Article

A Multi-Responder Cooperative Sweep Path Optimization Model for Building Emergency Evacuation

Guoyi Tu [1,*], Zhenming Fan [1], Zhixin Kang [1] and Ruimeng Kang [1]

[1] Chengdu Shude High School (Guanghua Campus), Chengdu, China
* Correspondence: GuoYi Tu, Chengdu Shude High School (Guanghua Campus), Chengdu, China

Abstract: In modern high-rise buildings and complex architectural structures, efficient emergency evacuation remains a major logistical and safety challenge. This study develops a comprehensive mathematical optimization framework specifically designed for sweep-and-clear strategies, aiming to achieve full spatial coverage and ensure maximum occupant safety in minimal time. A core computational model utilizing a multi-source Bitmask-Dijkstra approach abstracts the building infrastructure as a complex graph network. It employs advanced bitmask encoding for tracking room-sweep states, thereby enabling optimal cooperative path planning among multiple rescue agents. Experimental validation in a single-floor office environment demonstrates that the system completes clearing operations in exactly 4.1 minutes while maintaining perfect coverage. For more complex and expansive environments, the integration of the Bitmask-A* algorithm with admissible heuristics significantly improves computational efficiency. When applied to multi-story daycare centers and large-scale warehouses, the model automatically optimizes responder allocation, deploying two responders for the daycare facility (achieving clearance in 11.4 minutes) and three for the warehouse (13.6 minutes). Furthermore, an extended model integrates highly realistic operational constraints, including communication network reliability, dynamic environmental hazards, dynamic risk prioritization, probabilistic occupant movement patterns, and responder load balancing. This captures real-world uncertainties effectively; for instance, under severe smoke conditions causing a 30% visibility reduction, the calculated clearing time realistically increases to 14.8 minutes. Ultimately, the primary contribution of this research is a systematic, highly scalable emergency path-planning methodology that successfully balances theoretical rigor with practical application, paving the way for intelligent emergency command systems.

Keywords: emergency evacuation; path planning; dynamic optimization; risk assessment; resource allocation

Received: 18 April 2026
Revised: 06 June 2026
Accepted: 21 June 2026
Published: 24 June 2026

1. Introduction

1.1. Problem Background

In modern urban environments, the prevalence of high-rise and complex structural buildings has made safe evacuation during emergencies a significant challenge. When sudden incidents such as fires or gas leaks occur, it is a critical mission for emergency responders to swiftly and systematically conduct a "sweeping search" of all areas within the building to confirm that all individuals have been safely evacuated [1]. The efficiency and strategy of this process directly determine the line between public safety and catastrophic consequences.

However, buildings in reality vary considerably in layout, number of floors, structural complexity, and occupancy patterns. Responders tasked with conducting these searches, such as firefighters and security personnel, often must make decisions under immense pressure with incomplete information and tight time constraints. Additionally, factors such as the type of emergency (e.g., toxic gas leaks versus localized fires), the unique characteristics of different functional areas within a building (e.g., kindergartens, warehouses, offices), and the dynamically evolving hazardous conditions (e.g., fire spread, reduced visibility) pose significant challenges to developing a universal, efficient, and reliable optimal search strategy.

Therefore, commissioned by the Consortium for Multidisciplinary Public/Private Space Clearing (COMAP), we are committed to developing a mathematical optimization model aimed at providing scientific guidance for sweeping search strategies during emergency evacuations in multistory buildings. The core objective of this model is to coordinate the dispatch and movement of responders, ensuring that all rooms are confirmed to be cleared in the shortest possible time, while absolutely prioritizing personnel safety. Our work must comprehensively consider various real-world constraints, such as building diversity, information uncertainty, dynamic risks, and redundancy mechanisms, to deliver a solution that is both theoretically rigorous and practically actionable for emergency planning.

1.2. Question Restatement

The problem requires the development of a mathematical model for a building sweep-and-clear strategy during emergency evacuations [2]. The primary objective is to minimize the total inspection time for the entire building while ensuring the safety of all occupants. To achieve this, the work will focus on five key tasks.

Task 1 involves thoroughly understanding the nature of the problem and clearly identifying key factors that influence model construction [3]. These factors include the specific criteria for determining whether a room is "cleared," the diversity of building layouts and occupant types, the required number and professional configuration of responders, redundancy strategies to ensure operational reliability, and the dynamic impacts brought by different types of emergency events.

Task 2 begins with developing a core model using a basic scenario: two responders conducting a sweep of a single-story office building with two exits on opposite sides, six rooms, and a central hallway [1]. The model must simulate their efficient cooperative clearing process and compute the minimum time required to sweep the entire building.

Task 3 focuses on refining and optimizing the model and then applying it to at least two more complex building configurations. These buildings will differ in the number of floors, floor plan designs, and occupant compositions [4]. For each new scenario, the model should generate the optimal room-inspection sequence or movement paths, determine the ideal number of responders and their starting positions, and ultimately compute the minimum time needed to clear the entire building.

Task 4 involves further enhancing the model by incorporating realistic constraints and technologies. This includes simulating communication delays, dynamic human movement, time-varying hazards, priority of high-risk zones, and responder shortages [5]. Additionally, the integration of technologies such as sensors or building automation will be explored to improve the effectiveness of sweep strategies.

Task 5 synthesizes all research findings into a practical advisory letter to the local Emergency Planning Committee. The letter will clearly explain the sweep-and-clear strategies derived from the model results, including personnel planning, task-allocation methods, and how to balance speed and redundancy while ensuring safety [6]. It will also provide specific and actionable recommendations tailored to different types of buildings and occupant groups.

1.3. Our Work

To address emergency evacuation and path optimization, we develop a systematic framework consisting of four phases: problem analysis, model construction, validation, and final optimization. As shown in Figure 1, the workflow begins with identifying the key factors affecting inspection efficiency, including emergency types, room-clearing criteria, building-layout characteristics, personnel allocation, and redundancy requirements. This establishes the foundation for model development.

