



# Centralized exit allocation framework for system optimal evacuation in underground stations

Sunnie Haam<sup>a,1</sup>, Chiyeong Kang<sup>b,1</sup>, Hyunseok Kim<sup>b</sup>, Mintaek Yoo<sup>b,\*</sup>

<sup>a</sup> Department of Fire Safety Engineering and Disaster Management, University of Seoul, 163, Seoulsiripdae-ro, Dongdaemun-gu, Seoul, 02504, Republic of Korea

<sup>b</sup> Department of Civil and Environmental Engineering, Gachon University, 1342, Seongnam-daero, Sujeong-gu, Seongnam-si, Gyeonggi-do, 13120, Republic of Korea

## ARTICLE INFO

### Keywords:

Underground station  
Fire evacuation  
Centralized control  
User equilibrium  
System optimum  
Evacuation planning  
Exit allocation

## ABSTRACT

Underground stations represent high-risk environments where fires can result in mass casualties due to limited access points and enclosed spatial configurations. This study proposes a two-phase centralized evacuation guidance model structured sequentially around user equilibrium and system optimum, and comparatively analyzes the performance of the A\* algorithm and Dijkstra's algorithm under both conditions. A virtual three-basement-level station was modeled as a node-link graph with dynamic weights reflecting congestion levels and fire spread, and simulations were conducted across ten fire scenarios. Exit allocation ratios were explored at 1% increments to dynamically identify optimal distributions. Under the Phase 1 user equilibrium condition, the A\* algorithm demonstrated approximately 4.5 times greater computational efficiency than Dijkstra's algorithm and consistently achieved lower maximum evacuation times; however, bottlenecks arising from the concentration of evacuees at specific exits were observed. The inefficiency gap between user equilibrium and system optimum was quantified using the Price of Anarchy, which averaged 1.10 and 1.14 for the A\* and Dijkstra's algorithms, respectively. Under the Phase 2 system optimum condition, adjustments to exit allocation consistently reduced maximum evacuation times across all cases, and in bottleneck-free environments, Dijkstra's algorithm outperformed in certain cases. The findings of this study suggest that exit allocation strategy at the system level should be prioritized over algorithmic performance optimization in the design of evacuation guidance systems.

## 1. Introduction

According to the 2025 report by the International Association of Public Transport (UITP, 2025), the number of subway passengers in major cities worldwide has recovered rapidly following the COVID-19 pandemic, reaching 93% of pre-pandemic 2019 levels as of 2023, with ridership continuing on an upward trend. However, underground stations are characterized by a limited number of entrances and exits and are spatially constrained relative to the number of occupants. Due to the enclosed structural characteristics of such underground spaces, fires pose a significant risk of mass casualties. Notable historical incidents include the October 1995 Baku metro fire in Azerbaijan, which resulted in 289 fatalities and 265 injuries, and the February 2003 arson attack at Jungangno Station on the Daegu subway in South Korea, which caused 192 deaths and 148 injuries. Additionally, the January 2015 fire at L'Enfant Plaza Station in Washington, D.C. resulted in 1 fatality and 91

injuries, and approximately 50 smoke and fire incidents were reported on the D.C. Metro system in the first quarter of 2016 alone (Zhang et al., 2019). Furthermore, when a fire occurs in an underground station, smoke tends to spread upward due to buoyancy, which coincides with the direction of evacuation toward the surface, making it an even more critical impediment to evacuees. Tsukahara et al. (2011) conducted Fire Dynamics Simulator (FDS) simulations based on the 2003 Daegu subway fire and found that approximately 3.5 min after ignition, high concentrations of smoke and elevated temperatures develop throughout the upward evacuation stairways, rendering normal upward evacuation virtually impossible. Zhang et al. (2019) proposed a framework for identifying worst-case fire scenarios in underground stations using the concepts of Available Safe Egress Time (ASET) and Required Safe Egress Time (RSET), with heat release rate, fire location, and occupant load as key variables, confirming that exits function as the most critical bottleneck points during fire evacuation. Burnett (1984) identified

\* Corresponding author.

E-mail address: [mintaekyoo@gachon.ac.kr](mailto:mintaekyoo@gachon.ac.kr) (M. Yoo).

<sup>1</sup> These authors contributed equally to this work.

combustible materials in rolling stock, emergency ventilation systems, tunnel evacuation routes, and communication infrastructure as key risk factors for fire safety in underground rail systems, emphasizing that the structural characteristics of underground spaces make smoke propagation and toxic gas control extremely difficult during fires. Collectively, these prior studies indicate that underground station fires constitute high-risk environments in which smoke propagation, toxic gases, and reduced visibility interact in a compounding manner, underscoring the need for efficient evacuation route guidance systems.

In response to such emergency evacuation situations, a wide range of simulation models and guidance systems related to underground station evacuation have been actively investigated. Li et al. (2016) conducted evacuation simulations of underground stations using Pathfinder and identified staircases and fare gates as primary bottleneck points during the evacuation process. Zhou et al. (2022) proposed a Multi-Modal Cooperative Guidance (MMCG) system integrating three types of guidance means, signage, leaders, and broadcasts, and demonstrated through simulation that evacuation time exceeded 120 s without guidance, whereas the application of MMCG reduced the average evacuation time to 48.6 s. While research on evacuation simulation models and guidance systems has continued to advance, studies that simultaneously compare the performance of pathfinding algorithms under both user equilibrium and system optimum conditions, and quantitatively examine the effect of dynamically exploring fine-grained exit allocation ratios, remain limited.

Therefore, unlike prior studies, this research proposes a novel two-phase centralized evacuation guidance model structured sequentially around user equilibrium and system optimum and aims to simultaneously compare the performance of the A\* algorithm and Dijkstra's algorithm under both conditions. Furthermore, by dynamically searching for optimal exit allocation ratios at 1% increments, this study seeks to mitigate bottleneck phenomena and to quantitatively identify the effect of pathfinding algorithm performance on overall evacuation efficiency under the precondition of a system optimum environment.

## 2. Literature review

### 2.1. Fire evacuation simulation models

In research on fire evacuation in underground stations, a variety of simulation-based frameworks have been developed to quantitatively analyze the effects of fire on evacuation behavior. Jiang et al. (2025) constructed a four-story shopping mall in Chengdu as a digital platform using Building Information Modeling (BIM), and developed a new framework by integrating Pathfinder and PyroSim within this platform to execute fire scenarios. Through this approach, the study examined the effects of fire on evacuation and found that reduced visibility exerts a more widespread impact than increases in temperature and CO concentration. Kim et al. (2026) conducted a comparative study of the A\* algorithm and Dijkstra's algorithm for optimal path selection, targeting a six-basement-level underground station in Seoul. Evacuation simulation models were configured for static disasters, dynamic disasters, and real-time crowd congestion, and simulations were performed for eight disaster location cases per model. Schmidt-Polończyk (2024) defined three cases based on smoke conditions and pre-evacuation instructions, and conducted an evacuation experiment involving 50 participants in a road tunnel approximately 678 m in length in Poland. The experiment measured total evacuation time, pre-movement times, and travel speed, and analyzed the effects of smoke on movement time. The study demonstrated that travel speed under smoke conditions is influenced by a combination of factors, including evacuees' prior knowledge of the tunnel. Yang et al. (2022) proposed a route planning approach based on multiobjective robust optimization using the NSGA-II algorithm, integrating PyroSim and Pathfinder to address evacuation scenarios involving multiple simultaneous fires in an underground station. Li et al. (2024) integrated BIM, PyroSim, and Pathfinder to simulate multiple

fire scenarios across four fire zones in an underground commercial space (UCS) in Fuzhou, China, selected the most unfavorable fire scenario, and compared the evacuation efficiency of parallel double, straight, and cross-type staircases.

As demonstrated by these studies, research on the effects of fire on evacuation and the development of fire prediction models has been actively pursued. Collectively, these studies indicate that reduced visibility, smoke propagation, and the dynamic evolution of hazardous zones act in a compounding manner on evacuation efficiency.

### 2.2. Pathfinding models using optimal evacuation route algorithms

In the field of evacuation route planning, a variety of evacuation models based on optimal pathfinding algorithms including Dijkstra's algorithm and the A\* algorithm have been developed, and a growing body of research demonstrates that the criteria for algorithm selection vary depending on the application environment and constraints. Mirahadi and McCabe (2021) converted BIM-based data into a Geometric Network Graph (GNG) and developed EvacuSafe, a real-time fire evacuation management model, applying Dijkstra's algorithm to compute final evacuation routes. The study also defined multiple hazard indices, including structural collapse and door malfunction. Tian et al. (2025) developed an algorithm for real-time optimal route planning for oversized special vehicles subject to bridge and tunnel height constraints, targeting ultra-large-scale networks such as the Chinese national road network. By introducing a dual-layer road network model separating expressways from ordinary roads and a two-level spatial grid framework, the study achieved up to a 1,000-fold improvement in data retrieval speed compared to a single-layer approach. A parallelized bidirectional Dijkstra's algorithm, which initiates search simultaneously from both the origin and destination, was applied for route planning, and the bidirectional approach was found to be more effective in large-scale environments by reducing the search space by half. Zhu et al. (2025) predicted the inundation extent of a dam failure at the Fuchonggou tailings dam in Anhui Province, China, using numerical simulation combining the VOF method and the RNG k- $\epsilon$  turbulence model, and proposed an improved three-dimensional Dijkstra's algorithm for resident evacuation within the inundated zone. Gonzales et al. (2025) presented an evacuation route optimization method for resource-constrained environments lacking advanced GIS systems or real-time data infrastructure. The study manually converted the road network of Barranco, Lima, Peru into a graph and developed a static evacuation route computation model using WinQSB and Dijkstra's algorithm. Edge weights were defined based on distance, and travel time was calculated by applying an average walking speed of 1.2 m/s. The study adopted Dijkstra's algorithm on the grounds that the A\* algorithm requires precise heuristics and high-performance computing, rendering it unsuitable for the given environment. Wei et al. (2024) proposed the Modified Sparrow Search Algorithm (MSSA), an enhanced version of the Sparrow Search Algorithm (SSA), for multi-exit evacuation route planning in dense crowd environments. The SSA is a bio-inspired meta-heuristic optimization algorithm that assigns individual roles, producers, scroungers, and scouts, to derive routes, and is well-suited for route planning problems due to its fast convergence speed and high search precision. However, it was susceptible to convergence toward local optima. To address this limitation, the study incorporated a Golden Sine algorithm and a nonlinear weight factor optimization strategy into the producer position update phase to enhance global search capability, and integrated a Cauchy-Gaussian perturbation strategy with a mountain climbing method-based local search mechanism to strengthen the ability to escape local optima. Haam et al. (2025) developed a Dijkstra's algorithm-based evacuation route optimization model for a deep underground station with six basement levels in Seoul, and quantitatively analyzed the reduction in maximum evacuation time resulting from the installation of additional evacuation routes. The study introduced a dynamic edge weight update scheme that reflects a walking speed

reduction factor based on real-time congestion levels, thereby overcoming the limitations of prior studies that were confined to simple shortest-path searches. The study further demonstrated that the congestion dispersion effect varies depending on the location and configuration of additional routes. Additionally, based on evacuation simulation results, the concept of an "evacuation-vulnerable zone" was proposed to highlight the impact of structurally damaged areas on evacuation time during disasters.

As demonstrated above, model development and algorithm comparison studies utilizing optimal pathfinding algorithms such as Dijkstra's algorithm and the A\* algorithm have been actively conducted, and the findings collectively suggest that the required algorithmic characteristics vary depending on situational and environmental constraints.

