delivered to nodes in an isolated graph.

Figure 3 (C) shows the temporal evolution of the maximum eigenvalues  $\lambda^E$  and  $\lambda^W$  for Scenario 2. Compared with Fig. 3 (A), both  $\lambda^E$  and  $\lambda^W$  decrease significantly, particularly  $\lambda^W$ , because of graph connectivity loss.

Comparing Scenarios 1 and 2 is challenging because we must estimate  $p_f$  for Scenario 2, in a manner comparable to Scenario 1. To accomplish this, we used the median number of failed nodes at the end of the simulation for each scenario. We conducted 100 simulations by varying  $\epsilon_z^*$ ,  $\epsilon_u^*$ , and  $\epsilon_b^*$  from 0, 1, 2, 3, resulting in 6400 data points for both scenarios. The median values were compared. The same experiments were performed by varying  $p_f$ , which was estimated

to be  $p_f = 0.0006$ . Table 1 lists the minimum, median, mean, and maximum numbers of failed nodes at the end of the simulations for Scenarios 1 and 2 with  $p_f = 0.0006$ .

Table 1: Number of failure nodes observed in Scenarios 1 and 2.

Scenario	min	median	mean	max
1	0.0	7.0	8.0	23.0
2 ( $p_f = 0.0006$ )	0.0	7.0	7.6	20.0

We randomly sampled 100 data points from each dataset and conducted a Wilcoxon rank-sum test, which yielded  $W = 4861$  and a p-value of 0.73. Consequently, the null hypothesis was accepted, indicating no significant difference between the datasets when  $p_f = 0.0006$ .

We performed 1000 system failure simulations in which some components malfunctioned during fire evacuation in Scenarios 1 and 2. The chart on the left side of Fig. 5 shows the results for Scenario 1, whereas the right-hand chart shows the results for Scenario 2, as summarized in row B of Table 2. Lines 3–5 of Algorithm 2 were omitted from this analysis.

We performed 100 simulations with  $\epsilon_z^*$ ,  $\epsilon_u^*$ , and  $\epsilon_b^*$  varying from 0, 1, 2, 3, resulting in 64 cases to compare conditions with and without failure. The medians of  $T_{max}$  and  $\bar{\theta}$  were used for the evaluation, and Wilcoxon rank-sum tests were applied because of the non-Gaussian distributions. In Scenario 1, only 17 of the 64 cases were statistically significant, suggesting a minimal impact of failure in most instances. Conversely, in Scenario 2, all 64 cases were statistically significant for either  $T_{max}$  or  $\bar{\theta}$ , with 34 cases being significant for both, indicating a pronounced system failure effect.

Figure 6 shows the means and medians of  $T_{max}$  and  $\bar{\theta}$  for the 64 cases, with the means shown in blue and the medians in red as  $\epsilon_z^*$ ,  $\epsilon_u^*$ , and  $\epsilon_b^*$ , respectively. The diagonal line plots illustrate that the results with and without failures are similar. Given non-Gaussian distributions, medians serve as better summary statistics. However, considering rare but catastrophic events (RBCE), which are critical in evacuation studies (Tsurushima, 2024a), means should also be considered. These charts represent overlapping units (time and hazard) within a single chart.

## 6.1 Spatial Interpolation

Figure 6 illustrates that the means of  $\bar{\theta}$  are above the diagonal line in Scenario 1, and all plots (including the means and medians of  $T_{max}$  and  $\bar{\theta}$ ) are above the diagonal line in Scenario 2, indicating the significant

Table 2: Summary of the results.  $T_{max}^{\hat{}}$ ,  $T_{max}^{-}$ ,  $T_{max}^{\cdot}$  denotes the median, mean,  $AVaR_{0,2}$  of  $T_{max}$ , similar to  $\hat{\theta}$ ,  $\bar{\theta}$ ,  $\theta$ .

	Scenario 1						Scenario 2					
	$T_{max}^{\hat{}}$	$T_{max}^{-}$	$T_{max}^{\cdot}$	$\hat{\theta}$	$\bar{\theta}$	$\theta$	$T_{max}^{\hat{}}$	$T_{max}^{-}$	$T_{max}^{\cdot}$	$\hat{\theta}$	$\bar{\theta}$	$\theta$
A	443.00	457.26	614.25	33.53	92.66	299.84	-	-	-	-	-	-
B	462.00	481.56	669.36	48.62	174.51	621.35	568.00	631.88	1040.97	62.72	200.59	720.49
C	462.00	480.27	658.54	50.37	170.08	604.13	531.50	582.50	918.55	54.19	141.43	465.77
D	444.50	451.04	577.29	38.44	112.92	380.30	400.00	427.97	626.68	33.55	103.62	346.48