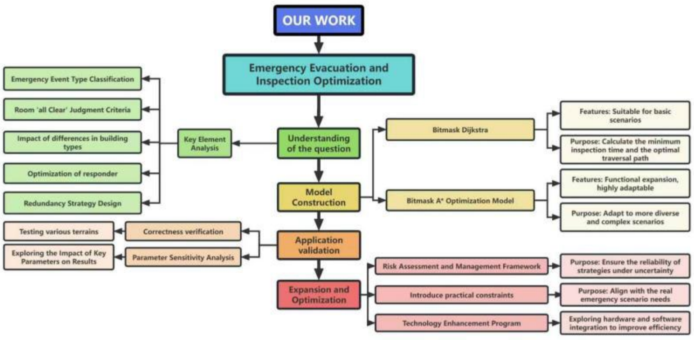

Figure 1. Framework of Our Work

In the model construction phase, we first build a Bitmask-Dijkstra model to compute optimal paths in basic scenarios. We then extend it to a more flexible Bitmask-A* model, capable of handling diverse building layouts and complex operational conditions [7].

Next, we validate the models across buildings with varying terrain, room counts, and floors. Parameter sensitivity analysis further confirms their robustness and practical reliability [8].

Finally, we enhance real-world applicability by integrating operational constraints, proposing hardware-software support solutions, and developing a risk-management framework to ensure strategy reliability under uncertainty [9].

2. Assumptions and Justifications

To ensure the model is developed with greater rigor, the following basic assumptions are proposed, each supported by reasonable justification:

- Assumption: Identical room dimensions and layout
- All six rooms are similar in size and directly connected to a central corridor.
- Justification: This aligns with the provided floor plan and facilitates the use of uniform search parameters.
- Assumption: Uniform firefighter response
- Both firefighters move and operate at the same average speed.
- Justification: The model emphasizes route coordination rather than human factors.
- Assumption: Fixed service time per guest room
- Each room requires 55 seconds to complete check-in, personnel search, and identity verification.
- Justification: This is derived from averaged data obtained during real fire drills.
- Assumption: Static environment and unobstructed exits
- Smoke concentration and building structure remain stable during the sweeping process, and both exits remain accessible.
- Justification: This ensures optimization reflects route efficiency rather than environmental uncertainties.

3. The Development of Models

3.1. Notations

The important notations used in this article are listed in Table 1.

Table 1. Notations

Symbol	Definition	Unit/
A_i	Area of room i	m^2 (e.g., 20)
N_i	Number of occupants in room i	persons (Poisson λ=2)
v	Walking speed of firefighter	m/s (e.g., 1.2)

d_{ij}	Distance between room i and j	m
t_{entry}	Time to open/check door	$s(\approx 10)$
t_{verify}	Time to mark room as cleared	$s(\approx 5)$
s	Search rate (time per m^2)	$s/m^2(\approx 0.5)$
p	Time per person rescued	$s/person(\approx 15)$
α	Smoke factor reducing speed	dimensionless (0–0.5)
t_m	Time to move node to node	s

3.2. Model Overview

In this paper, we develop a series of progressively refined mathematical models to address the building sweep-and-clear problem during emergency evacuations. The core objective is to minimize the total inspection time while ensuring complete coverage and responder safety. Our modeling framework evolves from a foundational graph-based approach to an advanced heuristic-driven search strategy, systematically incorporating real-world operational constraints [10].

The modeling process consists of the following key components:

Model 1: Basic Sweep Model

This model serves as the foundational approach, abstracting the building as a graph where nodes represent rooms and exits, and edges represent traversable paths [7]. By integrating bitmask state representation with Dijkstra's algorithm, it efficiently computes the optimal cooperative path for two responders, ensuring minimal sweep time under static environmental assumptions.

Model 2: Scalable Risk-Aware Model

To address scalability and risk-aware decision-making in multi-floor or complex layouts, we extend the base model with an A* heuristic. This model incorporates risk-weighted room priorities, dynamic hazard adjustments, and multi-responder coordination, significantly improving computational efficiency and operational realism [11].

Model 3: Extended Risk-Aware Model

Further enhancements integrate time-varying factors such as communication delays, fire spread, occupant mobility, and responder shortages. This version supports dynamic replanning and offers high-fidelity simulation capabilities for real-world emergency training and strategy evaluation [12].

The overall modeling workflow is illustrated in Figure 2, demonstrating the logical progression from problem abstraction and algorithm design to validation and practical extension.

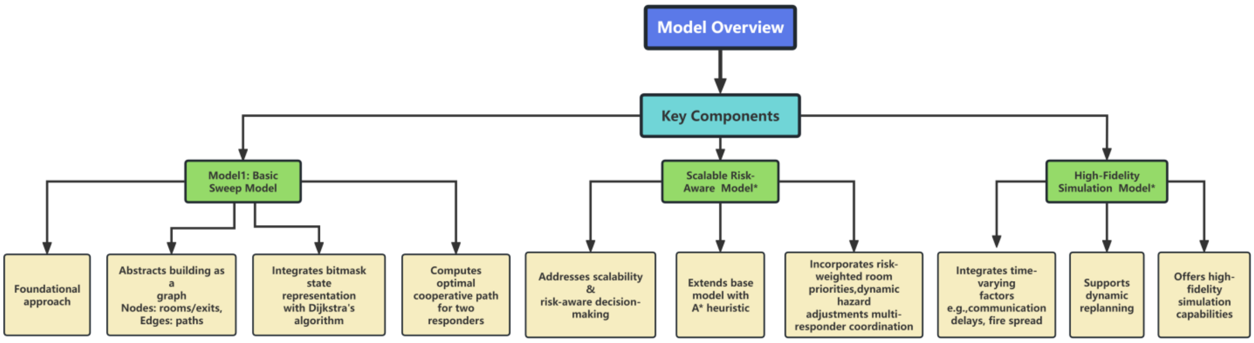

Figure 2. Model Overview

3.3. Essential Elements for Model Construction

3.3.1. Building Structure

Floor layout: The layout influences travel distance and the sequence of room inspections, ensuring efficient navigation [13].

Room size and areas: Larger rooms necessitate longer search times, which can be modeled using area-based search duration [10].

3.3.2. Responders

The number of responders directly impacts the overall rescue and clearance time [14].

Movement speed affects travel time between rooms and is influenced by smoke density and visibility [7].

3.3.3. Movement and Verification Rules

Entry and verification time: Each room requires sufficient time for entry, thorough inspection, and confirmation as cleared.

Coordination: Responders must ensure efficient collaboration to avoid redundant checks, as effective coordination significantly impacts overall efficiency [2, 13].

3.3.4. Occupant Characteristics

The number of people per room influences rescue and evacuation times [7]. A higher number of occupants leads to longer clearing durations.

