mathematics

MDPI

Article

Mathematical Programming for Optimal Evacuation in Industrial Facilities

Carmine Cerrone [1], Massimo Paolucci [2] and Anna Sciomachen [1,*]

[1] Department of Economics and Business Studies, University of Genova, 16126 Genova, Italy; carmine.cerrone@unige.it
[2] Department of Computer Science, Bioengineering, Robotics and Systems Engineering, University of Genova, 16145 Genova, Italy; massimo.paolucci@unige.it
* Correspondence: sciomach@economia.unige.it; Tel.: +39-010-209-55484

Abstract

This paper presents an optimization framework for determining safe and efficient evacuation paths in complex industrial facilities. The proposed approach models the evacuation process through a timed flow network that captures both the structural characteristics of the layout and the temporal evolution of emergency conditions. The formulation accommodates real-time updates, enabling dynamic re-routing when certain areas or connections become inaccessible. Computational experiments on large-scale instances demonstrate the scalability of the model and its ability to provide complete evacuation plans under increasing demand. The results confirm predictable relationships between population size, time horizon, and evacuation completion, supporting its use as a decision support tool for both strategic planning and operational response.

Keywords: evacuation planning; timed flow networks; mathematical programming

MSC: 90

1. Introduction

Recently, the safe evacuation of people from public areas or work environments has received considerable attention in the scientific literature. In the face of unpredictable situations such as fires, hurricanes, tsunamis, and toxic gas leaks, it is particularly important to move people to safe areas in the shortest possible time. In addition to studying human evacuation behavior through the observation of historical evacuation videos, computer simulation has become an important tool for the analysis of evacuation dynamics.

Research on evacuation under emergency conditions can be broadly divided into three distinct research streams, as reported in the survey by [1]. The first stream empirically studies pedestrian behavior and crowd dynamics. The second stream uses mathematical models to simulate and describe pedestrian behavior during evacuations. The third stream of research on emergency evacuation develops models that assist in determining the optimal evacuation plan or design solutions that facilitate evacuation processes. In the current literature, most research belongs to the first two streams.

Several studies have shown that, among all types of emergency, fire-related incidents occur with the highest frequency and cause the most severe casualties and material damage. Consequently, emergency evacuation during fires has received increasing attention from researchers. Despite continuous advances in fire prevention and protection, industrial fires still occur with almost the same frequency and impose a very high cost on companies and

Academic Editor: Janez Žerovnik

Received: 15 December 2025
Revised: 28 January 2026
Accepted: 10 February 2026
Published: 11 February 2026

governments every year. To give an idea, according to [2], structure fires accounted for 34% of incidents but 83% of total losses.

Among industrial facilities, chemical and process plants are particularly prone to fires due to the presence of large quantities of flammable chemicals and a wide variety of ignition sources. Fires in chemical and process plants can range from small incidents that can be safely managed with portable extinguishers to large-scale tank fires that require full plant evacuation and fire department intervention [3]. Fires in large public buildings with complex layouts and a high density of occupants often cause a large number of casualties [4].

Minimizing casualties is a classical yet essential research topic in the field of safety engineering [5]. The evacuation of people represents the most critical option [6], and its successful execution can ensure maximum safety for individuals. Therefore, ensuring that evacuees are aware of and use the safest and/or fastest evacuation routes available to reach a safe location is of fundamental importance. To achieve this goal, timely and appropriate actions must be established to guide people to evacuate in an orderly, rapid, and safe manner [7].

Currently, emergency evacuation plans in most public areas are mainly based on predefined static evacuation routes, which cannot take into account real hazards in advance. When trapped individuals have difficulty perceiving hazard information, the existing evacuation strategy may become problematic or even useless if a fire breaks out along the designated evacuation routes. To address this issue, evacuations are usually assisted by on-site personnel who provide instructions based on received hazard information [8]. However, this is a very slow operational process, involving the dissemination of hazard information, the need to make route decisions, and the coordination of evacuation assistance [9]. In summary, current evacuation route decision methods suffer from low efficiency, poor stability, and difficulty in managing real incident scenarios. These problems can be effectively solved by adopting an automatic system capable of providing evacuation routes that account for the real-time emergency situation.

This paper describes the design of the Evacuation Route Optimization Approach (EROA) that has been developed as a component of the project EMMA (Enhanced Management for Manufacturing Safety), an innovative Industry 4.0 project aimed at developing an integrated IoT-based platform for predictive safety, prevention, and emergency management in industrial environments. The system combines several patented subsystems, including human–machine anti-collision devices, automated evacuation management, hands-free emergency response applications, and a secure data aggregation platform. The overall objective is to increase workplace safety through real-time monitoring, intelligent decision-making, and advanced human–machine interaction.

EROA has been developed to determine safe paths for people within an industrial facility, guiding them from their initial locations to safety exits or safe areas during an emergency evacuation. As detailed below, the basic algorithm utilizes a flow network model, solvable using a linear mathematical programming solver.

EROA operates in two complementary modes. In the offline mode, it simulates possible emergency scenarios to pre-compute robust evacuation paths. In the real-time mode, it dynamically updates these routes based on sensor data collected throughout the plant, such as environmental parameters, accessibility conditions, and personnel status, allowing the decision support system to identify safe escape routes as the situation evolves. Data from the field, gathered by both fixed and wearable sensors, feed into a simulation environment with an advanced graphical interface, enabling continuous monitoring and forecasting of emergency development.

The remainder of the paper is organized as follows: Section 2 reviews the relevant literature on evacuation planning. Section 3 introduces the proposed evacuation framework, including the layout graph representation and the timed flow network formulation. Section 4 describes the real-time application of the model and the update mechanism based on sensor information. Section 5 presents the computational experiments and discusses the results obtained on both realistic and large-scale instances. Section 6 concludes the paper and outlines directions for future research. Finally, a flowchart summarizing the overall approach proposed in the paper is provided in Appendix A.

2. Related Works

From the beginning of this century, in the literature, the emergency evacuation optimization problem has been approached with different methodological approaches. An early work by Cepolina [10] introduces an optimization framework for evacuation and shows that bottleneck capacities may drop under high upstream densities.

Many studies address the problem by considering its similarity to the shortest path problem, proposing variants of Dijkstra's algorithm which, when applied in various emergency evacuation contexts, show better performance in terms of both computational time and evacuation plans. Among others, Khakzad [11] proposes an advanced methodology that integrates Dijkstra's algorithm with a mathematical programming model to design evacuation plans for major fires in process plants. The approach selects routes by assigning people to safety areas, considering their capacity, thermal level, and safety distance. The objective is to minimize the evacuation time of people. The study suggests the opportunity for further work combining optimization with fire simulation to extend the applicability to different plant layouts and hazard scenarios. Zhang et al. [12] propose a dynamic routing scheme that blends offline mapping of an evacuation network with online updates driven by real-time observations. A Dijkstra-based precomputed network accelerates search, while a breadth-first update adjusts to events such as local congestion. A study on an underground shopping center suggests that macroscopic representations are effective for global evacuation coordination. Mirahadi and McCabe [13] introduce EvacuSafe, which couples a modified Dijkstra cost based on a path risk index with building information modeling and an active dynamic signage system. The network representation of the building supports continuous updates of guidance as fire conditions change, including structural obstructions and door failures. Experiments on a two-story office show that prioritizing path safety, rather than travel time alone, yields safer trajectories and better outcomes for occupants. Maritime evacuation is considered in Liu et al. [14], which presents a time-varying equivalent-weight Dijkstra algorithm integrated with fire dynamics simulator data. The method incorporates speed reductions due to crowding, carbon monoxide exposure, limited visibility, and temperature, and adapts route weights as the incident progresses. Simulation results emphasize the need to integrate environmental and crowd risk in path selection. Jiang et al. [15] combine a modified Dijkstra procedure with a minimum cost maximum flow formulation to identify safe and efficient routes under time-dependent hazards.

Other variations of path planning algorithms have been successfully proposed in the literature for the emergency evacuation optimization problem. For maritime settings, the authors of Deng et al. [16] analyze evacuation on cruise ships with a dynamic method based on an improved A star search that is informed by fire dynamics simulator outputs. A ship-specific network is mapped in advance and then updated during an incident to avoid nodes affected by heat, smoke, or loss of visibility. Large-scale simulations are successfully performed. The findings support the role of combining predictive fire data with responsive routing to increase preparedness and resilience on

board. He et al. [17] proposes an evacuation path planning method for multi-hazard accidents in chemical industries based on a risk perception analysis as the distance criterion to improve the D* path planning algorithm. In chemical industrial parks, the authors of Chen et al. [18] present a bidirectional planning framework that coordinates rescue vehicle routing and pedestrian evacuation on a dynamic grid. An intelligent obstacle avoidance component suppresses road conflicts and accounts for time-varying capacities, while the planning layer optimizes both flows jointly rather than separately. Case studies for fire and toxic release illustrate improved clearance and reduced interference between rescue and evacuation streams.

Evacuation plans are often proposed in the literature in combination with simulation experiments that guide and analyze the identified routes. To mitigate congestion at exits, the authors of Kurdi et al. [19] propose the Balanced Emergency Evacuation algorithm that allocates evacuees across multiple exits using exit capacity, distance, and real-time conditions. Simulations benchmarked against Simulated Annealing and Depth-First Search report shorter total evacuation time and lower maximum crowding, especially in large facilities with few exits. The study suggests extensions with real-time sensing and automated monitoring to support implementation. Huang et al. [20] develop an automatic evacuation system that uses sensing and learning to recommend safe routes in real time. The system combines smoke and heat detection, crowd monitoring, and a data processing layer that infers route safety under changing conditions such as congestion, blocked doors, and smoke spread. Simulated experiments show faster and safer evacuations than standard rule-based methods, with gains attributable to continuous adaptation of guidance as conditions evolve. Tavares and Galea [21] combine evacuation simulation with design of experiments and response surface modeling to optimize exit configurations. Their method reduces simulation effort while enabling systematic evaluation of alternative layouts, illustrating how surrogate models can support efficient scenario exploration in evacuation planning. In Wang et al. [22], a strategy is proposed for fire evacuation with rescue guiders based on a hybrid modeling approach. The model has a bi-objective function considering time and cost and considers the layout of the guider. Subsequently, the model is solved by iterative optimization of the NSGA-II algorithm and verified with an FDS fire simulator.

