Smart and Resilient Transportation

Path planning algorithm for mobile robots in high-speed railway waiting halls under boarding-gate congestion

Wenhui Liu, Li Wang and Xiaoran Zhang
School of Traffic and Transportation, Beijing Jiaotong University, Beijing, China

Received 17 April 2026
Revised 11 June 2026
Accepted 7 July 2026

Abstract

Purpose – This paper aims to address the path-planning problem of a single mobile robot performing transport tasks in a high-speed railway waiting hall under boarding-gate congestion. Unlike general indoor navigation, this scenario involves schedule-driven localized congestion, semantic differences between target and non-target gates and a strict pre-departure deadline. The study seeks to develop a path-planning method that can improve execution efficiency, reduce unnecessary exposure to irrelevant congestion areas and better satisfy the operational requirements of in-station transport tasks.

Design/methodology/approach – A scenario-adapted path-planning method based on time-space A* is proposed. First, a gate-centered time-varying congestion model is constructed by integrating train departure schedules, ticket-checking time windows and spatial decay. Then, a task-oriented semantic mapping strategy is introduced to distinguish necessary approach to the target gate from avoidable exposure to non-target gates. Finally, a search framework combining time-varying traversal cost, semantic exposure penalty and deadline-aware pruning is developed for single-robot transport tasks in waiting halls.

Findings – Experimental results in normal, peak and dense scenarios with 30 random seeds show that the proposed method consistently achieves 100% on-time completion and feasible-path rates. Compared with static shortest-path methods, it reduces unnecessary exposure and improves execution efficiency. Compared with RRT*, it shows better feasibility and stability under deadline constraints. Compared with D* Lite, it achieves lower SimTime, much lower exposure and lower planning cost. The findings confirm the effectiveness of task-oriented time-space A* adaptation for predictable localized congestion in waiting halls.

Research limitations/implications – This study is limited to a single-robot, single-task setting in a simulated high-speed railway waiting-hall environment. Multi-robot coordination, task allocation, elevator resource competition and real-time sensing feedback are beyond the current scope. The congestion model is mainly timetable-driven and assumes known station layout, gate positions and train-to-gate assignments. These limitations suggest that future research should incorporate online perception, dynamic updates and more realistic deployment conditions to further strengthen the applicability of the proposed method.

Practical implications – The proposed method provides a practical planning framework for intelligent in-station transport tasks such as parcel transfer, small-item delivery and other short-distance logistics operations in high-speed railway waiting halls. By reducing unnecessary traversal through irrelevant congestion hotspots and improving deadline compliance, the method can enhance operational efficiency and planning quality in congestion-sensitive station environments. It also offers a useful reference for integrating timetable information and localized congestion priors into practical robotic navigation systems deployed in smart-station scenarios.

Funding: Systematic Major Project of China Railway Science and Technology Development plan no. P2025X001.

Smart and Resilient Transportation
Emerald Publishing Limited
e-ISSN: 2632-0495
p-ISSN: 2632-0487
DOI 10.1108/SRT-04-2026-0010

Social implications – This study contributes to the development of safer, more efficient and more intelligent railway-station service systems. Improved robot navigation under boarding-gate congestion can help reduce operational disruption, support more reliable in-station logistics and promote the coordinated use of robotic systems in public transport hubs. In the longer term, such methods may support smarter passenger-service infrastructure, improve service quality in busy railway environments and facilitate the digital transformation of high-speed railway stations toward more resilient and intelligent operation.

Originality/value – The originality of this study lies in adapting time-space A* to the specific operational characteristics of high-speed railway waiting halls. Unlike generic indoor navigation methods, the proposed approach explicitly models gate-centered timetable-driven congestion, distinguishes the semantic roles of target and non-target gates and integrates deadline-aware pruning into the search process. The study therefore provides a task-oriented path-planning framework tailored to waiting-hall transport tasks, offering both methodological value for congestion-aware robot navigation and practical value for intelligent railway-station applications.

Keywords High-speed railway station, Time-space A*, Path planning, Boarding-gate congestion, Semantic cost mapping

Paper type Technical paper

1. Introduction

1.1 Research background

With the development of high-speed rail networks and smart-station systems, high-speed railway stations are becoming integrated hubs for passenger service, intelligent operation and in-station logistics. Mobile robots have growing potential for short-distance transport tasks such as baggage delivery, parcel transfer and small-item distribution in waiting halls, offering better continuity, traceability and integration with station management platforms.

However, unlike warehouses or factories, high-speed railway waiting halls are open public spaces shared by passengers and robots. During ticket-checking periods, passengers gather around boarding gates and adjacent corridors, forming localized congestion that may increase robot travel time, reduce schedule slack and cause unnecessary exposure to unrelated passenger hotspots.

For the task considered in this paper, a robot must travel from an entrance or elevator-hall area to the elevator point behind a designated boarding gate before train departure. Therefore, path planning in this scenario is not simply a geometric shortest-path problem, but must jointly consider time-varying congestion, task-oriented gate semantics and deadline feasibility. This motivates a dedicated planning method for congestion-sensitive high-speed railway waiting halls.

1.2 Literature review

Dynamic path planning has been extensively studied in robotics. Classical graph-search methods such as Dijkstra and A* are effective in static environments, while dynamic and time-dependent extensions improve adaptability to changing obstacles and traversal costs. Recent studies have further moved path planning from purely geometric shortest-path search toward predictive, context-aware and uncertainty-aware navigation in dynamic environments (AbuJabal *et al.*, 2024; Lashin *et al.*, 2025). However, most of these methods are developed for generic indoor scenes or moving-obstacle settings, rather than for timetable-driven congestion anchored at fixed service facilities such as boarding gates in railway stations.

Another important research stream concerns robot navigation in crowded public spaces. Existing studies have shown that navigation in human-populated environments should consider not only collision avoidance, but also social compliance, interaction risk, evaluation

criteria and scenario complexity (Mavrogiannis *et al.*, 2023; Yu *et al.*, 2021).Nevertheless, most current crowd-aware and social-navigation methods treat dense areas as uniformly unfavorable and rarely distinguish between task-relevant congestion and task-irrelevant congestion. In the scenario studied in this paper, congestion around the target gate is necessary to approach because it is associated with task completion, whereas congestion around non-target gates should preferably be avoided. This task-oriented asymmetry remains insufficiently addressed in the existing literature.

A third related stream focuses on passenger-flow analysis and congestion modeling in rail transit hubs. Recent studies have shown that congestion in large stations is strongly associated with timetable organization, bottleneck distribution, passenger accumulation, queueing effects and node-level interactions (Zhao and Xu, 2026; Wei *et al.*, 2023). Researchers have investigated congestion propagation, station bottleneck mitigation, queueing-network-based control, spatial-temporal passenger-flow distribution and passenger-flow prediction using both simulation and learning-based methods (Zhao and Xu, 2026; Wei *et al.*, 2023). These studies provide important insight into where and when congestion emerges in rail hubs, but they mainly serve passenger organization, demand prediction and operational control rather than robot path planning. As a result, the structured congestion knowledge identified in rail-transit studies has not yet been sufficiently transformed into a time-varying cost representation for task-level robotic navigation.

Overall, three gaps remain: limited attention to gate-centered congestion, lack of semantic distinction between target and non-target gates and few works integrating deadline constraints into graph search. These gaps motivate our research.

1.3 Main contributions

The main contributions of this paper are as follows:

- A timetable-driven boarding-gate congestion field is constructed for high-speed railway waiting halls. By linking train departure schedules, ticket-checking time windows and gate-centered spatial decay, the model converts predictable passenger accumulation into time-varying traversal costs for robot path planning. Different from generic time-dependent edge-cost models, the dynamic costs in this study are generated from railway operating schedules.
- A congestion-aware time-space A*-type planning framework is developed by embedding the proposed congestion field into the edge-cost update process. The method adapts time-space A* to the gate-congestion characteristics of railway waiting halls rather than directly applying conventional A* or generic time-dependent search.
- A task-oriented adjustment mechanism is introduced by distinguishing target-gate and non-target-gate congestion and integrating deadline-aware pruning. This design reduces unnecessary exposure to task-irrelevant gate congestion while preserving destination accessibility and satisfying the pre-departure arrival constraint.