### 2.3. User-equilibrium and system-optimum approaches

The design approaches for evacuation route guidance systems can be broadly categorized into user equilibrium-based approaches, which aim to minimize the travel cost of individual users, and system optimum-based approaches, which aim to maximize the efficiency of the overall system. Studies analyzing the performance differences between these two approaches have been consistently conducted in the fields of transportation and evacuation. Matsui and Yoshihiro (2024) proposed FCR (Flow-based Capacity-aware Rerouting), a flow-based centralized route guidance system for mitigating urban traffic congestion. FCR computes rerouting at the level of traffic flow rather than individual vehicles, substantially reducing the computational burden of the centralized approach while simultaneously incorporating consideration of individual user benefit, an advantage associated with decentralized approaches. Tan et al. (2025) proposed the FRDT (Fire Resilience Digital Twin) framework, which provides real-time optimal evacuation routes for occupants present in a building during an indoor fire and enhances fire resilience. FRDT identifies the locations of all indoor occupants from a central server and computes and assigns routes for each individual. The study applied FRDT to simulate a scenario targeting an underground shopping mall in Beijing, suggesting that system optimization is effective in the early phases of evacuation. Zhao et al. (2022) proposed adaptive evacuation signage that optimizes exit directions based on current emergency conditions during fires and other hazardous events, as an alternative to conventional signage that directs evacuees in fixed directions. Two frameworks, centralized and decentralized, were developed for the adaptive evacuation signage system, and the effectiveness of both signage systems was validated by combining VR experiments and agent-based simulation. Tang et al. (2025) proposed the PAFA (Pedestrian flow Assignment and Facility Adjustment) model, which integrates pedestrian flow dynamics and facility operational state adjustment based on the Cell Transmission Model (CTM). This model is capable of accounting for the effects of facility layout and operational conditions, such as escalators and fare gates, on pedestrian movement. Numerical experiments demonstrated that the PAFA model under the system optimum (SO) condition reduced total travel time by 4.1–10.6% compared to the user equilibrium (UE) condition. Chow (2007) developed a comprehensive framework for analyzing and computing user equilibrium, system optimum, and externalities in time-dependent road networks, and numerically demonstrated from a transportation perspective that SO induces greater departure time dispersion than UE, thereby alleviating congestion and reducing system costs. Zhang et al. (2024a) proposed an efficient guidance planning model that combines CTM simulation with DRF optimization, reducing the number of computations to 1/50 to 1/80 of those required by conventional meta-heuristic approaches. In this model, routes proposed by the DRF are executed in CTM simulation and evaluated, after which the evaluation results are fed back into the DRF to update the routes; this iterative process enhances the efficiency of both initial route construction and route updates by incorporating the DRF into the model. Zhang et al. (2024b) proposed a model for optimal evacuation guidance using

dynamic exit signage, introducing Branch Grafting and Leaf Grafting algorithms within a DRF (Directed Rooted Forest)-based methodology. Validation through the Social Force Model (SFM) demonstrated approximately 100 times greater computational efficiency compared to genetic algorithms, and the practical realism of the simulation was enhanced by integrating the compliance rate into the optimization process.

Collectively, these studies suggest that system optimum conditions consistently outperform user equilibrium conditions in terms of total travel time and congestion, and demonstrate the need for system-level flow control to take precedence over individual route optimization in the design of evacuation guidance systems.

### 2.4. Evacuation route optimization models using reinforcement learning

Recent years have seen active research into artificial intelligence-based dynamic evacuation route guidance models, including reinforcement learning approaches, aimed at overcoming the limitations of static pathfinding algorithms. Lischka and Kulcsár (2026) defined the Bus Evacuation Orienteering Problem (BEOP), the problem of evacuating the maximum number of people by bus within a limited time, and proposed GREAT-EER, a deep reinforcement learning framework based on a Graph Edge Attention Network (GREAT). Reinforcement learning was applied to generate optimal evacuation routes that reflect the asymmetric travel time structure of real-world road networks, and a single offline training session enables inference within seconds for arbitrary evacuation scenarios, yielding practical advantages over exact solution methods. Huang et al. (2024) proposed RSARSA, which combines the SARSA algorithm with an exit radar and fire radar reward mechanism, to overcome the limitations of static evacuation planning. Reinforcement learning was applied to dynamic route selection that reflects real-time fire locations and pedestrian decision-making, enabling flexible responses to varying fire intensities and propagation speeds. In KTV scenario validation, RSARSA achieved an initial route search success time approximately 59% faster than standard SARSA. Xie et al. (2025) proposed a CTM-PyroSim-QMIX framework for real-time cooperative control of multiple dynamic signage systems in indoor fire evacuation. QMIX, a multi-agent reinforcement learning (MARL) algorithm, was applied to signage direction decisions that simultaneously optimize evacuation efficiency, congestion risk, and fire risk. The dynamic signage system proposed in the study demonstrated improvements over static signage across all metrics, including evacuation time, congestion, and fire damage indicators, in real space-based scenarios. Mas et al. (2024) proposed a reinforcement learning-based guidance system that places node agents at intersections and trains optimal route guidance policies using the SARSA algorithm, addressing the congestion problem associated with shortest-path strategies in tsunami evacuation. Reinforcement learning was applied to dynamically guide routes by incorporating real-time pedestrian density as a state variable, and the system achieved the evacuation of up to 3,012 additional evacuees in the early evacuation phase compared to the shortest-path approach in a tsunami scenario in Kochi City.

While these reinforcement learning-based studies have demonstrated improvements in route guidance performance in dynamic environments, limitations remain with respect to the applicability in real underground station environments and the transparency of result interpretation, owing to the structural complexity of the algorithms. Furthermore, direct performance comparisons with conventional graph-based algorithms remain insufficient.

### 2.5. Knowledge gaps and contributions

While prior studies have examined evacuation efficiency improvement from diverse perspectives, including fire evacuation simulation, optimal pathfinding algorithms, and centralized route guidance, the majority of research has focused on algorithm performance comparison

under user equilibrium conditions and on structural improvements to algorithms. Studies that quantitatively compare algorithm performance differences under system optimum conditions, including identical network configurations and forced exit allocation, remain limited. Furthermore, although studies conducted from user equilibrium and system optimum perspectives have consistently demonstrated that SO conditions improve overall evacuation efficiency relative to UE conditions, research directly comparing the performance differences among specific pathfinding algorithms under such system optimum conditions is scarce. Furthermore, while research proposing efficient system-level methodologies and evacuation guidance models utilizing advanced route optimization algorithms has been consistently conducted, studies that explicitly verify computation times at a level directly applicable to real evacuation situations remain limited. Reinforcement learning-based studies have demonstrated improvements in route guidance performance in dynamic environments. However, limitations remain regarding applicability and interpretability in real underground station environments due to the structural complexity of the algorithms. In summary, studies that simultaneously compare the A\* and Dijkstra's algorithms under both centralized control and user equilibrium conditions are nearly absent, and existing centralized control studies have rarely conducted simulations with granular exit allocation ratios as in the present study, leaving the effects of dynamically searching for optimal allocations through fine-grained exit allocation ratios insufficiently examined.

Based on these knowledge gaps, this study proposes a two-phase centralized evacuation guidance model structured sequentially around user equilibrium and system optimum, and comparatively analyzes the performance of the A\* and Dijkstra's algorithms under both conditions. Furthermore, by simulating with fine-grained exit allocation ratios and forcibly assigning evacuation routes, this study aims to quantitatively identify the effect of algorithmic optimization performance differences on evacuation efficiency in a bottleneck-mitigated environment.

### 3. Research methodology

#### 3.1. Evacuation route algorithm

##### 3.1.1. Dijkstra's algorithm

Dijkstra's algorithm is an optimal pathfinding algorithm that uses the

minimum cost from a source node to a target node, and employs an uninformed search strategy that explores all nodes. Due to the exhaustive nature of this all-node exploration, the algorithm exhibits superior path optimization performance and consistently derives routes that closely approximate the exact optimal solution. The operational procedure of Dijkstra's algorithm is as follows.

First, the distance of the source node is set to zero, and the distances of all other nodes are initialized to infinity. An unvisited node set is created, and distances to all nodes are explored until no unvisited nodes remain. During the process of exploring all nodes, the node  $u$  with the minimum distance among unvisited nodes is selected and marked as visited. All neighboring nodes  $v$  of node  $u$  are then examined, and a relaxation process is performed for each neighboring node. Specifically, it is checked whether the distance to a neighboring node  $v$  via the current node  $u$  is shorter than the previously recorded distance to  $v$ . If a shorter distance is confirmed, the distance of neighboring node  $v$  is updated to the shorter path distance, and the predecessor of  $v$  is recorded as  $u$ . This process is repeated for all neighboring nodes. The entire procedure is repeated until all nodes have been marked as visited, and upon termination of the algorithm, the shortest distance and path information to each node are stored (Dijkstra, 1959).

The execution procedure of Dijkstra's algorithm described above is illustrated in Fig. 1.

##### 3.1.2. A\* algorithm

The A\* algorithm is an optimal pathfinding algorithm that shares the same foundational objective as Dijkstra's algorithm. However, unlike the uninformed search strategy of Dijkstra's algorithm, the A\* algorithm employs a heuristic-based pathfinding approach. The heuristic is a function that estimates the expected cost to the destination, defined by Equation (1).

$$f(n) = g(n) + h(n) \quad (1)$$

Here,  $f(n)$  represents the total estimated cost of a path, computed as the sum of  $g(n)$ , the cost from the source node to the node currently being evaluated, and  $h(n)$ , the straight-line distance from the node currently being evaluated to the destination. The reason for adopting straight-line distance for  $h(n)$ , rather than Manhattan distance or pre-computed path-based distance values, is that straight-line distance incurs comparatively lower computational overhead than alternative

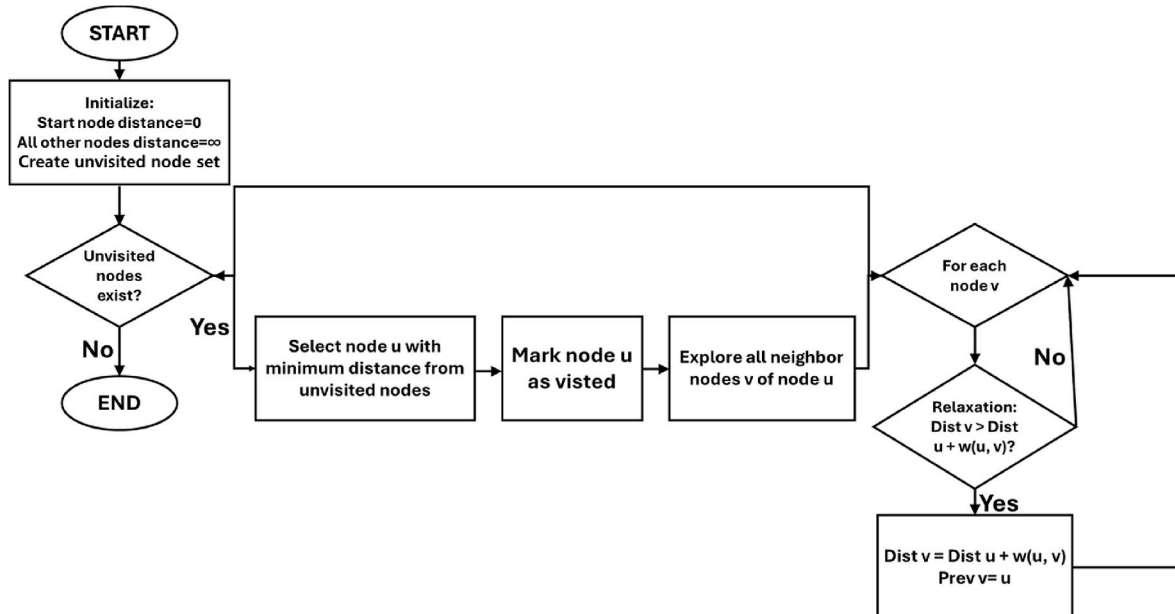


Fig. 1. Dijkstra's algorithm flow chart.

