2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Cooperative Path Planning for Four-Way Shuttle Vehicles in Storage and Retrieval Systems: A Hierarchically Dynamic Graph-Based Approach

Xingyao Han, Yuhong Tan, Siyuan Chen, Zhe Liu and Hesheng Wang

***Abstract*— Recently, Shuttle-based Storage and Retrieval Systems (SBS/RSs) have garnered significant attention from both academia and industry, owing to their high spatial utilization and rapid response speed. However, the weak connectivity of roadmaps in densely stored environments increases the likelihood of congestion and deadlocks when multiple four-way shuttles operate simultaneously, thereby imposing greater demands for collaborative path planning. Instead of exhaustively coordinating the shuttle motions during the off-line planning or online local control stages, we solve the cooperative path planning challenge from the perspective of altering the road graph structure dynamically. More specifically, we propose an approach to automatically transfer the typical road graph of SBS/RSs into a hierarchical graph with a reduced size, and then dynamically adjust its edge properties to prohibit any motion conflicts. In this manner, the planning problem of large-scale shuttle groups can be easily resolved and all the potential congestions can be eliminated inherently. Finally, we build a complete multi-shuttle cooperative path planning system adaptable for large-scale problems.**

I. Introduction

During the past decades, automatic warehousing systems based on multi-robot systems have developed rapidly [1]. Shuttle-Based Storage and Retrieval Systems (SBS/RSs) are widely recognized for their high space utilization, robustness, and rapid response speed [2]. ***Four-way shuttle vehicles*** (for short, shuttles) serve as operational devices within SBS/RSs, and their efficiency directly affects the performance of warehouses [3]. Due to weak connectivity in dense environments, congestion and deadlocks are more likely to occur when multiple shuttles operate simultaneously [4]. However, research on SBS/RSs mostly focuses on task scheduling [5]–[7] while overlooking the collaborative movements among shuttles.

Meanwhile, Multi-Robot Path Planning (MRPP) is a classic problem with numerous solvers [8]–[15]. However, existing solvers are defined in grid-like maps [8], with limited consideration for the connectivity of planning space. In contrast, shuttles operate in dense environments, characterized by weak connectivity [16] due to the lack of connections between *storage rows*. This difference may lead to decreased efficiency and resource waste when applying MRPP solvers directly [17]. Additionally, shuttles exhibit fast unidirectional movement but longer turning times [18], making them more suitable for rapid traversal on straight paths rather than frequent turning for obstacle avoidance. The graph in SBS/RS exhibits distinct features, with planning space divided into **horizontal *main aisles*** and **vertical *storage rows***. Leveraging the structure has the potential to boost planning performance. Unlike improving MRPP solvers, we propose a proactive method starting from the topology, endowing it with time-varying properties to preemptively avoid conflicts. *An apt analogy is that our method resembles laying down railway tracks while the shuttles move, ensuring seamless linear operations.* Our main contributions include:

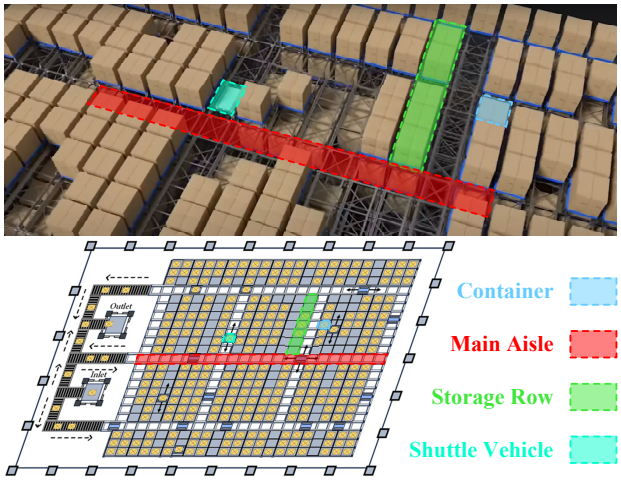

Fig. 1: The layout diagram of a SBS/RS. The layout is characterized by pathways categorized into main aisles and storage rows. Four-way shuttles move linearly along both the main aisles and the storage rows.