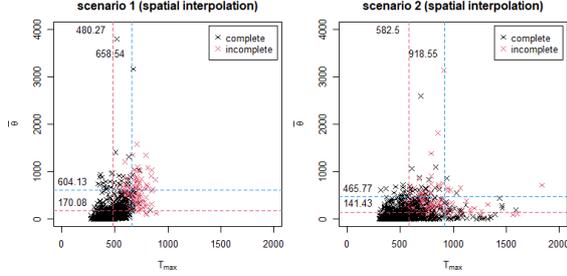
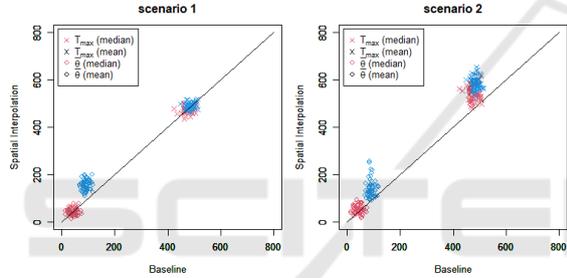
Figure 7: Spatial interpolation results with  $\epsilon_b^* = 3$ ,  $\epsilon_u^* = 3$ , &  $\epsilon_z^* = 2$ .

Figure 8: Spatial interpolation with and without failure for 64 drift cases.

effects of failures that need to be mitigated.

These adverse effects are generally caused by missing data regarding component failures, which can be estimated using spatial interpolation techniques. To enhance the evacuation guidance when component failures are identified,  $\tilde{W}^C$ , which is used to search for an optimal path in Algorithm 2, is modified using a spatial interpolation technique (lines 3–5). We assume that each signage agent determines an edge  $k = (i, j)$  as malfunctioning if both  $t - \hat{T}[i]$  and  $t - \hat{T}[j]$  exceed  $I_b^* + \epsilon_b^* + \epsilon_z^*$ , where  $t$  denotes the current time. The estimation of  $w_k^C$  is as follows:

$$\tilde{w}_k^C = \frac{\sum_{n \in \Phi(k)} d_{kn}^{-2} w_n^C}{\sum_{m \in \Phi(k)} d_{km}^{-2}}, \quad (5)$$

where  $\Phi(k)$  denotes the set of neighboring edges of  $k$ ; for instance,  $\Phi(1) = \{2, 13, 27, 36\}$ .  $w_k^C$  is updated by  $\tilde{w}_k^C$  when edge  $k$  is determined to have malfunctioned.

Figure 7 shows the results of 1000 simulations using a spatial interpolation. The results indicate that the mean and  $AVaR$  values are suppressed in Scenario

2, whereas they remain comparable in Scenario 1, A summary of these results can be found in row C of Table 2.

Figure 8 shows the impact of drifts ranging from zero to three on the medians and means of  $T_{max}$  and  $\theta$  when spatial interpolation is employed. Compared to Fig. 6, the plots are positioned closer to the diagonal lines, particularly in Scenario 2, indicating the positive impact of spatial interpolation in mitigating the adverse effects of component failures.

## 6.2 Persistent Guidance in Failure

Up to this point, we assumed that failed signage agents do not provide any guidance direction, leading to random behaviors by evacuation agents, because they can only make random selections. This assumption leads to prolonged evacuation times and increased fire hazards for evacuation agents, potentially resulting in unrealistic or overestimated values of  $T_{max}$  and  $\theta$ . An alternative assumption is that failed signs continue to display the last guidance direction, as shown before malfunctioning. This approach, known as persistent guidance, is feasible if physical indicators, rather than electrical devices, are used to indicate the guidance direction. Although the direction of guidance from failed signs may be inaccurate, evacuation agents do not resort to random behaviors.

We conducted 1000 simulations under the assumption of persistent guidance during sign failures. The results are presented in Fig. 9 and summarized in row D of Table 2 indicate that persistent guidance produces nearly comparable or, in some cases, better outcomes. This improvement is attributed to the absence of random behaviors among the agents, resulting in shorter evacuation times and potentially lower fire hazards.

Figure 10 shows the effect of drifts ranging from zero to three on the medians and means of  $T_{max}$  and  $\theta$  when persistent guidance is employed. In these figures, many fall on or below the diagonal lines, indicating the positive effects of persistent guidance. However, a few plots appear above the diagonal lines, with some showing exceptionally high values, suggesting rare but catastrophic outcomes that need to be mitigated.