methods and minimizing preprocessing computation for the heuristic is the most rational approach to fully leveraging the characteristics of the heuristic function. Furthermore, in the node-link network of this study, all movement is constrained to links, and the straight-line distance is therefore structurally always less than or equal to the actual travel distance. This satisfies the admissible heuristic condition, thereby theoretically guaranteeing the derivation of the optimal path by the A\* algorithm. In situations where a detour is necessary, such as when a staircase exists in the opposite direction of an exit, the heuristic value of the corresponding node is estimated to be relatively large, lowering its search priority and potentially increasing the number of nodes explored. However, this is solely a matter of search efficiency; since the algorithm continues searching until the target node is reached, the accuracy of the optimal path is not compromised. The fact that, in actual simulation results, the A\* algorithm explored approximately 4.5 times fewer nodes than Dijkstra's algorithm while recording equivalent evacuation times empirically supports the conclusion that the Euclidean heuristic maintains accuracy while possessing sufficient search guidance capability (Yan, 2023). This heuristic-based search approach explores only a subset of nodes, unlike the uninformed search strategy of Dijkstra's algorithm which exhaustively explores all nodes. In general, the optimality guarantee of the A\* algorithm is contingent on the admissibility of the heuristic function; when a non-admissible heuristic is employed, the globally optimal path cannot always be guaranteed.

### 3.2. Centralized control algorithm

#### 3.2.1. Two-phase Simulation

In this study, a centralized control approach was adopted for evacuation route guidance to address the bottleneck phenomena arising from individual-level route optimization. The evacuation guidance simulation was divided into two phases. According to Wardrop and Whitehead (1952), user equilibrium is defined as the state in which the travel costs on all used routes are equal and no less than those on any unused route, such that no individual traveler can reduce their own travel cost by unilaterally changing routes. System optimum, on the other hand, refers to the state in which the total travel cost across the entire network is minimized, which may require individual travelers to accept suboptimal routes for the benefit of the overall system. Based on these principles, Phase 1 serves as a dynamic approximation framework based on the user equilibrium principle of Wardrop and Whitehead (1952), in which clusters formed at each location independently select their optimal exit, and Phase 2 serves as a system optimum framework in which the centralized control system assigns exits to each cluster.

In Phase 1, each cluster autonomously selects the exit from which evacuation can be completed most rapidly from its current location. Throughout the simulation, routes are computed at every time step by reflecting congestion weights arising from per-second fluctuations in occupant flow and weights arising from fire spread, and routes are updated only when the route cost is lower than that of the current route; otherwise, the current route is maintained. The node-link network of this study consists of fixed link lengths of 5 m horizontally and 10 m vertically, such that any route change structurally results in a meaningful cost difference. This partially satisfies the core condition of Wardrop UE, in which a cluster maintains its route when it cannot benefit from unilaterally deviating from it. This individual optimization approach allows each cluster to make locally optimal decisions. However, since route selection among clusters does not account for one another, evacuation personnel may concentrate at specific exits, potentially degrading the efficiency of the overall system. It should be noted that this phase does not mathematically guarantee the achievement of global Wardrop UE through iterative convergence, and is interpreted as an approximation framework that emulates behavioral patterns based on the UE principle.

In Phase 2, exit utilization rates are analyzed based on the results of the Phase 1 simulation. The optimization objective function of this phase

is the minimization of the maximum evacuation time of the entire system, and the constraints are forced exit assignment by the centralized control system and the deactivation of autonomous route replanning by clusters. To derive the optimal exit allocation ratio, the centralized control system conducts simulations across 100 cases in 1% increments from 1% to 100%. For each case, the centralized control system reallocates clusters from overloaded exits to underutilized exits to balance exit utilization. Reallocated clusters are constrained to move exclusively toward the exit assigned by the centralized control system. The case yielding the shortest maximum evacuation time among the 100 cases is selected as the optimal allocation ratio; this constitutes a direct exhaustive search optimization of the SO objective function, guaranteeing the optimal solution within a 1% resolution. Through this process, individual autonomous choice is restricted while the evacuation efficiency of the overall system is improved.

Through these two phases, the performance improvement in user equilibrium resulting from system optimization is ultimately quantified, and the effectiveness of the centralized control system is evaluated.

#### 3.2.2. Exit distribution Strategy

In this study, the exit allocation mechanism is defined as follows.

The centralized control system reassigns cluster exits based on the exit utilization rates derived from the Phase 1 simulation results. To determine the optimal exit allocation ratio, simulations are conducted across 100 cases in 1% increments from 1% to 100%, and the exit assignment rules for each case are as follows.

First, the number of assignable exits is determined according to the target allocation ratio. For example, when the target allocation ratio is 1%, up to 100 exits could theoretically be assigned. However, since only 4 exits are available in practice, 1% is assigned to each of 3 exits and the remaining 97% is assigned to the 1 remaining exit. When the target allocation ratio is 20%, 20% is assigned to each of 3 exits and 40% is assigned to the remaining 1 exit. When the target allocation ratio cannot be distributed across all exits, for instance when  $40\% \times 4 = 160\%$  exceeds 100%, 40% is assigned to each of the maximum assignable 2 exits, and the remaining 20% is distributed equally among the other 2 exits.

The exits to be assigned the target allocation ratio are selected based on the distance between the Phase 1 utilization rate and the target allocation ratio. This distance is computed as the absolute difference between the Phase 1 utilization rate and the target allocation ratio, and higher priority is assigned to exits with a shorter distance. For example, if the utilization rates of each exit in Phase 1 are 20%, 20%, 10%, and 50%, and the target allocation ratio is 1%, the distances from each exit to 1% are 19%, 19%, 9%, and 49%, respectively. Consequently, Exit 3 is assigned the highest priority and Exits 1 and 2 are assigned the second priority, and these exits receive the target allocation ratio.

After the target allocation ratio has been assigned, clusters are reallocated from overloaded exits to underutilized exits. Clusters subject to reallocation are selected in descending order of evacuation time from the overloaded exit currently in use, and are reassigned to the nearest underutilized exit from their source node. This process is repeated across all 100 cases, and the case yielding the shortest maximum evacuation time is selected as the optimal allocation ratio.

### 3.3. Optimal evacuation guidance simulation algorithm

The evacuation guidance simulation model used in this study was developed in Python and simulates a scenario in which all occupants move in accordance with evacuation guidance.

In the initialization phase, the station network graph, occupant distribution by cluster, and the initial fire node ( $N_{fire}$ ) are configured. Based on the defined node and edge information, the selected path-finding algorithm (Dijkstra or A\*) is applied to each cluster to compute the initial optimal evacuation route.

The simulation runs in 1-s time steps until all clusters reach an exit, and at each time step, the following processes are executed sequentially:

fire propagation, travel speed update, route replanning, and cluster movement. In the fire propagation step, an adjacent node ignites when the distance traveled by fire along an edge equals or exceeds the length of that edge. In the travel speed update step, the congestion weight and fire weight at each cluster's current position are computed, and the actual travel speed is updated accordingly. The final travel speed is calculated by dividing the base speed by the product of the two weights. In the route replanning step, Phase 1 attempts route replanning at every second, computes a new optimal route based on the current graph state, and updates the existing route with the new route if a more efficient path is identified. In Phase 2, the centralized control system forcibly assigns an exit to each cluster for each case. Each cluster recomputes its evacuation route based on the assigned exit, searches for the optimal path to the assigned exit, and maintains that route throughout the remainder of the simulation. Upon arrival of all clusters at their respective exit nodes, the evacuation completion time is recorded. Upon completion of Phase 1, performance metrics including maximum evacuation time, exit utilization rates, and the number of route replannings are recorded and output. Upon completion of Phase 2, maximum evacuation time, optimal exit allocation ratio, and the degree of improvement in evacuation efficiency are output.

All simulations were performed on a desktop computer equipped with an Intel Core i5-13400 processor (13th generation, 10 cores) and 8 GB of RAM, running Windows 11 (64-bit). The simulation framework was implemented in Python 3.13, utilizing NumPy 2.2.6 for numerical computation, Matplotlib 3.10.3 for visualization, NetworkX 3.5 for graph-based network modeling, and pandas 2.3.0 for data management and result export.

The execution procedure of the evacuation guidance simulation model is illustrated in Fig. 2.

## 4. Model configuration and analysis conditions

### 4.1. Study site configuration

In this study, a virtual underground station with three basement levels located in Seoul was modeled as a node-link graph for use in the evacuation guidance simulation. The majority of domestic underground stations are small-to-medium-sized stations with two to three basement levels and four to six exits. The station structure used in this study designated six staircases connecting basement level 3 to basement level 2, four staircases connecting basement level 2 to basement level 1, and four exits connecting basement level 1 to the surface. The horizontal distance between nodes was set to 5 m, and the vertical distance between nodes across floors was set to 10 m. The node-link layout of the spatial model constructed as described above is illustrated in Fig. 3.

### 4.2. Pedestrian condition

The evacuation guidance simulation conducted in this study incorporates parameters such as evacuee placement, number of occupants, and evacuation speed. Among these, evacuation speed is one of the factors that most significantly influences the maximum evacuation time, which constitutes the primary outcome of this study. Accordingly, the evacuation speed values prescribed in the Design Guidelines for Urban Railway Stations and Transfer Convenience Facilities issued by the Ministry of Land, Infrastructure and Transport of the Republic of Korea were adopted (Ministry of LandInfrastructure and Transport, 2022). A horizontal travel speed of 60 m/min and a vertical travel speed of 15 m/min were applied, and the values used are summarized in Table 1.

Occupants were placed exclusively on basement level 3, simulating a congested scenario in which passengers waiting for trains and passengers who have just alighted from trains are concentrated in the same area, thereby facilitating the occurrence of high congestion levels. The number of occupants per node ranges from 200 to 500, with a total of 9,100 occupants distributed across 29 nodes. Occupant placement does

not influence the performance of any specific algorithm. In this study, a cluster is defined as a single movement unit comprising all occupants within the same node range. This definition is based on the premise that occupants initiating evacuation from the same location tend to move in the same direction psychologically, and that following the same evacuation route is rational from a system perspective as well. The size of each cluster is fixed at the time of initial formation at the start of the simulation, and each cluster is maintained as an independent unit without splitting or merging throughout the simulation. The occupant distribution is illustrated in Fig. 4.

### 4.3. Crowd congestion condition

In the evacuation guidance simulation of this study, dynamic route planning is performed by incorporating various conditions such as fire location and crowd density as dynamic weights. Among these, crowd density is a key variable that directly affects evacuation speed. When the number of occupants at a specific node exceeds a predefined threshold, weights are applied to the surrounding links to reduce the travel speed of evacuees.