Further studies face the evacuation problem as a multi-objective optimization problem. Georgiadou et al. [23] develop a multi-objective evolutionary framework for emergency response to major industrial accidents. Their approach identifies Pareto optimal evacuation and sheltering strategies that jointly consider health, socioeconomic, and operational impacts, demonstrating the effectiveness of optimization compared with traditional zone-based policies. Yuan and Sun [24] apply an adaptive ant colony algorithm into evacuation management considering the congestion degree. The study is applied to the case of a railway station, considering as objective functions the minimization of both the total evacuation time of all evacuees and the sum of the crowding degree of each channel in the evacuation process.

Starting in the 2020s, a relevant role is played by digital twins combined with evacuation plans in emergency situations. Among others, it is worth mentioning the review by [25] aimed at summarizing the potential of digital twin technology in this context. Han et al. [26] highlight the strategic role that digital twins have in the choice of safe routes in case of hurricane, allowing an accurate representation of real-world conditions. In [27], a fire emergency management system based on an informative digital twin to provide instantaneous fire situational information is presented, while [28] proposes an intelligent emergency digital twin system based on computer vision and deep learning. An emergency evacuation due to fire is also considered in [29], where a real-time digital twin controller acting with a Grid-Interactive Emergency Energy Management system for managing evac-

uation routes in case of building fires is presented. The authors show that the proposed integration guarantees better safety results and environmental advantages.

Some approaches have been specifically tailored to the type of emergency evacuation considered. For hazardous material incidents, protective actions include shelter in place and evacuation [30]. A stream of work models toxic dispersion, defines evacuation zones, and quantifies risk and social impacts [18]. In a related application to a nuclear power setting, the authors of Bolia et al. [31] develop models that link strategic and operational decisions and incorporate uncertainty in evacuation demand to obtain robust schedules.

Huang et al. [32] present a safety evaluation framework based on a dynamic data-driven evaluation methodology specifically designed for integrated metro–commercial complexes. Still regarding the issue of emergency evacuation in metro stations, a hybrid approach that integrates building information modeling, a fire dynamics simulator, and a multi-objective optimization algorithm is proposed in [33]. Reniers et al. [34] apply real options theory to evaluate optimal evacuation timing for chemical plants exposed to external fire hazards. Their empirical study shows that shutdown costs, uncertainty, and operational constraints substantially influence precautionary evacuation decisions, highlighting the importance of dynamic risk-based decision models in industrial contexts. Evacuation in chemical plant is also considered in [35], where different scenarios of the evacuation process in a chemical industrial park are simulated to analyze the impact of meteorological conditions on the dispersion of toxic gases. The study provides useful guidelines for efficient evacuation processes.

Other studies model the optimization evacuation problem by deriving a network flow model, as in the present case. Among others, Yuan et al. [36] present an integrated evacuation modeling approach that combines fine and coarse grid networks to simulate movement in large complex buildings with improved scalability. The hybrid model preserves detail where hazards are concentrated and efficiently identifies bottlenecks across extensive layouts. Their approach demonstrates the usefulness of multi-resolution network structures for evacuation analysis in large-scale facilities. Liu et al. [37] address buildings with several exits under smoke, fire, and capacity constraints by combining a heuristic search with network flow control to minimize total evacuation time.

The reviewed literature shows substantial progress in evacuation planning, including optimization-based routing, simulation-driven analysis, and data-rich digital twin approaches. Nevertheless, many existing contributions are developed for specific phases of the evacuation process or rely on heterogeneous modeling components, which limits their ability to coherently support planning, execution, and adaptation within a single framework. As a result, the integration of evacuation planning models with dynamically evolving system states remains a partially unresolved issue, particularly in large and complex industrial facilities.

In this context, the present article proposes a network flow-based evacuation model aimed at ensuring that the largest possible number of people reach safety in the shortest possible time. A key novelty of EROA lies in its ability to operate both at a strategic level, supporting offline evacuation planning, and at an operational level, where the same model is integrated with field sensors to enable real-time reoptimization of evacuation routes based on the actual positions of evacuees and the evolution of the emergency scenario. The following sections detail the structure of the proposed model, the optimization formulation, and the results obtained in a case study representative of real operational conditions.

3. The Model: Layout Graph and Timed Flow Network

The generation of the model used by EROA is based on the layout information of the facility, including the areas where people are located and those available for evacua-

tion paths. This layout is transformed into a graph, called the layout graph, specifically developed to represent the static structure of the facility.

Consider a layout graph $G = (V, A)$. V is the set of nodes corresponding to locations where people (workers, guests) may be within the facility. These nodes may be associated with spaces (offices, laboratories, workshops, service rooms, corridors, stairs). Depending on the size of these spaces and their topology (e.g., multiple doors), it may be necessary to consider multiple nodes to accurately track people's positions within the areas. A is the set of directed elementary paths connecting node pairs. In practice, if an arc connects the pair of nodes (u, v), people at node u can move directly to node v within a certain time without crossing other areas.

The static graph G representing the facility has each arc (u, v) associated with a transfer time t_{uv} and capacity c_{uv}. The transfer time may be an average time or depend on the person (or class of people) considered (e.g., evacuation of a department store with people of different ages and mobility). The capacity defines the number of people that can begin the transfer on the arc per time unit, denoted as Δt. For example, a capacity of 2 for an arc (u, v) indicates that, for every Δt, 2 people can start the transfer from u to v. If there are 10 people at u and the transfer time t_{uv} is 4, the total time for everyone at u to reach v is 20.

The layout graph G needs to be transformed into a dynamic timed graph, called the timed flow network $G_T = (V_T, A_T)$, to model the evacuation process's evolution. For this, a time horizon T is defined, composed of H discrete time steps Δt, which will simply be referred to as time units. Given $T = \{1, \ldots, H\}$, each node $v \in V$ in G is transformed into a set of timed nodes corresponding to pairs (i, t), where i is a location and t is a time step. Let $V^T = \{(v, t) : v \in V, t \in T\}$ denote the set of timed nodes.

Similarly, the arcs A in G are transformed into timed arcs, defining the set A^T. There are two subsets of timed arcs:

- Waiting arcs, $A^{TW} = \{((v, t), (v, t+1)) : (v, t), (v, t+1) \in V^T\}$, model the flow of people remaining in a location (e.g., a room) and correspond to arcs between two successive timed nodes for the same physical location;
- Movement arcs, $A^{TM} = \{((i, t), (j, t + T_{i,j})) : (i, t), (j, t + T_{i,j}) \in V^T\}$, model the transfer of people between different physical locations (e.g., from an office to a corridor or between two locations in a corridor if the corridor is sufficiently long).

For example, consider a simple layout as shown in Figure 1.

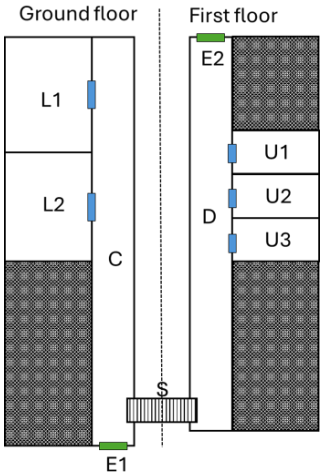

Figure 1. A simple layout example.

In Figure 1, two floors are connected by a staircase S. On the ground floor, two laboratories ($L1$ and $L2$) open into a corridor C, which leads to an emergency exit

$E1$. On the second floor, there are three offices ($U1$, $U2$, and $U3$), connected by a corridor D that has an emergency exit $E2$ on one side and the stairs S on the other. A graph representation of these environments is shown in Figure 2.

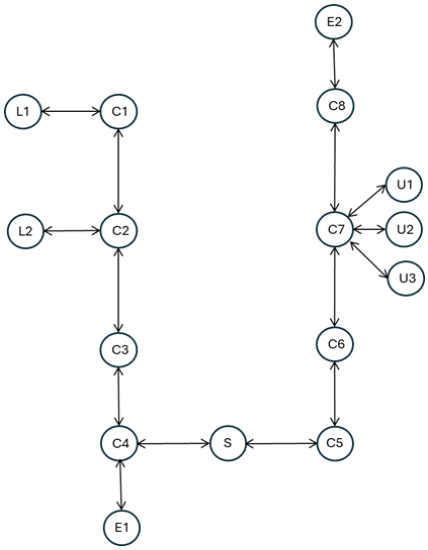

Figure 2. Graph representation of the layout from Figure 1.

For simplicity, each arc is represented as bidirectional, but in reality, it corresponds to a pair of directed arcs in both directions. This way, all possible movements between different areas are modeled. Each arc is associated with a transfer time and a capacity, which can differ in opposite directions to capture movement characteristics that depend on direction. For instance, arcs $(C5, S)$ and $(C5, S)$ represent movements on the same physical staircase but in opposite directions. The corresponding transfer times may therefore differ, as descending stairs generally allow faster movement than ascending, a distinction commonly adopted in evacuation and pedestrian flow modeling [38].

3.1. The Timed Flow Network

The timed network is constructed based on the time horizon T. Depending on the number of time units H considered, this network can become extensive and challenging to represent graphically. Figure 3 illustrates a small portion of the network for the example of Figures 1 and 2.

Figure 3a shows the portion of the graph representing the layout that includes the two laboratories located on the ground floor and two nodes corresponding to corridor C, with the associated traversal times for the arcs. Considering the defined time horizon, the corresponding time-expanded network is illustrated in Figure 3b. As can be observed, in this case, the time-stamped arcs are directed and model the movements between pairs of locations shown in Figure 3a, while accounting for the transfer times.