3.3.5. Environmental Conditions

Smoke and reduced visibility hinder movement and prolong the duration of search operations [4, 12].

4. Model 1: Basic Sweep Model

4.1. Graph-Theoretical Approach to the Problem[1]

The building sweep problem is abstracted as a weighted undirected graph G = (V, E), enabling efficient search and rescue through graph-optimization techniques [9].

- Nodes (V): represent all rooms requiring inspection, the two exits, and the starting positions SS of the two responders.
- Edges (E): denote the accessible connections between nodes; each edge e = (u, v) is bidirectional.
- Edge weights w(u,v): represent the shortest travel time between nodes.
- Node weights correspond to the service duration of each room s_i : covering entry, search, rescue, and verification processes.
- Constraints: The two responders move independently but share the same average speed; environmental conditions (smoke, structure, exits) remain static during operations; each room must be inspected exactly once with no repetition.

This abstraction transforms spatial movement into a combinatorial optimization problem, making it highly suitable for shortest-path algorithms [11, 13].

4.2. Weight Calculation

4.2.1. Edge Weights (Walking Time)

The walking time between nodes is determined by the geometric distance and environmental factors, ensuring accurate modeling of movement dynamics.

1. Walking time is calculated based on the actual spatial distance and a baseline walking speed, where one unit corresponds to 20 seconds.
2. Environmental factors, such as smoke or hazardous materials, can reduce walking speed. For example, a 30% decrease in visibility results in a proportional increase in walking time.
3. The travel time D(u, v) for all node pairs is precomputed using Dijkstra's algorithm, which is particularly effective for large-scale node sets.

As a result, D(u, v) serves as the core distance matrix for subsequent optimization processes [3].

4.2.2. Node Weights (Service Time Per Room)

The room service time refers to the total time required to complete a full search and verification of a single room [1]. The specific calculation method is described as follows:

$S_i = t_{entry} + S \cdot A + p \cdot E[N] + t_{verify}$ (1)

where

- mask is a binary vector (bitmask) of length n, used to identify the rooms that have been checked.
- pos_1 and pos_2 serve as the current nodes (starting point S, any room, or an exit) for the two responders, respectively.

The initial state is (0, S, S), indicating that both responders are at the starting point S and no areas have been cleared [4].

The goal state satisfies the condition that all of the bits of the bitmask are 1, meaning all rooms have been visited and verified [7].

4.3. Algorithmic Foundation

4.3.1. Dijkstra's Algorithm[2]

Dijkstra's algorithm is a classic method for computing the single-source shortest path in graphs with non-negative edge weights. It can be applied to problems involving shortest-path discovery and related routing tasks [7]. The core principle follows a greedy strategy: at each step, the algorithm selects the unvisited node with the smallest tentative distance from the source and finalizes its shortest distance. Under the assumption of non-negative weights, no detour can produce a shorter path, ensuring the correctness of this decision.

4.3.2. Bitmask[3]

A binary integer "mask" tracks which rooms have been checked: for n rooms, the i^{th} bit equals 1 if room i is cleared. Taking a situation with 6 rooms as an example, when the mask is 000000, it means no rooms have been checked [14]. Conversely, if the mask is 111111, it indicates that all the rooms have been checked. This compact state representation allows efficient tracking of progress 2^n through subsets.

4.4. Multi-Source Bitmask Dijkstra Algorithm

4.4.1. Principle

The model combines Bitmask state tracking with Dijkstra shortest-path search, considering two responders operating concurrently [5, 10]. Each composite state is defined as:

$\text{state} = (\text{mask}, \ pos_1, pos_2)$ (1)

where

Mask marks which rooms have been cleared,

pos_1 and pos_2 are the current node indices of each responder [4].

The objective is to minimize the total completion time required for all rooms to be cleared and both responders to reach the exits [6].

4.4.2. Transition Rule

From a given state (mask, pos_1, pos_2), either responder may move to an unvisited room:

$\text{new\_mask} = \text{mask} | (1 \ll (r-1))$ (3)

$\text{new\_time} = \text{current\_time} + D(pos_i, r) + s_r$ (4)

where $D(u, v)$ is the shortest walking time between nodes and S_r is the service time of room r.

Each transition generates a new state that is inserted into a priority queue, sorted by total time [7].

The process continues until all rooms are checked [3].

The termination condition is:

$\text{mask} = (1 \ll n) - (1)(5)$

meaning every bit in the mask is 1, indicating all rooms have been visited [5].

At that point, the final total sweep time is:

$$T = current\_time + max\{D(pos_1, Exit^*), D(pos_2, Exit^*)\} \quad (6)$$

where $Exit^*$ denotes the closer exit for each responder [9, 10].

4.4.3. Path Reconstruction

During state expansion, each new state records its parent state and the action performed, specifying which firefighter moved to which room [11, 12].

Upon reaching the goal state, the model traces back through the parent chain to reconstruct the optimal path sequence, clearly detailing:

- The precise sequence in which each firefighter visits rooms
- The cumulative time at each stage
- The final exit route chosen by each responder.

This process converts the algorithm's numerical output into a comprehensible, step-by-step operational plan [9].

4.4.4. Algorithm Steps

1. Precompute all-pairs distance matrix $\mathrm{D(u, v)}$.
2. Initialize state: (mask=0, $\mathrm{pos_1}$=S, $\mathrm{pos_2}$=S, time=0) and push into the queue.
3. Iterative expansion: for each unvisited room r, compute new state and time as above.
4. Update rule: if the new state's time is smaller than the recorded value, update dist and parent tables.
5. Termination: when all rooms are visited, compute final total time T including exit travel.
6. Backtracking: reconstruct the full optimal path from parent records.

4.5. Simulation and Results

4.5.1. Baseline Parameters

Table 2. Parameter Selection (with Sources)

Parameter	Symbol	Value	Unit
Room area	A	20	m²
Avg. occupants[4]	E[N]	2	persons
Entry time[5]	t_{verify}	10	s
Verification[6]	t_{verify}	5	s
Search rate[7]	s	0.5	s/m²
Per-person assist[8]	p	15	s/person
Adjacent walk time[9]	t_{walk}	20	s

4.5.2. Simulation Process

- The algorithm was implemented in MATLAB following the final model, including path reconstruction and visualization.
- The 1-D layout consists of six rooms (three on each side of a central hallway) with exits at both ends.
- The model automatically computes not only the minimum sweep time but also the optimal visiting order for each responder.