The relationship between crowd density and travel speed used for weight computation was adopted from values proposed by Zhou et al. (2020). Zhou et al. (2020) analyzed surveillance camera footage captured during an actual earthquake using Tracker software and quantitatively derived the relationship between crowd density and travel speed under evacuation conditions. The reliability of this study was validated through comparative analysis with seven existing pedestrian flow studies, including those by Hankin, Predtechenskii, and Weidmann, and it addressed the limitations of prior studies that were based on data from drill scenarios or failed to adequately reflect the influence of panic behavior, thereby providing data more representative of actual evacuation conditions. As shown in the resulting graph, evacuation speed decreases sharply as crowd density increases, with the most pronounced speed reduction observed in the range of 1 to 2 persons/m<sup>2</sup>. The evacuation speed graph as a function of crowd density is illustrated in Fig. 5.

To incorporate the weights derived from the graph above into the simulation, a threshold value for applying speed reduction must be defined. Crowd density is normalized to 1 when the number of evacuees reaches the maximum capacity of a corridor, calculated based on corridor width and evacuation capacity per meter of corridor width (persons/m). This occupant count is set as the threshold value, against which speed reduction is applied. The evacuation capacity values for corridors were adopted from the Design Guidelines for Urban Railway Stations and Transfer Convenience Facilities issued by the Ministry of Land, Infrastructure and Transport (Ministry of LandInfrastructure and Transport, 2022). The maximum evacuation capacity is computed as the product of corridor width and evacuation capacity per meter of width, and the minimum widths specified in the design guidelines were applied: 8 m for horizontal corridors and 6 m for vertical corridors. Accordingly, the threshold value for horizontal movement elements was calculated as 640 persons, the product of the minimum width of 8 m and the evacuation capacity of 80 persons/m, and the threshold value for vertical movement elements was calculated as 360 persons, the product of the width of 6 m and the evacuation capacity of 60 persons/m. In addition, in the speed-density model, the actual travel speed is reduced by dividing the base travel speed by a composite weight that reflects both congestion and fire conditions, and all clusters travel along links with this composite weight applied. The link travel time ( $T_{link}$ ) is defined as follows.

$$T_{link} = \frac{d}{v_{actual}}, v_{actual} = \frac{v_{base}}{w_{total}}, w_{total} = w_{congestion} \times w_{fire}$$

where  $d$  is the link distance (m),  $v_{base}$  is the base travel speed, and  $w_{total}$  is the composite weight. All clusters move according to this computation

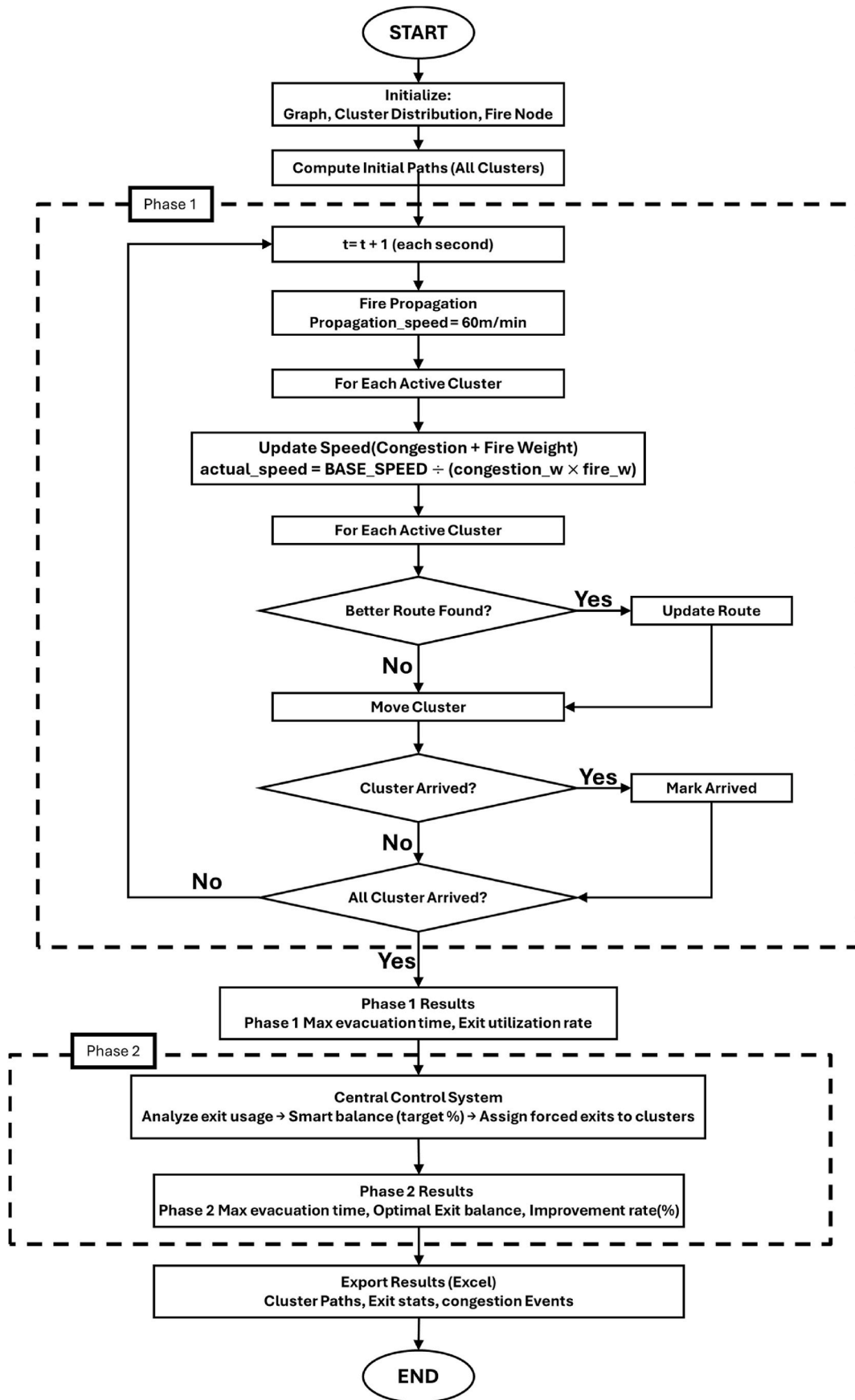


Fig. 2. Optimal evacuation guidance simulation model flow chart.

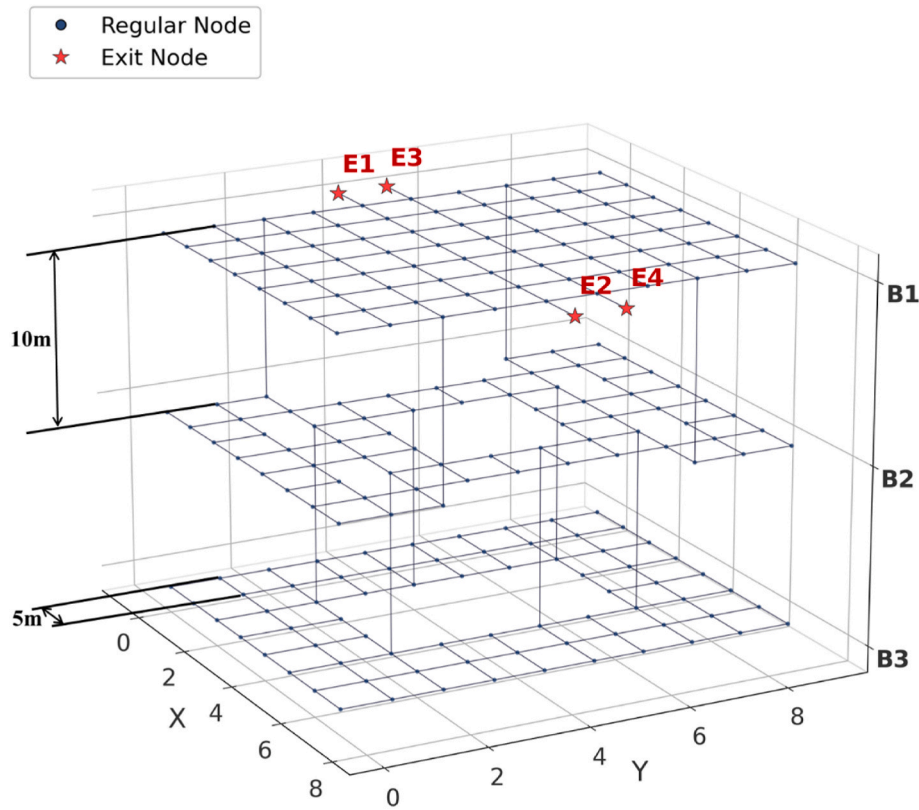


Fig. 3. Node-link layout of the underground station.

Table 1

Evacuation speeds by evacuation factor.

| Evacuation Factor                                           | Evacuation Speed |
|-------------------------------------------------------------|------------------|
| Horizontal transportation(platform, concourse and corridor) | 60 m/min         |
| Vertical transportation(stairs, stationary escalator)       | 15 m/min         |

method, and the operational procedure of the model is as follows. At every second, the number of occupants at each node is assessed, and the speed reduction weight corresponding to the applicable range based on the occupant threshold values in Table 3 is applied. The weight is applied to all links adjacent to the node in question, and when two nodes sharing a link simultaneously exhibit congestion, the weight corresponding to the higher congestion level is applied. The travel speed as a function of evacuation capacity and number of evacuees is summarized in Tables 2 and 3, respectively.

All input variables described above were configured based on values derived from design guidelines and prior studies, and were applied identically across all algorithms and cases. No algorithm-specific tuning parameters were employed.

#### 4.4. Fire spread model

In this study, fire originates at the initial ignition node and spreads to all connected adjacent nodes. The propagation speed and weights were adopted from values proposed by Haam et al. (2025). Accordingly, the fire propagation speed was set to 60 m/min, and since the edge length between nodes is 5 m, fire spreads to adjacent nodes every 6 s. The fire weight was set to 2, such that evacuees passing through a fire-affected node experience a reduction in travel speed by a factor of one-half. An example of fire propagation is illustrated in Fig. 6.

Ten fire origin locations were selected to reflect a variety of spatial conditions. Spatial diversity was ensured by designating four nodes at

the extremities and one node at the center of the spaces on basement levels 2 and 3. Basement level 1 was excluded from the scenarios because the simulations in this study are predicated on a scenario in which occupants concentrated on the basement level 3 platform evacuate to the surface; in the event of a fire on basement level 1, that floor, being adjacent to the surface exits, would fall within the fire's zone of influence, potentially resulting in evacuation routes being blocked or hazard conditions being intensified. Accordingly, evacuation guidance simulations were conducted targeting basement levels 2 and 3, which present higher route complexity and congestion likelihood, with five cases designated on each floor for a total of ten cases. The fire origin node for each case is summarized in Table 4.

## 5. Analysis results

In this study, the evacuation guidance simulation was conducted in two phases: Phase 1 corresponding to user equilibrium and Phase 2 corresponding to system optimum.