2. Scenario description and problem formulation

2.1 Scenario description and task definition

This study investigates mobile-robot path planning for transport tasks in a high-speed railway station waiting hall. The hall layout approximates a realistic concourse, including an entrance or elevator-hall area, waiting-seat islands, guiding obstacles, two rows of boarding

gates and elevator points behind each gate. Robots move entirely within publicly accessible circulation spaces.

Under a given task set, each task is planned and evaluated independently. For any task r, the task is jointly determined by a start point s_r, a departure time t_r^0, a target gate g_r and the departure time t_r^d of the train associated with that gate. The destination of the task is not the gate itself, but the unique elevator point bound to the target gate, denoted by q_r. The start point is selected from several candidate locations near the entrance or elevator-hall area to simulate transport demands originating from different station service locations. In addition, the departure time of each task is generated only within the effective service window before train departure; tasks arising after the departure-induced dissipation period are not considered.

Unlike conventional indoor planning, the main challenge is not static obstacle avoidance but localized congestion from ticket-checking. During pre-departure boarding, passengers form time-varying hotspots in front of gates and along nearby corridors, which reduce traversal efficiency. The goal is therefore to generate feasible paths that align with temporal gate congestion while satisfying task deadlines. Figure 1 shows the semantic grid map of the hall, including entrance areas, waiting-seat islands, boarding gates and elevator points.

2.2 Problem characteristics and research scope

The path-planning problem in this paper has three key characteristics. First, path cost is explicitly time-dependent: congestion around boarding gates changes with departure time, so the optimal path varies over time. Second, task semantics are important: the robot must reach the target gate and its elevator point, while avoiding non-target gates, which play an asymmetric role. Third, strict temporal deadlines apply: the robot must reach the destination no later than 3 min before train departure, a hard constraint.

To focus on the core impact of gate congestion, this study considers a single robot performing a single task. Multi-robot coordination, task allocation and elevator resource competition are excluded. The waiting hall layout, gate locations and train-to-gate assignments are assumed known. Gate congestion is modeled as a localized reduction in traversal efficiency, not an impassable barrier. Under these assumptions, the problem is: given a start point, departure time, target gate and train schedule, how can time-varying gate congestion be exploited to generate a suitable path.

2.3 Problem formulation

The waiting hall is discretized into a semantic grid map and further abstracted as a search graph:

$$G = (V, E), \tag{1}$$

where V denotes the set of traversable nodes and E denotes the set of feasible edges between adjacent nodes. For any edge $e \in E$, let $\ell(e)$ denote its geometric length. Under uncongested conditions, the nominal traversal time of edge e is defined as follows:

$$\tau_0(e) = \frac{\ell(e)}{v}, \tag{2}$$

where v is the nominal moving speed of the robot.

For any task r, let the destination elevator point associated with the target gate g_r be q_r. A candidate path can then be represented as follows:

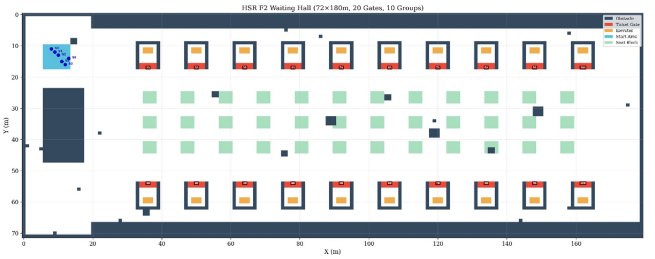


Figure 1. Semantic grid map of the high-speed railway waiting hall

$$P_r = \langle v_0, v_1, \ldots, v_n \rangle, \tag{3}$$

where $v_0 = s_r$, $v_n = q_r$ and each consecutive pair (v_k, v_{k+1}) corresponds to a feasible edge $e_k \in E$. Since congestion near the gates varies over time, the actual traversal cost of an edge is no longer constant. Let t_k denote the time at which the robot starts traversing edge e_k. Then the actual traversal time of e_k under task r is written as follows:

$$\tau_r(e_k, t_k) = \tau_0(e_k) + \Delta\tau_r(e_k, t_k), \tag{4}$$

where $\Delta\tau_r(e_k, t_k)$ is the additional time cost caused by local gate-induced congestion. If the edge passes through hotspot regions related to non-target gates, an additional exposure penalty may also be incurred to characterize the undesirability of crossing irrelevant congested areas.

Accordingly, the expected travel time of path P_r is defined as follows:

$$T(P_r) = \sum_{k=0}^{n-1} \tau_r(e_k, t_k). \tag{5}$$

To assess the degree to which the path traverses congestion around non-target gates, the non-target-gate exposure of the path is defined as follows:

$$X(P_r) = \sum_{k=0}^{n-1} \xi_r(e_k, t_k), \tag{6}$$

where $\xi_r(e_k, t_k)$ denotes the exposure intensity of edge e_k to hotspot regions associated with non-target gates at traversal time t_k. This quantity does not replace travel time as the primary execution metric, but provides an additional measure for evaluating whether the planned path unnecessarily enters irrelevant congested areas.

For task r, the generated path must first satisfy the deadline constraint:

$$t_r^0 + T(P_r) \le t_r^d - 3\,\text{min}. \tag{7}$$

That is, the robot must arrive at the destination elevator point at least 3 min before the departure of the associated train. Within the set of feasible paths satisfying this temporal

constraint, the objective is to obtain a path with lower expected travel time and lower non-target-gate exposure. Therefore, the problem can be stated as follows:

Given the start point s_r, departure time t_r^0, target gate g_r and train departure time t_r^d, determine an optimal feasible path from s_r to q_r such that the path satisfies the pre-departure arrival constraint while minimizing expected travel time and reducing unnecessary exposure to congestion around non-target gates.

It should be noted that, although the experiments in later chapters are organized in the form of task sets, the planning and simulation procedures are still carried out independently for each task rather than through joint assignment or global scheduling over multiple tasks. The above formulation therefore serves as the foundation for the subsequent modeling of time-varying gate congestion and the development of the proposed path-planning method.

3. Time-varying modeling of boarding-gate congestion

3.1 Modeling framework

Congestion in high-speed railway waiting halls arises from ticket-checking processes before train departure. Passengers gradually gather at boarding gates and adjacent corridors, creating localized high-density regions that reduce traversal efficiency but do not form impassable obstacles. In this study, gate congestion is modeled as a time-varying traversal cost rather than a hard barrier.

A three-layer framework is adopted: temporal evolution, spatial decay and task relevance. Temporal evolution follows train schedules and ticket-checking patterns; spatial decay captures crowd diffusion around gates; task relevance distinguishes target from non-target gates. This framework transforms localized congestion into a time-dependent, task-aware cost suitable for path planning.

The model is designed under the assumptions established in Section 2. Specifically, the station layout, gate positions and train-to-gate assignments are known in advance; the robot performs a single task at a time; and only the pre-departure congestion buildup is considered, while the post-departure dissipation process is ignored. Under these assumptions, the purpose of this chapter is to construct a gate-centered spatiotemporal congestion field that can be directly embedded into the search process of the path-planning algorithm.

3.2 Temporal modeling of congestion based on train schedules

Let $\mathcal{M}$ denote the set of trains in the study period and G denote the set of boarding gates. For any train $m \in \mathcal{M}$, let t_m^d denote its departure time and $g(m) \in G$ the boarding gate serving that train. Because the tasks considered in this paper all correspond to pre-departure service demands, only the congestion growth process before departure is modeled. The dissipation of passenger flow after departure is not included.