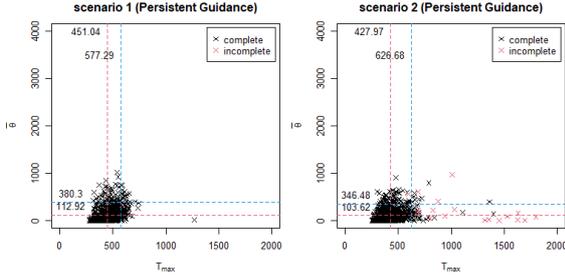


Figure 9: Persistent guidance results with  $\epsilon_b^* = 3$ ,  $\epsilon_u^* = 3$ , &  $\epsilon_z^* = 2$ .

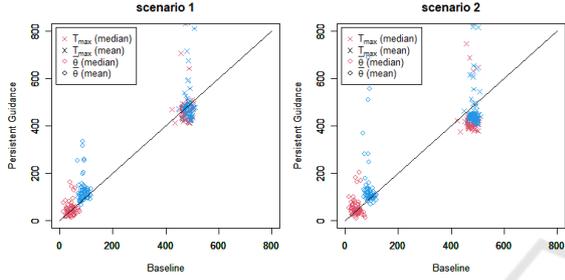


Figure 10: Persistent guidance with and without failure for 64 drift cases.

To assess the impact of drifts, we performed a multiple regression analysis using  $T_{max}/T_{max}^*$  and  $\theta/\theta^*$  as dependent variables and  $\epsilon_b^*$ ,  $\epsilon_u^*$ , and  $\epsilon_z^*$  as independent variables. Here,  $T_{max}$  and  $\theta$  denote the results with persistent guidance, whereas  $T_{max}^*$  and  $\theta^*$  represent the baseline results. The results are summarized in Table 3, indicating the negative impact of  $\epsilon_b^*$  and the positive impact of  $\epsilon_z^*$  (p values for Scenarios 1 and 2 were  $2.2e-16$  and  $3.6e-13$ , respectively).

Considering the negative and positive impacts of  $\epsilon_b^*$  and  $\epsilon_z^*$ , assuming a range for  $\epsilon_b^*$  of 1, 2, 3, and for  $\epsilon_z^*$  of 0, 1, 2, we filtered 36 of the 64 cases depicted in Fig. 11. In this filtered set, nearly all the plots fell on or below the diagonal lines, indicating that persistent guidance with limited time drifts and delays yielded a performance comparable to that of an ideal system.

## 7 DISCUSSION

In this study, we investigated two types of uncertainties in a distributed dynamic evacuation guidance

Table 3: Results of multiple regression analysis highlighting the effects of variables on guidance performance.

	Scenario 1			Scenario 2		
	coeff	p-value		coeff	p-value	
$\epsilon_u^*$	-0.0053	0.752		0.0345	0.577	
$\epsilon_b^*$	-0.1495	2e-16	***	-0.3950	7.31e-10	***
$\epsilon_z^*$	1.1076	1.14e-13	***	0.3237	2e-16	***

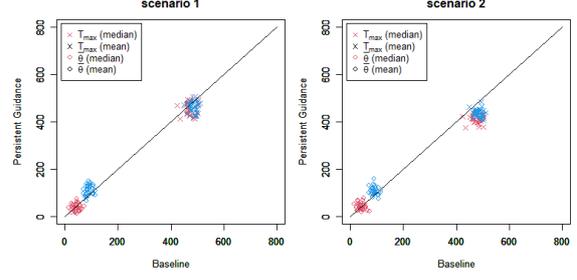


Figure 11: Persistent guidance with limited drifts, with and without failure, for 64 drift cases.

system: missing information owing to failed system components and inaccurate information caused by delays and drifts in the message-passing mechanism used for information sharing among components. Spatial interpolation was tested to mitigate the former uncertainty, which resulted in positive effects in both scenarios. However, the results did not reach baseline levels (Fig. 8). Figure 10 demonstrates that persistent guidance is promising, because most plots lie on or below the diagonal lines in both scenarios, indicating comparable or better results than the baseline. Although these two methods cannot be directly compared owing to their differing assumptions, these experiments may provide valuable insights for real-world applications.

This figure also highlights a critical issue not observed in other cases: a small number of plots deviate significantly from the diagonal lines, indicating the presence of RBCE. Unlike in other cases, delays and drifts in the distributed system adversely affect the guidance results for persistent guidance. Multiple regression analysis revealed intriguing insights:  $\epsilon_b^*$  had a negative effect, whereas  $\epsilon_z^*$  had a positive effect on the guidance performance. The negative effect of  $\epsilon_b^*$ , such as when  $\epsilon_b^* = 0$ , is counterintuitive, because this condition implies that all nodes estimate their local information simultaneously, ensuring consistency, provided  $\epsilon_z^* = 0$ . However, when  $\epsilon_z^* \neq 0$ ,  $\epsilon_b^* = 0$  causes inconsistencies in the graph information, resulting in several cycles on the graph. In persistent guidance, all agents adhered to the sign directions without exhibiting random behaviors, which led to catastrophic outcomes when the graph contained cycles.