i) Instead of coordinating motions by designing planners, we resolve the cooperative path planning challenges from the perspective of altering the road graph structure dynamically, thus achieving collision-free MRPP inherently.

ii) We propose an approach to automatically transfer the typical road graph of SBS/RSs into a hierarchical graph with a reduced size, and then dynamically adjust its edge properties to prohibit any motion conflicts. The proposed approach can be used for any SBS/RSs.

iii) We build a cooperative path-planning system adaptable to large-scale SBS/RSs. Comprehensive simulations validate our efficiency and optimality, while experiments demonstrate a 20% increase in throughput in a real-world SBS/RS.

II. Related Work

A. Multi-Robot Path Planning for Warehousing Systems

MRPP typically involves planning collision-free paths for robots [19]–[21]. In practical warehousing systems, the MRPP problem is often defined as a lifelong problem [22]. Under the definition, when a robot reaches a pre-assigned

The first two authors contributed equally. This work was supported by the Natural Science Foundation of China under Grant 62303307 and the Shanghai Municipal Science and Technology Major Project under Grant 2021SHZDZX0102. X. Han and Z. Liu are with the MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University. Y. Tan, S. Chen, and H. Wang are with the Department of Automation, Shanghai Jiao Tong University. Corresponding authors: Zhe Liu and Hesheng Wang.

goal, it immediately moves towards the next assigned goal [23]. Lifelong problems are typically resolved through temporal decoupling [22] and corresponding solvers have been widely applied in warehousing systems. However, the work above focuses on simulated agents in grids, neglecting the graph structure. *Given the specific road topology of SBS/RSs, shall we tackle the cooperative path planning challenges by altering the road graph to achieve collision-free operation inherently?* This is exactly what we aim for in this paper.

B. Shuttle-based Storage and Retrieval Systems (SBS/RSs)

SBS/RSs are extensively adopted automated material handling systems in warehouses and distribution centers. Previous research [5]–[7] on SBS/RSs has predominantly focused on task scheduling, with a particular emphasis on the influence of task sequencing and allocation. Few stuties [17], [24] have addressed collaborative path planning, but they are limited to small-scale instances (not more than 20 units). Meanwhile, they overlook the structure inherent to SBS/RSs themselves. *How to accommodate large-scale shuttles (over 100) and leverage the topological features remains an unresolved challenge.*

III. Problem Formulation

A. Multi-Robot Path Planning (MRPP)

MRPP is formulated based on an undirected graph $\mathscr{G} = (\mathscr{V}, \mathscr{E})$. There are N_r robots on $\mathscr{G}$, denoted as $\mathscr{R} = \{r_1, r_2, \cdots, r_{N_r}\}$. The start and pre-assigned goal nodes of $\mathscr{R}$ are respectively denoted as $\mathscr{V}_s = \{s_1, s_2, \cdots, s_{N_r}\}$ and $\mathscr{V}_g = \{g_1, g_2, \cdots, g_{N_r}\}$. Clearly, $\mathscr{V}_s, \mathscr{V}_g \subset \mathscr{V}$. For $0 \le t \le T$, r_i occupies $u \in \mathscr{V}$, denoted as $q_i^t = u$. The objective of MRPP is to plan collision-free paths $\mathscr{P} = \{p_i = [q_i^0, q_i^1, \cdots, q_i^T] | i = 1, 2, \cdots, N_r\}$, satisfying for all $1 \le t \le T$ and $1 \le i, j \le N_r$:

$$i) q_i^t \neq q_j^t; \quad ii) (q_i^{t-1}, q_i^t) \neq (q_j^t, q_j^{t-1}). \tag{1}$$

where *i)* and *ii)* ensure there are no node and edge conflicts, respectively. In practical systems, robots are continuously assigned new goal nodes after reaching pre-assigned goals, referred to as the lifelong MRPP problem [22].