The time-expanded network also includes information on the initial position of people at the beginning of the time horizon ($t = 0$). This information is associated with the time-stamped nodes $(v, 0)$ and is represented by the parameter $b_{(v,0)}$, which can take a non-negative value. In general, each time-stamped node (v, t) is associated with a parameter $b_{(v,t)}$, which equals zero for $t > 0$.

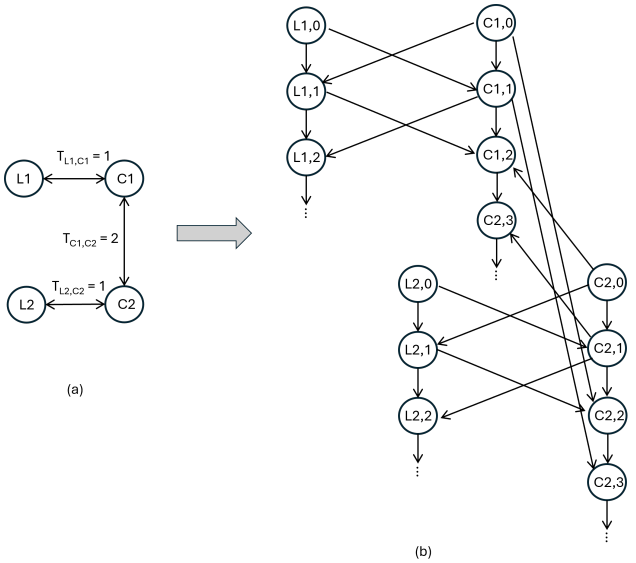

Figure 3. A portion of the timed network (**b**) for a portion of the layout graph (**a**) in Figure 2.

To correctly model the time-expanded flow network, it is necessary to introduce an additional node, denoted by F. This node represents a sink whose purpose is to ensure that all individuals present in the facility can reach either a safe exit or a designated safe area. Specifically, each exit or safe area is connected to node F through an arc whose traversal time is irrelevant and whose capacity is considered infinite. Since F is the only sink in the network, its associated parameter b_F is set to balance the network, that is,

$$b_F = -\sum_{(v,t)\in V^T} b_{(v,t)}$$

However, it is also necessary to consider the possibility that not all individuals can reach an exit within the assumed time horizon. For this reason, additional arcs must be introduced between nodes initially occupied by people and the final sink node F.

Figure 4 provides a concise representation of the complete network after the addition of node F. In particular, the figure shows, for the layout graph of the example in Figure 4, the nodes associated with the laboratories and offices at time $t = 0$, i.e., at the beginning of the planning horizon. These nodes represent the sources of the flow of people within the facility. The figure also displays the time-stamped nodes corresponding to the two emergency exits (specifically, t_{E1} and t_{E2} represent the minimum time within which a person can reach exits E1 and E2, respectively).

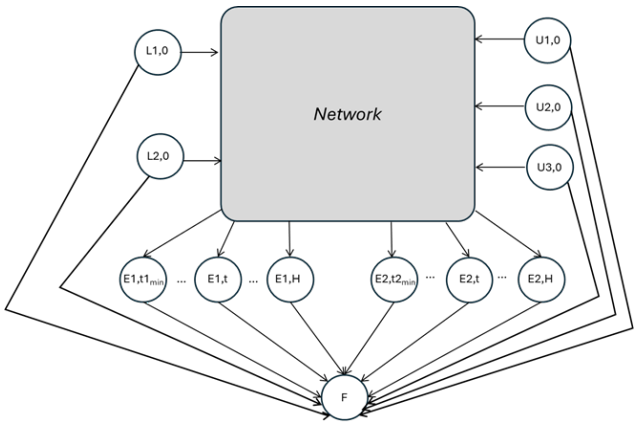

Figure 4. Synthetic representation of the time-expanded network for the layout graph shown in Figure 2.

Furthermore, the figure shows the arcs connecting the time-stamped nodes of the exits to the final node F, as well as the arcs modeling the potential flow of individuals who are unable to reach an exit within the assumed time horizon, namely, those connecting the nodes where people are initially located to the final sink node F.

3.2. Mathematical Model for the Timed Flow Network

The mathematical programming formulation introduced in this section is a linear programming (LP) network flow model, as all decision variables, which are binary in nature, indicating whether or not the corresponding arc has been crossed, are continuous, and the objective function and constraints are linear. As such, the problem can be solved efficiently using standard LP solvers, even for large time-expanded networks. Before defining the formulation, we formally introduce the sets, parameters, and variables used.

Sets

- V: Set of physical locations in the layout (nodes).
- $S \subset V$: Subset of safety exits or safe areas.
- E: Set of direct connections between physical locations (arcs).
- T: Time horizon.
- $V^T = \{(v,t) : v \in V, t \in T\} \cup \{F\}$: Set of timed nodes extended with the final fictitious node F.
- $A^{TW} = \{((v,t),(v,t+1)) : (v,t),(v,t+1) \in V^T\}$: Set of timed waiting arcs.
- $A^{TM} = \{((i,t),(j,t+T_{i,j})) : (i,t),(j,t+T_{i,j}) \in V^T\}$: Set of timed movement arcs.
- $A^T = A^{TW} \cup A^{TM} \cup \{((e,t),F) : e \in E,(e,t) \in V^T\}$: Set of all timed arcs in the network.

Parameters

- $b_{(v,t)}, (v,t) \in V^T$: Flow parameter for timed nodes (number of people present in physical locations at time t).
- $b_F = -\sum_{(v,0=\in V^T} b_{(v,0)}$: Flow parameter for the fictitious final node.
- $T_{i,j}, (i,j) \in A$: Transfer time between two physical locations connected by arc (i,j).
- $C_a, a \in A^{TM}$: Capacity of movement arcs.
- M: big-M, sufficiently large constant.

Variables

- $x_a, a \in A^T$: Flow variables.
- $u_{(i,t)} \geq 0, (i,t) \in V^T : b_{(i,t)} > 0$: Flow variables for the arcs between nodes where people are initially present and the final node F.

Formulation

$$min \quad Z = \sum_{((e,t),F) \in A^T} t \cdot x_{((e,t),F)} + M \cdot \sum_{(i,t) \in V^t : b_{(i,t)} > 0} u_{(i,t)} \tag{1}$$

Subject to constraints:

$$\begin{aligned} &\sum_{((i,t),(j,t+T_{i,j})) \in A^T} x_{((i,t),(j,t+T_{i,j}))} - \\ &\sum_{((j,t-T_{j,i}),(i,t)) \in A^T} x_{((j,t-T_{j,i}),(i,t))} + u_{(i,t)} = b_{(i,t)} \quad (i,t) \in V^T, b_{(i,t)} > 0 \end{aligned} \tag{2}$$

$$\sum_{((i,t),(j,t+T_{i,j}))\in A^T} x_{((i,t),(j,t+T_{i,j}))} - \sum_{((j,t-T_{j,i}),(i,t))\in A^T} x_{((j,t-T_{j,i}),(i,t))} = 0 \quad (i,t) \in V^T, b_{(i,t)} > 0 \tag{3}$$

$$- \sum_{((e,t),F)\in A^T} x_{((e,t),F)} - \sum_{(i,t)\in V^T : b_{(i,t)}>0} u_{(i,t)} = b_F \tag{4}$$

$$x_a \leq C_a \quad a \in A^{TW} \cup A^{TM} \tag{5}$$

$$\begin{aligned} &x_a \geq 0 \quad a \in A^T \\ &u_{(i,t)} \geq 0 \quad (i,t) \in V^T : b_{(i,t)} > 0 \end{aligned} \tag{6}$$

The objective function (1) expresses the minimization of the weighted evacuation time of people reaching a safe exit or area. The product $t \cdot x_{((e,t),F)}$ weights the evacuation time t of a group of people by the number of individuals composing that group. The second term in (1) strongly penalizes the sum of the flows in arcs $u_{(i,t)}$, which corresponds to the total number of people who are unable to reach a safe exit or area within the time horizon T. Consequently, the solver aims to ensure that the largest possible number of people reach safety in the shortest possible time.
Constraints (2) and (3) express the flow conservation conditions for all network nodes of type (i,t), that is, for all nodes except the fictitious final node F.
Constraint (4) instead enforces the flow conservation for node F.
Constraints (5) impose the maximum flow capacity for both waiting and movement arcs.
Finally, constraints (6) define the model variables and their non-negativity conditions.

Although the proposed model is based on a time-expanded network representation, its novelty lies in its use as a real-time, data-driven re-planning framework. The formulation is explicitly designed to be repeatedly updated using sensor information on evacuees' positions and network accessibility, enabling fast reoptimization under evolving emergency conditions. This distinguishes the approach from static time-expanded evacuation models typically used for one-shot planning.

3.3. An Alternative Formulation

Instead of minimizing the weighted average time for people reaching safety, one might aim to minimize the time at which the last person reaches a safe location (evacuation makespan). This requires introducing two new variables, one being a binary variable as follows:

- c_{max}: The last time step when the final person reaches a safe location.
- $y_{e,t} \in \{0,1\}, (e,t) \in A^T$: A binary variable indicating whether there is positive flow in arc $((e,t),F)$.

The objective function changes accordingly:

$$min \quad z = c_{max} + B \cdot \sum_{(i,t)\in V^T : b_{(i,t)}>0} u_{(i,t)} \tag{7}$$

It has the following additional constraints:

$$c_{max} \geq t \cdot y_{(e,t)} \quad (e,t) \in V^T, e \in E \tag{8}$$

$$x_{((e,t),F)} \leq -b_F \cdot y_{(e,t)} \quad ((e,t),F) \in A^T \tag{9}$$

Constraints (8) enforce that the evacuation makespan c_{max} cannot be smaller than the time t if there exists a positive flow on arc $((e,t),F)$, that is, if at least one person reaches the safe location e at time t.