The techniques employed in this study are relatively straightforward, including the *Broadcast* and *UpdateSign* algorithms, as well as spatial interpolation. Although these techniques can be enhanced using more advanced methods, doing so may pose challenges. We developed a pretrained denoising autoencoder that learns fire-spread patterns in the environment and incorporates them into signs for spatial interpolation. However, this approach resulted in an in-

ferior performance compared to the simple spatial interpolation technique proposed in this study.

Furthermore, the optimal placement of signage agents and their connections within the floor plan is crucial but presents challenges. Although simple agents have been used for system evaluation, studying the impact of human factors on evacuation guidance performance is also crucial. Addressing these issues will be essential for future studies.

## 8 CONCLUSION

A distributed dynamic evacuation guidance system utilizing *Broadcast* and *UpdateSign* algorithms was proposed. System performance was evaluated using an asynchronous multiagent simulation framework. Two strategies were introduced to mitigate the adverse effects of component failure. Remarkably, the system with failed components not only maintained functionality, but also performed comparably to a fully operational system. This advancement represents a significant step toward developing evacuation support systems capable of operating effectively under disaster conditions, offering valuable insights into creating resilient systems that can sustain functionality in harsh environments.

## ACKNOWLEDGEMENT

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

## REFERENCES

- Baidal, C., Arreaga, N., and Padilla, V. (2020). Design and testing of a dynamic reactive signage network towards fire emergency evacuations. *International Journal of Electrical and Computer Engineering (IJECE)*, 10:5853.
- Galea, E., Xie, H., Deere, S., Cooney, D., and Filippidis, L. (2017). Evaluating the effectiveness of an improved active dynamic signage system using full scale evacuation trials. *Fire Safety Journal*, 91.
- Galea, R. E., Xie, H., and Lawrence, J. P. (2014). Experimental and survey studies on the effectiveness of dynamic signage systems. *Fire Safety Science*, 11:1129–1143.
- Haghani, M. (2020a). Empirical methods in pedestrian, crowd and evacuation dynamics: Part I. experimental methods and merging topics. *Safety Science*, 129:104743.
- Haghani, M. (2020b). Empirical methods in pedestrian, crowd and evacuation dynamics: Part II. field methods and controversial topics. *Safety Science*, 129:104760.
- Helbing, D., Farkas, I., and Vicsek, T. (2000). Simulating dynamical features of escape panic. *Nature*, 407(28):487–490.
- Kawahara, J., Hara, T., and Sasabe, M. (2023). On robustness against evacuees' unexpected movement in automatic evacuation guiding. *Computers and Electrical Engineering*, 105:108531.
- Lin, H.-M., Chen, S.-H., Kao, J., Lee, Y.-M., Lin, C.-Y., and Hsiao, G. (2017). Applying active dynamic signage system in complex underground construction. *International Journal of Scientific & Engineering Research*, 8.
- Lovreglio, R., Fonzone, A., and dell'Olio, L. (2016). A mixed logit model for predicting exit choice during building evacuations. *Transportation Research Part A: Policy and Practice*, 92:59–75.
- Lujak, M., Billhardt, H., Dunkel, J., Fernández, A., Hermoso, R., and Ossowski, S. (2017). A distributed architecture for real-time evacuation guidance in large smart buildings. *Computer Science and Information Systems*, 14:257–282.
- Nguyen, V.-Q., Vu, H.-T., Nguyen, V.-H., and Kim, K. (2022). A smart evacuation guidance system for large buildings. *Electronics*, 11:2938.
- Rust, P., Picard, G., and Ramparany, F. (2020). Resilient distributed constraint optimization in physical multi-agent systems. *24th European Conference on Artificial Intelligence - ECAI 2020*, pages 195–202.
- Tsurushima, A. (2024a). Integrated simulation approach for dynamic distributed evacuation guidance under fire spread and rare but catastrophic events. *Proceedings of the 16th International Conference on Agents and Artificial Intelligence - Volume 1 (ICAART2024)*, pages 105–116.
- Tsurushima, A. (2024b). Simulation analysis of evacuation guidance using dynamic distributed signage. *Proceedings of the 16th International Conference on Agents and Artificial Intelligence - Volume 1 (ICAART2024)*, pages 179–188.
- Wilensky, U. (1999). Netlogo. Center for Connected Learning and Computer-Based Modeling, Northwestern 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.
- Zhao, H., Winter, S., and Tomko, M. (2017). Integrating decentralized indoor evacuation with information repositories in the field. *ISPRS International Journal of Geo-Information*, 6(7):213.
- Zu, Y. and Dai, R. (2017). Distributed path planning for building evacuation guidance. *Cyber-Physical Systems*, 3(1-4):1–21.