B. Senario and Environment Model in SBS/RSs

Our work is motivated by MRPP in SBS/RSs, as illustrated in Figure 1. A typical SBS/RS consists of a transfer area with automated outbound and inbound stations (hoister stations) on the left side and a storage area with numerous storage stations on the right side. The planning space (also referred to as ***roadmap***) resides within the storage area, denoted as $\mathscr{G} = (\mathscr{V}, \mathscr{E})$. Transportation tasks are generated and published online by the Warehouse Management System (WMS), each consisting of a pickup station and a hoister station. Each shuttle c_i moves to the pickup station upon task assignment and then proceeds to the hoister station to complete the task.

C. Topology Analysis in SBS/RSs

The *roadmap* $\mathscr{G}$ can be divided into **horizontal *main aisles*** (denoted as S_h) and **vertical *storage rows*** (denoted as S_v), respectively. For $u \in \mathscr{V}$, we denote its degree as ***deg***. It's evident in SBS/RS that if $deg(u) > 2$, $u \in S_h$; otherwise, $u \in S_v$. We introduce the boolean function $\mathbb{I}(u)$:

$$\mathbb{I}(u) = 1, \text{ if } deg(u) > 2, \text{ otherwise } 0. \tag{2}$$

where u belongs to S_h if $\mathbb{I}(u) = 1$, otherwise S_v. Merging u_i, u_j that are adjacent and share the same $\mathbb{I}(u)$, i.e.

$$\forall u_i, u_j \in \mathscr{V}, (u_i, u_j) \in \mathscr{E} \ \& \ \mathbb{I}(u_i) = \mathbb{I}(u_j), u_i, u_j \in S_{h/v}^k. \tag{3}$$

we can obtain ***aisle sectors*** $\mathscr{S}_h = \{S_h^1, S_h^2, \cdots, S_h^{N_h}\}$ and ***row sectors*** $\mathscr{S}_v = \{S_v^1, S_v^2, \cdots, S_v^{N_v}\}$, where N_h and N_v are the numbers of ***main aisles*** and ***storage rows***[1]. The shuttle moves within $S_{h/v}$ in one dimension, facilitating high-speed unidirectional movement[2]. Further considering each $S_{h/v}$ as a node, we can construct an undirected sector-level graph $\mathscr{G}_s$:

$$\mathscr{G}_s = (\mathscr{V}_s, \mathscr{E}_s), \mathscr{V}_s = \{S_h^i | i \in [1, N_h]\} \cup \{S_v^i | i \in [1, N_v]\}. \tag{4}$$

The degree of S_v in $\mathscr{G}_s$ must satisfy $deg(S_v) = 1$ (defined as *single row sector*) or $deg(S_v) = 2$ (defined as *double row sector*. We introduce another boolean function $\mathbb{J}$ for S_v:

$$\mathbb{J}(S_v) = 1, \text{ if } deg(S_v) > 1, \text{ otherwise } 0. \tag{5}$$

Considering the topological structure of SBS/RS, we observe the following properties of $\mathscr{G}_s$:

Property 1: S_h and S_v on $\mathscr{G}_s$ are not adjacent:

$$\forall S^i \in \mathscr{S}_h, \ S^j \in \mathscr{S}_v, \ (S^i, S^j) \notin \mathscr{E}_s. \tag{6}$$

Property 2: For any S_v on $\mathscr{G}_s$, there always exists an adjacent S_h as below, and vice versa.

$$\forall S_v, \exists S_h, \text{ s.t. } (S_v, S_h) \in \mathscr{E}_s. \tag{7}$$

Note: $\mathscr{G}_s$ can be regarded as the connectivity between *aisle sectors* S_h and *row sectors* S_v. The transitions of c_i between sectors are determined by planning results on $\mathscr{G}_s$. Based on the properties, we can derive the following lemma:

Lemma 1. *The sector-level path* Φ_i *planned for shuttle* c_i *on* $\mathscr{G}_s$ *necessarily takes the form* $(S_h^1, S_v^2, S_h^3, S_v^4, \cdots, S_h^n)$.