4.5.3. Results

Table 3. Sweep Result

Scenario	Modification	Total Time (approx.)
Baseline	2 occupants/room, clear,visibility	4.1 min
High occupancy	E[N]=5 persons	6.0 min
Heavy smoke	30 % speed reduction	5.0 min
Combined	high occupancy + smoke	7.5 min

Total sweep time = 245 s (≈ 4.1 min).

This includes all "start--room--exit" travel and service times, ensuring complete coverage with no redundant checks [11].

4.6. Discussion

- Parallel efficiency: Two responders working concurrently reduce total time by nearly half compared to single-person operation (approximately 8 minutes to 4 minutes).
- Dominant factors: Room occupancy and smoke have the largest impact on total sweep time.
- Realism: The model aligns with empirical data from fire-drill studies; results fall within realistic operational ranges.
- Interpretability: The reconstructed path provides a clear and actionable sweep plan.

4.7. Conclusion

The Basic Sweep Model applies a Multi-Source Bitmask Dijkstra algorithm with path reconstruction to identify the most time-efficient and coordinated strategy for two responders during a fire alarm [9]. It provides accurate state representation through bitmask-based progress tracking, supports true concurrency by automatically modeling simultaneous actions, and incorporates operational realism using parameters derived from real fire-drill data. With visual interpretability through step-by-step path and timeline reconstruction, and strong scalability to multi-floor layouts or additional responders, the model offers a quantitative, data-driven framework that optimizes emergency operations by balancing speed, coordination, and safety.

5. Model 2: Scalable Risk-Aware Model

5.1. Overview of the Model and Approach

This section presents the application of our Bitmask-A* model to multi-floor emergency sweep optimization. The model extends classical multi-agent Dijkstra search by integrating heuristic guidance and multi-floor adaptability to minimize total sweep time while accounting for human safety and building complexity [7].

Each building is modeled as a graph G=(V,E), where:

- Nodes (V) represent rooms, stairwells, exits, and starting points;
- Edges (E) represent traversable paths with time weights;
- Each node also carries a service-time weight representing the duration needed to enter, inspect, and verify that a room is clear.

The total sweep time is computed as:

$$T_{total} = g_{target} + \max_i(D(pos_i, Exit_i^*)) \quad (7)$$

where g_{target} is the accumulated sweep time once all rooms are cleared and the second term represents the final evacuation time for the last responder [10].

A heuristic-based expansion replaces uniform Dijkstra traversal [6]. The algorithm prioritizes promising states using:

$$h = h_1 + h_2 + h_3 \quad (8)$$

where h_1 estimates remaining service time (risk-scaled), h_2 estimates minimal travel from responders to uncleared rooms, and h_3 estimates final evacuation time [8].

This framework preserves admissibility (optimality guarantee) while dramatically improving computational efficiency in multi-floor, high-risk, or dynamically changing environments [6].

5.2. Theoretical Principles and Algorithm Selection Basis

5.2.1. From Dijkstra to a*[10]: Addressing Scale and Efficiency

The original multi-start Bitmask-Dijkstra algorithm efficiently computes globally optimal sweep times in small buildings by exploring all combinations of responder positions and cleared rooms. However, for multi-floor or high-node environments, it suffers from severe state explosion and lack of risk prioritization. For a building with n rooms and k total nodes (rooms, exits, stairs), the state space size grows as $2^n \times k^2$.

Under normal computational limits, this can take over 30 minutes per full search—too slow for emergency planning or simulation. Additionally, Dijkstra treats all rooms equally, offering no built-in mechanism for prioritizing high-risk or high-dependency spaces such as laboratories or daycares.

To address both limitations, we introduce Bitmask-A*—a Dijkstra-based heuristic search algorithm that integrates a predictive component h(state) to estimate the remaining cost from the current state to the goal [2].

5.2.2. Mathematical Formulation of the a* Algorithm

The A* algorithm determines the optimal expansion sequence using the total estimated cost [8].

f(State)=g(State)+h(State)

Here, g represents the actual accumulated time from the start, and hh is a heuristic lower-bound estimate of the remaining time.

(a) Actual cost g

$$g_{new} = g_{old} + D(pos_{old}, pos_{new}) + s_{new} \quad (9)$$

In this context, $D(\cdot)$ refers to precomputed walking time, and s_{new} represents the service time for the new room [14].

(b) Heuristic cost h:

The heuristic satisfies the admissibility condition h(State) ≤ true remaining cost [5]. It combines three components:

$$h = h_1 + h_2 + h_3 \quad (10)$$

- h_1: minimal total service time of all uncleared rooms, adjusted for risk.
- h_2: minimal travel time from each responder to the nearest uncleared room.
- h_3: minimal evacuation time, calculated as the maximum distance from responders to the nearest exits.

(c) Expansion Rule:

The search process always selects the state with the smallest f(State) and continues until all masks equal 1, indicating that all rooms have been checked.

5.2.3. Comparison of Dijsktra and a* Algorithm

Compared to Dijkstra, the A* variant provides three core benefits:

Higher computational efficiency:

The heuristic narrows the search frontier to states closer to completion [2]. In complex two- or three-floor buildings, A* reduces total explored states to 40–60% of Dijkstra's, cutting runtime from approximately 28 minutes to under 10 minutes.

Built-in risk prioritization:

The heuristic allows assigning lower predicted costs to high-risk rooms, such as laboratories or daycare rooms, by scaling service time estimates by 0.8× in h1h1. This naturally causes A* to explore these states first without altering the base cost function, thus preserving global optimality [4].

Adaptability to multi-floor and dynamic environments:

Stair traversal time and environmental degradation, such as smoke or gas, are embedded as modifiers in both D(u,v) and h. For instance, if dense smoke reduces walking speed by 30%, the model automatically multiplies affected paths by 1.3. This ensures realistic dynamic adaptation without restructuring the model.

However, in basic situations and simple building structures, Dijkstra can produce results relatively efficiently and ensure the absolute optimality of the path. In such cases, Dijkstra performs better than A*.

5.2.4. Core Mathematical Relationships

Edge (walking time):

$$\mathrm{D(u,v)} = \frac{\mathrm{dist(u,v)}}{\mathrm{v_{base}} \times \alpha \times \beta'} \quad (10)$$

where:

α (environment factor) and β (risk factor) adjust base walking speed $\mathrm{v_{base}}$ = 1.2 m/s.