For a gate g, the temporal congestion intensity at time t is denoted by $C_g(t)$. To describe the fact that congestion gradually strengthens before departure, a normalized pre-departure time-window function is introduced. For a single train m assigned to gate g, let t_m^s be the time when noticeable congestion begins to emerge and let t_m^p be the time when local congestion enters a high-intensity stage. Then the congestion contribution of train m to gate g at time t is defined as follows:

$$C_{g,m}(t) = \begin{cases} 0, & t < t_m^s, \\ \left(\frac{t - t_m^s}{t_m^p - t_m^s}\right)^{\kappa}, & t_m^s \le t < t_m^p, \\ 1, & t_m^p \le t \le t_m^d, \\ 0, & t > t_m^d, \end{cases} \tag{8}$$

where $\kappa > 0$ is a shape parameter controlling the steepness of congestion growth. This formulation indicates that the congestion effect is nearly zero before the checking process starts, gradually increases during the boarding buildup window and stays at a high level close to departure.

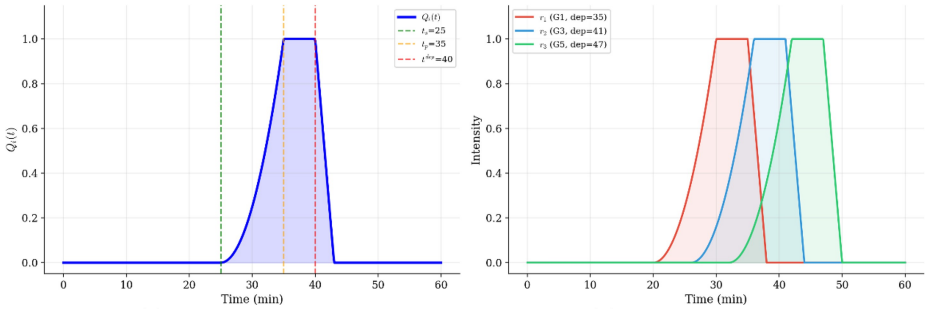
When multiple trains are assigned to the same gate during the study period, the temporal congestion intensity of that gate can be expressed as the superposition of train-specific effects:

$$C_g(t) = \sum_{m \in \mathcal{M}_g} \omega_m C_{g,m}(t), \tag{9}$$

where $\mathcal{M}_g$ is the set of trains associated with gate g and ω_m is the influence weight of train m. In this study, the influence weight is set to 1 for all trains as passenger-volume differences are not explicitly modeled; future work may use ticket sales or real-time passenger-flow data to assign variable weights. This superposition mechanism allows the temporal congestion profile of each gate to be generated directly from the train timetable, thereby reflecting the predictable nature of localized congestion in high-speed railway stations. Figure 2 illustrates the temporal evolution of boarding-gate congestion for a single train and the overlap of congestion windows when multiple trains are assigned to the same gate.

3.3 Spatial influence modeling of gate congestion

In addition to temporal variation, gate congestion also exhibits strong spatial locality. Passenger accumulation mainly occurs in front of the gate and along nearby circulation channels. As the distance from the gate increases, the congestion influence gradually



(a) Single Train Congestion Window **(b)** Multi-Train Overlapping Windows

Figure 2. Illustration of boarding-gate congestion windows: (a) congestion evolution for a single train and (b) overlapping congestion windows for multiple trains served by the same gate

weakens. Therefore, the temporal congestion intensity must be further projected onto the spatial domain of the waiting hall.

Let p_g denote the spatial position of gate g and let $d(x, p_g)$ denote the effective distance between any location x in the waiting hall and gate g. To characterize the attenuation of congestion influence with increasing distance, the spatial influence function of gate g at location x is defined as follows:

$$S_g(x) = \exp\big(-\beta\, d(x, p_g)\big), \tag{10}$$

where $\beta > 0$ is a decay coefficient. This function satisfies two intuitive properties: locations close to the gate receive strong congestion influence, whereas locations farther away receive much weaker influence. Compared with a simple rectangular hotspot definition, the exponential-decay form yields a smoother and more realistic representation of how local congestion affects surrounding corridors and accessible areas.

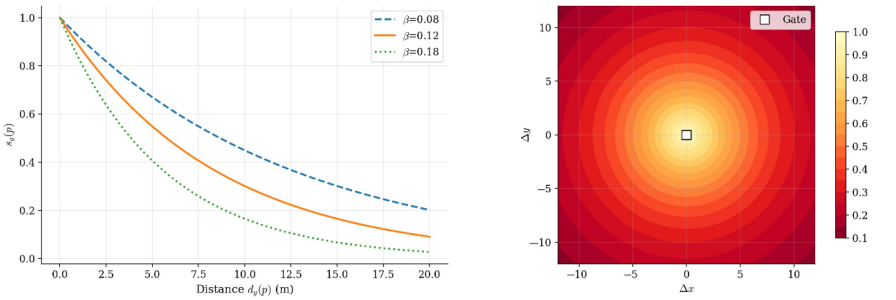
By combining the temporal evolution and the spatial decay, the spatiotemporal congestion intensity generated by gate g at location x and time t is defined as follows:

$$H_g(x, t) = C_g(t) S_g(x). \tag{11}$$

This formulation unifies the two key questions of congestion modeling, namely, *when* congestion becomes stronger and *where* its influence is more significant. Because several gates may be simultaneously active in the waiting hall, the overall congestion intensity at location x and time t is written as follows:

$$H(x, t) = \sum_{g \in \mathcal{G}} H_g(x, t). \tag{12}$$

Therefore, the local congestion status of any location in the waiting hall at any time can be obtained from $H(x, t)$, which provides the basis for subsequent edge-cost computation in the search graph. As shown in Figure 3, the spatial influence of gate congestion decays with distance and forms a localized congestion field around each boarding gate.



(a) Exponential Decay **(b)** 2D Spatial Field (β=0.12)

Figure 3. Spatial decay modeling of gate congestion: (a) decay curves under different values of β and (b) two-dimensional spatial influence field of a boarding gate

3.4 Task-oriented differential congestion mapping

For path planning, not all congestion should be treated identically. For a given task r, the target gate g_r and the elevator point behind it constitute the destination region that must be approached. Even if congestion near the target gate is relatively strong, it should not be regarded as a region to be strongly avoided, because the robot must eventually reach that area to complete the task. By contrast, congestion around non-target gates is unrelated to the current task and should be avoided as much as possible. To reflect this semantic asymmetry, a task-oriented differential congestion mapping is constructed.

For task r, the task-relevant congestion intensity at location x and time t is defined as follows:

$$H_r(x, t) = \alpha_{\text{tar}} H_{g_r}(x, t) + \alpha_{\text{non}} \sum_{g \in \mathcal{G},\, g \neq g_r} H_g(x, t), \tag{13}$$

where α_{tar} and α_{non} are the influence coefficients of the target gate and non-target gates, respectively and they satisfy:

$$0 \leq \alpha_{\text{tar}} < \alpha_{\text{non}}. \tag{14}$$

In the experiments, α_{tar} and α_{non} are set to 0.05 and 1.00, respectively. This setting implies that necessary accessibility to the target gate is preserved, while a stronger avoidance tendency is imposed on congestion associated with non-target gates. In this way, path planning is no longer based on a uniform judgment of whether an area is congested, but instead distinguishes between congestion that should be approached and congestion that should be avoided according to task semantics.

On the discrete search graph $G = (V, E)$, consider any edge $e \in E$ and let x_e denote the center position of that edge. The congestion-induced traversal increment of edge e for task r at time t is defined as:

$$\Delta\tau_r(e, t) = \gamma H_r(x_e, t), \tag{15}$$

where $\gamma > 0$ is a congestion amplification coefficient. Accordingly, the time-varying traversal cost of edge e under task r can be written as follows:

$$\tau_r(e, t) = \tau_0(e) + \Delta\tau_r(e, t), \tag{16}$$

where $\tau_0(e)$ is the nominal traversal time defined in Section 2. This equation shows that when an edge lies in a highly congested region and is traversed near the boarding peak, its traversal cost increases significantly; when the edge is far from the congestion hotspot or is traversed during a low-impact period, the cost remains close to the uncongested baseline.