Constraints (9) link the variables $y_{(e,t)}$ to the flow variables outgoing from nodes i and entering the fictitious final node F.

While this alternative formulation provides a more direct representation of the evacuation makespan objective, the introduction of binary variables $y_{(e,t)}$ transforms the model from a continuous LP into a Mixed Integer Linear Program (MILP), significantly increasing its computational complexity. As a consequence, for large time-expanded networks, this formulation may entail a computational burden such that the model becomes unsuitable for real-time applications.

4. Real-Time Model Application

The previously presented model can be directly utilized to identify the paths that different groups of people should follow to quickly reach safety exits or designated safe areas. The parameters b associated with the temporally indexed nodes at $t = 0$ indeed represent the number of people present in the facility at the onset of an emergency. However, the model is flexible and can also determine evacuation paths as the emergency unfolds. Specifically, whenever the system responsible for gathering data from sensors within the facility detects at a given time $\hat{t}$ modified conditions with respect to the current plan (e.g., if certain spaces or routes become hazardous, or if evacuees' positions differ from those expected), this information can be transmitted to the real-time optimization module, which will then be capable of instantiating an updated model.

The real-time information may concern the following situations:

- The current location of individuals within the facility at time $\hat{t}$;
- One or more locations are no longer accessible;
- One or more locations will become inaccessible after a certain time instant;
- One or more connections between locations are no longer passable;
- One or more connections between locations will become impassable after a certain moment in time.

In these cases, the model is redefined as follows:

1. The flows determined by the prior resolution of the model, before the event altering the scenario, are considered. Subsequently, the temporal network and the individuals' locations within it are updated as of time $\hat{t}$. Let $\hat{V}_t^T$ and $\hat{A}_t^T$ represent, respectively, the nodes and edges of the temporal network remaining after updating at time $\hat{t}$, with the following steps undertaken:
   - Any node $(i,t) \in V^T$ with $t < \hat{t}$ is removed from $\hat{V}_t^T$ as these nodes represent past moments;
   - Any edge $((i,t),(j,t+T_{i,j}))$ in A^T with $t + T_{i,j} < \hat{t}$ is removed from $\hat{A}_t^T$ since it also connects past nodes;
   - For each node $h \in V$ associated with an area that has become, or will become, inaccessible at time $\hat{t}$ or later at time t^*, the nodes $(h,t) \in V^T$ with $t > \hat{t}$ or $t \geq t^*$ are removed, along with all outbound edges $((h,t),(j,t+T_{h,j})) \in A^T$;
   - For each edge $(h,k) \in A$ corresponding to a connection between areas that has become, or will become, impassable at $\hat{t}$ or at a later time t^*, the edges $((h,t),(k,t+T_{h,k})) \in A^T$ with $t > \hat{t}$ or $t \geq t^*$ are removed.

2. For updating the current position of individuals at time $\hat{t}$, if received via field sensors, the values of parameters $b_{(i,\hat{t})}$ for each node $(i,\hat{t}) \in \hat{V}_t^T$ should be adjusted based on the received information.
   In the absence of such field data, as described below, the parameters of the temporally indexed nodes can be updated by assuming that until time $\hat{t}$, individuals have followed the flows determined by the prior optimization model. The parameters $b_{(i,\hat{t})}$ for each node $(i,\hat{t}) \in \hat{V}_t^T$ are then updated as follows:
   - For each $(j,t) \in \hat{V}_t^T$

$$b_{(j,\hat{t})} = \sum_{((i,\hat{t}-T_{i,j}),(j,\hat{t}))\in A_t^T} x_{((i,\hat{t}-T_{i,j}),(j,\hat{t}))} \tag{10}$$

     where $x_{((i,\hat{t}-T_{i,j}),(j,\hat{t}))}$ represents the flow along edges from previous nodes;
   - For edges $((i,t_1),(j,t_2)) \in \hat{A}_t^T$ such that $t_1 < \hat{t}$ and $t_2 > \hat{t}$, the node parameter for the destination node (j,t_2) is updated as

$$b_{(j,t_2)} = \sum_{((i,t_1),(j,t_2))\in \hat{A}_t^T : t_1<\hat{t}\wedge t_2>\hat{t}} x_{((i,t_1),(j,t_2))} \tag{11}$$

   - If there is a flow of people starting from node i at time $t_1 < \hat{t}$ toward a node j that becomes inaccessible after $\hat{t}$ and that would be reached at time $t_2 > \hat{t}$, then it is assumed that such people must return to the departure node i, which will be reached again at time $2\hat{t} - t_1$, provided that i remains accessible after $\hat{t}$. In this case, the node parameter will be updated as

$$b_{(i,2\hat{t}-t_1)} = x_{((i,t_1),(j,t_2))} \tag{12}$$

   - The same update must be performed if, in the evacuation plan determined by the previous solution of the model, there exists a non-zero flow on an arc $((i,t_1),(j,t_2)) \notin \hat{A}_t^T$ that is no longer accessible, with $t_1 < \hat{t}$ and $t_2 > \hat{t}$;
   - Finally, in the previous situation, if the node i also becomes inaccessible, the model signals the impossibility of bringing to safety the people located on the arc that has become inaccessible.

Note that the procedure described in this section does not aim to simulate individual evacuation behavior. Instead, it defines a real-time model update mechanism driven by sensor data, providing the actual positions of evacuees and the current availability of locations and connections. Deviations from previously suggested routes, e.g., due to congestion, hesitation, counterflows, or non-compliant behavior, are therefore implicitly captured through the updated system state and reflected in subsequent reoptimizations, allowing the evacuation plan to continuously adapt to the observed evolution of the emergency.

5. Experimental Results

In this section, we first introduce and discuss a test case represented by a small industrial building, and subsequently, we present an experimental campaign considering random layouts of large and very large dimensions. All computational experiments were carried out using IBM ILOG CPLEX (version 22.1.1.0), with the optimization models implemented in Python 3.10. The tests were executed on a standard laptop equipped with a 13th Gen Intel® Core™ i7–1355U processor running at 1.70 GHz, featuring 10 physical cores and 12 logical processors.

5.1. An Industrial Test Case

The industrial test case used to validate the evacuation optimization algorithm concerns an industrial building designed for pharmaceutical, cosmetic, or light chemical production. The facility consists of two levels, as shown in Figure 5: the ground floor dedicated to production activities and the first floor reserved for offices and administrative functions.

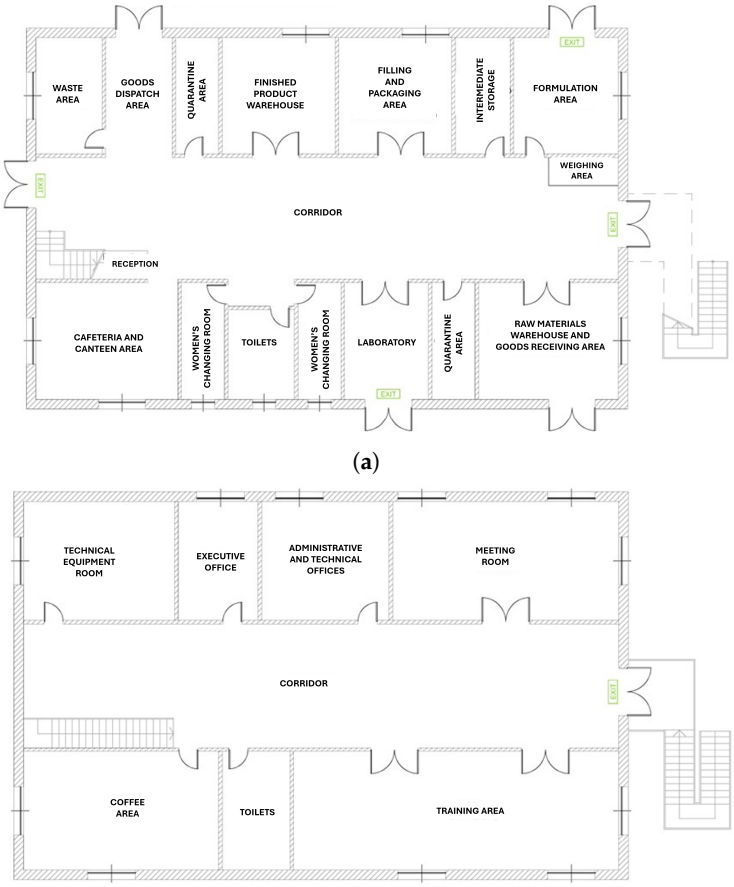

Figure 5. Layout of the industrial plant. (**a**) Ground floor; (**b**) first floor.

The layout of each floor was represented as a directed graph $G = (V, A)$. These graphs describe the static structure of the facility and serve as the input for the evacuation optimizer. Arcs in A represent directed connections between nodes, while nodes in V correspond to functional areas of the plant, identified as follows:

- C are the corridors;
- B are the bathrooms;
- E are the emergency exits;
- K are the bar/canteen areas;
- L are the laboratories;
- R are the rooms;
- S are the staircases;
- U are the offices;
- W are the warehouses.

The following Figures 6 and 7 represents respectively the layout graphs of the ground floor and first floor of the facility where rooms are depicted in gray, corridors in white, stairs in orange and emergency exits in green. The graph for the ground floor in Figure 6 includes corridors C_1–C_9, rooms R_1–R_{10}, the bar/canteen K_1, bathroom B_1, lab-

oratory L_1, and warehouses W_1, W_2, W_3. The actual spaces are associated with the node identifiers as follows: secretariat (R_1), waste area (R_2), shipping area (R_3), quarantine area 2 (R_4), bottle filling and packaging (R_5), formulation area (product mixing) (R_6), weighing/dosing area (R_7), female changing room (R_8), male changing room (R_9), and quarantine area 1 (R_{10}). The warehouse nodes respectively represent the finished product warehouse (W_1), intermediate storage (W_2), and raw materials warehouse (W_3) with incoming goods. The ground and first floors are connected by the internal staircase S_1. Four emergency exits are available: E_1 (main entrance), E_2 (near the laboratory), E_3 (near the formulation area), and E_4 (at the end of corridor C_7).

