

Dynamic Network Topology Analysis, Design, and Evaluation for Multi-Robot Vehicle Transfer in High-Density Storage Yards

Lin Zhang, Qiyu Cai, Runjiao Bao, Tianwei Niu, Yongkang Xu, Jing Si,
Shoukun Wang, and Junzheng Wang

Abstract—With the rapid advancement of intelligent manufacturing and the rise of emerging markets, global automobile exports have surged, placing unprecedented demands on logistics infrastructure. Efficient coordination of multiple robots for vehicle autonomous transfer is essential in high-density storage environments. However, conventional navigation mode, where autonomous robots navigate the entire space, often leads to inefficiencies, congestion, and increased safety risks. To address these challenges, this paper proposes a dynamic network topology framework to optimize large-scale vehicle transfers in high-density environments. The approach models free space as a network graph with directional, weighted movement costs. Leveraging yard operational characteristics, real-time transfer conditions, and robot specific capabilities, we introduce an event-triggered mechanism to update the network topology dynamically. This method continuously refines drivable space, effectively integrating yard areas with roadways to enhance routing flexibility in robot scheduling. Scenario-based evaluations demonstrate that the proposed approach reduces traveled distance by up to 12.3% and task completion time by 19.3% compared to traditional operational networks, leading to lower operational costs and improved task efficiency. Notably, these benefits become more pronounced as the number of robots increases and the operational environment grows more complex.

I. INTRODUCTION

The rapid advancement of intelligent and digital manufacturing is transforming the global automotive industry, while emerging markets are driving unprecedented growth in automobile exports, placing significant demands on logistics infrastructure [1], [2]. Multi-robot logistics systems, known for their efficiency, flexibility, and scalability, are widely deployed in intelligent warehouses [3], automated container terminals [4], and autonomous mining operations [5]. However, vehicle autonomous transfer remains in its early stages, with most research focused on indoor robotic valet parking [6]. As production and exports expand, large-scale outdoor vehicle autonomous transfer is becoming crucial for manufacturers and Roll-on/Roll-off (Ro-Ro) terminals.

Large-scale vehicle transfers require efficiently moving a high volume of vehicles between areas in the shortest time. High-density storage yards, commonly used by Ro-Ro terminals and automotive logistics hubs, optimize space by eliminating traditional roadways and aisles, increasing land usage by up to 60% compared to conventional parking lots

This work is supported by National Natural Science Foundation of China (No.62473044), the BIT Research and Innovation Promoting Project (No. 2024YCXZ007), and the Fundamental Research Funds for the Central Universities (No.2024CX06023). (Corresponding author: Shoukun Wang)

The authors are with the School of Automation, Beijing Institute of Technology, Beijing 100081, China. (e-mail: bit.zhanglin@bit.edu.cn; bitwsk@bit.edu.cn)



Fig. 1: Vehicle autonomous transfer scenario: (a) High-density storage yard transfer; (b) Robots entering the yard to pick up vehicles; (c) Robot straddling the vehicle.

[7], [8]. Additionally, as illustrated in Fig. 1, a straddle-type transfer robot with independent four-wheel drive and steering is better suited for outdoor heavy-load, high-speed operations, and maneuvering in storage yards than existing transfer robots [9]. Therefore, ensuring the safe and efficient scheduling and planning of straddle-type robots in high-density yards has become a critical challenge.

Due to the complex environment and dense vehicle storage, robot operations are restricted, and planning across the entire available space would be inefficient and unsafe. A common approach in multi-robot systems is to define a network topology (road network map) based on the layout [10], [11], which is crucial for large-scale vehicle transfers. However, manually designing network topologies is time-consuming and often leads to suboptimal solutions. Unlike fixed loading/unloading points in warehouses, vehicle logistics involves open storage areas without predefined transfer locations, where each vehicle is a “dynamic” obstacle. Additionally, vehicle transfers involve larger sizes, higher value, and tighter timelines [12], [13]. Therefore, creating a network topology in such environments requires addressing spatial, safety, and efficiency concerns, with the challenge being to automatically generate and dynamically adjust it based on layout, operational conditions, and robot capabilities.

A. Literature review

Network topology maps, also known as road maps for free space, were initially developed for motion planning of a single robot in static environments. For instance, V. Digani et al. [14] proposed a method to construct road maps for Automated Guided Vehicles (AGVs) in industrial logistics, enhancing system flexibility and efficiency by optimizing path coverage, connectivity, and redundancy. Building on this, S. Uttendorf et al. [15] introduced an expert system that combined human expertise with fuzzy logic and optimization, leveraging a blend of the A* algorithm and Bellman-Ford method for effective route mapping. Later, P. Beinschob et al. [16] proposed an automatic approach using 3D laser scanning to generate semantic maps for route design based on geometric and semantic data. H. Hu et al. [17] developed a node network for container port layouts to assist robot path planning using magnetic pin-guided navigation. While these methods automate route map generation, they primarily focus on layout design and spatial utilization in fixed scenarios, allowing robots to follow predefined routes. Although C. Yang et al. [18] enhanced route map construction using Euclidean coordinates and multidimensional edge attributes stored in a dictionary, challenges persist in multi-origin-destination systems. Identifying multiple feasible paths often leads to suboptimal solutions, and real-world deployment, particularly in dynamic path optimization across complex networks, remains inadequately addressed.

With the advancement of multi-robot systems, route map design has increasingly focused on collision avoidance. For instance, D. Kozjek et al. [19] enhanced fleet coordination through reinforcement learning, enabling automatic route generation that avoids conflicts and improves system performance. J. Stenzel et al. [20] developed an algorithm to create a roadmap graph based on map or CAD files and robot dimensions, ensuring no overlap of graph edges and vertices to prevent collisions. R. Vrabi et al. [21] proposed a grid graph with directed, weighted edges, optimizing the weights via an Ant Colony Optimization (ACO) algorithm to reduce conflicts. While these methods effectively minimize collisions, they prioritize maximizing available paths, leading to redundancy and underutilization of the road network. As a result, real-time adjustments based on environmental and operational changes are crucial. T. Zuzcek et al. [22] introduced an ACO-based method for generating route maps tailored to operational scenario changes. However, this method is computationally expensive and sensitive to initialization and optimization parameters. Current road generation approaches largely focus on resolving layout conflicts and coordinating multiple robots, but often overlook the specific operational characteristics of the robots and the yard environment itself.