To further evaluate the avoidance performance of a planned path with respect to non-target-gate hotspots, the non-target exposure of path P_r is defined as follows:

$$X(P_r) = \sum_{e \in P_r} \xi_r(e, t), \tag{17}$$

where t denotes the time when the robot traverses edge e and $\xi_r(e, t)$ denotes the exposure intensity of that edge to congestion generated by non-target gates. This indicator does not replace travel time as the primary optimization objective. Instead, it serves as an auxiliary

evaluation metric in the experiments, quantifying the extent to which a path unnecessarily passes through irrelevant congestion hotspots.

Through the above construction, a gate-centered congestion representation is obtained that is simultaneously time-dependent, spatially localized and task-sensitive. This representation not only reflects the operational characteristics of boarding-gate congestion in high-speed railway waiting halls, but also provides a mathematically tractable basis for dynamic edge-cost assignment in the subsequent path-planning algorithm. Figure 4 presents heat maps of congestion intensity at different time instants, showing how local congestion hotspots evolve over time in the waiting hall.

4. Proposed path-planning method

4.1 Methodological rationale

Based on the time-varying gate-congestion model developed in Section 3, this section proposes a path-planning method for a single robot performing a transport task in the waiting hall. The robot must travel from an entrance or elevator-hall area to the elevator point behind the target gate and arrive no later than 3 min before train departure. Thus, the search process must consider spatial reachability, time-varying traversal cost, non-target-gate exposure and deadline feasibility.

Classical A* is effective for static graphs but cannot directly handle timetable-driven gate congestion. Although time-dependent search is more suitable, a generic application does not distinguish target-gate and non-target-gate congestion or explicitly enforce the pre-departure deadline. Therefore, this study embeds the task-relevant congestion field into edge traversal costs, adds a semantic exposure penalty for non-target gates and applies deadline-aware pruning during search. This forms a congestion-aware, task-oriented time-space A*-type planning framework for high-speed railway waiting halls.

4.2 Search state and optimization objective

4.2.1 Time-augmented state representation. Let the search graph of the waiting hall be:

$$G = (V, E), \tag{18}$$

where V is the set of traversable nodes and E is the set of feasible edges. For a task r, let s_r denote the start node, q_r the destination elevator point, t_r^0 the task departure time, g_r the target gate and t_r^d the departure time of the associated train. Because edge costs vary with traversal time, the search state cannot be described only by spatial position. Instead, the state must also include the time at which the robot reaches that position. Accordingly, the state is defined as follows:

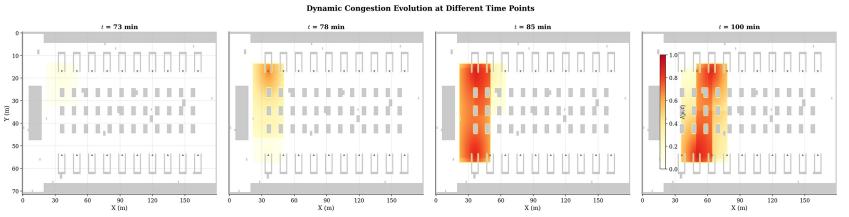


Figure 4. Heat maps of congestion intensity at different time instants

$$z = (v, t), \tag{19}$$

where $v \in V$ denotes the current node and t denotes the arrival time at node v. This time-augmented representation captures the key property of the present problem, namely, that the future cost from the same spatial node may differ when the arrival time is different.

To keep the search process tractable, the proposed method adopts a label-based structure on the time-augmented graph. Each feasible state label records the current node, the arrival time, the accumulated search cost and the predecessor relation used for path reconstruction. Thus, the search operates on time-stamped labels rather than on spatial nodes alone. This representation is more consistent with the time-varying edge-cost setting established in Section 3 and avoids the ambiguity that would arise if only one cost record were maintained for each node regardless of arrival time.

4.2.2 Path objective. For task r, let a candidate path be written as follows:

$$P_r = \langle v_0, v_1, \ldots, v_n \rangle, \tag{20}$$

where $v_0 = s_r$ and $v_n = q_r$. Let $e_k = (v_k, v_{k+1})$ denote the kth edge on the path and let t_k be the time at which the robot starts traversing e_k. According to Section 3, the time-varying traversal time of edge e_k under task r is as follows:

$$\tau_r(e_k, t_k) = \tau_0(e_k) + \Delta\tau_r(e_k, t_k), \tag{21}$$

where $\tau_0(e_k)$ is the nominal traversal time and $\Delta\tau_r(e_k, t_k)$ is the additional delay induced by the task-relevant congestion field. The expected travel time of path P_r is therefore:

$$T(P_r) = \sum_{k=0}^{n-1} \tau_r(e_k, t_k). \tag{22}$$

In addition, let $\xi_r(e_k, t_k)$ denote the non-target-gate exposure intensity of edge e_k at traversal time t_k. The cumulative exposure of the path is defined as follows:

$$X(P_r) = \sum_{k=0}^{n-1} \xi_r(e_k, t_k). \tag{23}$$

The generated path must first satisfy the deadline constraint:

$$t_r^0 + T(P_r) \le t_r^d - 3\ \text{min}. \tag{24}$$

Within the set of feasible paths satisfying this hard temporal constraint, the planning objective is defined as follows:

$$\min_{P_r} J(P_r) = T(P_r) + \lambda X(P_r), \tag{25}$$

where $\lambda \ge 0$ is the exposure-penalty weight. This formulation reflects two requirements simultaneously. The first is execution efficiency, represented by the expected travel time. The second is task-oriented avoidance of irrelevant congestion, represented by the non-

target-gate exposure term. By combining the two, the search process is guided toward paths that are both temporally efficient and semantically suitable for the given station transport task.

4.3 Dynamic cost update and search evaluation

4.3.1 Edge cost and label update. For any edge $e \in E$, the nominal traversal time is defined as follows:

$$\tau_0(e) = \frac{\ell(e)}{v}, \tag{26}$$

where $\ell(e)$ is the edge length and v is the nominal robot speed. Let x_e denote the center position of edge e. Based on the task-oriented congestion field established in Section 3, the congestion-induced delay on edge e for task r at time t is as follows:

$$\Delta\tau_r(e, t) = \gamma H_r(x_e, t), \tag{27}$$

where $H_r(x_e, t)$ is the task-relevant congestion intensity and $\gamma > 0$ is the congestion amplification coefficient. Therefore, the time-varying traversal time becomes:

$$\tau_r(e, t) = \tau_0(e) + \Delta\tau_r(e, t). \tag{28}$$

To reflect task semantics more explicitly, a non-target-gate exposure penalty is further introduced into the search cost. Let $\xi_r(e, t)$ be the exposure intensity of edge e to non-target-gate hotspots at time t. Then the composite search cost of traversing edge e at time t is as follows:

$$c_r(e, t) = \tau_r(e, t) + \lambda\xi_r(e, t). \tag{29}$$

Suppose the current label is $z_i = (v_i, t_i)$ with accumulated search cost $g(z_i)$. When the search expands from node v_i to its adjacent node v_j through edge e_{ij}, the temporary accumulated cost is updated as follows:

$$g'(z_j) = g(z_i) + c_r(e_{ij}, t_i), \tag{30}$$

and the predicted arrival time at the new label is updated as follows:

$$t_j = t_i + \tau_r(e_{ij}, t_i). \tag{31}$$

It should be emphasized that the arrival-time update is determined only by the actual time-varying traversal time $\tau_r(e_{ij}, t_i)$, rather than by the full composite search cost. The exposure term is introduced to guide path preference, but it does not alter the physical execution time of the robot. This separation preserves the consistency between real temporal evolution and semantic search bias.