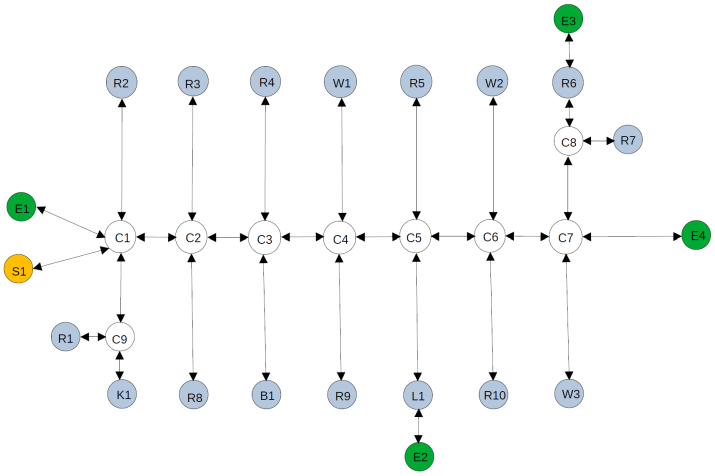

Figure 6. Layout graph representation of the ground floor .

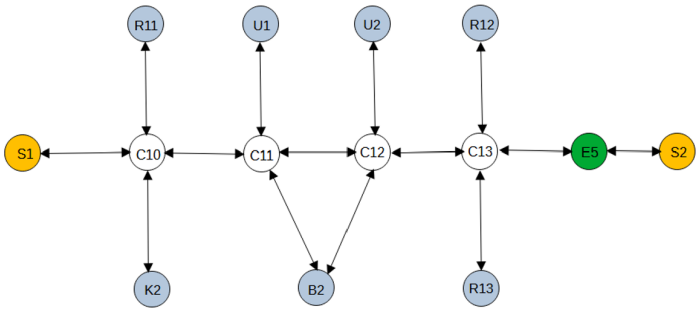

Figure 7. Graph representation of the first floor.

The first floor in Figure 7 extends along corridors C_{10}–C_{13} and includes the technical room (R_{11}), the management and administrative offices (U_1, U_2), the meeting room (R_{12}), the training room (R_{13}), bathroom (B_2), and a coffee area (K_2). A fire escape (E_5) is located at the end of the main corridor and leads to an external staircase (S_2).

For each graph node, the number of people that are initially present is given, denoted as $b_{v,0}, v \in V$. For each arc $(i, j) \in A$, the values of the following parameters are given:

- T_{ij}, the traversal time between nodes i and j;
- C_{ij}, the maximum capacity of the connection (people simultaneously traversing it).

We performed several tests for this case study, first considering the formulation in Section 3.2. We generated the evacuation paths assuming that 50 people were inside the facility at time $t = 0$. In particular, all individuals were initially placed inside rooms rather than in corridors, staircases, or exits. We executed three preliminary tests by progressively increasing the simulation horizon H in order to analyze both the model size and its computational performance.

Table 1 summarizes the initial population, the chosen time horizon, and the resulting model dimensions in terms of variables and constraints. As expected, an increase in the time horizon leads to an increase in the number of variables and constraints. The table also reports the two components of the objective function (1): the weighted average evacuation time (WAET), which corresponds to the first term of (1) normalized by the number of people to evacuate, and the number of unsaved people, i.e., the ones who are unable to reach an emergency exit within the given horizon. The *MKS* column reports the evacuation makespan, while *CPU* denotes the computation time required by the MILP solver.

The first test shows that a horizon of five time steps is insufficient, leaving 14 people unable to reach an exit. When the horizon is increased to 15 time steps, the second test indicates that all individuals reach safety, with a makespan of 7 time steps. Further extending the horizon to 25 time steps does not modify the evacuation outcome, although both the model size and the computational effort increase. The last row of Table 1 reports the effect of doubling the initial population to 100 individuals. In this case, the model dimension remains identical to the 50-person scenario, and the solution time is comparable. However, both the makespan and the WAET increase considerably. Overall, for all experiments, the computation times remain below one second, demonstrating the effectiveness of the formulation.

Table 1. Offline tests for the industrial plant with formulation in Section 3.2.

Population	Horizon	Variables	Constraints	WAET	Unsaved	MKS	CPU
50	5	529	799	2.38	14	5	0.09
50	15	1549	2299	4.16	0	7	0.27
50	25	2569	3799	4.16	0	7	0.63
100	15	1549	2299	6.16	0	12	0.25

To estimate pedestrian flows and the number of evacuees per unit of time, we relied on the results reported in Vanumu et al. [38], which provide a comprehensive review of fundamental diagrams for pedestrian flow characteristics. In order to derive the main instance parameters, particularly flow capacities and travel times, values were generated assuming an average pedestrian walking speed of approximately 1 m/s. A deliberately conservative approach was adopted to avoid any potential overestimation of pedestrian flows. Under this setting, the real-world instance resulted in a discretization where each time step corresponds to approximately 12 s, implying that the minimum meaningful time horizon is on the order of one minute. While realistic instances are essential to assess the usability and practical relevance of the proposed model, obtaining accurate solutions tailored to a specific facility or location would ultimately require experimental measurements or detailed simulation studies. Such analyses would allow the generation of more realistic and reliable input data, leading to a more faithful representation of actual evacuation dynamics.

Then, we verified the influence of changing the maximum capacity of the connections, considering the case with 50 people and five time steps as the horizon. In particular, we generated different scenarios by multiplying the original capacity by a factor of $cm \in \{0.5, 0.8, 1.0, 1.2, 1.5, 1.7, 2.0, 2.5, 3.0\}$. Figure 8 shows the variations in the WAET and the number of unsaved people for the different values of cm.

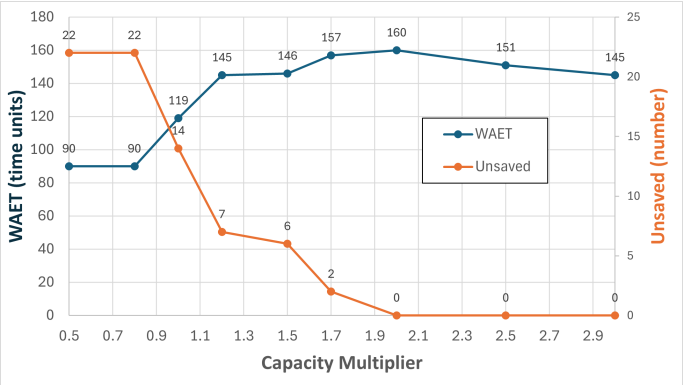

Figure 8. Analysis of maximum capacity changes.

Due to the short time horizon, the WAET increases for cm $\leq$ 2.0, since a significant portion of evacuees are forced to reach the exits close to the end of the time horizon. When cm = 2.0, the increased capacities allow all evacuees to reach a safe exit within the time horizon. For larger values of cm, the WAET decreases because a higher number of evacuees are able to reach safety earlier. In the instance, all evacuees are initially located inside their own offices. This represents a particularly challenging scenario, as no individual is assumed to be casually positioned near an exit at the beginning of the evacuation process. To analyze the impact of this initial spatial distribution, a set of 25 scenarios was generated starting from a baseline instance with 100 evacuees and a time horizon of 15 time steps. In each scenario, a subset of evacuees was relocated from offices to corridor areas. For each scenario, five independent instances were generated, with the proportion of relocated evacuees ranging from 10% to 90% of the total population. The results, summarized in Figure 9, clearly show that allowing a fraction of evacuees to be initially located outside their offices leads to a reduction in the average evacuation time of approximately 14% on average. Moreover, this reduction is not linearly related to the number of relocated evacuees. The most significant improvements are observed when 50% of the population is located outside their offices. Beyond this threshold, the benefits tend to diminish, likely due to increased congestion effects caused by a higher number of evacuees simultaneously present in shared spaces such as corridors.

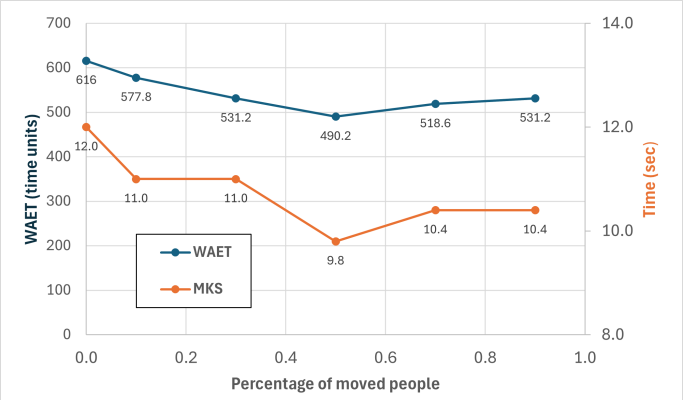

Figure 9. Analysis of scenarios changing the initial location of people in the facility.

Figure 10 reports the cumulative percentage of evacuees that reach a safe exit over time, obtained by averaging the results over 30 scenarios with different initial spatial distributions of 100 evacuees within the facility, considering a fixed time horizon of 15 time steps. For each scenario, only flows reaching the safe exit within the horizon are considered, and the cumulative curve is computed with respect to the total number of evacuated individuals. The resulting curve exhibits a smooth and nearly linear growth, with the final fraction of evacuees reaching

safety shortly after the majority, thus showing no evidence of a pronounced tail. This suggests that minimizing the WAET does not systematically penalize the last evacuees.