A dynamically adjustable route topology, which considers both robot behaviors and real-time operational situations, has excellent potential to improve efficiency, reduce costs, and enhance scalability in large-scale multi-robot systems. This approach enables smarter, adaptive systems that can address complex, real-time challenges in logistics operations.

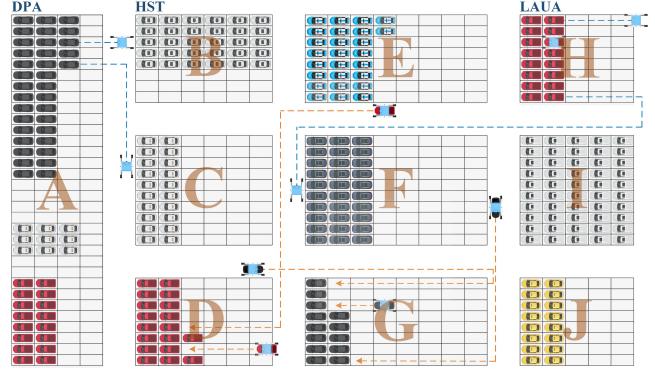


Fig. 2: Vehicle autonomous transfer scenario in high-density storage yard: 10 transfer robots transfer vehicles from Zones A and H to Zones G and D.

B. Contribution

This paper addresses the key challenges of autonomously generating and dynamically updating drivable roads for vehicle transfers in high-density storage yards. By analyzing the requirements of high-density outdoor vehicle transfer and the characteristics of the straddle-type robot, we abstract the free space as a grid map with directional weighted movement costs. We propose a method for automatic network topology generation and dynamic updates utilizing an event-triggered mechanism. Specifically, this method takes into account the yard layout, real-time transfer conditions, and the specific capabilities of the robots. By dynamically updating the drivable space during vehicle transfers, it effectively integrates the yard and road boundaries, enabling the robots to select routes more flexibly during scheduling planning. This strategy reduces unnecessary detours and alleviates congestion.

The paper is structured as follows: Section II analyzes the problem, Section III presents the method, Section IV provides validation, and Section V concludes.

II. PROBLEM STATEMENT

A. Problem overview

As shown in Fig. 2, vehicle transfer in high-density yards is common in Ro-Ro terminals, which typically consist of specialized zones such as loading and unloading areas (LAUA), high-density storage yards (HSY), and distribution port areas (DPA). The LAUA accommodates newly imported and export-ready vehicles, while the HSY stores vehicles awaiting export or departure. The DPA facilitates vehicle exchange between the terminal and external rolling stock. During import and export operations, vehicles must often be transferred between these areas within a short time. To replace the costly and inefficient manual transfer process with the multi-robot collaborative transfer, two key challenges must be addressed: (1) Construct the robot's initial navigation topology based on the terminal layout and vehicle distribution location; (2) Considering yard characteristics, robot capabilities, and real-time transfer conditions, dynamically adjust the network topology to improve efficiency.

B. Problem formulation

Consider a vehicles transfer operational environment with multiple vehicles straddle-type transfer robots $\mathcal{K} = \{r_1, r_2, \dots, r_m\}$, where the operational environment is defined by a time-varying workspace $\mathcal{W}(t) \in \mathbb{R}^2$. This workspace includes impassable regions \mathcal{O} (e.g., buildings, fixed obstacles), parked vehicles $\mathcal{D}_t = \{d_1(t), \dots, d_n(t)\}$ with time-dependent positions, and the free navigable workspace $\mathcal{F}(t) = \mathcal{W} \setminus (\mathcal{O} \cup \bigcup_{d_k \in \mathcal{D}_t} \mathcal{R}(d_k(t)))$, where $\mathcal{R}(\cdot)$ represent the area occupied. The navigation environment is encoded as a network topology graph $\mathcal{G}_t(\mathcal{V}_t, \mathcal{E}_t, \Omega_t)$, where:

- $\mathcal{V}_t = \{v_{1,1}(t), \dots, v_{m,n}(t)\}$ represents the set of vertices of the central graph, which correspond to the free nodes within the yard. Each $v_{m,n}$ denotes the nodes (x_m, y_n) .
- $\mathcal{E}_t = \{e_{(v_{1,1}, v_{1,2})(t)}, \dots, e_{(v_{i,j}, v_{p,q})(t)}\}$ represents the set of ordered vertex pairs, called edges, with the road segment volume expressed as $\mathcal{R}(p_e)$. This set corresponds to transitions between adjacent free space units: $\mathcal{E} \subseteq \{(v_{i,j}(t), v_{p,q}(t)) \mid (v_{i,j}(t), v_{p,q}(t)) \in \mathcal{V}_t^2, v_{i,j}(t) \sim v_{p,q}(t)\}$.
- $\Omega_t = \{\omega_{v_{1,1} \rightarrow v_{1,2}}(t), \dots, \omega_{v_{i,j} \rightarrow v_{p,q}}(t)\}$ represents the set of weights that determine the actual cost of moving along each edge. If the weight $\omega_{v_{i,j} \rightarrow v_{p,q}}(t) = \infty$, it indicates that movement from vertex $v_{i,j}$ (node $[x_i, y_i]$) to vertex $v_{p,q}$ (node $[x_p, y_q]$) is impossible.