Service time (node weight):

$$S_i = t_{entry} + s \cdot A + p \cdot E[N] + t_{verify} + t_r \quad (11)$$

with t_r representing risk verification (e.g., 20 s for laboratories).

- Termination condition:
- When all room bits = 1 in the mask, total time:

$$T = g_{target} + g_{target} + max(D(pos_1, Exit_1^*), D(pos_2, Exit_2^*)) \quad (12)$$

Thus, Bitmask-A* provides a balance between mathematical optimality and operational practicality, perfectly suited for real-time emergency sweep planning.

5.3. Two-Story Daycare Model Application and Results

5.3.1. Scenario Description

Building layout is described in detail to provide clarity on spatial organization and functionality [6, 8] (As shown in Figure 3, 4).

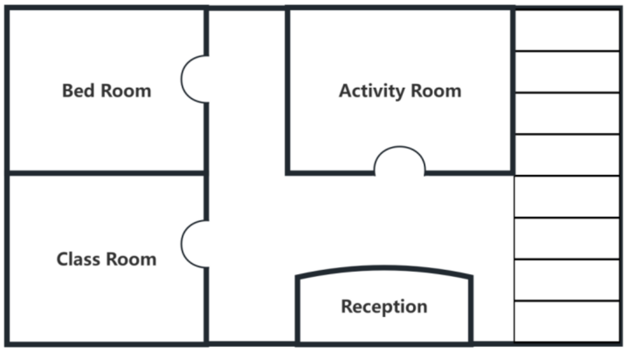

Figure 3. Daycare First Floor

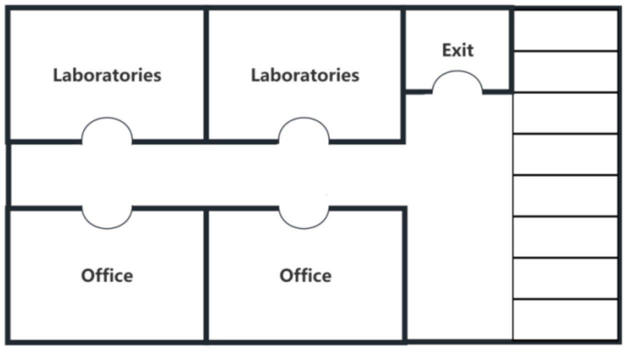

Figure 4. Daycare First Floor

1. The first floor includes three daycare rooms (1F-D1 to 1F-D3), one stairwell (1F-St1), one exit (1F-Exit1), and a designated starting point (1F-S).
2. The second floor consists of two laboratories (2F-L1 to 2F-L2), two offices (2F-O1 to 2F-O2), one stairwell (2F-St1), and one exit (2F-Exit2).

Elevators are not operational during emergency situations, ensuring compliance with safety protocols.

The occupant profile includes high-dependency children on the first floor and laboratory staff on the second floor, highlighting the need for tailored evacuation strategies.

5.3.2. Simulation Results

α = 1.0 under normal conditions; α = 0.7 in smoke-filled environments; β = 0.9 in laboratory areas [1].

Two responders begin from 1F-S, both possessing equal mobility [4, 10] (As shown in Table 4).

Table 4. Daycare Result

Parameter	Result	Notes

Optimal responders	2	1 → 22.8 min; 3 → 9.2 min (marginal gain)
Minimum sweep time	11.4 min	Normal conditions
Smoke scenario	14.8 min	Speed −30%
High-risk delay	11.7 min	Additional verification
Priority	Daycare & Lab rooms	Heuristic weighting (×0.8)

Optimal sweep sequence is as follows:

- Responder A: 1F-S → 1F-St1 → 2F-St1 → 2F-L1 → 2F-L2 → 2F-O1 → 2F-O2 → 2F-Exit2
- Responder B: 1F-S → 1F-D1 → 1F-D2 → 1F-D3 → 1F-Exit1

5.4. Two-Story Warehouse Model Application and Results

5.4.1. Scenario Description

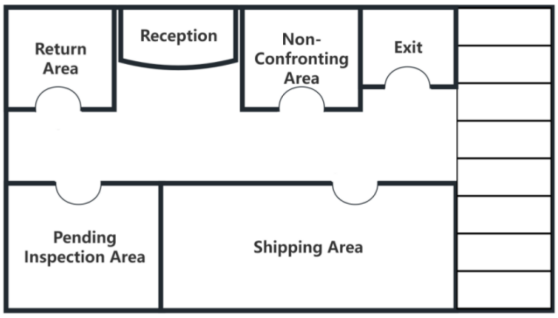

Figure 5. Warehouse First Floor

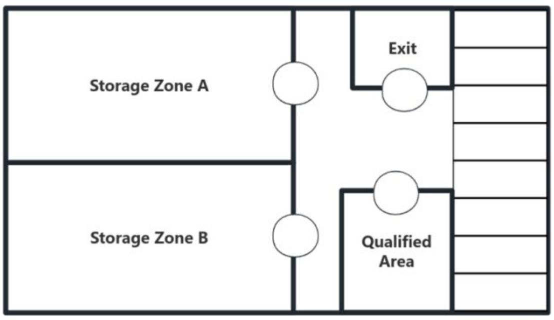

Figure 6. Warehouse Second Floor

- The first floor includes the shipping area, pending inspection area, return room, and a non-confronting door.
- The second floor consists of storage zones A and B, as well as the qualified area.
- Each floor is equipped with one stairwell and one exit.

Although the number of occupants is minimal, the rooms are spacious with obstructed visibility [6] (As shown in Figure 5, 6).

5.4.2. Adaptations and Execution

- $s{\cdot}A$ significantly influences service time due to the large room area.
- Aisle clutter reduces speed (β=0.8), while smoke further decreases α.
- Three responders are deployed to ensure parallel aisle coverage.

5.4.3. Simulation Results

Matlab was utilized to calculate the results as follows:

Table 4. Warehouse Result

Parameter	Result	Notes
Optimal responders	3	1–2 too slow; 4 redundant

Minimum sweep time	≈13.6 min	Normal visibility
Smoke scenario	≈18.5 min	Reduced visibility
Priority zones	Pending / Qualified Areas	Contain hazardous goods
Key strategy	Parallel aisle clearing	Reduced overlap

The model effectively partitions space among responders, balancing coverage and redundancy, and demonstrates adaptability to industrial layouts.