### 5.1. User-equilibrium results

The Phase 1 simulation was conducted based on user equilibrium, and the maximum evacuation time by algorithm and exit utilization rate by exit are summarized in Tables 5 and 6, respectively. In Phase 1, the maximum evacuation time ranged from 574 to 741 s for the A\* algorithm and from 590 to 722 s for Dijkstra's algorithm, with the A\* algorithm consistently recording lower maximum evacuation times across all ten cases. Regarding exit utilization rates, both algorithms exhibited the highest concentration at Exit E3, ranging from 43.7% to 49.5%, suggesting that evacuee concentration at a specific exit had occurred. In the case of the A\* algorithm, the utilization rate of Exit E2 averaged 13.2%, which was higher than the 7.0% recorded by Dijkstra's algorithm, indicating that the uncertainty inherent in heuristic-based path-finding inadvertently induced an exit dispersion effect. Meanwhile, in

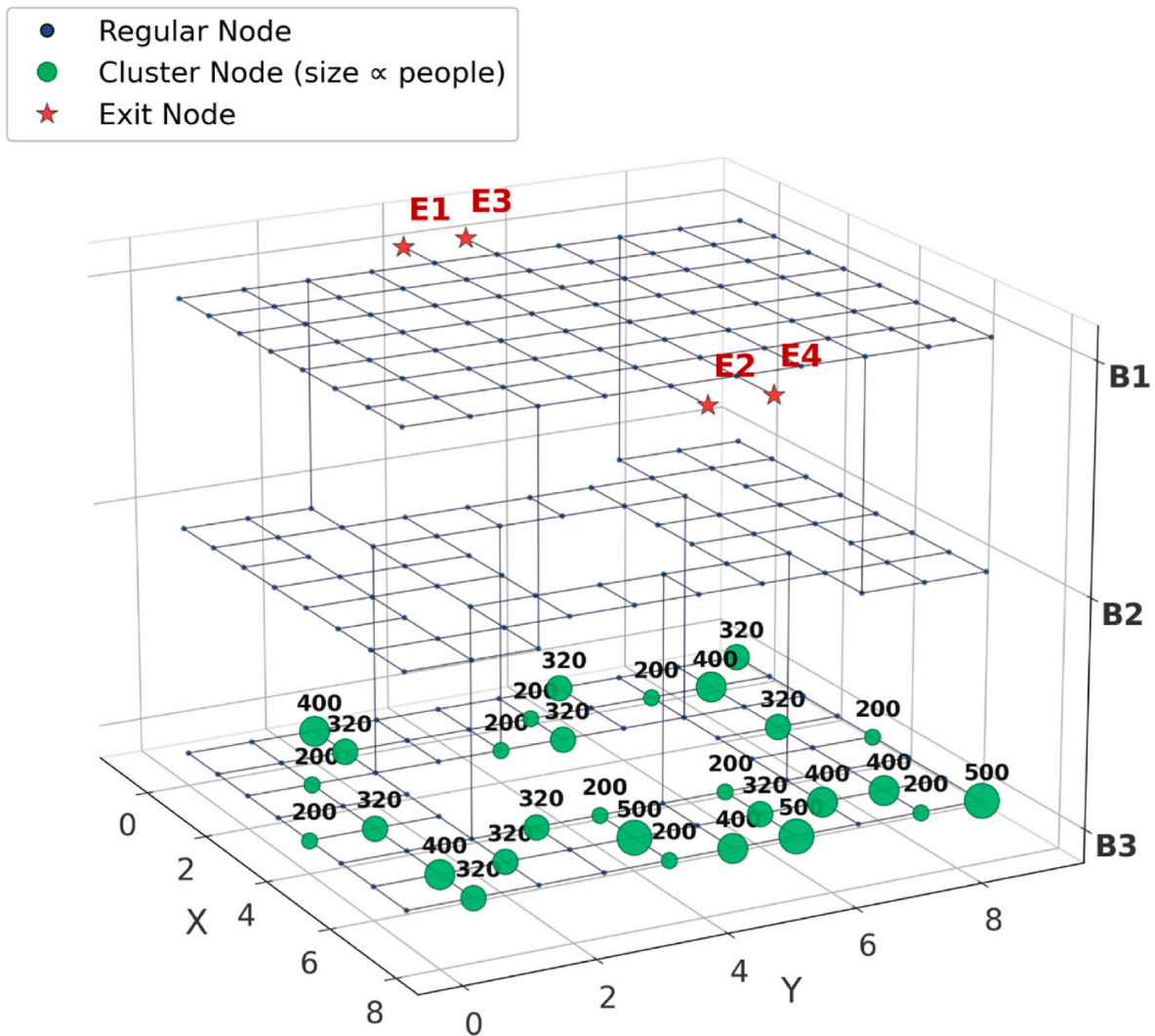


Fig. 4. Occupant distribution across cluster nodes in the underground station

Case 4, where Exit E3 was blocked by fire, both algorithms exhibited intensified concentration at Exit E4, recording utilization rates of 74.9% for the A\* algorithm and 69.2% for Dijkstra's algorithm, which resulted in both algorithms recording the longest maximum evacuation time among the ten cases.

### 5.2. System-optimum results

The Phase 2 simulation was conducted based on system optimum, and the maximum evacuation time by algorithm and exit utilization rate by exit are summarized in Tables 7 and 8, respectively.

In Phase 2, the maximum evacuation time ranged from 545 to 637 s for the A\* algorithm and from 547 to 634 s for Dijkstra's algorithm. In the majority of cases, the difference between the two algorithms was within 1%, indicating nearly identical performance, while in certain cases Dijkstra's algorithm recorded lower values. This indicates a tendency for the performance gap between the two algorithms to narrow considerably or to reverse, in contrast to Phase 1.

Regarding exit utilization rates, the average utilization rates of E1 and E2, which had remained at 22.8% for Dijkstra's algorithm and 24.9% for the A\* algorithm in Phase 1, increased to 39.7% and 36.5%, respectively, in Phase 2. In contrast, the average utilization rates of E3 and E4, which had been 71.7% for Dijkstra's algorithm and 75.04% for the A\* algorithm, decreased to 60.2% and 63.5%, respectively. These figures numerically demonstrate that a dispersion effect had occurred. In

particular, in Case 4 where E3 was blocked, the utilization rate of E4 decreased from 69.2% to 43.5% for Dijkstra's algorithm and from 74.9% to 38.7% for the A\* algorithm, while the combined utilization rate of E1 and E2 increased from 30.7% to 56.8% for Dijkstra's algorithm and from 25.0% to 61.4% for the A\* algorithm. As exit utilization rates became more dispersed in this manner, the maximum evacuation time also decreased from 722 s to 617 s for Dijkstra's algorithm and from 741 s to 637 s for the A\* algorithm, respectively.

### 5.3. Comparative analysis

Table 9 presents a comparative summary of the results of the A\* algorithm and Dijkstra's algorithm across ten fire scenarios, and Table 10 records the execution time and number of explored nodes for each fire scenario to summarize the computational efficiency of the model. First, examining the Phase 2 simulation results with the centralized exit allocation system applied, maximum evacuation time decreased consistently across all cases. The A\* algorithm showed an average reduction of approximately 59.8 s (8.8%), and Dijkstra's algorithm showed an average reduction of approximately 83.3 s (12.4%), demonstrating that proactive congestion mitigation through exit allocation directly contributes to improvements in overall system evacuation efficiency.

Based on this system-level optimization effect, a comparison of algorithm performance revealed a tendency contrary to that observed in

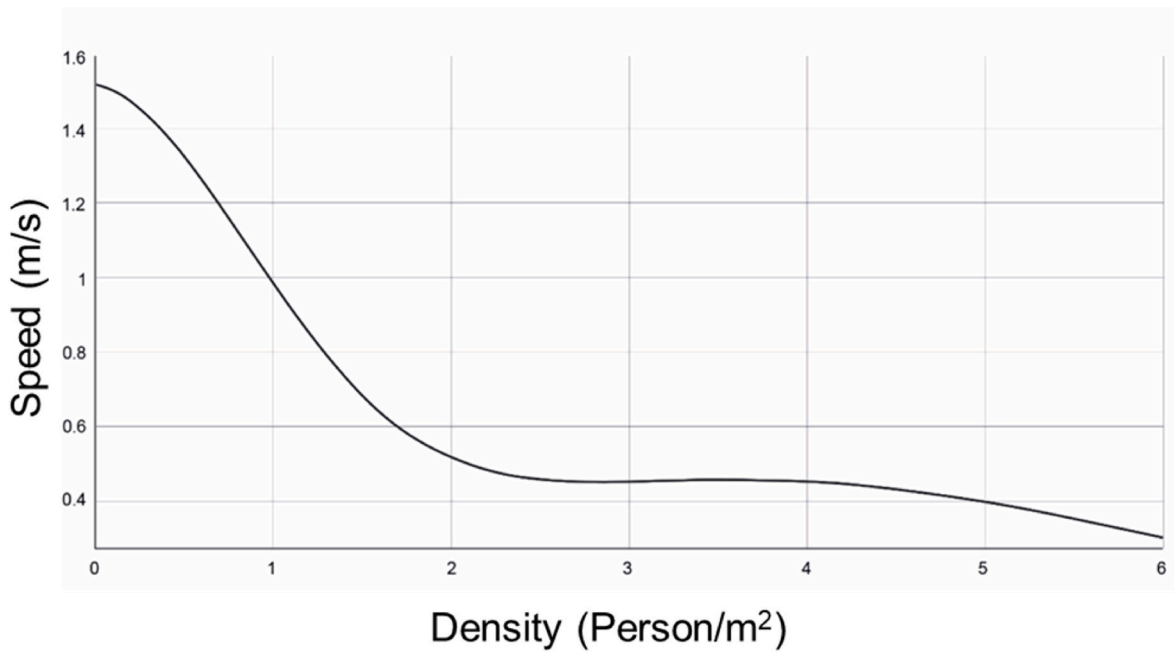


Fig. 5. The fitting curve of the evacuation speed under different density (Zhou et al., 2020).

Table 2

Minimum evacuation capacity per unit Width(m).

| Evacuation Factor                                           | Evacuation Capacity |
|-------------------------------------------------------------|---------------------|
| Horizontal transportation(platform, concourse and corridor) | 80 persons/m•min    |
| Vertical transportation(stairs, stationary escalator)       | 60 persons/m•min    |

Table 3

Walking speed reduction table based on crowd density.

| Evacuation Factor                                           | Population (person) | Multiplier |
|-------------------------------------------------------------|---------------------|------------|
| Horizontal transportation(platform, concourse and corridor) | 640                 | 2          |
| Horizontal transportation(platform, concourse and corridor) | 1,280               | 2.5        |
| Horizontal transportation(platform, concourse and corridor) | 1,920               | 3          |
| Vertical transportation(stairs, stationary escalator)       | 360                 | 2          |
| Vertical transportation(stairs, stationary escalator)       | 720                 | 2.5        |
| Vertical transportation(stairs, stationary escalator)       | 1,080               | 3          |

Phase 1. In the Phase 1 simulation under the user equilibrium mechanism, the A\* algorithm demonstrated consistent superiority in terms of

computational efficiency and maximum evacuation time. The A\* algorithm, employing a heuristic-based search approach, explored an average of approximately 170,000 to 190,000 nodes, achieving more than approximately four times the computational efficiency of Dijkstra's algorithm, and recorded shorter maximum evacuation times in all cases except one. In contrast, in the Phase 2 simulation, route planning was conducted under conditions in which congestion had been proactively mitigated through forced exit allocation based on system optimum, resulting in a tendency for the performance gap between the two algorithms to narrow considerably or to reverse. In particular, in Cases 3, 4, and 8, Dijkstra's algorithm recorded faster evacuation times by 6.16%,

Table 4

Simulation cases based on disaster node locations.

| Case | Disaster Node |
|------|---------------|
| 1    | N242          |
| 2    | N200          |
| 3    | N207          |
| 4    | N270          |
| 5    | N297          |
| 6    | N336          |
| 7    | N300          |
| 8    | N307          |
| 9    | N370          |
| 10   | N397          |

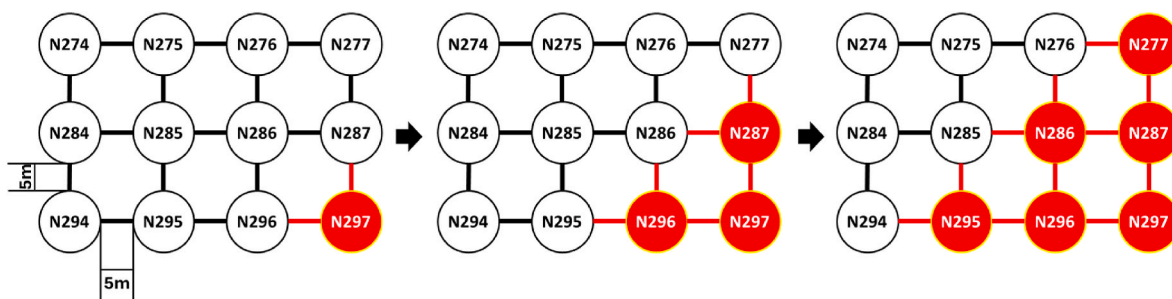


Fig. 6. Example of fire propagation model.