The constructed network topology must ensure that each road segment in the yard avoids intersecting impassable areas. In addition, due to the size limitations of the robots, their movements cannot interact with these areas. The constraints are as follows:

$$\mathcal{R}(p_e) \cap o = \emptyset \quad \forall e \in \mathcal{E}, \forall o \in \mathcal{O}, \quad (1)$$

$$\mathcal{R}(r_i) \cap o = \emptyset \quad \forall r_i \in \mathcal{K}, \forall o \in \mathcal{O}, \quad (2)$$

$$\mathcal{R}(r_i) \cap \mathcal{R}(p_e) \neq \emptyset \quad \forall r_i \in \mathcal{K}, \forall e \in \mathcal{E}. \quad (3)$$

Additionally, parked vehicles in the yard are “dynamic” obstacles whose positions change during the transfer process and depend on the task type-acting as destinations for pick-up tasks and as impassable areas for drop-off tasks. Moreover, according to vehicle distribution in the yard and real-time transfer conditions, some new networks can also be added inside the yard to improve efficiency. These networks need to dynamically adjust the network topology according to the occupancy state $s_{i,j}(t)$ of the node $v_{i,j}(t)$ at time t . Moreover, as shown in Fig. 1c, depending on the robot’s state (whether it is carrying a vehicle), as it can “straddle” vehicles, introducing unconventional navigable routes. Therefore, designing an initial network topology $\mathcal{G}_0 = (\mathcal{V}_0, \mathcal{E}_0, \Omega_0)$ that considers the layout characteristics of the storage yard and vehicle positions is a critical challenge. Additionally, dynamically updating the network topology $\mathcal{G}_t = (\mathcal{V}_t, \mathcal{E}_t, \Omega_t)$ based on real-time transfer conditions and robot capabilities, while ensuring safety and improving transfer efficiency, constitutes a complex dynamic optimization problem.

III. METHODS

This section first presents a method for constructing the initial network topology based on the layout of the storage yard. Subsequently, dynamic update criteria for the network topology are proposed, utilizing an event-triggered mechanism that takes into account the yard layout, real-time transfer conditions, and the specific capabilities of the robots.

A. Construction of the initial network topology map

In a high-density storage area, the free space consists of three main components: the yards \mathcal{Y} , roads \mathcal{P} , and intersections \mathcal{I} . Fig. 3 illustrates a simple example of such a storage area. The first step in constructing the initial network topology is identifying the distribution of free space \mathcal{L} . The occupied space is then discretized into a two-dimensional (2D) grid, with each grid center serving as a node. Nodes are connected by edges based on the length d_h and width d_v of the parking spaces. The edge weight is initialized according to the port operation scenario and the direction of the robot’s movement along the roads. The pseudocode for initializing the network topology is shown in Alg. 1.

Algorithm 1 Initial network topology construction.

Require: Yard layout \mathcal{L} ; Parking space length d_h ; Parking space width d_v ;

Ensure: Initial network topology $\mathcal{G}_0 = (\mathcal{V}_0, \mathcal{E}_0, \Omega_0)$;

```

1: Initialize:  $\mathcal{V}_0 \leftarrow \emptyset, \mathcal{E}_0 \leftarrow \emptyset, \Omega_0 \leftarrow \emptyset$ ;
2: for each entity  $\mathcal{B}_k \in \{\mathcal{Y}, \mathcal{P}, \mathcal{I}\}$  do
3:   Generate a non-uniform grid based on  $\mathcal{B}_k$ ’ geometry;
4:   Create node  $v_{i,j}$  at the center of each grid cell;
5:   Update  $\mathcal{V}_0 \leftarrow \mathcal{V}_0 \cup \{v_{i,j}\}$ ;
6: end for
7: for each node  $v_{i,j} \in \mathcal{P} \cup \mathcal{I}, v_{p,q} \in \mathcal{P} \cup \mathcal{I}$  do
8:   if  $p = i \wedge \|v_{i,j} - v_{p,q}\| \leq d_v$  or  $q = j \wedge \|v_{i,j} - v_{p,q}\| \leq d_h$  then
9:     Update  $\mathcal{E}_0 \leftarrow \mathcal{E}_0 \cup \{e_{(v_{i,j}, v_{p,q})}(0)\}$ ;
10:    Update  $\omega_{v_{i,j} \rightarrow v_{p,q}}(0)$  via road rules;
11:   end if
12: end for
13: for each node  $v_{i,j} \in \mathcal{Y}, v_{p,q} \in \mathcal{Y}$  do
14:   if  $p = i \wedge \|v_{i,j} - v_{p,q}\| \leq d_v$  then
15:     Update  $\mathcal{E}_0 \leftarrow \mathcal{E}_0 \cup \{e_{(v_{i,j}, v_{p,q})}(0)\}$ ;
16:     Update  $\omega_{v_{i,j} \rightarrow v_{p,q}}(0)$  via Eq. (4)-(5);
17:   end if
18: end for
19: Update  $\Omega_0 \leftarrow \Omega_0 \cup \{\omega_{v_{i,j} \rightarrow v_{p,q}}(0)\}$ ;
20: return  $\mathcal{G}_0 = (\mathcal{V}_0, \mathcal{E}_0, \Omega_0)$ .

```

The irregular shape of vehicles is considered during discretization, as they are not square, leading to a non-uniform grid design that aligns with typical parking orientations. The grid prioritizes the yard space, with the road grid following the yard’s layout. Additionally, the robot’s size and the distance between parked vehicles are factored in to prevent collisions. Intersections between roads are handled by allowing robots to rotate and move laterally at intersections, with no additional processing for key intersection nodes.

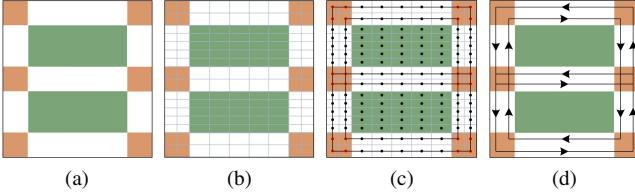


Fig. 3: Initial topology construction process of a high-density storage network: (a) Free space identification; (b) Space discretization into a 2D grid; (c) Edge connection based on grid centers; (d) Weight assignment for feasible routes.