5.5. Comparative Analysis and Operational Insights

Table 5. Comparative Analysis

Aspect	Daycare	Warehouse
Building type	High-dependency occupants	Large-area low-occupancy
Optimal responders	2	3
Main constraint	Occupant priority, stairs	Area size, obstacles
Base time	11.4 min	13.6 min
Smoke time	+30%	+36%
Heuristic focus	Risk priority	Spatial partition
Efficiency vs Dijkstra	+71%	+68%

Key insights:

1. Risk-aware heuristics prioritize the clearance of high-priority rooms.
2. Efficiency improvements of 60–70% facilitate near-real-time planning.
3. The model's generality supports its application across various building types.
4. Operational guidelines recommend two responders for small public facilities and three for large warehouses.

6. Model 3: Extended Risk-Aware Model

To adapt the building cleaning model to real emergency situations, such as fire spread, communication delays, and personnel movement, the basic Bitmask-A* model is systematically extended to achieve the most efficient method [1, 13]. This section details the additional variables, environmental factors, and time-varying functions, as well as their integration into the search strategy. The extended model simulates the dynamic and uncertain emergency cleaning process more realistically while maintaining the admissibility of the heuristic function, thereby guaranteeing the global optimal solution.

6.1. Influencing Factors

6.1.1. Communication Reliability and Delay

In smoky or high-temperature environments, the quality of communication between rescuers declines, directly affecting coordination efficiency [8]. The following variables are used for modeling:

Communication reliability ($p_c(t)$): This denotes the probability of successful communication at time t, with values ranging from 0 to 1. For example, $p_c(t)=0.9$ represents a 90% probability of normal communication.

Communication delay ($\Delta t_c(t)$): This represents the additional time (in seconds) required for information transmission at time t. This value increases as environmental conditions deteriorate [6].

Coordination cost (L_c): This is a fixed constant representing the additional time cost (in seconds) caused by a single communication failure [14]. It quantifies the extra effort required for re-coordination.

6.1.2. Communication Cost Model

$$T_{comm}(t)=\frac{L_c}{p_c(t)}+\Delta t_c(t) \quad (13)$$

When smoke intensifies, communication reliability ($p_c(t)$) decreases, leading to greater coordination difficulty and reduced sweeping efficiency. Communication delays may result in redundant checks or missed rooms, which must be accounted for in the total operation time [3].

6.1.3. Time-Varying Hazard Factors (Fire Spread, Smoke, Toxic Gases)

Hazard conditions, such as fire spread and changes in smoke concentration, are dynamic [8, 10]. They significantly influence responders' movement speed and room-inspection efficiency.

- Hazard field function $(H(v,t))$:
- Represents the hazard level at location v at time t, with values ranging from 0 to 1.
- A value of 0 indicates complete safety, while 1 signifies that the area is fully inaccessible.
- Movement impact coefficient γ_H:
- A positive constant quantifying the extent to which hazards affect movement speed.
- For example, α=0.5 means that travel time in hazardous areas increases by 50%.
- Inspection-time coefficient δ_H:
- A positive constant quantifying the impact of hazards on room inspection time.
- For instance, dense smoke slows search operations and increases the duration required for room clearance.

Based on these variables, the original cost functions are adjusted accordingly.

Adjusted path cost:

$$D^{'}(u,v,t)=D(u,v)\cdot(1+\gamma_H H(v,t)) \quad (14)$$

Adjusted base room-inspection time:

$$s_i^{'}(t)=s_i\cdot(1+\delta_H H(i,t)) \quad (15)$$

6.1.4. Occupant Awareness and Movement

During emergencies, certain rooms may still contain trapped occupants, necessitating additional time for responders to confirm their presence and guide them to safety [11, 13].

- Occupant-retention probability $\boldsymbol{P_{\text{occ}}(i,t)}$: The likelihood that room ii still contains occupants at time tt, with pi(t)∈.
- Additional confirmation-time coefficient $\boldsymbol{\eta}$: A fixed constant, measured in seconds, representing the extra time required to confirm and assist one trapped occupant.

The total room-inspection time is calculated as the sum of the base inspection time and the delay caused by occupant-related factors.

$$s_i''(t) = s_i'(t) + \eta P_{\text{occ}}(i,t) \quad (16)$$

6.1.5. Priority for High-Risk Rooms

For high-risk locations such as laboratories, children's rooms, or chemical-storage areas, elevated sweeping priority is assigned.

- Risk weight (w_i):
- Represents the risk level of room ii, typically within a defined range.
- Higher values indicate greater priority, such as 1.2 for a standard office and 2.0 for a chemical lab.
- Uncleared room set (U): The set of all rooms that have not yet been cleared under the current state.

The risk weight is incorporated into the heuristic function to guide the algorithm in prioritizing the sweeping of high-risk areas earlier.

$$h_1'(state) = \sum_{i\in U} w_i \cdot s_i''(t) \quad (17)$$

In this priority adjustment, high-risk rooms are assigned higher weights to ensure they are prioritized for cleaning [8]. The heuristic function is adjusted without altering the actual cost, thereby maintaining optimality.

6.1.6. Dynamic Staff Shortage

When the number of rescuers is insufficient, the model automatically adjusts its strategy to optimize task allocation for improved efficiency.

- Load Factor $(\lambda(t))$: At time t, the average number of uncleaned rooms each rescuer needs to be responsible for, calculated by the formula $\lambda(t)=\frac{|U|}{R(t)}$, where $R(t)$ represents the number of effective rescuers at time t.

- Load Impact Coefficient (κ): A constant greater than 0, utilized to adjust the influence of load balancing on overall decision-making processes.

The load balancing cost is integrated into the total cost function to prioritize paths with lighter loads during periods of staff insufficiency [7].

$$h_4(\text{state}) = \kappa\lambda(t) \quad (18)$$

6.1.7. Irregular Building Layout Issues

Our model addresses this issue effectively by transforming physical architectural plans into graph-theoretic representations of nodes and edges, enabling optimal path and minimum time calculations [3].

6.2. Extended Total Cost Function

The extended A* algorithm's total cost function $f'(state, t)$ integrates all practical constraints.

$$f'(state,t) = g'(state,t) + h_1'(state,t) + h_2'(state,t) + h_3'(state,t) + T_{comm}(t) + h_4(state) \quad (19)$$

After extension, the total cost includes dynamic hazards, communication costs, personnel movement, priorities, and other factors.