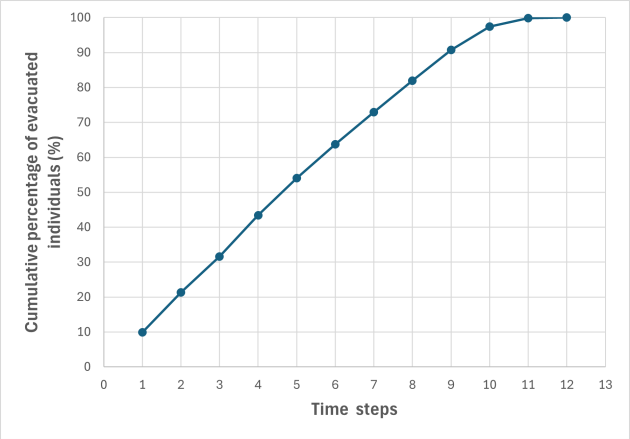

Figure 10. Average cumulative percentage of evacuees reaching an exit over time.

Tables 2 and 3 present the tests performed by first considering three offline scenarios and then, for each of them, three possible real-time disruptions. Scenario 1 in Table 2 corresponds to an initial distribution of 50 people inside the facility, which is modified in Scenarios 2 and 3 by randomly relocating 20% of the population. Table 3 reports the results obtained by considering the following disruptions for the three scenarios:

- A portion of the first floor corridor providing access to exit E5 becomes unavailable (node C13 is removed) starting from time step 3;
- The same situation occurs starting from time step 5;
- The staircase S1 cannot be accessed starting from time step 1.

Table 2. Offline planning for the three real-time scenarios.

Scenario	Initial Population	Percentage of Moved People	Horizon	WAET	Unsaved	Makespan	CPU Time
1	50	0.0	15	4.4	0	8	0.16
2	50	0.2	15	4.1	1	7	0.14
3	50	0.2	15	3.8	1	7	0.17

Table 3. Real-time re-planning for the three real-time scenarios.

Scenario	Disruption Instant	Unavailable Nodes	People in at Disruption	WAET	Unsaved	Makespan	CPU Time
1	3	C13	33	7.7	0	13	0.30
2	3	C13	31	7.7	2	13	0.22
3	3	C13	28	7.8	1	13	0.20
1	5	C13	16	10.4	0	13	0.16
2	5	C13	14	11.1	1	13	0.17
3	5	C13	11	12.0	1	13	0.17
1	1	S1	44	5.2	0	9	0.24
2	1	S1	44	4.8	1	9	0.24
3	1	S1	42	4.6	1	8	0.24

The model is updated to account for both the current positions of the evacuees still inside the facility at disruption time (column People in at disruption in Table 3) and the

unavailable nodes of the layout, as described in Section 4. The updated optimization problem is then solved again to generate new evacuation paths. As shown in Table 3, the disruptions affect the evacuation dynamics: in particular, both the makespan and the WAET always increase, and the number of people who are not able to reach a safe exit never improves. Despite the structural disruption, the computational times remain very small, confirming the suitability of the proposed approach with real-time applications.

The last test, performed on the facility with a population of 100 evacuees, aims to evaluate the alternative formulation introduced in Section 3.3. Table 4 reports the results obtained for the initial distribution of the evacuees and for five possible relocations involving different percentages of the population, as indicated in the first column. The values reported in the other columns represent the percentage deviations of the results obtained using the alternative formulation that minimizes the makespan (*MKS*) with respect to those obtained by minimizing the WAET, computed as $100 \cdot (ValCol^{MKS} - ValCol^{WAET}) / ValCol^{WAET}$.

As expected, minimizing the makespan leads to larger WAET values, since the arrival times at exits of the evacuees occurring before the makespan are not explicitly taken into account by the objective function. On the other hand, both the MKS and the number of unsaved evacuees remain unchanged when minimizing the makespan, whereas the computation time is significantly higher due to the combinatorial nature of the alternative formulation. Therefore, minimizing the WAET appears to be a more reliable option for evacuation path planning, even though this objective may produce solutions in which small groups of evacuees experience longer evacuation times.

Table 4. Comparison of results for the two proposed formulations.

Percentage of Moved People	WAET	Unsaved	Makespan	CPU Time
0.0	17.79%	0.0%	0.0%	141.8%
0.1	19.96%	0.0%	0.0%	1150.7%
0.3	21.07%	0.0%	0.0%	794.3%
0.5	22.52%	0.0%	0.0%	391.6%
0.7	25.42%	0.0%	0.0%	916.7%
0.9	24.97%	0.0%	0.0%	233.5%
Average	21.95%	0.0%	0.0%	604.8%

5.2. Random Instance Generator and Computational Experiments

To further assess the performance and scalability of the proposed algorithm beyond the realistic industrial test case, we developed a dedicated random instance generator. Its purpose is to create synthetic evacuation scenarios of increasing size, allowing us to evaluate the operational limits of the model and understand how different parameter settings affect both solution quality and computational effort.

The generator produces instances starting from a grid-based structure. For a given parameter n, representing the grid side length, the generator constructs a square grid composed of $n \times n$ transit nodes. Figure 11 illustrates a generated instance with $n = 3$. Transit nodes are shown as light blue circles, while the four red circles located at the corners represent the available emergency exits. Between each pair of adjacent transit nodes, the generator creates a rectangular node representing a room. Each room is associated with an integer indicating the number of occupants (shown in parentheses). Horizontal and vertical arcs connect adjacent nodes in the grid; each arc is characterized by a traversal time and a capacity, i.e., the maximum number of people allowed to enter the arc per time unit.

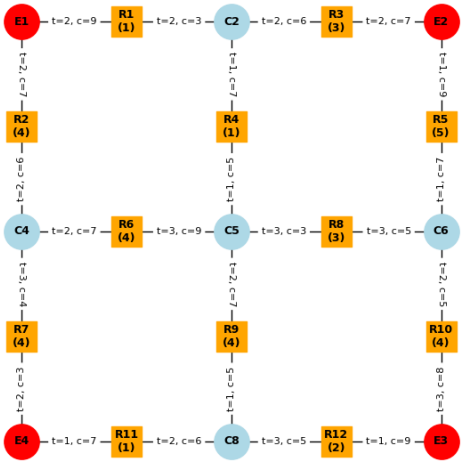

Figure 11. Example of a randomly generated grid layout with $n = 3$.

Using this generator, we created instances whose total number of nodes corresponds to

$$280,\ 408,\ 560,\ 736,\ 936,\ 1160,$$

obtained by varying the grid side from 10 to 20 with a step of 2. The number of people per room was set to one of the values $\{3, 6, 9\}$, while the time horizon was tested at $\{30, 60, 90\}$. For each combination of parameters, two independent instances were generated.

We first analyzed the computational time as the network size increases, fixing the number of people per room. The results are reported in Figure 12. Up to approximately 736 nodes, the computational times of the three scenarios (three, six, and nine people per room) remain almost identical. Beyond this point, however, the scenario with nine occupants per room becomes significantly more demanding, reaching an average running time of about 117 s for instances with 1160 nodes.

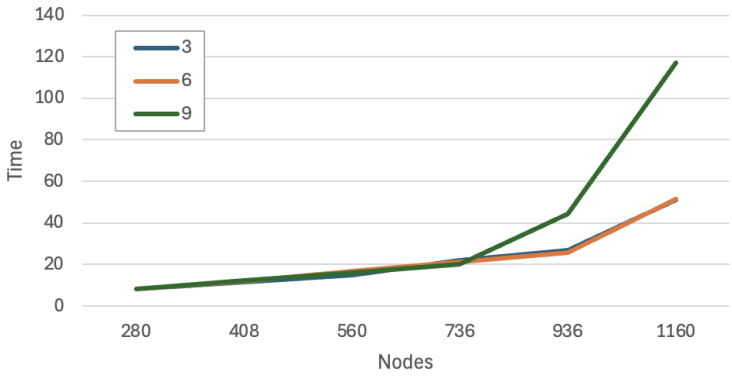

Figure 12. Average CPU time as a function of the number of nodes, for different values of the average number of people per room (3, 6, and 9).

A second experiment investigated computational times while fixing the time horizon to 30, 60, and 90 time units. The corresponding results are shown in Figure 13. In this case, the increase in computational effort is clearly proportional to the time horizon, as expected due to the enlarged time-expanded network. The number of variables grows almost linearly with the horizon, resulting in a correspondingly linear increase in solution times.

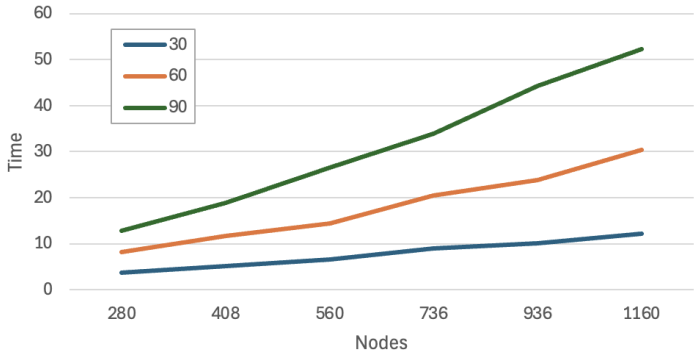

Figure 13. Average CPU time as a function of the number of nodes, for different values of the time horizon (30, 60, and 90 time units).

Overall, these large-scale experiments confirm that the proposed formulation remains computationally tractable for medium and large networks, especially for shorter time horizons and moderate occupancy levels. They also highlight how congestion, modeled through room populations and arc capacities, plays a crucial role in determining problem complexity, particularly when evacuation flows become more intense in denser scenarios.

Figure 14 shows the results obtained by progressively increasing the time horizon from 30 to 210 time units, with a step of 30, for large instances composed of 1160 nodes. Three scenarios were analyzed, corresponding to an increasing number of people to evacuate: 2280, 4560, and 6840 individuals. The radar plot displays, for each scenario, the percentage of people successfully evacuated over time. As expected, the total evacuation time grows with the population size. In particular, when the number of evacuees increases, the system requires more time intervals to achieve complete evacuation, confirming the scalability and consistency of the proposed model under larger-load conditions.