Next, the nodes of pick-up yards $\mathcal{Y}_{\text{pick-up}}$ and drop-off yards $\mathcal{Y}_{\text{drop-off}}$ are added to the network topology based on vehicle distribution, while free yards $\mathcal{Y}_{\text{free}}$ without tasks are temporarily excluded. To standardize operations, vehicles are typically parked sequentially within the yard, aligned with the robot's entry and exit direction, with one side of the yard connected to the road. Based on the simplified coordinate system in Fig. 4a, the connection of the pick-up and drop-off yard is as follows, and the edge weight is dynamically adjusted according to the position of the vehicle to ensure the smooth operation of the robot. When a yard is connected to a nearby two-way road, the connection is considered fully connected by default without being limited to the side of the road closest to the yard. The connection modes of the pick-up yards and drop-off yards are as follows:

$$\omega_{v_{i,j} \leftrightarrow v_{i+1,j}}^{\text{pick-up}}(0) = \begin{cases} 1, & \text{if } s_{p,j}(0) \\ & \triangleleft s_{p+1,j}(0), \quad \forall i \in [p, m], \\ \infty, & \text{otherwise,} \quad \forall j \in [1, n], \end{cases} \quad (4)$$

$$\omega_{v_{i,j} \leftrightarrow v_{i+1,j}}^{\text{drop-off}}(0) = \begin{cases} 1, & \text{if } s_{p,j}(0) \\ & \triangleright s_{p+1,j}(0), \quad \forall i \in [p, m], \\ \infty, & \text{otherwise} \quad \forall j \in [1, n], \end{cases} \quad (5)$$

where $\omega_{v_{i,j} \leftrightarrow v_{i+1,j}}^{\text{pick-up}}(0)$ and $\omega_{v_{i,j} \leftrightarrow v_{i+1,j}}^{\text{drop-off}}(0)$ represent the bidirectional weights between node $v_{i,j}$ and $v_{i+1,j}$ in the pick-up and drop-off yards at initialization, respectively. $s_{p,q}(0)$ indicates the occupancy state of the node $[x_p, y_q]$ at initialization, where 1 represents occupied and 0 is unoccupied. The state difference operator $a \triangleleft b \equiv a \wedge \neg b$ is true if a is true and b is false. In contrast, the state co-extinction operator $a \triangleright b \equiv \neg a \wedge \neg b$ is true when both a and b are false.

In practice, storage yards and roads may not be perfectly aligned along horizontal or vertical axes. Therefore, adjustments are necessary for each position within the established network topology. The formula for calculating the new position is as follows:

$$\mathbf{p}' = \mathbf{c} + \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \cdot (\mathbf{p} - \mathbf{c}), \quad (6)$$

where $\mathbf{p} = [x, y]^T$ is the position vector in 2D space, $\mathbf{c} = [c_x, c_y]^T$ is the center of rotation, θ is the rotation angle of the entire space, and $\mathbf{p} - \mathbf{c}$ represents the offset relative to the center of rotation \mathbf{c} .

B. Dynamic network topology update criteria

To ensure efficient vehicle pick-up and drop-off operations while minimizing computational resource consumption, we design dynamic network topology update criteria based on an event-triggered mechanism, which contains three alternative criteria, as shown in Alg. 2. Its essence lies in updating the network by adjusting the weights of each initial state based on the changes in node occupancy within the yard. Compared to real-time updates, this method saves computational resources and is better suited for practical deployment. The update criteria consider the yard's characteristics, real-time transfer conditions, and the robot's specific capabilities. The design criteria are as follows.

Algorithm 2 Network topology dynamic update.

```

Require: Initial network topology  $\mathcal{G}_0 = (\mathcal{V}_0, \mathcal{E}_0, \Omega_0)$ ; Node occupation state  $s_{i,j}(t-1)$ ;
Ensure: Updated network topology  $\mathcal{G}_t = (\mathcal{V}_t, \mathcal{E}_t, \Omega_t)$ ;
1: Initialize:  $\mathcal{V}_{t-1} \leftarrow \mathcal{V}_0, \mathcal{E}_{t-1} \leftarrow \mathcal{E}_0, \Omega_{t-1} \leftarrow \Omega_0$ ;
2: while  $\exists \Delta s_{i,j}(t) \neq 0$  do
3:   Update  $s_{i,j}(t) \leftarrow s_{i,j}(t-1) + \Delta s_{i,j}(t)$ ;
4:   if Criterion A is enabled and satisfied then
5:     for each free yard  $y \in \mathcal{Y}_{\text{free}}$  do
6:       Update  $\omega_{v_{i,j} \rightarrow v_{p,q}}(t)$  via Eq. (7)-(8);
7:     end for
8:     for each pick-up yard  $y \in \mathcal{Y}_{\text{pick-up}}$  do
9:       Update  $\omega_{v_{i,j} \rightarrow v_{p,q}}(t)$  via Eq. (9);
10:    end for
11:   end if
12:   if Criterion B is enabled and satisfied then
13:     for each yard  $y \in \mathcal{Y}_{\text{pick-up}}$  and  $y \in \mathcal{Y}_{\text{drop-off}}$  do
14:       if  $p = i \wedge \|v_{i,j} - v_{p,q}\| \leq d_v$  then
15:         Update  $\mathcal{E}_t \leftarrow \mathcal{E}_{t-1} \cup e_{(v_{i,j}, v_{p,q})}(t)$ ;
16:       end if
17:       Update  $\omega_{v_{i,j} \rightarrow v_{p,q}}(t)$  via Eq. (10)-(11);
18:     end for
19:   end if
20:   if Criterion C is enabled and satisfied then
21:     for each yard  $y \in \mathcal{Y}_{\text{pick-up}}$  and  $y \in \mathcal{Y}_{\text{drop-off}}$  do
22:       Update  $\omega_{v_{i,j} \rightarrow v_{p,q}}(t)$  via Eq. (12)-(13);
23:     end for
24:   end if
25:   Update  $\Omega_t \leftarrow \Omega_{t-1} \cup \{\omega_{v_{i,j} \rightarrow v_{p,q}}(t)\}$ ;
26:   return  $\mathcal{G}_t = (\mathcal{V}_t, \mathcal{E}_t, \Omega_t)$ ;
27: end while

```

Criterion A: When three consecutive spaces (rows or columns) are available in non-operational yards, the central row or column is designated as the accessible route. In the pick-up yard, the front and rear main roads are connected to locations within the yard where vehicles awaiting transfer are positioned, providing direct access to the main road.