4.3.2 Heuristic evaluation function. The proposed method retains the A*-style evaluation framework. For any time-augmented label $z = (v, t)$, the evaluation function is defined as follows:

$$f(z) = g(z) + h(v), \tag{32}$$

where $g(z)$ is the accumulated search cost from the start node to label z and $h(v)$ is the heuristic estimate of the remaining cost from node v to the destination. Because the congestion increment and the exposure penalty are both nonnegative, the uncongested travel-time estimate based on Euclidean distance can be used as an admissible optimistic heuristic:

$$h(v) = \frac{d(v, q_r)}{v}, \tag{33}$$

where $d(v, q_r)$ denotes the Euclidean distance from node v to the destination node q_r. This heuristic ignores future congestion and exposure and therefore provides a lower-bound estimate of the remaining travel time while maintaining low computational complexity.

The use of this heuristic offers two advantages. First, it preserves the directional guidance of A* and avoids the inefficiency of uninformed exhaustive expansion. Second, because the heuristic is independent of the semantic penalty term, it does not distort the feasibility judgment of the deadline constraint. Thus, the heuristic remains simple, efficient and compatible with the underlying task setting.

4.3.3 Deadline-oriented feasibility pruning. For the transport task considered in this paper, a path is not acceptable merely because it is spatially reachable. It must also satisfy the hard requirement that the robot arrive at the destination elevator point no later than 3 min before train departure. If infeasible branches are left in the search until the final stage, a substantial amount of computation may be wasted on paths that can never satisfy the deadline. To avoid this, the temporal feasibility constraint is integrated directly into the expansion process.

For task r, define the latest allowable arrival time as follows:

$$t_r^{\max} = t_r^d - 3\ \text{min}. \tag{34}$$

When the current label $z_i = (v_i, t_i)$ is expanded to a successor node v_j, the predicted arrival time is as follows:

$$t_j = t_i + \tau_r(e_{ij}, t_i). \tag{35}$$

Let the optimistic remaining travel time from v_j to the destination be:

$$\widehat{h}(v_j) = \frac{d(v_j, q_r)}{v}. \tag{36}$$

If:

$$t_j + \widehat{h}(v_j) > t_r^{\max}, \tag{37}$$

then even under the most optimistic uncongested continuation, the branch cannot reach the destination before the deadline. In that case, the branch is pruned immediately and is not inserted into the open list. This deadline-oriented pruning mechanism significantly reduces invalid search effort and improves the adaptation of the planning process to pre-departure transport tasks in high-speed railway stations.

4.3.4 Label management and dominance rule. Because multiple labels may reach the same spatial node at different times, the proposed method uses a dominance rule to control redundant expansion. For two labels $z_a = (v, t_a)$ and $z_b = (v, t_b)$ associated with the same node v, label z_a is said to dominate z_b if:

$$t_a \leq t_b \quad \text{and } g(z_a) \leq g(z_b), \tag{38}$$

with at least one strict inequality. In this case, label z_b can be safely discarded because it is no better than z_a in either arrival time or accumulated search cost. By applying this rule during insertion and update, the search can preserve the necessary temporal diversity of labels while avoiding uncontrolled growth of the state space. This mechanism makes the time-augmented search more suitable for time-varying cost environments than a single-record node expansion strategy.

4.4 Algorithm description

Based on the above definitions, the proposed method can be described as a deadline-aware, time-augmented A*-type search on the semantic grid graph. The algorithm takes as input the search graph G, the start node s_r, the destination node q_r, the task departure time t_r^0, the target gate g_r and the train departure time t_r^d. The output is the optimal feasible path under the composite objective defined in Section 4.2.2.

At initialization, an open list and a closed label set are created. The initial label is defined as follows:

$$z_0 = \left(s_r, t_r^0\right), \tag{39}$$

with:

$$g(z_0) = 0, f(z_0) = h(s_r). \tag{40}$$

The predecessor pointer of the initial label is set to null. The latest allowable arrival time is computed as $t_r^{\max} = t_r^d - 3$ min and the initial label is inserted into the open list.

During each iteration, the algorithm extracts from the open list the label with the minimum value of f. If the corresponding node is the destination q_r, the path is reconstructed by backtracking along the predecessor relation and the resulting path is returned. Otherwise, the label is expanded to all feasible adjacent nodes. For each adjacent edge, the algorithm computes the task-relevant congestion delay, the time-varying traversal time, the non-target-gate exposure penalty and the updated accumulated cost. Then it evaluates the predicted arrival time and applies the deadline-oriented pruning rule. If the successor label is temporally infeasible, it is discarded immediately. If the successor label is feasible but dominated by an existing label at the same node, it is also discarded. Otherwise, the successor label is updated and inserted into the open list. The procedure continues until either a feasible destination label is found or the open list becomes empty.

For clarity, the procedure can be summarized as follows:

Algorithm 1 Proposed deadline − aware congestion − sensitive path − planning method

Input: search graph $G = (V, E)$, start node s_r, destination node q_r, departure time t_r^0, target gate g_r, train departure time t_r^d

Output: optimal feasible path P_r^*

1. Initialize the open list O and the closed label set C.
2. Create the initial label $z_0 = (s_r, t_r^0)$, set $g(z_0) = 0$, $f(z_0) = h(s_r)$, and predecessor $(z_0) = \emptyset$.
3. Compute the latest allowable arrival time $t_r^{max} = t_r^d - 3$ min.
4. Insert z_0 into O.
5. **While** $O \neq \emptyset$ **do:**
   1. Extract the label $z_i = (v_i, t_i)$ with the minimum $f-$value from O.
   2. **If** $v_i = q_r$ **then:** reconstruct the path by predecessor tracing and return it.
   3. Insert z_i into C.
   4. **For each** adjacent node v_j of v_i **do:**
      1. **If** v_i is an obstacle or the edge e_{ij} is infeasible, **then continue**.
      2. Compute the congestion increment $\Delta\tau_r(e_{ij}, t_i)$.
      3. Compute the time − varying traversal time :
      $$\tau_r(e_{ij}, t_i) = \tau_0(e_{ij}) + \Delta\tau_r(e_{ij}, t_i).$$
      4. Compute the exposure penalty $\xi_r(e_{ij}, t_i)$.
      5. Update the temporary cost:
      $$g'(z_j) = g(z_i) + \tau_r(e_{ij}, t_i) + \lambda\xi_r(e_{ij}, t_i).$$
      6. Update the arrival time :
      $$t_j = t_i + \tau_r(e_{ij}, t_i).$$
      7. **If** $t_j + \widehat{h}(v_j) > t_r^{max}$, **then continue**.

8. **If** the new label is dominated by an existing label at v_j, **then continue**

9. Otherwise, create or update label $z_j = (v_j, t_j)$, set predecessor $(z_j) = z_i$, compute :

$$f(z_j) = g'(z_j) + h(v_j),$$

and insert z_j into O.

6. **If** $O = \emptyset$ and no feasible destination label has been found, **then return** failure.

The worst-case computational complexity of the time-space A* is O(|E|·|T|), where |E| is the number of edges and |T| the number of discrete time steps. In practice, deadline pruning and label dominance reduce the search space, enabling near real-time path planning for single-robot tasks.

Remark on theoretical properties. The proposed method is an A*-type search on a time-augmented graph and its optimality is defined with respect to the composite objective in equations (25) rather than the conventional static shortest path. Since all congestion-induced traversal costs and exposure penalties are nonnegative and the heuristic in equations (33) is based on uncongested Euclidean travel time, it provides an optimistic lower bound for the remaining cost. Therefore, under finite grid resolution and retained non-dominated labels, the algorithm preserves A*-style optimality among feasible labels. The semantic exposure penalty only changes the search preference and does not affect the physical arrival-time update. The deadline pruning removes a branch only when its current arrival time plus the optimistic remaining travel time exceeds the latest allowable arrival time; therefore, it does not discard any path that could still satisfy the deadline. Under these assumptions, the method is complete in the discretized search space: if a feasible on-time path exists and is not dominated, it can be found; otherwise, the algorithm returns failure.