Proof. The pickup and drop-off nodes of c_i must necessarily be located within *storage row*[3], thus the starting and ending nodes of Φ_i must belong to $\mathscr{S}_h$. According to Equation. 6, two *row sectors* are necessarily non-adjacent, hence Φ_i must satisfy at least $\Phi_i = (S_h^1)$. When traversing through one S_v, Φ_i takes the form $\Phi_i = (S_h^1, S_v^2, S_h^3)$. Similarly, when passing through k S_v, Φ_i becomes $(S_h^1, S_v^2, S_h^3, S_v^4, \cdots, S_h^{2k+1})$. □

IV. Main Approach

A. Hierarchical Sector-Level Graph Extraction

In this section, we propose a sector-level topological graph extractor $\mathbb{E}$ capable of directly generating $\mathscr{G}_s$ from $\mathscr{G}$, i.e., $\mathscr{G}_s \leftarrow \mathbb{E}(\mathscr{G})$. With $\mathscr{G}_s$ obtained, we establish a hierarchical

[1] S_v can be divided into vertical highways and storage rows shown by the white and gray in Figure 1, which we collectively simplify as storage rows.

[2] Existing factories often configure all or parts of $S_{h/v}$ as the single road to facilitate subsequent planning and high-speed shuttle movement.

[3] The pickup nodes are naturally within storage rows. After task completion, hoister stations cannot serve as shuttle parking nodes, thus the drop-off nodes are distributed in the endpoints of storage rows.

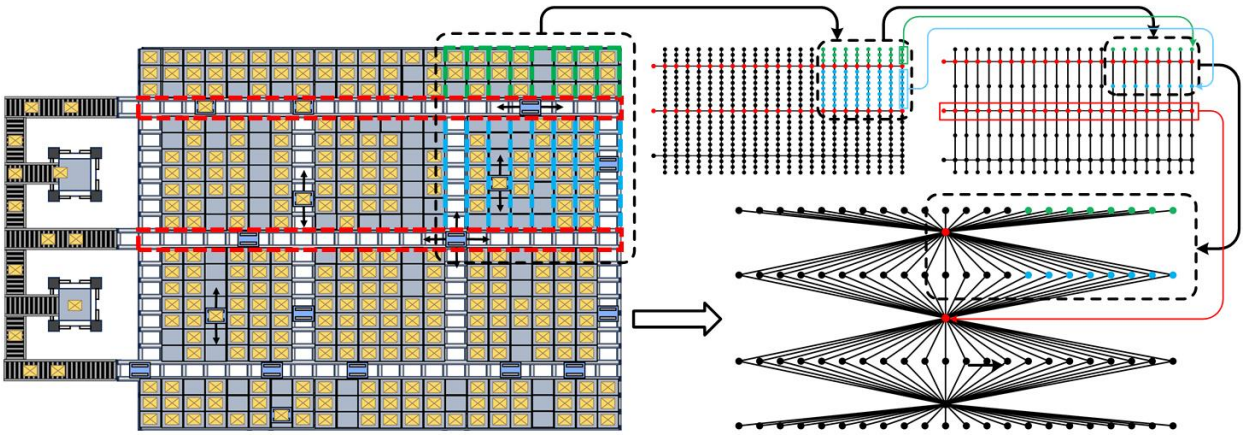

Fig. 2: Extraction of the sector-level graph. On the left is the original *roadmap*, with the top-right illustrating the extraction process. Meanwhile, the extracted sector-level graph is depicted on the bottom right. The red, blue, and green dots represent ***Main Aisles***, ***Double Rows*** and ***Single Rows***, respectively.