As shown in Fig. 4, we have removed the boundary between “roads” and “storage areas” based on the yard layout and characteristics, providing more routing options while ensuring safety. Additionally, by redefining the small

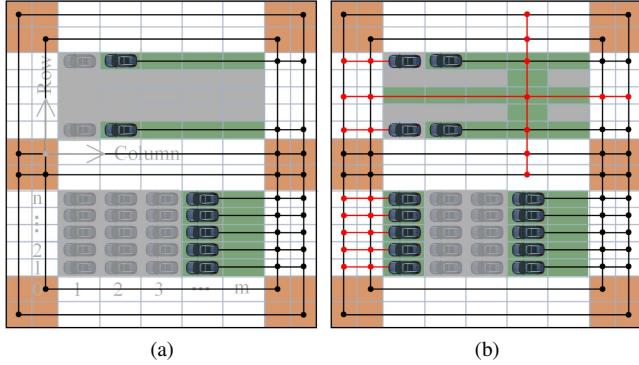


Fig. 4: The network topology before and after applying criterion A: (a) Initial network topology map: green indicates drivable areas, gray indicates non-drivable areas. (b) Network topology map after application of Criterion A: red indicates new driveable roads.

stacking retrieval scheme for high-density yards [23], robots can now pick up vehicles from both the front and rear of the same yard, reducing conflicts and improving transfer efficiency. This criterion is expressed as:

$$\omega_{v_{i,j} \leftrightarrow v_{i+1,j}}(t) = \begin{cases} 1, & \text{if } (s_{i,j-1}(t) \\ & \nabla s_{i,j}(t)) \quad \forall i \in [0, m], \\ & \triangleleft s_{i,j+1}(t), \quad \forall j \in [1, n], \\ \infty, & \text{otherwise,} \end{cases} \quad (7)$$

$$\omega_{v_{i,j} \leftrightarrow v_{i,j+1}}(t) = \begin{cases} 1, & \text{if } (s_{i-1,j}(t) \\ & \nabla s_{i,j}(t)) \quad \forall i \in [1, m], \\ & \triangleleft s_{i+1,j}(t), \quad \forall j \in [0, n], \\ \infty, & \text{otherwise,} \end{cases} \quad (8)$$

$$\omega_{v_{i,j} \leftrightarrow v_{i+1,j}}^{\text{pick-up}}(t) = \begin{cases} 1, & \text{if } s_{p,j}(t) \\ & \triangleleft s_{p-1,j}(t), \quad \forall i \in [0, p-1], \\ \infty, & \text{otherwise,} \end{cases} \quad (9)$$

Criterion B: In the vehicle pick-up and drop-off yards, if a vehicle is positioned in front of the operation area, and there are no vehicles on either side or behind, the space behind the operation position on the side without a vehicle can be designated as an accessible route.

As shown in Fig. 5, this criterion, based on real-time transfer conditions, enables the robot to move laterally onto the main road after reversing within the available safe area, providing more exit options. However, if there are no vehicles on the sides after the operation, directly connecting the lateral movement road could cause a collision with the vehicle in front. Therefore, the robot must reverse at least one parking space before connecting to the lateral road. To ensure safety, robots can only exit laterally and cannot switch modes within the yard. The lateral movement road is allowed only for exit, not entry. The updated criteria for both directions follow the same approach. Therefore, the specific

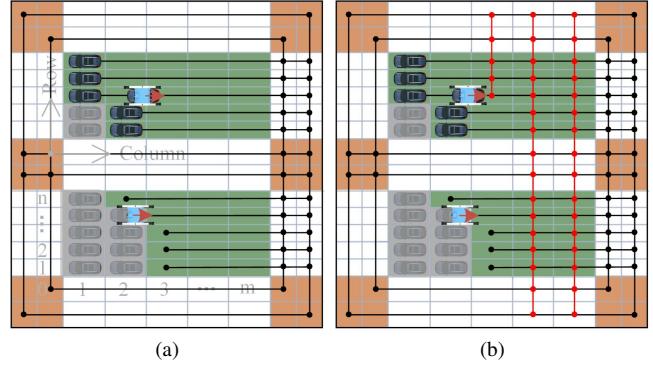


Fig. 5: The network topology before and after applying criterion B: (a) Initial network topology map: the upper side yard is the pick-up yard, and the lower side yard is the drop-off yard. (b) Network topology map after application of Criterion B: red indicates new driveable roads.

update criteria for upward horizontal movement are listed as follows:

$$\omega_{v_{i+1,j} \rightarrow v_{i+1,j+1}}^{\text{pick-up}}(t) = \begin{cases} 1, & \text{if } (s_{p,q}(t-1) \\ & \triangleleft s_{p,q}(t)) \quad \forall i \in [p, m], \\ & \triangleleft s_{i,j}(t), \quad \forall j \in [q, n], \\ \infty, & \text{otherwise,} \end{cases} \quad (10)$$

$$\omega_{v_{i+2,j} \rightarrow v_{i+2,j+1}}^{\text{pick-off}}(t) = \begin{cases} 1, & \text{if } (s_{p,q}(t) \\ & \triangleleft s_{p,q}(t-1)) \quad \forall i \in [p, m], \\ & \triangleleft s_{i,j+1}(t), \quad \forall j \in [q, n], \\ \infty, & \text{otherwise,} \end{cases} \quad (11)$$

Criterion C: In the vehicle pick-up yard, the network topology can be updated to allow the robot to navigate around parked vehicles, reach the front row, pick up the vehicle, and exit the yard. Similarly, after completing the drop-off, the robot can navigate through parked vehicles to exit the drop-off yard.