**Table 5**  
Dijkstra's algorithm results.

| Case | Maximum Evacuation time(sec) | Exit Utilization Rate |        |        |        |
|------|------------------------------|-----------------------|--------|--------|--------|
|      |                              | E1 (%)                | E2 (%) | E3 (%) | E4 (%) |
| 1    | 686                          | 6.6                   | 7.0    | 49.5   | 36.9   |
| 2    | 590                          | 14.5                  | 7.0    | 43.7   | 34.7   |
| 3    | 676                          | 14.5                  | 7.0    | 43.7   | 34.7   |
| 4    | 722                          | 23.7                  | 7.0    | 0      | 69.2   |
| 5    | 676                          | 14.5                  | 7.0    | 43.7   | 34.7   |
| 6    | 712                          | 25.9                  | 7.0    | 43.7   | 23.3   |
| 7    | 593                          | 14.5                  | 7.0    | 16.7   | 34.7   |
| 8    | 685                          | 14.5                  | 7.0    | 16.7   | 34.7   |
| 9    | 680                          | 14.5                  | 7.0    | 43.7   | 34.7   |
| 10   | 681                          | 14.5                  | 7.0    | 43.7   | 34.7   |

**Table 6**  
A\* algorithm results.

| Case | Maximum Evacuation time(sec) | Exit Utilization Rate |        |        |        |
|------|------------------------------|-----------------------|--------|--------|--------|
|      |                              | E1 (%)                | E2 (%) | E3 (%) | E4 (%) |
| 1    | 674                          | 6.6                   | 12.7   | 49.5   | 31.2   |
| 2    | 574                          | 12.3                  | 12.7   | 43.7   | 31.2   |
| 3    | 661                          | 12.3                  | 12.7   | 43.7   | 31.2   |
| 4    | 741                          | 12.3                  | 12.7   | 0      | 74.9   |
| 5    | 665                          | 12.3                  | 12.7   | 43.7   | 31.2   |
| 6    | 670                          | 12.3                  | 17.2   | 43.7   | 26.8   |
| 7    | 574                          | 12.3                  | 12.7   | 43.7   | 31.2   |
| 8    | 653                          | 12.3                  | 12.7   | 43.7   | 31.2   |
| 9    | 668                          | 12.3                  | 12.7   | 43.7   | 31.2   |
| 10   | 663                          | 12.3                  | 12.7   | 43.7   | 31.2   |

**Table 7**  
Dijkstra's algorithm results.

| Case | Maximum Evacuation time(sec) | Exit Utilization Rate |        |        |        |
|------|------------------------------|-----------------------|--------|--------|--------|
|      |                              | E1 (%)                | E2 (%) | E3 (%) | E4 (%) |
| 1    | 620                          | 16.7                  | 29.2   | 27.3   | 26.8   |
| 2    | 547                          | 14.5                  | 13.6   | 31.6   | 40.2   |
| 3    | 548                          | 29.5                  | 23.5   | 23.7   | 23.3   |
| 4    | 617                          | 47.3                  | 9.2    | 0      | 43.5   |
| 5    | 619                          | 14.5                  | 11.4   | 33.8   | 40.2   |
| 6    | 591                          | 23.7                  | 23.5   | 27.3   | 25.5   |
| 7    | 547                          | 14.5                  | 13.6   | 31.6   | 40.2   |
| 8    | 548                          | 29.5                  | 23.5   | 23.7   | 23.3   |
| 9    | 634                          | 14.5                  | 19.1   | 31.6   | 34.7   |
| 10   | 597                          | 14.5                  | 11.4   | 33.8   | 40.2   |

**Table 8**  
A\* algorithm results.

| Case | Maximum Evacuation time(sec) | Exit Utilization Rate |        |        |        |
|------|------------------------------|-----------------------|--------|--------|--------|
|      |                              | E1 (%)                | E2 (%) | E3 (%) | E4 (%) |
| 1    | 620                          | 14.5                  | 29.2   | 27.3   | 29.0   |
| 2    | 545                          | 12.3                  | 19.3   | 31.7   | 36.7   |
| 3    | 584                          | 7.9                   | 7.1    | 77.1   | 7.9    |
| 4    | 637                          | 29.5                  | 31.9   | 0      | 38.7   |
| 5    | 618                          | 2.2                   | 29.5   | 31.6   | 36.7   |
| 6    | 587                          | 20.2                  | 24.8   | 36.0   | 18.9   |
| 7    | 545                          | 12.3                  | 19.3   | 31.6   | 36.7   |
| 8    | 582                          | 16.7                  | 16.3   | 59.1   | 7.9    |
| 9    | 629                          | 7.9                   | 32.7   | 28.1   | 31.2   |
| 10   | 598                          | 12.3                  | 19.3   | 31.6   | 36.7   |

3.14%, and 5.84%, respectively.

This outcome stems from differences in the manner in which congestion forms under user optimization-based route selection versus centralized control-based exit allocation. In Phase 1, a user equilibrium-based approach is applied in which each cluster independently selects its

optimal route, and in this process, occupant concentration along identical or similar routes leads to increased congestion. In particular, Dijkstra's algorithm, which performs uninformed search, was more severely affected by this route concentration phenomenon, resulting in intensified congestion. Consequently, the heuristic-based A\* algorithm demonstrated relatively superior performance. In Phase 2, by contrast, exits were proactively distributed through the centralized exit allocation system in a manner that mitigates congestion, and route selection was subsequently performed under these conditions. As a result, route planning was conducted with bottleneck phenomena substantially reduced, and under these conditions Dijkstra's algorithm, which is advantageous for exact optimal solution search, demonstrated superior performance. Therefore, to improve overall evacuation performance, it is important to first mitigate congestion through system-level optimization based on centralized control, prior to improving the performance of individual pathfinding algorithms, and it can be confirmed that an approach that subsequently improves pathfinding algorithm performance is most effective.

#### 5.4. Sensitivity analysis of crowd size

To verify the robustness of the two-phase centralized control system with respect to variations in crowd size, simulations were conducted under the Case 1 condition with crowd sizes set to  $\times 0.7$  (6,370 persons),  $\times 1.0$  (9,100 persons), and  $\times 1.5$  (13,650 persons) relative to the baseline occupant count (9,100 persons), for both the A\* algorithm and Dijkstra's algorithm.

As presented in Table 11, as crowd size increased, the Phase 1 maximum evacuation time increased from 666 s to 753 s for the A\* algorithm and from 669 s to 759 s for Dijkstra's algorithm. This is attributable to the increased frequency of congestion weight occurrence and the intensification of bottleneck phenomena as crowd density rises. However, the evacuation time reduction effect following the application of Phase 2 centralized control appeared consistently across all three crowd size conditions. Reduction rates of 8.0% to 9.8% were recorded for the A\* algorithm and 8.8% to 9.6% for Dijkstra's algorithm, and the variation in reduction rate across crowd sizes was highly limited, remaining within approximately 1 to 2 percentage points.

These results suggest that the effectiveness of the two-phase centralized control system proposed in this study is not dependent on any specific crowd density condition. Provided that crowd density exceeds the level at which congestion weights are substantively generated, the evacuation time reduction effect achieved through centralized control-based exit allocation is stably maintained regardless of the crowd scale. Therefore, the conclusions of this study are not confined to a single density condition and are considered to be universally applicable across a range of crowd size environments.

## 6. Discussion

Unlike prior studies that focused on comparisons of pathfinding performance of a single algorithm or simulations conducted under fixed exit allocation conditions, this study subdivided exit allocation ratios into 100 increments and applied them across ten fire cases to perform dynamic optimal allocation ratio searching, and sequentially structured Phase 1 and Phase 2 to directly and quantitatively compare the performance difference between individual optimum and system optimum under identical network conditions. The algorithm-specific performance derived at each phase and the maximum evacuation time results by phase suggest that system-level flow control should take precedence over individual pathfinding efficiency in the design of evacuation guidance systems.

The situation in Phase 1, in which evacuees each select their own shortest path, corresponds to the user equilibrium state as defined in transportation system theory, and is conceptually equivalent to the user equilibrium state defined by Wardrop and Whitehead (1952). In this

**Table 9**

Comparative analysis of evacuation time and reduction rate by algorithm.

| Case | Algorithm | Phase 1 (s) | Phase 2 (s) | Optimal Ratio (%) | Reduction {s (%)} | Phase 1 Best | Phase 2 Best                |
|------|-----------|-------------|-------------|-------------------|-------------------|--------------|-----------------------------|
| 1    | A*        | 674         | 620         | 35                | 54 (8.0%)         | A*           | Tie ( $\Delta = 0s$ )       |
|      | Dijkstra  | 686         | 620         | 29                | 66 (9.6%)         |              |                             |
| 2    | A*        | 574         | 545         | 34                | 29 (5.1%)         | A*           | A* ( $\Delta = 2s$ )        |
|      | Dijkstra  | 590         | 547         | 35                | 43 (7.3%)         |              |                             |
| 3    | A*        | 661         | 584         | 8                 | 77 (11.6%)        | A*           | Dijkstra ( $\Delta = 36s$ ) |
|      | Dijkstra  | 676         | 548         | 26                | 128 (18.9%)       |              |                             |
| 4    | A*        | 741         | 637         | 29                | 104 (14.0%)       | Dijkstra     | Dijkstra ( $\Delta = 20s$ ) |
|      | Dijkstra  | 722         | 617         | 46                | 105 (14.5%)       |              |                             |
| 5    | A*        | 665         | 618         | 32                | 47 (7.1%)         | A*           | Tie ( $\Delta = 1s$ )       |
|      | Dijkstra  | 676         | 619         | 35                | 57 (8.4%)         |              |                             |
| 6    | A*        | 670         | 587         | 21                | 83 (10.7%)        | A*           | Tie ( $\Delta = 4s$ )       |
|      | Dijkstra  | 712         | 591         | 24                | 121 (16.8%)       |              |                             |
| 7    | A*        | 574         | 545         | 34                | 29 (5.1%)         | A*           | Tie ( $\Delta = 0s$ )       |
|      | Dijkstra  | 593         | 547         | 35                | 46 (7.8%)         |              |                             |
| 8    | A*        | 653         | 582         | 13                | 71 (10.9%)        | A*           | Dijkstra ( $\Delta = 34s$ ) |
|      | Dijkstra  | 685         | 548         | 26                | 137 (20.0%)       |              |                             |
| 9    | A*        | 668         | 629         | 31                | 39 (5.8%)         | A*           | Tie ( $\Delta = 5s$ )       |
|      | Dijkstra  | 680         | 634         | 35                | 46 (6.8%)         |              |                             |
| 10   | A*        | 663         | 598         | 34                | 65 (9.8%)         | A*           | Tie ( $\Delta = 1s$ )       |
|      | Dijkstra  | 681         | 597         | 36                | 84 (12.3%)        |              |                             |