planning framework $(\mathscr{G}_s, \mathscr{G})$, which can guide and coordinate large-scale shuttles effectively. Given *roadmap* $\mathscr{G} = (\mathscr{V}, \mathscr{E})$, we traverse $u_i \in V$ and check $\mathbb{I}(u_i)$. Open-set $\mathscr{O}$ and close-set $\mathscr{C}$ are introduced as intermediate variables. $\mathscr{O}$ is used to traverse the nodes that have not been assigned to any sector S^j. For each $u_i \in \mathscr{O}$, if $\mathbb{I}(u_i) = 1$, then u_i belongs to *main aisles*. Based on this, we create the *aisle sector* $S_h^i = \{u_i\}$. Otherwise, we create the *row sector* $S_v^i = \{u_i\}$. Then, we add u_i to $\mathscr{C}$ and expand based on it. For the expanded u_j, we traverse $u_k \in neighbor(u_j) \setminus \mathscr{C}$. If $\exists u_k \in neighbor(u_j)$ s.t. $\mathbb{I}(u_k) = \mathbb{I}(u_j)$, then u_k is added to the already created $S_{h/v}^i$ and assigned as the node for next expansion. Otherwise, $S_{h/v}^i$ is complete and the next u_i is retrieved from $\mathscr{O}$, and the above process is repeated until $\mathscr{O} = \varnothing$. At this juncture, $\mathscr{V}_s$ is constructed and $\mathscr{E}_s$ is added as follows:

$$\forall u_i, u_j \in S_{h/v}^i, S_{h/v}^j, \text{ s.t. } (u_i, u_j) \in \mathscr{E}, (S_{h/v}^i, S_{h/v}^j) \in \mathscr{E}_s. \quad (8)$$

Thus far, we have extracted $\mathscr{G}_s$ from $\mathscr{G}$. Additionally, we have observed that $\mathscr{G}_s$ satisfies the following lemma:

Lemma 2. *If path planning on $\mathscr{G}$ is feasible, then the path planning on the extracted $\mathscr{G}_s$ is also feasible.*

Proof. Suppose shuttle c_i has a feasible plan denoted as $p_i = [q_i^0, q_i^1, \cdots, q_i^T]$ on $\mathscr{G}$. $\forall (q_i^k, q_i^{k+1})$, if $\mathbb{I}(q_i^k) = \mathbb{I}(q_i^{k+1})$, then according to Equation 3, they belong to the same S^k. If $\mathbb{I}(q_i^k) \neq \mathbb{I}(q_i^{k+1})$, then based on Equation 8, S^k and S^{k+1} satisfy $(S^k, S^{k+1}) \in \mathscr{E}_s$. Therefore, p_i can be transformed into $\Phi_i = [S_i^0, S_i^1, \cdots, S_i^T]$. Merging the same (S^k, S^{k+1}), we obtain $\Phi_i' = [S_i^0, S_i^1, \cdots, S_i^{T'}]$. $\forall 0 \leq k \leq T' - 1, (S^k, S^{k+1}) \in \mathscr{E}_s$. Thus, planning on $\mathscr{G}_s$ is also feasible. □

According to **Lemma 2**, we obtain the sector-level graph $\mathscr{G}_s$ without compromising feasibility and constructed a hierarchical planning framework $(\mathscr{G}_s, \mathscr{G})$. In comparison to $\mathscr{G}$, planning on $(\mathscr{G}_s, \mathscr{G})$ offers the following advantages: *i)* $\mathscr{G}_s$ features a reduced node number, facilitating faster searches and scalability to large-scale systems; *ii)* $\mathscr{G}_s$ can seamlessly incorporate temporal variations based on the *row sectors*, thereby influencing the topological structure of $\mathscr{G}$. This allows for pruning on the high level to assist the low-level planning, as elaborated in Section. IV-B; *iii)* $\mathscr{G}_s$ serves as a crucial data structure for extracting sector-level information, such as traffic flow or shuttle malfunctions, facilitating the diversion of four-way shuttles as discussed in Section IV-C.