As shown in Fig. 6, we leverage the specific capabilities of the robot to move across vehicles in the “internal” area when no vehicles are present. This enables the updating of driving routes in the pick-up and drop-off yards, ensuring that the robot can complete tasks without the need to switch modes or change directions. This approach provides more flexible options for the robot in the storage yard, challenging traditional operational methods. The updated criteria are as follows:

$$\omega_{v_{i+1,j} \rightarrow v_{i,j}}^{\text{pick-off}}(t) = \begin{cases} 1, & \text{if } s_{p,q}(t) \\ & \triangleleft s_{p,q}(t-1), \quad \forall i \in [0, p-1], \\ \infty, & \text{otherwise,} \end{cases} \quad (12)$$

$$\omega_{v_{i+1,j} \rightarrow v_{i,j}}^{\text{pick-up}}(t) = \begin{cases} 1, & \text{if } s_{p,j}(t) \\ & \triangleleft s_{p-1,j}(t), \quad \forall i \in [p, m], \\ \infty, & \text{otherwise,} \end{cases} \quad (13)$$

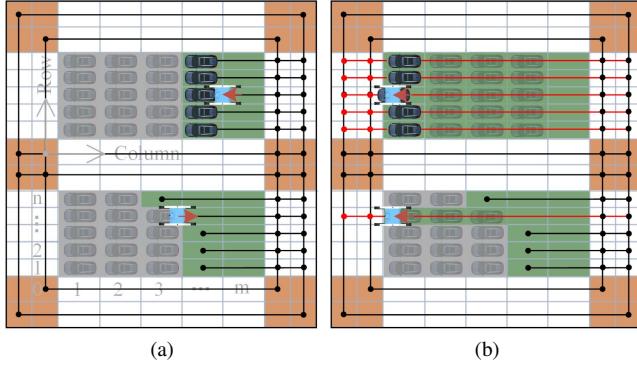


Fig. 6: The network topology before and after applying criterion C: (a) Initial network topology map; the upper side yard is the pick-up yard, and the lower side yard is the drop-off yard. (b) Network topology map after application of Criterion C.

As described above, Criterion A enables navigation in empty yard areas across all operational modes; Criterion B introduces exit roads to reduce congestion at entry/exit points; and Criterion C separates entry and exit routes to eliminate reversing. These criteria provide additional road options, improving efficiency and reducing transfer costs. While Criterion A can be combined with both B and C, combining B and C increases scheduling complexity and deployment challenges due to operational mode differences. Therefore, we focus on evaluating combinations of Criterion A with B and C, excluding B and C together.

IV. VERIFICATION AND EVALUATION

We designed small-scale, medium-scale, and large-scale scenarios to validate and evaluate the feasibility and effectiveness of the proposed methods and analyzed how different scenarios impact update criteria. All algorithms were implemented in Python and simulated on a PC equipped with an AMD R7 8845HS CPU and 24GB of RAM.

A. Simulation design

The simulation is based on the layout of the Yantai Port Ro-Ro terminal, as shown in Fig. 2, although the actual area of each yard is larger. A unified task allocation and path planning algorithm was used to evaluate the effectiveness of the proposed network topology update method in different scenarios. Given the high randomness of robot allocation in the yard and the significant impact of initial scheduling algorithms under various scenarios, we specified the tasks for each robot within designated yards. A staggered allocation strategy was adopted to avoid collisions when robots operate simultaneously in adjacent rows. Path planning was executed using the spatiotemporal A* algorithm, and robot collision detection was carried out through time-window conflict detection.

As shown in Tab. I, we define the operating time and travel speed of the robots during the simulation and establish small-scale, medium-scale, and large-scale scenarios.

In these scenarios, transfers between different storage yards were set up, with five robots in each yard assigned to execute 25 tasks. The simulation records the total traveled distance, average number of turns, task completion time, and average waiting time of the robots, which are used to evaluate the performance of the different criteria further.

TABLE I: Robot parameters and scenarios setting.

Description	Value
Pick-up vehicle time	10 s
Drop-off vehicle time	5 s
Robot mode-switching time	2 s
Robot 90° rotation time	3 s
Road driving speed	5 m/s ²
Yard driving speed	2.5 m/s ²
Small-scale scenario	$B \rightarrow G, H \rightarrow D$
Medium-scale scenario	$B \rightarrow G, D \rightarrow H, J \rightarrow C$
Large-scale scenario	$A \rightarrow H, B \rightarrow G, D \rightarrow E, J \rightarrow C$

B. Validation of dynamic network topology criteria

To evaluate the effectiveness of the proposed network topology update criteria and minimize the impact of scenarios on the algorithm, tests were conducted across the three designed scenarios. The evaluation methods included the traditional method without network topology update (WNTU), network topology update Criteria A (NTUA), B (NTUB), and C (NTUC). Additionally, combinations of criteria were tested, specifically NTUAB (the combination of NTUA and NTUB) and NTUAC (the combination of NTUA and NTUC). The comparison results are presented in Tab.II.

The results show that the proposed network topology update criteria outperform traditional vehicle transfer methods across all scales. In the three scenarios, NTUA, NTUB, and NTUC reduce traveled distance by an average of 6.8%, 6.7%, and 10.6%, respectively, while cutting task completion time by 7.7%, 2.9%, and 17.0%. NTUC achieves the most significant efficiency gains and cost reduction. NTUA and NTUC, which enable intra-yard traversal, significantly reduce turns compared to NTUB and the traditional method. However, although NTUB reduces total distance and alleviates congestion at yard entrances and exits, high robot traffic on the main road may still cause bottlenecks and delays. This phenomenon occurs in medium-sized scenarios, where a high concentration of robots on the main roads above Yards D, G, and J leads to congestion when moving laterally onto the main road from these yards. As a result, the total completion and waiting times can sometimes exceed those of the traditional method.

When combining different criteria, NTUAB and NTUAC reduce traveled distance by 12.3% and 10.7%, respectively, and completion time by 12.9% and 19.3% compared to the traditional method. This demonstrates that combined criteria improve cost efficiency more effectively than individual ones. Notably, NTUAB achieves shorter travel distances than NTUAC, differing from the individual NTUB and NTUC results. This is because integrating NTUB with NTUA enables front-and-rear vehicle pick-up, enhancing flexibility

TABLE II: Simulation results of dynamic network topology in different scenarios.