5. Experimental results and analysis

5.1 Experimental setup

Experiments were conducted in the HSR F2 waiting hall with a size of 72 m × 180 m, 20 boarding gates and 6 × 6 cuboid seat blocks as internal obstacles. The proposed method was compared with Dijkstra, A*, RRT* and D* Lite under the same map, timetable-driven congestion setting and task-generation rules. The multi-task experiments were repeated with 30 random seeds under three scenarios: Normal (15 tasks), Peak (30 tasks) and Dense (45 tasks).

The evaluation metrics include OnTime%, PFR%, SimTime, Margin, Exposure, PlanMs and PathLen. Here, *SimTime* denotes the realized execution time, *Margin* denotes the remaining time before the departure deadline, *Exposure* denotes cumulative exposure to non-target-gate congestion regions, *PlanMs* denotes planning time and *PathLen* denotes path length. It should be noted that D* Lite does not use prior congestion knowledge in this experiment; instead, it discovers congestion during execution and replans incrementally. By

contrast, the proposed method performs proactive time-space planning based on predictable gate-congestion priors.

5.2 Mechanism verification

Two representative tasks were selected for mechanism verification: M1 (lower gate 8B) and M2 (upper gate 10A). The results are shown in Table 1. As shown in Figure 5, different methods exhibit clearly different path-selection behaviors in Task M1. Figure 6 further shows the path differences among methods in Task M2, especially in terms of congestion avoidance and route detour.

For both tasks, static shortest-path methods produced short geometric routes but suffered from high exposure and long execution time. RRT* reduced exposure compared with Dijkstra and A*, but its execution time remained high. D* Lite improved performance through online replanning, yet still required multiple replanning operations. The proposed method achieved the lowest or near-lowest SimTime and the lowest Exposure in both tasks, indicating that proactive planning with prior congestion knowledge is more suitable for this scenario than purely static or purely reactive mechanisms.

Table 1. Mechanism verification results

Task	Method	SimTime	Margin	Exposure	PlanMs	PathLen	OnTime	Replan
M1	Dijkstra	20.6	12.1	22.94	30.9	144.6	Y	0
M1	A*	21.26	11.44	28.19	30.6	144.6	Y	0
M1	RRT*	23.83	8.87	7.31	459.8	170.5	Y	0
M1	D* Lite	17.38	15.32	2.35	392.6	166.8	Y	11
M1	Proposed	16.1	16.6	0.75	1164.2	158.4	Y	0
M2	Dijkstra	22.51	15.74	29.83	33.8	160.3	Y	0
M2	A*	22.53	15.72	29.82	17.7	160.3	Y	0
M2	RRT*	33.46	4.79	13.95	421.5	204	Y	0
M2	D* Lite	18.32	19.93	4.29	723.1	173.3	Y	8
M2	Proposed	16.6	21.65	0	628	166	Y	0

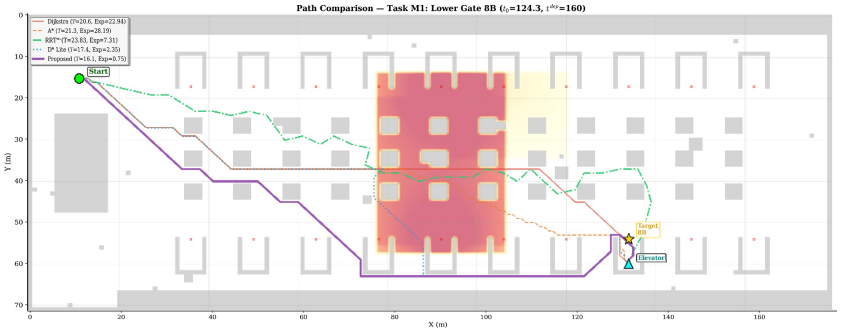


Figure 5. Path comparison of different methods for task M1

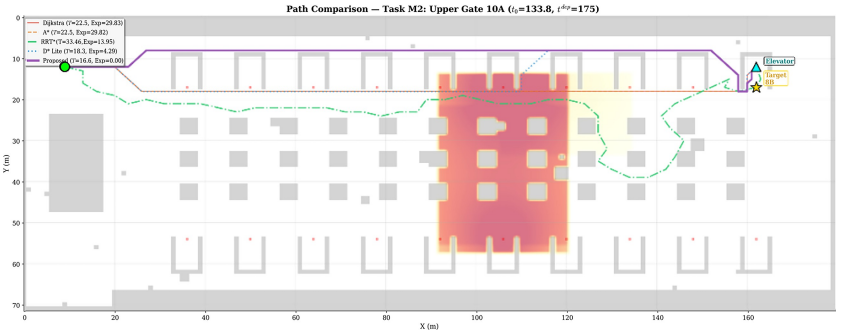


Figure 6. Path comparison of different methods for task M2

5.3 Multi-task comparative experiments

5.3.1 Normal scenario. In the normal scenario (Table 2), all graph-search methods achieved 100% OnTime and PFR, whereas RRT* showed limited robustness. Compared with Dijkstra and A*, the proposed method substantially reduced both SimTime and Exposure. Compared with D* Lite, the proposed method further reduced SimTime from 11.24 to 10.72, reduced Exposure from 1.65 to 0.23 and shortened planning time from 441.5 ms to 282.7 ms. These results indicate that when congestion is predictable, proactive time-space planning yields both better execution quality and lower planning overhead.

5.3.2 Peak scenario. As congestion intensified (Table 3), the gap between static planning and congestion-aware planning became larger. The proposed method achieved the best overall result, with the lowest SimTime, the highest Margin and extremely low Exposure. Relative to D* Lite, the proposed method reduced SimTime by about 7.4% and Exposure by

Table 2. Results in the normal scenario (15 tasks, 30 seeds)

Method	OnTime(%)	PFR(%)	SimTime	Margin	Exposure	PlanMs	PathLen
Dijkstra	100.00	100.00	12.69 ± 1.35	24.38 ± 2.03	10.11 ± 2.32	21.6 ± 4.2	102.8
A*	100.00	100.00	12.57 ± 1.34	24.50 ± 2.03	9.66 ± 2.38	15.6 ± 3.6	102.8
RRT*	73.10	73.10	14.63 ± 2.02	22.35 ± 2.54	3.00 ± 0.77	427.2 ± 68.6	116.5
D* Lite	100.00	100.00	11.24 ± 1.21	25.83 ± 1.92	1.65 ± 0.47	441.5 ± 209.7	107.4
Proposed	100.00	100.00	10.72 ± 1.12	26.35 ± 1.84	0.23 ± 0.14	282.7 ± 79.3	106.5

Table 3. Results in the peak scenario (30 tasks, 30 seeds)

Method	OnTime(%)	PFR(%)	SimTime	Margin	Exposure	PlanMs	PathLen
Dijkstra	100.00	100.00	14.58 ± 1.03	22.33 ± 1.11	17.48 ± 1.70	31.5 ± 6.0	103.9
A*	100.00	100.00	14.76 ± 1.05	22.14 ± 1.13	18.37 ± 1.76	22.3 ± 5.3	103.9
RRT*	73.30	73.90	16.91 ± 1.47	19.95 ± 1.64	5.39 ± 0.69	577.2 ± 139.2	118.3
D* lite	100.00	100.00	12.35 ± 0.89	24.56 ± 1.05	1.90 ± 0.36	771.8 ± 254.3	117.9
Proposed	100.00	100.00	11.43 ± 0.76	25.48 ± 0.99	0.15 ± 0.07	693.2 ± 201.4	113.8

about 92.2%. This suggests that in peak congestion, prior congestion knowledge is more beneficial than purely reactive replanning.