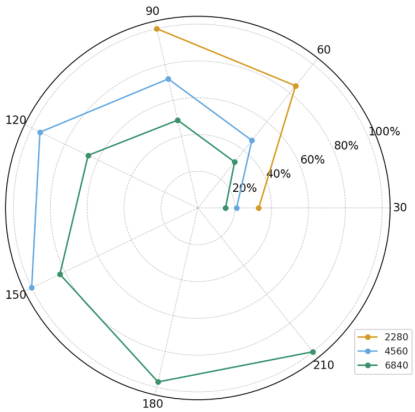

Figure 14. Percentage of people evacuated over time for increasing population sizes.

6. Discussion

The results presented in this study highlight the effectiveness of the proposed evacuation optimization framework when applied to complex industrial layouts. The formulation based on a timed flow network provides a flexible structure capable of representing both static topological constraints and the temporal evolution of an emergency scenario. The capacity to integrate real-time information further strengthens the practical applicability of the model, allowing the system to adapt to evolving conditions such as blocked routes, inaccessible areas, or updated crowd distributions.

The computational experiments demonstrate that the model scales to large instances and reacts predictably to increasing demand. In particular, radar plot analyses for scenarios

with 2280, 4560, and 6840 evacuees confirm a monotonic relationship between population size and evacuation completion time. This behavior is expected in flow-based models but nonetheless essential for validating the robustness of the formulation. The results also show that increasing the time horizon allows the solver to identify feasible and complete evacuation plans even in heavily populated facilities, offering insights into how to dimension operational time windows during emergency planning.

In this context, it is important to clarify the practical implications of the computational times observed in the large-scale synthetic experiments. While the most demanding synthetic instances lead to computation times of up to approximately 117 s, these scenarios were deliberately designed to stress the model and explore its computational limits. Their size is at least two orders of magnitude larger than the real industrial case study considered in this work. For instances up to approximately ten times the size of the real case, the proposed approach consistently terminates within a few seconds, making it suitable for operational use in realistic emergency scenarios. For extremely large-scale settings of the type considered in the stress tests, the adoption of dedicated heuristic strategies or more powerful computing resources represents a natural and practical direction.

From a managerial perspective, the framework provides decision makers with a tool that can support both preventive planning and real-time response. The ability to recompute evacuation strategies on the basis of updated field data makes it suitable for integration with sensor networks and automated safety systems. This capability can significantly improve emergency preparedness and reduce risks in high-density or high-hazard industrial environments.

Future research may explore several extensions of the model. First, incorporating uncertainty into transfer times, arc availability, or population behavior would allow the development of more robust evacuation plans. Second, hybrid approaches that combine simulation-based fire or smoke models with optimization could enrich the representation of risk along evacuation paths. Third, the integration of behavioral models may provide more accurate predictions in scenarios involving panic, limited mobility, or high congestion.

Author Contributions: Conceptualization, A.S.; methodology, M.P. and A.S.; software, C.C. and M.P.; validation, C.C. and M.P.; investigation, A.S.; data curation, C.C.; writing—original draft, C.C. and M.P.; writing—review and editing, A.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was carried out within the EMMA (emergency management for workers and machines) project, funded by the FESR 2021–2027 Liguria region program.

Data Availability Statement: The original data presented in the study are openly available at https://github.com/CarmineCerrone/OptimalEvacuationIndustrialFacilities (accessed on 14 December 2025).

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A. Flowchart Description of the Evacuation Optimization Algorithm

Figure A1 summarizes the workflow of the proposed evacuation optimization approach. The diagram represents the online (real-time) optimization algorithm. If the decision block Emergency-Induced Disruption? is ignored, the same diagram also describes the offline planning algorithm, since no reoptimization cycle is triggered.

The flowchart in Figure A1 provides a structured representation of the main phases of the proposed evacuation optimization algorithm. In the following, each block of the diagram is described in the order in which it is executed, highlighting its role within the overall workflow.

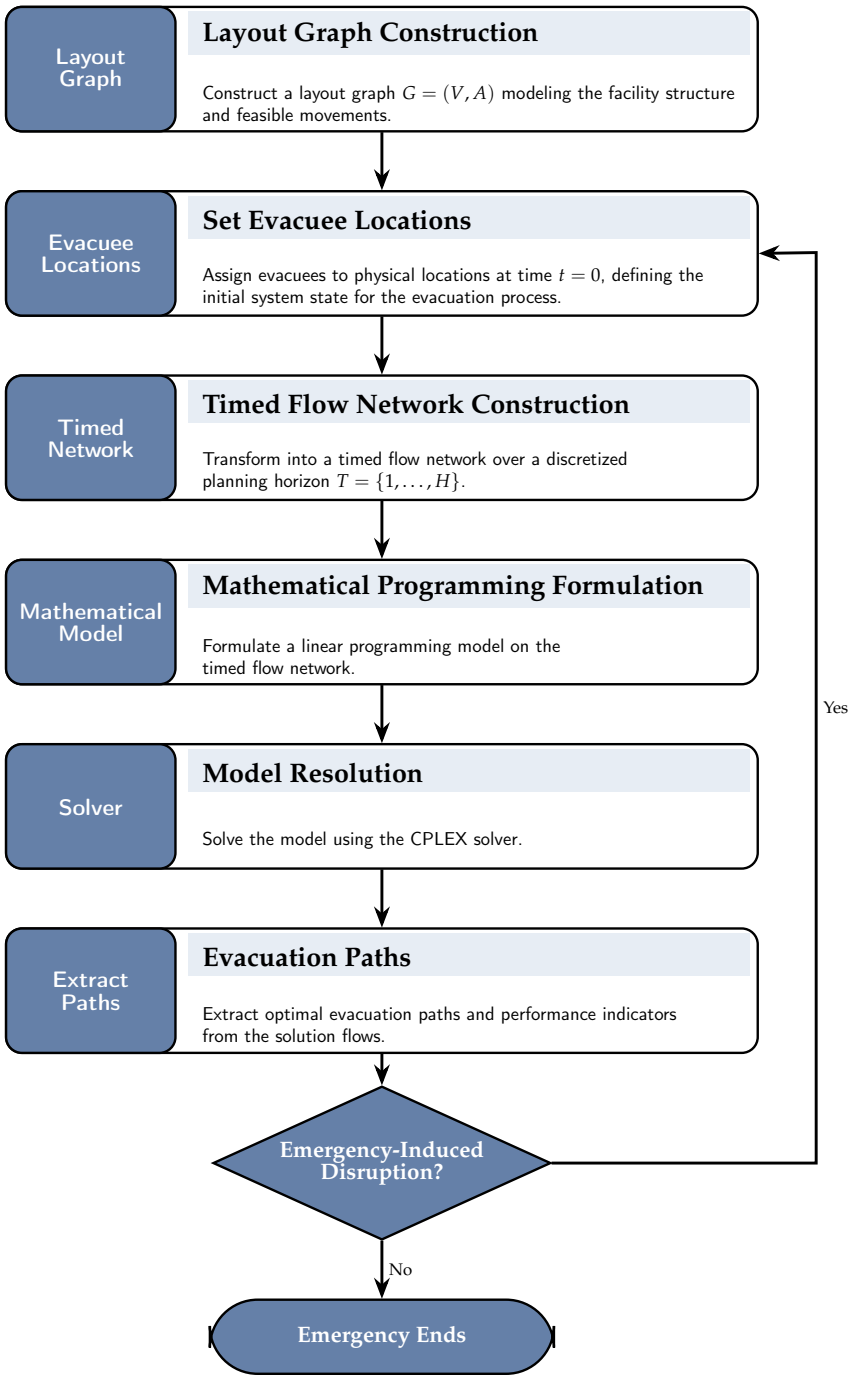

Figure A1. Flowchart of the online evacuation optimization algorithm (ignoring the decision block, the diagram also represents the offline algorithm).

1. **Layout Graph Construction.** The process starts by building the layout graph $G = (V, A)$, which represents the facility topology. Nodes correspond to physical locations (e.g., rooms, corridors, staircases), and directed arcs represent feasible movements between locations. Each arc is associated with traversal-time and capacity information, capturing the static constraints of the environment.
2. **Set Evacuee Locations.** Evacuees are assigned to physical locations at the beginning of the emergency, i.e., at time $t = 0$. This step defines the initial system state and corresponds to setting the initial node supplies (e.g., $b_{(i,0)}$) used by the subsequent time-expanded formulation.
3. **Timed Flow Network Construction.** Given a discretized planning horizon $T = \{1, \ldots, H\}$, the layout graph is transformed into a timed (time-expanded) flow

network. In this representation, each physical node is replicated across time and arcs are expanded accordingly, allowing the temporal evolution of evacuation flows to be explicitly modeled (including both waiting and movement dynamics).

4. **Mathematical Programming Formulation.** An optimization model is then formulated on the timed network as a (network) linear programming model. The formulation enforces flow conservation at timed nodes and capacity constraints on arcs, and it includes an objective function that promotes fast evacuation while penalizing evacuees who cannot reach a safe exit (or safe area) within the considered horizon.
5. **Model Resolution.** The resulting mathematical program is solved using the CPLEX solver. The solution returns optimal flow values on the timed network, thereby determining the recommended evacuation routing over time.
6. **Evacuation Paths.** Evacuation routes are extracted from the optimal flow solution and translated into actionable paths for groups of evacuees. In addition, performance indicators are derived from the solution, such as the weighted evacuation time and the number of unsaved individuals (and, when relevant, the evacuation makespan).

The final part of the flowchart introduces the decision mechanism that enables the online behavior of the algorithm. This mechanism governs the detection of emergency-induced disruptions and the possible activation of a reoptimization cycle, as described below.