B. Time-Varying Hierarchical Graph

In the preceding section, we extract $\mathscr{G}_s$ from $\mathscr{G}$, with its topological connections represented as depicted in Figure 2. Each *sector* constitutes a one-dimensional motion space internally. To ensure non-conflicting interactions among shuttle vehicles, it is paramount to guarantee conflict-free movements within *sectors*. In SBS/RS, the *main aisles* are typically configured as unidirectional, mitigating conflict occurrences. However, *storage rows* are often set as bidirectional, based on actual warehousing requirements, leading to possible deadlocks. For instance, when $\mathbb{J}(S_v) = 0$, a later-arriving vehicle entering S_v might block a preceding one, resulting in edge conflicts. Similarly, when $\mathbb{J}(S_v) = 1$, two vehicles at opposite ends may attempt to traverse to the other side simultaneously. **Hence, the crucial challenge lies in ensuring no conflicts occur within *row sectors*.**

Due to the direct correlation between the direction of edges in $\mathscr{G}_s$ and the orientation within S_v, we manage the orientation of S_v through the introduction of a time dimension to $\mathscr{G}_s$. Specifically, we define $\mathscr{G}_s^t = (\mathscr{V}_s, \mathscr{E}_s^t)$, with $\mathscr{G}_s^0 = \mathscr{G}_s$. We introduce temporal dynamics by modifying $\mathscr{E}_s^t$ and transforming $\mathscr{G}_s$ from an undirected graph into a time-varying directed graph $\mathscr{G}_s^t$. Building upon this, we further modify the dynamic *roadmap* $\mathscr{G} = \mathscr{G}^t = (\mathscr{V}, \mathscr{E}^t)$ according to Equation. 8, enhancing planning flexibility while ensuring safety. As transitions between sectors in $\mathscr{G}_s^t$, shuttle vehicles can anticipate the upcoming S_v and determine the orientation of $e_s^t = (S_h, S_v)|_t$. Below, we elucidate the specific mechanism of the time-varying e_s^t based on different $\mathbb{J}(S_v)$:

i) If $\mathbb{J}(S_v) = 0$, there exists only one S_h adjacent to S_v, making S_v a leaf node in $\mathscr{G}_s$. When c_i enters S_v from S_h, it must inevitably return to S_h. Therefore, upon the reservation and entry of c_i into S_v, the following holds:

$$(S_v, S_h) \in \mathscr{E}_s^t, (S_h, S_v) \notin \mathscr{E}_s^t. \quad (9)$$

ii) If $\mathbb{J}(S_v) = 1$, there exist two ***aisle sectors*** $S_h^{i,j}$ adjacent to S_v, rendering S_v a passageway connecting S_h^i and S_h^j. When c_i enters S_v from S_h^i, its objective is to either return to S_h^i

or traverse through to S_h^j. ***ii-a)*** If the destination is S_h^i, then its reservation aligns with *i)* and conflicts could be avoided. ***ii-b)*** If the destination is S_h^j, then its reservation leads to:

$$\begin{cases} (S_h^i, S_v) \in E_s, \quad (S_v, S_h^i) \notin E_s, \\ (S_v, S_h^j) \in E_s, \quad (S_h^j, S_v) \notin E_s. \end{cases} \tag{10}$$

Here, we ensure that planning on $\mathscr{G}_s^t$ is collision-free within row sectors, and we provide the following lemma:

Lemma 3. *The reservation mechanism based on i) and ii) guarantees collision-free planning within row sectors.*

Proof: We organize the proof according to $\mathbb{J}(S_v)$ and suppose S_v has been reserved by c_i. If $\mathbb{J}(S_v) = 0$, any other c_j are prohibited from entering S_v, preventing conflicts within S_v. If $\mathbb{J}(S_v) = 1$, shuttles within S_h^i, such as c_j, can follow c_i to enter S_v, which leads to no collision. However, shuttles from the other sector S_h^j, such as c_k, cannot enter S_v. Hence, conflicts within S_v are also avoided. □

The advantages of the reservation-based dynamic graph are evident: On one hand, compared to undirected graphs, $\mathscr{G}_s^t$ fully leverages the unidirectional speed advantage of shuttle vehicles, ensuring smooth and unobstructed travel on one-way paths. On the other hand, compared to unidirectional roadmaps, the dynamic characteristics of the time-varying graph preserve the flexibility of the planning space.