5.3.3 Dense scenario. In the dense scenario (Table 4), the proposed method remained stable, maintaining 100% OnTime and PFR while achieving the lowest SimTime and Exposure. Compared with D* Lite, the proposed method reduced SimTime by about 7.4%, reduced Exposure by about 91.5% and reduced planning time by about 29.3%. This confirms that the proposed scenario-adapted planner scales better as congestion becomes stronger and more widespread.

5.3.4 Overall discussion. Across all three scenarios, Dijkstra and A* remained feasible but suffered from large exposure to irrelevant gate hotspots. RRT* showed weaker robustness and lower feasibility under deadline constraints. D* Lite served as a strong online replanning baseline, but its reactive mechanism still produced higher exposure and higher planning cost than the proposed method. Overall, the results indicate that the contribution of this paper is not to replace all existing dynamic planners, but to adapt time-space A* to the gate-congestion characteristics of high-speed railway waiting halls through predictive congestion modeling, semantic differentiation and deadline-aware search. The relative improvement of the proposed method over the compared methods across evaluation metrics is summarized in Figure 7. A cross-scenario comparison of different methods is further provided in Figure 8. Although the proposed method achieves 100% OnTime in all tested scenarios, this reflects controlled single-robot conditions with pre-generated tasks; real deployment with multi-robot interactions or stochastic passenger behavior may be more challenging.

5.4 Ablation study

An ablation study was conducted in the peak scenario to evaluate the contribution of each component. The results are shown in Table 5.

Table 4. Results in the dense scenario (45 tasks, 30 seeds)

Method	OnTime(%)	PFR(%)	SimTime	Margin	Exposure	PlanMs	PathLen
Dijkstra	100.00	100.00	15.98 ± 1.07	21.08 ± 1.24	28.04 ± 2.75	23.0 ± 1.9	103.4
A*	100.00	100.00	16.18 ± 1.10	20.88 ± 1.25	29.22 ± 2.79	14.5 ± 1.8	103.4
RRT*	73.30	76.50	19.01 ± 1.41	18.09 ± 1.72	8.65 ± 1.17	340.8 ± 9.2	119.4
D* lite	100.00	100.00	12.17 ± 0.75	24.89 ± 1.01	1.79 ± 0.22	609.7 ± 87.2	116.5
Proposed	100.00	100.00	11.27 ± 0.66	25.80 ± 0.95	0.15 ± 0.04	431.1 ± 49.1	112.2

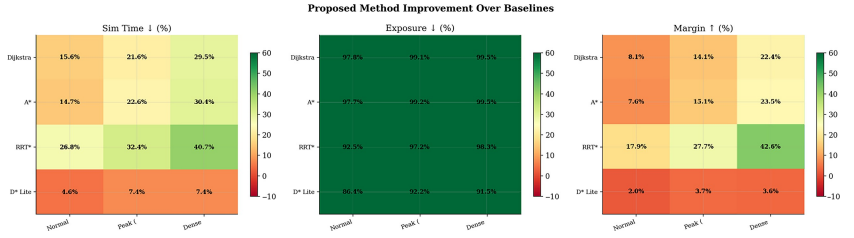


Figure 7. Heatmap of relative improvement of the proposed method across evaluation metrics

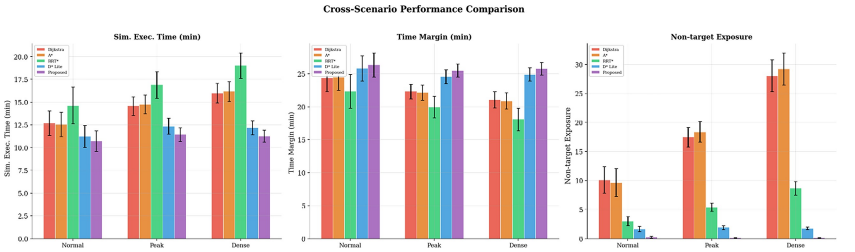


Figure 8. Cross-scenario comparison of different methods

Table 5. Ablation results in the peak scenario

Method	OnTime(%)	PFR(%)	SimTime	Exposure
Full method	100.00	100.00	11.43 ± 0.76	0.15 ± 0.07
W/o time-window	100.00	100.00	12.37 ± 0.86	0.42 ± 0.14
W/o dynamic-area	100.00	100.00	12.86 ± 1.02	8.14 ± 1.74
W/o semantic	100.00	100.00	11.40 ± 0.76	0.49 ± 0.15

Removing the time-window module increased both SimTime and Exposure, showing that temporal evolution modeling is necessary. Removing the dynamic-area module caused the largest deterioration, especially in Exposure, indicating that spatial hotspot representation is essential. Removing the semantic module barely changed SimTime but increased Exposure sharply, which confirms that semantic differentiation mainly improves task relevance rather than raw travel speed. The contribution of each module is further illustrated in Figure 9.

5.5 Statistical significance analysis

Statistical significance results are reported in Table 6 using D* Lite as the best baseline in this setting. Across all three scenarios, the proposed method achieved significantly lower SimTime and Exposure than D* Lite, with $p < 0.001$ under paired statistical tests across 30 random seeds. This indicates that the observed improvements are not due to random fluctuation but reflect a stable advantage of proactive congestion-aware planning under predictable gate-congestion conditions.

5.6 Parameter sensitivity analysis

A representative parameter perturbation analysis was conducted in the peak scenario to examine the influence of three key parameters: the spatial decay coefficient β in equations (10), the congestion amplification coefficient γ in equations (15) and the exposure-penalty weight λ in equations (25). Each parameter was varied around its default value, while the other two parameters were fixed. The default values were $\beta_0 = 0.06$, $\gamma_0 = 1.8$, $\lambda_0 = 0.06$ $\beta_0 = 0.3$.

Table 7 shows that the proposed method maintains 100% on-time rate and 100% path-finding rate under all tested settings. The variations in SimTime, Margin, Exposure and PathLen are small, indicating that the method is relatively stable to moderate parameter perturbations. For β, a larger value slightly reduces non-target exposure because the congestion influence becomes more localized. For γ, the tested range produces only minor

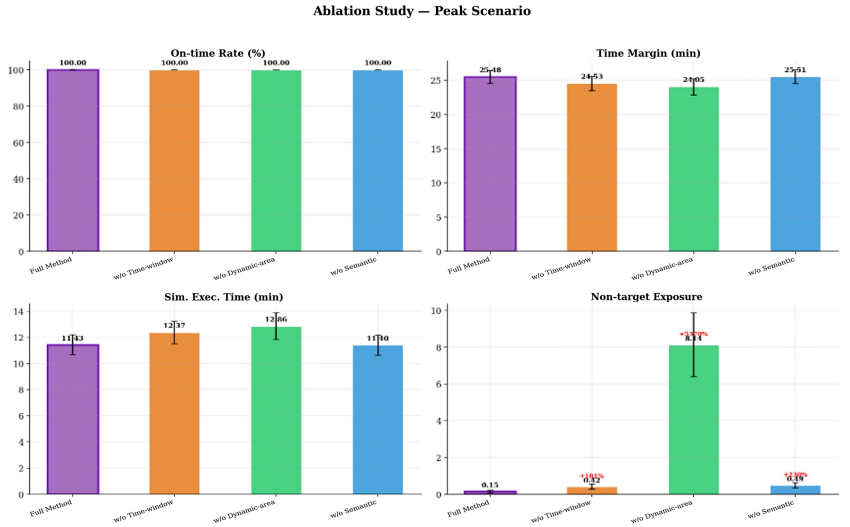


Figure 9. Bar-chart results of the ablation study in the peak scenario

Table 6. Statistical significance analysis

Scenario	Metric	Best Base	Proposed	Baseline	*p*-value
Normal	SimTime	D* Lite	10.717 ± 1.117	11.238 ± 1.210	0
Normal	Exposure	D* Lite	0.226 ± 0.144	1.654 ± 0.472	0
Peak	SimTime	D* Lite	11.429 ± 0.759	12.346 ± 0.893	0
Peak	Exposure	D* Lite	0.149 ± 0.069	1.904 ± 0.358	0
Dense	SimTime	D* Lite	11.268 ± 0.656	12.174 ± 0.750	0
Dense	Exposure	D* Lite	0.151 ± 0.044	1.786 ± 0.217	0