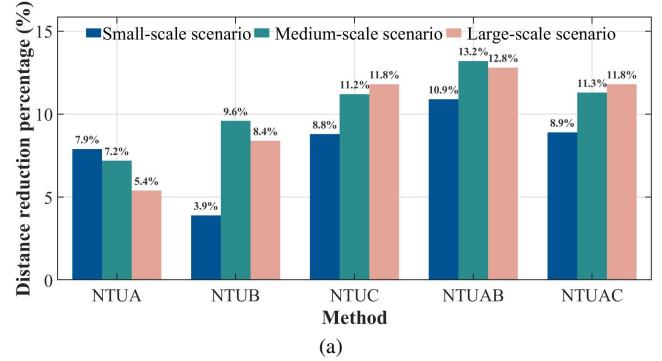
Scenario (robots-tasks)	Method	Total traveled distance (m)	Average number of turns	Task completion time (s)	Average avoidance waiting time (s)
Small-scale scenario (10-50)	WNTU	4680.50	44.00	2040.38	306.72
	NTUA	4307.66	39.00	1826.72	300.81
	NTUB	4495.74	44.00	1997.77	327.51
	NTUC	4266.42	43.00	1601.84	173.93
	NTUAB	4169.42	40.00	1756.79	281.79
	NTUAC	4262.00	36.40	1644.50	247.54
Medium-scale scenario (15-75)	WNTU	6463.25	44.00	2363.84	543.86
	NTUA	5995.91	35.73	2353.07	542.42
	NTUB	5840.27	44.00	2472.45	603.58
	NTUC	5739.69	40.67	2164.33	519.70
	NTUAB	5611.53	36.53	2177.88	434.32
	NTUAC	5732.49	34.80	2056.02	365.28
Large-scale scenario (20-100)	WNTU	8438.70	43.50	2851.36	562.91
	NTUA	7980.76	40.00	2508.11	505.81
	NTUB	7733.10	43.50	2565.04	641.45
	NTUC	7439.92	39.00	2250.76	520.94
	NTUAB	7356.86	39.75	2365.70	475.87
	NTUAC	7438.84	35.60	2119.14	369.25

and reducing travel distance. In contrast, adding NTUA to NTUC primarily decreases the number of turns but has little effect on total distance. In medium-sized yards, NTUB alone results in long waiting times due to congestion on the main road. However, combining NTUB with NTUA allows direct intra-yard traversal, reducing main-road congestion and significantly improving task completion time. Despite this, NTUAC consistently outperforms NTUAB by an average of 6.4% across all scenarios, prioritizing faster task completion at the cost of longer travel distances. Thus, selection should align with operational priorities: NTUAC is preferable when efficiency is critical, while NTUAB is more suitable for minimizing operational costs.

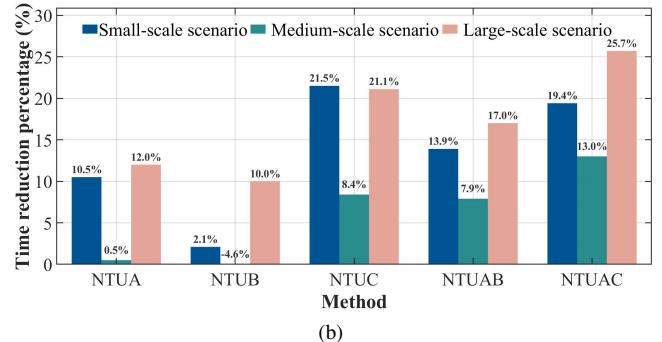
C. Impact of scenarios on different update criteria

To evaluate the performance of different update criteria across various scenarios, we calculated the percentage improvements in total traveled distance and task completion time for each criterion compared to the conventional approach, based on small-, medium-, and large-scale scenarios, as shown in Fig. 7.

Fig. 7 illustrates that the effectiveness of the criteria varies across scenarios. While nearly all criteria enhance efficiency and reduce costs compared to the conventional approach, the combined criterion demonstrates the most significant benefits. A comparative analysis reveals that the proposed criteria become increasingly effective in minimizing travel distance as scenario complexity rises, particularly for NTUAC. Specifically, NTUA optimizes vehicle operation sequences within the yard to reduce traveled distance, but its effectiveness depends on yard layout. In small-scale scenarios, front-to-back operations yield substantial distance reductions. However, as complexity increases, the overall traveled distance grows, diminishing NTUA's relative impact. In contrast, NTUB and NTUC introduce lateral transfers and



(a)



(b)

Fig. 7: Performance improvement percentages of different criteria compared to the traditional model across various scenarios: (a) Reduction in traveled distance. (b) Reduction in completion time.

cross-row movements, leading to progressively greater reductions in travel distance as scenario complexity increases.

In terms of reducing task completion time, although the increase in scenario scale and the number of robots leads to more conflicts and waiting during operations, the performance gains of the proposed criteria remain unaffected.

Instead, task time reduction is mainly driven by the specific characteristics of the scenario. Compared to the traditional method, nearly all methods show improvements; however, in medium-scale scenarios, the concentrated layout of operation yards leads to additional waiting times due to conflicts, making the upgrades less pronounced than in small and large scenarios. In contrast, large-scale scenarios are not hindered by the increased number of robots. On the contrary, as scenario complexity rises, robots benefit from a greater variety of available routes, and the introduction of the new operational mode facilitates a smoother workflow. This effect is most notable for NTUAC.

In summary, while the performance of different criteria varies across scenarios, their effectiveness remains strong even as complexity and the number of robots increase. The proposed dynamic network topology update enhances performance by offering more routing options in complex scenarios. Future work will investigate how yard distribution impacts road network update criteria, enabling the autonomous selection of the optimal approach for improved efficiency and cost-effectiveness.

V. CONCLUSION

This paper presents a method for generating and dynamically updating network topologies for multi-robot collaborative vehicle transfers in high-density yards, addressing challenges in free-space scheduling to enhance efficiency and safety. By abstracting free space into a direction-weighted network map and updating it through an event-triggered mechanism, the approach improves scheduling flexibility and efficiency. Experimental results show that the dynamic network topology reduces travel distance by up to 12.3% and task completion time by 19.3% compared to traditional operation networks, resulting in lower costs and improved efficiency. Furthermore, its performance increases with the number of robots and scenario complexity, demonstrating scalability and reliability. This work optimizes multi-robot systems in high-density yards and offers valuable insights for large-scale autonomous vehicle transfer operations.