**Table 10**

Comparative analysis of computational efficiency by algorithm.

| Case | Algorithm | Execution Time (s) | Explored Nodes | Time Ratio (A*/D) | Node Ratio (A*/D) |
|------|-----------|--------------------|----------------|-------------------|-------------------|
| 1    | A*        | 5.36               | 170,603        | 0.369             | 0.214             |
|      | Dijkstra  | 6.15               | 796,003        |                   |                   |
| 2    | A*        | 5.06               | 175,216        | 0.422             | 0.220             |
|      | Dijkstra  | 5.50               | 796,562        |                   |                   |
| 3    | A*        | 5.04               | 175,687        | 0.430             | 0.220             |
|      | Dijkstra  | 5.42               | 797,233        |                   |                   |
| 4    | A*        | 5.47               | 192,887        | 0.504             | 0.250             |
|      | Dijkstra  | 5.53               | 772,059        |                   |                   |
| 5    | A*        | 5.05               | 175,887        | 0.393             | 0.221             |
|      | Dijkstra  | 5.58               | 797,327        |                   |                   |
| 6    | A*        | 5.41               | 171,839        | 0.388             | 0.215             |
|      | Dijkstra  | 5.75               | 799,097        |                   |                   |
| 7    | A*        | 5.07               | 174,861        | 0.366             | 0.219             |
|      | Dijkstra  | 5.72               | 796,673        |                   |                   |
| 8    | A*        | 4.89               | 175,304        | 0.406             | 0.220             |
|      | Dijkstra  | 5.29               | 797,249        |                   |                   |
| 9    | A*        | 5.56               | 173,162        | 0.440             | 0.217             |
|      | Dijkstra  | 5.59               | 797,143        |                   |                   |
| 10   | A*        | 5.05               | 175,262        | 0.404             | 0.220             |
|      | Dijkstra  | 5.55               | 797,216        |                   |                   |

state, the A\* algorithm, which employs heuristics to achieve efficient computation and derive approximate solutions to the optimal path, provides efficient routes from the perspective of individual evacuees, and the A\* algorithm consistently recorded lower maximum evacuation times than Dijkstra's algorithm across Phase 1 of all ten cases. However, evacuation guidance based on such user equilibrium results in excessive concentration of occupants on specific primary evacuation routes. In the case of Dijkstra's algorithm, despite the theoretical guarantee of deriving the optimal solution through exhaustive node exploration, the phenomenon of multiple clusters converging on identical or similar optimal routes occurred at a higher frequency compared to the A\* algorithm

**Table 11**

Sensitivity analysis results by occupant scale.

| Crowd Scale                                  | A* Algorithm  |               |                    | Dijkstra's Algorithm |               |                    |
|----------------------------------------------|---------------|---------------|--------------------|----------------------|---------------|--------------------|
|                                              | Phase 1 (sec) | Phase 2 (sec) | Reduction rate (%) | Phase 1 (sec)        | Phase 2 (sec) | Reduction rate (%) |
| <b>Low (<math>\times 0.7</math>) 6,370</b>   | 666           | 604           | 9.3%               | 669                  | 608           | 9.1%               |
| <b>Base (<math>\times 1.0</math>) 9,100</b>  | 674           | 620           | 8.0%               | 686                  | 620           | 9.6%               |
| <b>High (<math>\times 1.5</math>) 13,650</b> | 753           | 679           | 9.8%               | 759                  | 692           | 8.8%               |

when multiple clusters initiated evacuation simultaneously. As a result, greater concentrations of clusters in specific corridors and staircases led to bottleneck phenomena and increased congestion, which constituted a primary cause of delays in overall evacuation time.

This phenomenon is defined in the present study as the "Optimization Paradox." The Optimization Paradox refers to the phenomenon in which higher path optimization performance of a pathfinding algorithm, such as Dijkstra's algorithm, causes a greater number of clusters to converge on identical routes, paradoxically degrading the overall evacuation efficiency of the system. This shares a structurally analogous mechanism with the Braess Paradox in transportation system theory (Braess et al., 2005). The two paradoxes share the same underlying structure in that individually rational choices by users result in systemic inefficiency, and that increased congestion due to route concentration serves as the primary contributing factor. However, the present study's scenario differs from the Braess Paradox in that the cause is not the addition of a new route but rather the optimization performance of the algorithm itself, and in that route assignment is performed by a guidance system rather than through voluntary choice by individual users.

Meanwhile, this inefficiency gap between user equilibrium and system optimum can be quantified using the Price of Anarchy (PoA) concept proposed by Papadimitriou (2001). PoA is defined as the ratio of the cost under user equilibrium to the cost under system optimum, as expressed in Equation (2).

$$PoA = \frac{C(\text{Nash Equilibrium})}{C(\text{System Optimum})} \quad (2)$$

The Price of Anarchy has also been widely used in prior evacuation simulation research. Fries et al. (2022) quantified the performance difference between system equilibrium and selfish equilibrium in evacuation scenarios targeting the Wood River road network in Illinois, USA, using VISSIM simulation. Crociani and Lämmel (2016) proposed a cellular automaton (CA)-based pedestrian simulation model and verified the process by which pedestrian route assignment converges to Nash

Equilibrium or System Optimum through iterative learning cycles, employing the PoA as a quantitative measure of system inefficiency. As demonstrated by these studies, the PoA has been widely utilized as an indicator for quantifying the inefficiency gap between user equilibrium and system optimum, and the present study applies it in the same context.

In this study, the maximum evacuation time of Phase 1 was mapped to the cost under user equilibrium, and the maximum evacuation time of Phase 2 was mapped to the cost under system optimum, and the PoA for each case was computed using Equation (2).

$$PoA = \frac{T_{Phase1}}{T_{Phase2}} \quad (3)$$

The computed PoA values averaged 1.10 for the A\* algorithm and 1.14 for Dijkstra's algorithm, indicating that individual optimum-based route guidance induces inefficiencies of approximately 10% and 14%, respectively, relative to system optimum.

Phase 2 of the two-phase centralized exit allocation system proposed in this study is implemented by forcibly adjusting exit allocation ratios to proactively disperse occupant concentration, thereby resolving the bottleneck phenomena that occurred in Phase 1. Maximum evacuation time decreased consistently across all ten cases following the application of Phase 2, with average reductions of approximately 59.8 s (8.8%) for the A\* algorithm and approximately 83.3 s (12.4%) for Dijkstra's algorithm. These figures numerically confirm that the autonomous route selection of individual evacuees in Phase 1 induced occupant concentration at specific exits, and that exits were not sufficiently utilized in a distributed manner. In particular, in Cases 3 and 8, Dijkstra's algorithm recorded a Phase 2 maximum evacuation time of 548 s, outperforming the A\* algorithm's 584 s and 582 s by 36 s and 34 s, respectively, and in Case 4, Dijkstra's algorithm also demonstrated a 20-s advantage with 617 s compared to the A\* algorithm's 637 s. This is attributable to the fact that in an environment where bottlenecks were mitigated through forced exit allocation adjustment under system optimization, the path-finding performance of Dijkstra's algorithm, which guarantees the optimal solution through uninformed search, manifested substantively. The inefficient route concentration under Phase 1 conditions was mitigated in the Phase 2 environment predicated on system optimization, resulting in improved overall evacuation efficiency, and it is numerically confirmed that the sum of individual optimal solutions does not always converge to the globally optimal solution.

In terms of computational efficiency, the two algorithms exhibit a clear difference. Based on the virtual station model used in this study (3 floors, 29 clusters), the total model execution time and number of explored nodes including both Phase 1 and Phase 2 were, on average, 5.20 s and approximately 175,000 nodes for the A\* algorithm, compared to 5.61 s and approximately 797,000 nodes for Dijkstra's algorithm a difference of 0.41 s in computation time and approximately 4.5 times in the number of explored nodes. This is considered to be at a level applicable to real-world use in the event of an actual fire. Furthermore, this study was conducted on a general desktop environment (Intel Core i5-13400, 8 GB RAM), and computation time could be further reduced if server-grade computers are utilized in an actual operational environment.

However, since the simulations in this study were conducted on a small-scale virtual station, computation time may increase with the growth in the number of nodes and links when applied to an actual large-scale station. Investigation into computational efficiency improvements and the feasibility of application to large-scale networks is left as a subject for future research.

These findings suggest that improving the performance of individual pathfinding algorithms alone has limited effect on enhancing overall system evacuation efficiency, and that the introduction of a centralized control system that proactively adjusts occupant allocation by exit must be prioritized. Furthermore, it can be confirmed that the criteria for algorithm selection may vary depending on system operating

conditions. In environments where only individual route planning is available without centralized control, the A\* algorithm employing a heuristic-based search approach is advantageous. However, in environments where centralized exit allocation is presupposed, Dijkstra's algorithm employing an uninformed search approach can demonstrate superior evacuation performance. Moreover, the exit allocation approach using 100 cases applied in this study does not rely on empirical configuration but derives real-time optimal allocation ratios based on simulation data, confirming that it constitutes a universally applicable approach across diverse station configurations and fire scenarios.

However, this study has additional limitations. The fire propagation model was simplified such that factors including visibility reduction due to smoke and temperature variation over time were not reflected in route weights. In addition, the congestion model considers only speed reduction as a function of occupant density, and psychological and physical factors that arise in actual crowd behavior, such as panic responses, herding behavior, and velocity interference between clusters at flow intersections, were not sufficiently incorporated. Furthermore, injured or mobility-impaired individuals at the fire scene were not included in the model, and consequently secondary bottlenecks and evacuation flow disruptions caused by such individuals were not accounted for. Finally, the spatial model used for simulation in this study represents a small-to-medium-sized underground structure with three basement levels, and if the model were implemented based on an actual large-scale station, Dijkstra's algorithm, which explores more than four times as many nodes as the A\* algorithm, may require substantially greater computational cost and time, necessitating further research on this matter. Additionally, since the optimal allocation ratio differed between the two algorithms in Phase 2, the effects of algorithm choice and allocation strategy are not fully isolated; future work employing a controlled, identical allocation ratio across both algorithms would more rigorously disentangle these two effects. Regarding the exclusion of smoke visibility noted above, its likely effect would be to intensify congestion under UE conditions more than under SO, since the SO framework proactively disperses occupants before congestion builds. The UE–SO performance gap would therefore be expected to widen rather than narrow, suggesting that the conclusions of this study remain qualitatively robust.

Accordingly, future research should advance the simulation framework toward a more realistic representation by integrating fire simulation programs such as FDS to reflect walking speed reduction due to smoke propagation and visibility degradation, incorporating psychological and physical factors such as panic states of evacuating clusters and velocity interference at crossing flows, integrating dynamic fire propagation models with real-world crowd behavior data, and adding scenarios that account for mobility-impaired individuals. Furthermore, combining dynamic route replanning functionality linked to real-time sensor data would substantially enhance the practical applicability of the two-phase system proposed in this study.

## 7. Conclusion

This study proposed a two-phase centralized evacuation guidance model structured sequentially around user equilibrium and system optimum for fire evacuation in underground stations, and quantitatively compared the performance of the A\* algorithm and Dijkstra's algorithm under both conditions. Simulation results across ten fire scenarios demonstrated that under the Phase 1 user equilibrium condition, the A\* algorithm consistently outperformed Dijkstra's algorithm in both computational efficiency and maximum evacuation time, whereas under the Phase 2 system optimum condition, the maximum evacuation times of both algorithms converged to comparable levels or Dijkstra's algorithm demonstrated superiority in certain cases as exit allocation was forcibly adjusted. These results numerically confirmed that system-level exit allocation strategy exerts a more decisive influence on overall evacuation efficiency than the optimization performance of individual

pathfinding algorithms.