Table 7. Parameter sensitivity analysis in the peak scenario

Parameter	Value	OnTime(%)	SimTime	Margin	Exposure	PathLen
β	0.048	100.00 ± 0.00	12.85 ± 0.09	20.56 ± 1.57	0.15 ± 0.07	128.03 ± 0.73
β	0.06	100.00 ± 0.00	12.85 ± 0.09	20.57 ± 1.58	0.13 ± 0.06	128.03 ± 0.73
β	0.072	100.00 ± 0.00	12.84 ± 0.09	20.57 ± 1.58	0.11 ± 0.05	128.03 ± 0.73
γ	1.44	100.00 ± 0.00	12.84 ± 0.09	20.57 ± 1.58	0.13 ± 0.06	128.03 ± 0.73
γ	1.8	100.00 ± 0.00	12.85 ± 0.09	20.57 ± 1.58	0.13 ± 0.06	128.03 ± 0.73
γ	2.16	100.00 ± 0.00	12.85 ± 0.10	20.56 ± 1.57	0.13 ± 0.06	128.03 ± 0.73
λ	0.24	100.00 ± 0.00	12.85 ± 0.09	20.57 ± 1.58	0.13 ± 0.06	128.03 ± 0.73
λ	0.3	100.00 ± 0.00	12.85 ± 0.09	20.57 ± 1.58	0.13 ± 0.06	128.03 ± 0.73
λ	0.36	100.00 ± 0.00	12.85 ± 0.09	20.57 ± 1.58	0.13 ± 0.06	128.03 ± 0.73

changes in execution performance, suggesting that the planner is not overly sensitive to moderate congestion amplification. For λ, the final path metrics remain almost unchanged within the tested range, which indicates that the default semantic exposure penalty is already sufficient to guide non-target-gate avoidance in the tested tasks. Overall, the default parameter setting provides stable and balanced performance.

6. Conclusion and future work

6.1 Conclusion

This paper investigates single-robot path planning for transport tasks in high-speed railway waiting halls under boarding-gate congestion. Different from general indoor navigation, the scenario features schedule-driven congestion, target/non-target gate semantics and a strict pre-departure deadline. To address these characteristics, a scenario-adapted method based on time-space A* is proposed.

The method combines a gate-centered time-varying congestion model, a task-oriented semantic mapping strategy and a search framework integrating time-varying traversal cost, semantic exposure penalty and deadline-aware pruning. Experimental results show that the proposed method reduces unnecessary exposure and improves execution efficiency over static shortest-path methods, achieves better feasibility and stability than RRT* and outperforms DLite in SimTime, Exposure and planning cost. This demonstrates the effectiveness and practical value of task-oriented time-space A adaptation for waiting-hall transport tasks.

6.2 Future work

Several issues deserve further study. First, this paper considered only the single-robot, single-task setting. Future work may extend the method to multi-robot coordination, including conflict avoidance and task allocation. Second, the current congestion model is mainly timetable-driven; future work may integrate gate passenger counts or camera-based crowd-density data to update the congestion field online and adjust edge costs dynamically. Third, the present study was validated in simulation; more realistic station layouts and practical deployment tests should be explored in subsequent work.

References

AbuJabal, N., Mohd Basri, M.A., Danapalasingam, K.A., Mohd Zain, Z., Abd Aziz, N.A., Abdullah, N. S. and Muda, Z. (2024), "A comprehensive study of recent path-planning techniques in dynamic environments for autonomous robots", *Sensors*, Vol. 24 No. 24, p. 8089.

Lashin, M., El-Mashad, S.Y. and Elgammal, A.T. (2025), "Real-time path planning in dynamic environments using LSTM-augmented a search*", *Results in Engineering*, Vol. 27, p. 106324.

Mavrogiannis, C., Baldini, F., Wang, A., Tian, Y., Knepper, R.A. and Steinfeld, A. (2023), "Core challenges of social robot navigation: a survey", *ACM Transactions on Human-Robot Interaction*, Vol. 12 No. 3, p. 36.

Wei, L., Guo, D., Chen, Z., Yang, J. and Feng, T. (2023), "Forecasting short-term passenger flow of subway stations based on the temporal pattern attention mechanism and the long short-term memory network.", *ISPRS: International Journal of Geo-Information*, Vol. 12 No. 1, p. 25.

Yu, C., Li, H., Xu, X. and Sun, Q. (2021), "Estimating left behind patterns in congested metro systems: a Bayesian model", *Smart and Resilient Transportation*, Vol. 3 No. 2, pp. 149-161.

Zhao, W. and Xu, M. (2026), "A passenger flow congestion propagation Bayesian network model for urban rail transit hubs", *Reliability Engineering and System Safety*, Vol. 266, p. 111655.

Further reading

Alyassi, R., Abouelazm, M., Almeer, M.H., Elsersy, M., Al-Hammadi, Y., Werghi, N. and Elrefaei, L.A. (2025), "Social robot navigation: a review and benchmarking of current methods", *Frontiers in Robotics and AI*, Vol. 12, p. 1658643.

Francis, A., Pérez-D'Arpino, C., Li, C., *et al* (2025), "Principles and guidelines for evaluating social robot navigation algorithms", *ACM Transactions on Human-Robot Interaction*, Vol. 14 No. 2, p. 34.

Karwowski, J., Szynkiewicz, W. and Niewiadomska-Szynkiewicz, E. (2024), "Bridging requirements, planning, and evaluation: a review of social robot navigation", *Sensors*, Vol. 24 No. 9, p. 2794.

Mirsky, R., Xiao, X., Hart, J. and Stone, P. (2024), "Conflict avoidance in social navigation—a survey", *ACM Transactions on Human-Robot Interaction*, Vol. 13 No. 1, pp. 1-36.

Peng, J., Wei, Z., Li, J., Guo, X. and Wang, S. (2024), "Passenger flow bottleneck decongestion in subway stations: a simulation study", *SIMULATION*, Vol. 100 No. 10, pp. 981-995.

Song, B., Tang, S. and Li, Y. (2024), "A new path planning strategy integrating improved ACO and DWA algorithms for mobile robots in dynamic environments", *Mathematical Biosciences and Engineering*, No. 2, pp. 2189-2211.

Wang, H., Guo, L., Chen, T., Li, Y. and Zhou, G. (2026), "Quantification and evaluation for resilience of railway transportation system based on the cloud model", *Smart and Resilient Transportation*, Vol. 8 No. 1, pp. 57-78.

Wang, W., Ji, Y., Zhao, Z., Gao, X., Liu, Y. and Zhang, J. (2024), "Simulation optimization of Station-Level control of Large-Scale passenger flow based on queueing network and surrogate model", *Sustainability*, Vol. 16 No. 17, p. 7502.

Wang, S., Gong, G., Liu, Y. and Guo, F. (2025), "Passenger flow prediction between subway stations based on congestion state parameters and composite neural network", *Applied Soft Computing*, Vol. 185 No. Part B, p. 113991.

Zhao, X., Li, C., Zou, X., Du, X. and Ismail, A. (2024), "Passenger flow prediction for rail transit stations based on an improved SSA-LSTM model", *Mathematics*, Vol. 12 No. 22, p. 3556.

Zhiwen, L., Qin, Y. and Wang, M. (2022), "Research on high-speed railway operation adjustment model based on priority", *Smart and Resilient Transportation*, Vol. 4 No. 1, pp. 12-21.

Corresponding author

Wenhui Liu can be contacted at: whliu1005@163.com