7. **Emergency-Induced Disruption? (Decision block).** During the emergency, the system checks whether an emergency-induced disruption has occurred, including: (i) changes in evacuee locations with respect to the expected plan, and/or (ii) changes in network availability (e.g., locations becoming inaccessible, connections becoming impassable). If a disruption is detected, the algorithm loops back to Set Evacuee Locations, updates the current system state using the available information, and re-runs Steps 3–6 to compute updated evacuation paths.
8. **Emergency Ends.** If no disruption is detected (or the emergency is declared terminated), the algorithm stops.

References

1. Vermuyten, H.; Beliën, J.; De Boeck, L.; Reniers, G.; Wauters, T. A review of optimisation models for pedestrian evacuation and design problems. *Saf. Sci.* **2016**, *87*, 167–178. [CrossRef]
2. National Fire Protection Association (NFPA). Fire Loss in the United States During 2023. Statistical Report, 2024. Available online: https//www.nfpa.org/education-and-research/research/nfpa-research/fire-statistical-reports/fire-loss-in-the-united-states (accessed on 3 December 2025).
3. Khakzad, N. Optimal firefighting to prevent domino effects: Methodologies based on dynamic influence diagram and mathematical programming. *Reliab. Eng. Syst. Saf.* **2021**, *212*, 107577. [CrossRef]
4. Huang, P.; Chen, M.; Chen, K.; Ye, S.; Yu, L. Study on an emergency evacuation model considering information transfer and rerouting: Taking a simplified H-shape metro station hall as an example. *Tunn. Undergr. Space Technol.* **2022**, *124*, 104485. [CrossRef]
5. Goerlandt, F.; Li, J.; Reniers, G.; Boustras, G. Safety science: A bibliographic synopsis of publications in 2020. *Saf. Sci.* **2021**, *139*, e105242. [CrossRef]
6. Fu, L.; Qin, H.; Shi, Q.; Zhang, Y.; Shi, Y. An experimental study on evacuation dynamics including individuals with simulated disabilities. *Saf. Sci.* **2022**, *155*, 105878. [CrossRef]
7. Balboa, A.; González-Villa, J.; Cuesta, A.; Abreu, O.; Alvear, D. Testing a real-time intelligent evacuation guiding system for complex buildings. *Saf. Sci.* **2020**, *132*, 104970. [CrossRef]
8. Zhang, P.; Yang, L.; Lo, S.; Wang, D.; Li, M.; Jiang, J.; Jiang, N. Experimental study on evacuation behavior with guidance under high and low urgency conditions. *Saf. Sci.* **2022**, *154*, 105865. [CrossRef]
9. Chen, J.; Liu, C.; Meng, Y.; Zhong, M. Multi-Dimensional evacuation risk evaluation in standard subway station. *Saf. Sci.* **2021**, *142*, 105392. [CrossRef]

10. Cepolina, E.M. Phased evacuation: An optimisation model which takes into account the capacity drop phenomenon in pedestrian flows. *Fire Saf. J.* **2009**, *44*, 532–544. [CrossRef]
11. Khakzad, N. A methodology based on Dijkstra's algorithm and mathematical programming for optimal evacuation in process plants in the event of major tank fires. *Reliab. Eng. Syst. Saf.* **2023**, *236*, 109291. [CrossRef]
12. Zhang, W.; Liu, X.; Huang, Z.; Zhu, J. Dynamic parameters identification for sliding joints of surface grinder based on deep neural network modeling. *Adv. Mech. Eng.* **2021**, *13*, 1687814021992181. [CrossRef]
13. Mirahadi, F.; McCabe, B.Y. EvacuSafe: A real-time model for building evacuation based on Dijkstra's algorithm. *J. Build. Eng.* **2021**, *34*, 101687. [CrossRef]
14. Liu, L.; Zhang, H.; Shi, J.; Geng, J. Real-time evacuation route optimization in the fire scenarios of cruise ships. *Simul. Model. Pract. Theory* **2023**, *129*, 102843. [CrossRef]
15. Jiang, Y.; Xu, K.; Gai, W.; Salhi, S. Emergency response for tackling major accidental toxic gas releases: What should be done and when? *Saf. Sci.* **2022**, *154*, 105819. [CrossRef]
16. Deng, K.; Zhang, Q.; Zhang, H.; Xiao, P.; Chen, J. Optimal emergency evacuation route planning model based on fire prediction data. *Mathematics* **2022**, *10*, 3146. [CrossRef]
17. He, Z.; Shen, K.; Lan, M.; Weng, W. An evacuation path planning method for multi-hazard accidents in chemical industries based on risk perception. *Reliab. Eng. Syst. Saf.* **2024**, *244*, 109912. [CrossRef]
18. Chen, P.; Chen, G.; Wang, L.; Reniers, G. Optimizing emergency rescue and evacuation planning with intelligent obstacle avoidance in a chemical industrial park. *J. Loss Prev. Process Ind.* **2018**, *56*, 119–127. [CrossRef]
19. Kurdi, H.; Almulifi, A.; Al-Megren, S.; Youcef-Toumi, K. A balanced evacuation algorithm for facilities with multiple exits. *Eur. J. Oper. Res.* **2021**, *289*, 285–296. [CrossRef]
20. Huang, P.; Lin, X.; Liu, C.; Fu, L.; Yu, L. A real-time automatic fire emergency evacuation route selection model based on decision-making processes of pedestrians. *Saf. Sci.* **2024**, *169*, 106332. [CrossRef]
21. Tavares, R.M.; Galea, E.R. Evacuation modelling analysis within the operational research context: A combined approach for improving enclosure designs. *Build. Environ.* **2009**, *44*, 1005–1016. [CrossRef]
22. Wang, K.; Yuan, W.; Liang, W.; Yao, Y. An optimal guidance strategy for fire evacuations: A hybrid modeling approach. *J. Build. Eng.* **2023**, *73*, 106796. [CrossRef]
23. Georgiadou, P.S.; Papazoglou, I.A.; Kiranoudis, C.T. Multi objective evolutionary emergency response optimization in industrial accident scenarios. *Comput. Oper. Res.* **2010**, *37*, 1257–1267.
24. Yuan, J.; Sun, B. Multi-Objective Optimization Model for Emergency Evacuation Based on Adaptive Ant Colony Algorithm. *AI* **2025**, *6*, 1–15. [CrossRef]
25. Lin, J.R.; Chen, K.Y.; Song, S.Y.; Cai, Y.H.; Pan, P.; Deng, Y.C. Digital twin of buildings and occupants for emergency evacuation: Framework, technologies, applications and trends. *Adv. Eng. Inform.* **2025**, *66*, 103419. [CrossRef]
26. Han, Y.; Bian, R.; Meng, X. Improving a Regional-Scale Hurricane Evacuation Traffic Simulation Framework for Digital Twin Creation. *Transp. Res. Rec.* **2025**, *2679*, 171–185. [CrossRef]
27. Kim, Y.J.; Kim, H.; Ha, B.; Kim, W.T. Advanced fire emergency management based on potential fire risk assessment with informative digital twins. *Autom. Constr.* **2024**, *167*, 105722. [CrossRef]
28. Ding, Y.; Zhang, Y.; Huang, X. Intelligent emergency digital twin system for monitoring building fire evacuation. *J. Build. Eng.* **2023**, *77*, 107416. [CrossRef]
29. Alghamdi, M.; Abadleh, A.; Mnasri, S.; Alrashidi, M.; Alkhazi, I.S.; Alghamdi, A.; Albelwi, S. Hazard- and Fairness-Aware Evacuation with Grid-Interactive Energy Management: A Digital-Twin Controller for Life Safety and Sustainability. *Sustainability* **2026**, *18*, 133. [CrossRef]
30. Sorensen, J.H.; Shumpert, B.L.; Vogt, B.M. Planning for protective action decision making: evacuate or shelter-in-place. *J. Hazard. Mater.* **2004**, *109*, 1–11. [CrossRef]
31. Bolia, N.B. Robust scheduling for large scale evacuation planning. *Socio-Econ. Plan. Sci.* **2020**, *71*, 100756.
32. Huang, Y.; Yu, H.; Ju, X.; Pan, X. Safety Evaluation and Management Optimization Strategies for Building Operations Under the Integrated Metro Station–Commercial Development Model: A Case Study. *Systems* **2025**, *13*, 1081. [CrossRef]
33. Feng, Z.; Hou, H.; Chen, H.; Liu, Y. Fire simulation and evacuation optimization for metro stations with interpretability: A hybrid BIM, fire dynamics simulator and multiobjectives optimization algorithm approach. *J. Build. Eng.* **2026**, *117*, 114798. [CrossRef]
34. Reniers, G.; Audenaert, A.; Pauwels, N.; Soudan, K. Empirical validation of a real options theory based method for optimizing evacuation decisions within chemical plants. *J. Hazard. Mater.* **2011**, *186*, 779–787. [CrossRef] [PubMed]
35. Song, Y.; Wang, B.; Wang, X.; Zhang, Y.; Zhang, J.; Wang, Y. Hazardous Chemical Accident Evacuation Simulation and Analysis of Results. *Sustainability* **2025**, *17*, 6415. [CrossRef]
36. Yuan, F.; Han, L.D.; Chin, S.M.; Hwang, H. Integrated network approach of evacuation simulation for large complex buildings. *Saf. Sci.* **2009**, *47*, 698–706. [CrossRef]

37. Liu, C.; Mao, Z.L.; Fu, Z.M. Emergency evacuation model and algorithm in the building with several exits. *Procedia Eng.* **2016**, *135*, 12–18. [CrossRef]
38. Vanumu, L.D.; Ramachandra Rao, K.; Tiwari, G. Fundamental diagrams of pedestrian flow characteristics: A review. *Eur. Transp. Res. Rev.* **2017**, *9*, 49. [CrossRef]