Future work will explore the impact of yard distribution in complex scenarios on various road network update criteria, enabling the autonomous selection of the optimal approach based on scenario-specific characteristics. Additionally, we are deploying and testing the method at Yantai Port in Shandong to validate its practicality in real-world operations.

REFERENCES

- [1] S. Montoya-Zapata, N. Klement, C. Silva, O. Gibaru, and M. Lafou, “Multi-agent system for perturbations in the kitting process of an automotive assembly line,” *Engineering Applications of Artificial Intelligence*, vol. 135, p. 108679, 2024.
- [2] A. Abbasi-Pooya and M. T. Lash, “The third party logistics provider freight management problem: a framework and deep reinforcement learning approach,” *Annals of Operations Research*, vol. 339, no. 1, pp. 965–1024, 2024.
- [3] A. Bhargava, M. Suhaib, and A. S. Singholi, “A review of recent advances, techniques, and control algorithms for automated guided vehicle systems,” *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, vol. 46, no. 7, p. 419, 2024.
- [4] P. Z. Sun, J. You, S. Qiu, E. Q. Wu, P. Xiong, A. Song, H. Zhang, and T. Lu, “Agv-based vehicle transportation in automated container terminals: A survey,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 1, pp. 341–356, 2022.
- [5] S. Ge, F.-Y. Wang, J. Yang, Z. Ding, X. Wang, Y. Li, S. Teng, Z. Liu, Y. Ai, and L. Chen, “Making standards for smart mining operations: Intelligent vehicles for autonomous mining transportation,” *IEEE Transactions on Intelligent Vehicles*, vol. 7, no. 3, pp. 413–416, 2022.
- [6] M. Choi, G. Kang, and S. Lee, “Autonomous driving parking robot systems for urban environmental benefit evaluation,” *Journal of Cleaner Production*, vol. 469, p. 143215, 2024.
- [7] M. Nourinejad, S. Bahrami, and M. J. Roorda, “Designing parking facilities for autonomous vehicles,” *Transportation Research Part B: Methodological*, vol. 109, pp. 110–127, 2018.
- [8] J. Hou, G. Chen, Z. Li, W. He, S. Gu, A. Knoll, and C. Jiang, “Hybrid residual multiexpert reinforcement learning for spatial scheduling of high-density parking lots,” *IEEE transactions on cybernetics*, 2023.
- [9] Y. Xu, L. Zhang, Z. Liu, S. Wang, and J. Wang, “Design and development of a new autonomous transportation robot for finished vehicles docking transportation in ro/ro logistics terminal,” *Advanced Engineering Informatics*, vol. 66, p. 103391, 2025.
- [10] A. Kleiner, D. Sun, and D. Meyerdelius, “Armo: Adaptive road map optimization for large robot teams,” in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011.
- [11] Z. Song, K. Sheng, P. Zhang, Z. Li, B. Chen, and X. Qiu, “An integrated network modeling for road maps,” in *Model Design and Simulation Analysis: 15th International Conference, AsiaSim 2015, Jeju, Korea, November 4-7, 2015, Revised Selected Papers 15*. Springer, 2016, pp. 17–27.
- [12] X. Zhang, X. Ming, and Z. Chen, “Integration of ai technologies and logistics robots in unmanned port: A framework and application,” in *Proceedings of the 4th International Conference on Robotics and Artificial Intelligence*, 2018, pp. 82–86.
- [13] L. Zhang, Y. Xu, J. Si, R. Bao, Y. An, S. Wang, and J. Wang, “Autonomous transfer robot system for commercial vehicles at ro/ro terminals,” *Expert Systems with Applications*, vol. 289, p. 128347, 2025.
- [14] V. Digani, L. Sabattini, C. Secchi, and C. Fantuzzi, “An automatic approach for the generation of the roadmap for multi-agv systems in an industrial environment,” in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*.
- [15] S. Uttendorf, B. Eilert, and L. Overmeyer, “A fuzzy logic expert system for the automated generation of roadmaps for automated guided vehicle systems,” *IEEE*, 2016.
- [16] P. Beinschob, M. Meyer, C. Reinke, V. Digani, and L. Sabattini, “Semi-automated map creation for fast deployment of agv fleets in modern logistics,” *Robotics and Autonomous Systems*, vol. 87, 2016.
- [17] H. Hu, X. Yang, S. Xiao, and F. Wang, “Anti-conflict agv path planning in automated container terminals based on multi-agent reinforcement learning,” *International Journal of Production Research*, vol. 61, no. 1, pp. 65–80, 2023.
- [18] C. Yang and Z. Lin, “Dictionary-labeled a*: Optimal path-and-posture planning for mobile robots with turning radius constraints,” *IEEE Robotics and Automation Letters*, 2024.
- [19] D. Kozjek, A. Malus, and R. Vrabič, “Reinforcement-learning-based route generation for heavy-traffic autonomous mobile robot systems,” *Sensors*, vol. 21, no. 14, p. 4809, 2021.
- [20] J. Stenzel and L. Schmitz, “Automated roadmap graph creation and mapf benchmarking for large agv fleets,” in *2022 8th International Conference on Automation, Robotics and Applications (ICARA)*. IEEE, 2022, pp. 146–153.
- [21] R. Vrabič, T. Uek, G. Kulj, I. Banfi, and V. Zaletelj, “Improving theflow inmulti-robot logistic systems through optimization of layout roadmaps,” 2023.
- [22] T. Žužek, A. Zdešar, G. Škulj, I. Banfi, M. Bošnak, V. Zaletelj, G. Klančar et al., “Simulation-based approach for automatic roadmap design in multi-agv systems,” *IEEE Transactions on Automation Science and Engineering*, 2023.
- [23] G. Chen, J. Hou, J. Dong, Z. Li, S. Gu, B. Zhang, J. Yu, and A. Knoll, “Multiobjective scheduling strategy with genetic algorithm and time-enhanced a* planning for autonomous parking robotics in high-density unmanned parking lots,” *IEEE/ASME Transactions on Mechatronics*, vol. 26, no. 3, pp. 1547–1557, 2020.