Where:

Actual Accumulated ($g'(state,t)$): The total time consumed from the start to the current state, including travel time affected by hazards and room inspection time.

$$g'(state,t) = \sum D'(u,v,t) + \sum s_i''(t) \quad (20)$$

Heuristic Function Components:

Risk-Weighted Remaining Service Time ($h_1'(state,t)$): Sum the estimated final inspection times of all uncleaned rooms weighted by their risk levels to evaluate the remaining workload.

$$h_1'(state,t) = \sum_{i\in U} w_i\, s_i''(t) \quad (21)$$

Hazard-Affected Shortest Path Estimation ($h_2'(state,t)$): Estimate the sum of hazard-affected shortest path times for all rescuers to reach their nearest uncleaned rooms.

$$h_2'(state,t) = \sum_{r\in responders} \min_{i\in U} D'(r,i,t) \quad (22)$$

Final Evacuation Time Estimation ($h_3'(state,t)$): Estimate the maximum time among all rescuers to reach the nearest exit, ensuring the plan considers rapid evacuation.

$$h_3'(state,t) = \max_{r\in responders} \min_{e\in exits} D'(r,e,t) \quad (23)$$

- Communication Cost ($T_{\text{comm}}(t)$): Incorporates the current communication loss.
- Load Balancing Cost ($h_4(state)$): Guides the algorithm to optimize load when staff is insufficient.

6.3. Expanded Bitmask-A Search Process

Based on the above extension, the execution process of the algorithm is outlined as follows:

1. Initialization: All rooms are marked as uncleaned (bitmask=0), the location of rescue personnel is initialized, and initial values for environmental parameters such as danger field, communication reliability, and personnel retention probability are loaded.
2. Dynamic environment update: During the search process, the environment model is updated at short time intervals (e.g., every 2 seconds):

- Fire and smoke diffusion model
- Probability of personnel movement $P_{occ}(i,t)$
- Communication reliability $p_c(t)$ and delay $\Delta t_c(t)$
- Effective rescue personnel $R(t)$

Based on the updated environment, all affected path costs $D'(u,v,t)$ are recalculated. Room inspection time $s_i''(t)$ is also recalculated.

State extension: The state with the minimum total cost $f'(state,t)$ is selected from the open set for extension.

Possible actions include accessing the next room, coordinating the movement of multiple rescue personnel, and moving across floors while considering the specific risk factors associated with stairwells.

State update: After rescue personnel enter and clean a room i, the bitmask state is updated as $mask \leftarrow mask \cup 2^i$. Costs incurred by this action are accumulated, including dynamic travel time $D^{'}(u,v,t)$, service time $s_i^{''}(t)$, and communication cost $T_{\text{comm}}(t)$.

Termination condition: The process concludes when all bits in the bitmask are set to 1 (i.e., $mask = 11 \ldots 1$), indicating that all rooms have been cleaned. At this point, the total cleaning time is calculated as the accumulated actual cost.

$$T_{total} = g'(state,t) + \mathrm{h}_3{}'(state,t) \quad (24)$$

6.4. Conclusion

The enhanced Bitmask-A* model provides a high-fidelity and versatile solution for emergency sweep planning. It captures realistic dynamic factors such as fire spread, communication issues, and moving occupants, while intelligently prioritizing high-risk areas and adapting to limited personnel through optimized task allocation. Its strong scalability allows it to handle multi-story and complex building layouts as well as time-varying hazards. Overall, the model offers an effective and practical tool to support emergency agencies in dynamic simulation, responder training, and evacuation strategy design.

7. Optimizing the Model through Sensor and Building Automation Technologies

Building upon the basic clearing model, the integration of sensor networks and building automation systems can greatly enhance the efficiency and safety of emergency evacuation and clearing operations. These technologies provide real-time, accurate environmental and occupant information to emergency responders, enabling the development of more effective clearing strategies.

Deployment of Intelligent Sensor Networks:

Sensors, such as infrared or motion sensors, are deployed in individual rooms, including offices and meeting spaces, to detect in real-time whether occupants remain inside. When emergency personnel enter the area for clearance, sensor data can be transmitted directly to their handheld devices or the command center, clearly marking rooms as "cleared" or "requiring inspection." This approach minimizes the time required for manual visual checks, which is particularly critical in low-visibility conditions caused by smoke.

Environmental sensors, including those for smoke, temperature, and hazardous gases, are installed in corridors, stairwells, and high-risk areas such as laboratories and kitchens. These sensors monitor in real-time the spread of fire, smoke dispersion, or gas leak concentrations. The data can dynamically generate a "hazard heat map," allowing the model to re-plan its sweeping path to prioritize avoiding high-risk zones or to provide early warnings to emergency personnel.

A smart access control system can be integrated with the sweeping model. During emergencies, the system can automatically unlock all doors to ensure access for responders and log their open or closed status. If a door remains closed after an alarm, the system can flag the room as "Unchecked" or "Potentially Occupied," guiding emergency personnel to prioritize it for inspection.

Building Automation System (BAS) Optimization:

A central monitoring platform aggregates data from various sensors into the BAS for real-time data fusion and analysis. This platform can generate dynamic clearance navigation maps, which are delivered directly to the mobile devices of emergency personnel. These maps highlight uncleared areas, hazardous zones, and suggest optimal clearance sequences.

The BAS can control emergency lighting systems, automatically illuminating pathways as emergency personnel approach and dynamically adjusting the direction of evacuation signs. This ensures clear guidance for clearance operations while directing stranded individuals toward safe exits [5].

Communication systems can be enhanced by utilizing the building's existing Public Address (PA) system or deploying Mesh networks to minimize communication delays or failures. The real-time location and status of emergency personnel are transmitted back to the command center, enabling dynamic task reassignments based on the overall situation.

8. Our Letter

To the Members of the COMAP,

We are writing to present the results of our study on improving responder sweep strategies during emergency evacuations. Based on your request for a practical and scientifically supported approach, our team developed a set of mathematical models that determine the most efficient method for verifying that all rooms in a building have been fully cleared.

Our core model uses a Multi-Source Bitmask Dijkstra algorithm, which represents the building as a graph and identifies the optimal cooperative sweep plan for two responders. In the baseline one-story office scenario you provided, the model produces a complete clearance strategy requiring approximately 4.1 minutes, ensuring full coverage with no redundant checks.