The user equilibrium and system optimum conditions, which had previously been addressed independently in prior studies, were sequentially structured within a single simulation and directly compared under identical network conditions. The bottleneck mitigation effect was quantitatively identified through a dynamic optimal allocation ratio search approach in which exit allocation ratios were subdivided into 1% increments. These findings suggest that system-level flow control should be prioritized over algorithm performance optimization in the design of underground station evacuation guidance systems, and it is anticipated that the results can serve as design criteria for the development of future centralized evacuation guidance systems.

As this study is a simulation-based investigation targeting a virtual underground station, additional validation across large-scale transfer stations and multiple real-world stations is required for the generalization of the findings. Future research plans to apply the two-phase centralized control framework proposed in this study to real station environments and to verify practical applicability through extension to dynamic route replanning linked to real-time sensor data and AI-based intelligent evacuation guidance systems.

### CRedit authorship contribution statement

**Sunnie Haam:** Software, Writing – original draft. **Chiyeong Kang:** Methodology, Writing – original draft. **Hyunseok Kim:** Conceptualization, Formal analysis. **Mintaek Yoo:** Resources, Supervision, Writing – review & editing.

### Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

### Acknowledgements

**Funding:** This research was supported by the Korea Agency for Infrastructure Technology Advancement (KAIA) grant funded by the Ministry of Land, Infrastructure and Transport (Grant No. RS-2023-00238458) and by the 2026 Research Fund of the University of Seoul for Sunnie Haam. We appreciate the kind of support provided.

### Data availability

Data will be made available on request.

### References

- Braess, D., Nagurney, A., Wakolbinger, T., 2005. On a paradox of traffic planning. *Transp. Sci.* 39 (4), 446–450. <https://doi.org/10.1287/trsc.1050.0127>.
- Burnett, J., 1984. Fire safety concerns for rail rapid transit systems. *Fire Saf. J.* 8 (1), 3–7. [https://doi.org/10.1016/0379-7112\(84\)90048-1](https://doi.org/10.1016/0379-7112(84)90048-1).
- Chow, A.H., 2007. *User Equilibrium, System Optimum, and Externalities in Time-dependent Road Networks*. University College London.
- Crociani, L., Lämmel, G., 2016. Multidestination pedestrian flows in equilibrium: a cellular automaton-based approach. *Comput. Aided Civ. Infrastruct. Eng.* 31 (6), 432–448. <https://doi.org/10.1111/mice.12209>.
- Dijkstra, E.W., 1959. A note on two problems in connexion with graphs. *Numer. Math.* 1 (1), 269–271. <https://doi.org/10.1007/BF01386390>.
- Fries, R., Vennu, R., Qi, Y., Raj Sah, U., 2022. A microsimulation-based analysis of the price of anarchy in transportation networks during an evacuation. In: *International Conference on Transportation and Development 2022*, pp. 158–167. <https://doi.org/10.1061/9780784484326.016>.
- Gonzales, S.H., Vega-Huerta, H., Maquen-Niño, G.L.E., De-la-Cruz-VdV, P., Moquillaza-Henríquez, S.D., Benito-Pacheco, O., et al., 2025. Optimized earthquake evacuation routes using Dijkstra's Algorithm and WinQSB. <https://doi.org/10.3844/jcssp.2025.2349.2360>.
- Haam, S., Kim, H., Yoo, M., Song, W.S., 2025. Optimal additional evacuation route analysis in deep underground station structures using Dijkstra's algorithm. *Developments in the Built Environment*, 100784. <https://doi.org/10.1016/j.dibe.2025.100784>.
- Huang, P., Lin, X., Liu, C., Fu, L., Yu, L., 2024. A real-time automatic fire emergency evacuation route selection model based on decision-making processes of pedestrians. *Saf. Sci.* 169, 106332. <https://doi.org/10.1016/j.ssci.2023.106332>.
- Jiang, Y., Liu, G., Du, Y., Cai, S., Si, Z., He, J., Zhou, X., 2025. Analysis and optimization of fire evacuation safety performance in large urban complexes. *PLoS One* 20 (11), e0336964. <https://doi.org/10.1371/journal.pone.0336964>.
- Kim, H., Haam, S., Yoo, M., Song, W.S., 2026. Algorithmic evaluation of fire evacuation efficiency under dynamic crowd and smoke conditions. *Fire* 9 (1), 32. <https://doi.org/10.3390/fire9010032>.
- Li, X., Chen, R., Zhu, Y., Jim, C.Y., 2024. Emergency fire evacuation simulation of underground commercial street. *Simulat. Model. Pract. Theor.* 134, 102929. <https://doi.org/10.1016/j.simpat.2024.102929>.
- Li, Z.Y., Tang, M., Liang, D., Zhao, Z., 2016. Numerical simulation of evacuation in a subway station. *Procedia Eng.* 135, 616–621. <https://doi.org/10.1016/j.proeng.2016.01.126>.
- Lischka, A., Kulcsár, B., 2026. GREAT-EER: graph edge attention network for emergency evacuation responses. *arXiv preprint arXiv:2602.14676*. <https://doi.org/10.48550/arXiv.2602.14676>.
- Mas, E., Moya, L., Gonzales, E., Koshimura, S., 2024. Reinforcement learning-based tsunami evacuation guidance system. *Int. J. Disaster Risk Reduct.* 115, 105023. <https://doi.org/10.1016/j.ijdrr.2024.105023>.
- Matsui, Y., Yoshihiro, T., 2024. A flow-based centralized route guidance system for traffic congestion mitigation. *IEEE Trans. Intell. Transport. Syst.* 25 (8), 8681–8694. <https://doi.org/10.1109/TITS.2024.3375267>.
- Ministry of Land, Infrastructure and Transport (MOLIT), 2022. *Design Guidelines for Urban Railway Stations and Transfer & Convenience Facilities (MOLIT Notification No. 2022-78)*. Sejong, South Korea [In Korean].
- Mirahadi, F., McCabe, B.Y., 2021. EvacuSafe: a real-time model for building evacuation based on Dijkstra's algorithm. *J. Build. Eng.* 34, 101687. <https://doi.org/10.1016/j.jobbe.2020.101687>.
- Papadimitriou, C., 2001. Algorithms, games, and the internet. In: *Proceedings of the thirty-third Annual ACM Symposium on Theory of Computing*, pp. 749–753. <https://doi.org/10.1145/380752.380883>.
- Schmidt-Poloczyk, N., 2024. Full-scale experimental study on influence of smoke on pedestrian movement during evacuation in road tunnel. *Buildings* 14 (11), 3654. <https://doi.org/10.3390/buildings14113654>.
- Tan, J.K.N., Zhang, S., Law, A.W.K., Cheung, S.H., 2025. Digital-twin enabled evacuation to improve individual and community resilience of building occupants against indoor fires: framework and route-finding algorithm. *Dev. Built Environ.* 22, 100672. <https://doi.org/10.1016/j.dibe.2025.100672>.
- Tang, Q., Qu, Y., Sun, H., Li, X., Wu, J., 2025. Flexible crowd evacuation optimization in public places: a cell-based approach considering adjustable facilities. *Transport. Res. E Logist. Transport. Rev.* 201, 104222. <https://doi.org/10.1016/j.tre.2025.104222>.
- Tian, T., Wu, H., Wei, H., Wu, F., Xu, M., 2025. An efficient route planning algorithm for special vehicles with large-scale road network data. *ISPRS Int. J. GeoInf.* 14 (2), 71. <https://doi.org/10.3390/ijgi14020071>.
- Tsukahara, M., Koshiba, Y., Ohtani, H., 2011. Effectiveness of downward evacuation in a large-scale subway fire using fire dynamics simulator. *Tunn. Undergr. Space Technol.* 26 (4), 573–581. <https://doi.org/10.1016/j.tust.2011.02.002>.
- UITP, 2025. *Public Transport: Global Urban Mobility Data*. UITP News. <https://www.uitp.org/news/public-transport-global-urban-mobility-data/>.
- Wardrop, J.G., Whitehead, J.L., 1952. Correspondence. *some theoretical aspects of road traffic research*. *Proc. Inst. Civ. Eng.* 1 (5), 767–768.
- Wei, X., Zhang, Y., Song, H., Qin, H., Zhao, G., 2024. Research on evacuation path planning based on improved sparrow search algorithm. *Comput. Model. Eng. Sci.* 139 (2). <https://doi.org/10.32604/cmescs.2023.045096>.
- Xie, C.Z.T., Chen, Q., Zhu, B., Lee, E.W.M., Tang, T.Q., Yin, X., Yuan, Z., Zhang, B., 2025. Coordinating dynamic signage for evacuation guidance: a multi-agent reinforcement learning approach integrating mesoscopic crowd modeling and fire propagation. *Chaos Solitons Fractals* 194, 116246. <https://doi.org/10.1016/j.chaos.2025.116246>.
- Yan, Y., 2023. Research on the A star algorithm for finding shortest path. *Highlights in science. Eng. Technol.* 46, 154–161.
- Yang, X., Zhang, R., Li, Y., Pan, F., 2022. Passenger evacuation path planning in subway station under multiple fires based on multiobjective robust optimization. *IEEE Trans. Intell. Transport. Syst.* 23 (11), 21915–21931. <https://doi.org/10.1109/TITS.2022.3190291>.
- Zhang, B., Lo, J.T., Fang, H., Xie, C., Tang, T., Lo, S., 2024a. Coupled simulation-optimization model for pedestrian evacuation guidance planning. *Simulat. Model. Pract. Theor.* 134, 102922. <https://doi.org/10.1016/j.simpat.2024.102922>.
- Zhang, B., Lo, J.T., Fang, H., Xie, C., Tang, T., Lo, S., 2024b. Directed rooted forest based direction setting method: a step toward automated dynamic exit signs. *J. Build. Eng.* 85, 108504. <https://doi.org/10.1016/j.jobbe.2024.108504>.
- Zhang, L., Wu, X., Liu, M., Liu, W., Ashuri, B., 2019. Discovering worst fire scenarios in subway stations: a simulation approach. *Autom. Construct.* 99, 183–196. <https://doi.org/10.1016/j.autcon.2018.12.007>.
- Zhao, H., Schwabe, A., Schläfli, F., Thrash, T., Aguilar, L., Dubey, R.K., et al., 2022. Fire evacuation supported by centralized and decentralized visual guidance systems. *Saf. Sci.* 145, 105451. <https://doi.org/10.1016/j.ssci.2021.105451>.

- Zhou, J., Li, S., Nie, G., Fan, X., Xia, C., 2020. Developing a revised social force model for pedestrians' earthquake emergency evacuation. *Geomatics. Nat. Hazards Risk* 11 (1), 335–356. <https://doi.org/10.1080/19475705.2020.1724202>.
- Zhou, M., Dong, H., Ioannou, P.A., Wang, F.Y., 2022. Crowd evacuation with multi-modal cooperative guidance in subway stations: computational experiments and optimization. *IEEE Trans. Comput. Social Syst.* 10 (5), 2536–2545. <https://doi.org/10.1109/TCSS.2022.3188135>.
- Zhu, Y., Zhou, Z., Zhou, J., Xu, X., Wu, X., Nie, W., 2025. An improved 3D Dijkstra algorithm of evacuation route considering tailings dam failure. *Nat. Hazards* 121 (3), 2483–2505. <https://doi.org/10.1007/s11069-024-06883-9>.