C. Multi-Shuttle Cooperative Path Planning

In this section, we divide the movement of shuttle vehicles into two stages: planning and execution.

In the planning stage, we employ A* on $\mathscr{G}_s^t$ to plan sector-level path Φ_i for each c_i. It's important to highlight that the edge weight of (S_h, S_v) is contingent on c_i's position. This consideration arises because the cost for a shuttle located at the left of S_h to travel to the left-end S_v is lower than the right-end S_v'. Furthermore, in large-scale multi-shuttle path planning, local congestion is a significant factor limiting efficiency. Therefore, building upon our previous work [12], we integrate traffic congestion cost with passage cost to create a composite cost. Specifically, we traverse the nodes on $\mathscr{G}_s^t$, calculating the congestion level $\gamma^t = \frac{N_i^t}{||S_v||}$, where N_i^t represents the number of shuttles, and $||S_v||$ denotes the number of nodes in the sector (the theoretical maximum shuttle capacity). Thus, $0 \leq \gamma^t \leq 1$. Assuming the passage cost of (S_h^i, S_v^j) is $d_{i,j}$, the composite passage cost $h_{i,j}^t$ is given by $h_{i,j}^t = d_{i,j} \times (1 + \gamma^t)$. We obtain sector-level paths through this approach, guiding multiple shuttles to minimize local congestion while maintaining nearly constant distances.

In the execution stage, we determine the starting and ending nodes for c_i based on Φ_i. With $\mathscr{G}^t$, Φ_i is expanded into road nodes. As demonstrated in Section. IV-B, conflicts within sectors are impossible. Therefore, we focus on c_i's sector transition and categorize it based on $\mathbb{J}(S_v)$:

i) If $\mathbb{J}(S_v) = 0$, the subsequent c_j cannot enter S_v according to Equation. 9. In this situation, c_j seeks a nearby temporary avoidance sector S_v'. Thus, $\Phi_j = \{S_h, S_v\}$ transitions to $\Phi_j = \{S_h, S_v', S_h, S_v\}$, waiting at S_v' until S_v releases the reservation;

TABLE I. PERFORMANCE OF ONE-SHOT MULTI-SHUTTLE PATH PLANNING

Evaluation Metrics	Methods	Shuttle Vehicle Number			
		20	40	60	80
Makespan ↓	**HCA* [9]**	**204.40**	**231.60**	271.00	-
	WHCA* [9]	211.20	250.50	313.00	-
	Manuel	234.86	258.78	255.83	277.20
	HTG [25]	229.32	240.56	248.28	263.52
	Ours	219.72	232.68	**247.84**	**262.25**
EO (%) ↓	**HCA* [9]**	111.00	115.2	136.80	-
	WHCA* [9]	107.93	121.6	139.9	-
	Manual	108.70	111.62	113.86	125.18
	HTG [25]	105.65	107.56	111.96	122.13
	Ours	**102.96**	**105.72**	**111.12**	**120.85**
DO (%) ↓	**HCA* [9]**	113.10	119.00	131.33	-
	WHCA* [9]	106.73	118.33	134.36	-
	Manual	107.5	108.88	106.47	107.44
	HTG [25]	104.77	104.64	104.62	104.61
	Ours	**102.12**	**102.90**	**102.90**	**103.06**
NoT ↓	**HCA* [9]**	298.2	771.8	1357.0	-
	WHCA* [9]	313.4	832.8	1485.0	-
	Manual	305.0	554.0	829.7	1119.3
	HTG [25]	294.0	595.4	893.4	1204.6
	Ours	**239.5**	**501.1**	**760.4**	**1023.6**
SR (%) ↑	**HCA* [9]**	**100**	80	20	0
	WHCA* [9]	**100**	80	60	0
	Manual	88	72	48	60
	HTG [25]	**100**	**100**	**100**	**100**
	Ours	**100**	**100**	**100**	**100**

- means deadlocks occurred and tasks were not entirely accomplished.