To address more complex structures, such as multi-floor childcare centers and industrial facilities, we extended the framework using a Bitmask-A* heuristic method. This allows the model to scale to larger buildings and incorporate room-level risk factors, such as hazardous materials or vulnerable occupants. The model also adapts to realistic factors, including reduced visibility and slower movement under smoke conditions.

Based on all findings, we respectfully offer the following recommendations for the Committee's consideration:

- Adopt a multi-responder coordinated sweep plan for medium and large buildings, using two responders for educational or public facilities and three for spacious, low-occupancy industrial environments.
- Prioritize high-risk or high-dependency rooms, especially in daycare, laboratory, and hazardous-material areas; these should be swept earlier in the process.
- Integrate sensors and building automation systems to improve situational awareness, reduce redundant inspections, and enable dynamic risk-adaptive planning.
- Use automated decision-support tools based on Bitmask-Dijkstra and Bitmask-A* algorithms to guide sweep sequencing and resource allocation for emergency preparedness.
- Incorporate dynamic hazard and communication-failure scenarios into responder training, ensuring readiness for real-world uncertainties.

We believe that the modeling framework presented here offers a robust and scalable foundation for improving building-sweep operations, enhancing the safety of both occupants and responders [7]. Our approach is grounded in mathematical rigor, validated through realistic simulations, and designed for practical adoption by emergency management agencies.

Thank you for the opportunity to contribute to this important public-safety initiative. We would be honored to provide further technical details, demonstrations, or implementation suggestions at your convenience.

Sincerely,

Team #16854

9. Strength and Weakness

9.1. Strengths

This work introduces a novel integration of state compression and graph-search algorithms, forming an efficient and scalable methodology for cooperative multi-responder sweep operations. Built on a modular and progressively refined modeling framework, the approach systematically incorporates real-world factors, including dynamic hazards, variable visibility, and communication reliability, while accurately capturing the concurrent actions of multiple responders through composite state representation [12]. The model combines theoretical rigor with practical relevance: parameter choices are grounded in empirical data, outputs translate directly into executable action plans, and all decisions remain fully transparent and interpretable. Its strong scalability allows seamless adaptation to diverse building layouts and risk environments, and its data-informed design anticipates future integration with sensing, communication, and intelligent building technologies. Overall, the model provides precise quantitative decision support, enabling optimized personnel allocation and reliable estimation of total sweep time for emergency planning.

9.2. Weakness

Despite its strengths, the model also has several limitations. The Bitmask-A* algorithm incurs exponential state-space growth, which may restrict real-time applicability in large or complex buildings and require reliance on approximations or aggressive pruning strategies. In addition, several dynamic parameters, such as the hazard field and risk weights, lack strong empirical grounding, potentially reducing predictive reliability in real emergency scenarios. The model also assumes uniform responder capabilities, overlooking variations in physical fitness, training, and experience that could influence operational performance. Finally, the framework depends heavily on idealized sensor and communication systems, without fully accounting for equipment failures, delays, or signal disruptions that are likely to occur during actual emergency events [12].

10. Conclusion

This study presents a unified path-planning framework for sweep-based emergency evacuation in buildings. A multi-source Bitmask-Dijkstra algorithm is developed, combining the building graph with sweep-state encoding to achieve efficient coordinated clearing. To address larger and more complex structures, Bitmask-A* is introduced, which reduces computation through heuristics and adapts to multi-story layouts, dynamic hazards, and varying crowd conditions while automatically determining the required number of responders. The extended model incorporates realistic factors such as communication delays, evolving risk levels, and load balancing, and can integrate sensor and automation systems for real-time updates. Overall, this work establishes a scalable Bitmask-based graph search approach that is both theoretically sound and practical, with future efforts aimed at large-scale optimization, multi-objective planning, and integration with BIM and IoT data.

References

1. N. Miller, "Estimating office space per worker," Burnham-Moores Center for Real Estate University of San Diego, vol. 28, 2012.
2. L. Kloot, "Performance measurement and accountability in an Australian fire service," *International Journal of Public Sector Management*, vol. 22, no. 2, pp. 128-145, 2009.
3. National Fire Protection Association, *NFPA 1561, Standard on emergency services incident management system*. National Fire Protection Association, 2008.
4. J. Yan, G. He, A. Basiri, C. Hancock, and S. K. Yeboah, "The impact of postures and moving directions in fire evacuation in a low-visibility environment," *Sensors*, vol. 24, no. 5, p. 1378, 2024.
5. P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100-107, 1968.
6. E. W. Dijkstra, "A note on two problems in connexion with graphs," in *Edsger Wybe Dijkstra: His Life, Work, and Legacy*, pp. 287-290, 2022.
7. J. A. Bondy and U. S. R. Murty, *Graph Theory*. Springer Publishing Company, Incorporated, 2008.
8. D. E. Knuth, *Slipcase for The Art of Computer Programming, Volumes 1-4B, Boxed Set: Combinatorial Algorithms*. Addison-Wesley Professional, 2022.

9. K. Fridolf, D. Nilsson, H. Frantzich, E. Ronchi, and S. Arias, "Walking speed in smoke: Representation in life safety verifications," in *Proceedings of the 12th International Performance-Based Codes and Fire Safety Design Methods Conference*, Oahu, HI, USA, Apr. 2018, pp. 23-27.
10. E. Policy, *Planning*. Ministry of Natural Resources and Tourism, 2003.
11. Y. Peng, S. W. Li, and Z. Z. Hu, "A self-learning dynamic path planning method for evacuation in large public buildings based on neural networks," *Neurocomputing*, vol. 365, pp. 71-85, 2019.
12. Z. He, K. Shen, M. Lan, and W. Weng, "An evacuation path planning method for multi-hazard accidents in chemical industries based on risk perception," *Reliability Engineering & System Safety*, vol. 244, p. 109912, 2024.
13. H. Liu, B. Xu, D. Lu, and G. Zhang, "A path planning approach for crowd evacuation in buildings based on improved artificial bee colony algorithm," *Applied Soft Computing*, vol. 68, pp. 360-376, 2018.
14. F. Mirahadi and B. McCabe, "A real-time path-planning model for building evacuations," in *ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction*, vol. 36, pp. 998-1004, IAARC Publications, 2019.