ii-a) If $\mathbb{J}(S_v) = 1$ and c_i returns to S_h^i, the switching mechanism for c_j remains the same as *i)*;

ii-b) Otherwise, c_i traverses S_v from S_h^i to S_h^j, according to Equation. 10, the co-directional c_i' follows c_i from S_h^i to S_v while the counter-directional c_j cannot enter S_v from S_h^j. In this case, c_j undergoes a replanning process. According to $\mathscr{G}_s^t$, its search path must avoid S_v. As a result, its sector-level path $\Phi_j = \{S_h^i, S_v, S_h^j\}$ switches to $\Phi_j = \{S_h^i, S_v', S_h^j\}$. By transitioning through S_v', the shuttle naturally achieves diversion, thereby avoiding local conflicts.

In summary, we have achieved collaborative path planning for four-way shuttle vehicles based on the hierarchically dynamic graph $(\mathscr{G}^t, \mathscr{G}_s^t)$. Our planning approach combines the stability of methods based on unidirectional paths with the flexibility of those based on bidirectional paths. From a graph perspective, we provide a novel planning solution.

V. EXPERIMENTAL EVALUATION

In this work, we evaluate the efficiency and performance of various methods using a real-world factory map comprised of 4,833 nodes and 9,602 edges. Existing planning methods for shuttles rely on unidirectional ***roadmaps***, which are designed to facilitate rapid linear movement. The generation of *roadmaps* is critical for *roadmap*-based methods. We utilized manually designed *roadmaps* (previously deployed in the factory) and Hopcroft-Tarjan-generated *roadmaps* (denoted as HTG [25]). We then applied the improved A* algorithm [17] to compare dynamic *roadmaps* with offline fixed *roadmaps*.

TABLE II. PERFORMANCE OF LIFELONG MULTI-SHUTTLE PATH PLANNING

Evaluation Metrics	Methods	Shuttle Vehicle Number								
		20	**30**	**40**	**50**	**60**	**70**	**80**	**90**	**100**
Runtime (ms) ↓	**HCA* [9]**	488.05	1481.30	1685.24	-	-	-	-	-	-
	WHCA* [9]	421.23	1418.93	1599.58	3548.59	-	-	-	-	-
	Manual	146.22	149.35	152.25	153.57	166.22	172.07	-	-	-
	HTG [25]	149.60	153.48	153.94	152.03	157.03	163.92	162.04	-	-
	Ours	**142.64** + 23.52	**149.17** + 24.51	**145.51** + 22.29	**149.14** + 23.64	**150.66** + 23.96	**156.2** + 24.72	**158.65** + 24.86	**157.74** + 24.71	**166.65** + 25.09
Throughput ↑	**HCA* [9]**	0.1077	0.1463	0.1948	-	-	-	-	-	-
	WHCA* [9]	0.1019	0.1554	0.1866	0.2225	-	-	-	-	-
	Manual	0.1133	0.1602	0.2246	0.2784	0.3175	0.3663	-	-	-
	HTG [25]	0.1129	0.1693	0.2246	0.2914	0.3294	0.3776	0.4186	-	-
	Ours	**0.1155**	**0.1750**	**0.2305**	**0.2955**	**0.3371**	**0.3953**	**0.4301**	**0.4712**	**0.5160**
EO (%) ↓	**HCA* [9]**	110.78	118.80	121.87	-	-	-	-	-	-
	WHCA* [9]	110.27	115.50	119.82	121.68	-	-	-	-	-
	Manual	104.68	112.87	107.16	114.63	114.95	116.44	-	-	-
	HTG [25]	105.10	106.03	107.18	108.14	110.18	113.61	115.29	-	-
	Ours	**102.92**	**103.77**	**104.95**	**105.21**	**107.46**	**108.90**	**113.91**	**118.16**	**123.42**
DO (%) ↓	**HCA* [9]**	110.15	116.92	120.31	-	-	-	-	-	-
	WHCA* [9]	109.82	114.43	117.96	120.11	-	-	-	-	-
	Manual	103.93	111.31	104.94	111.38	109.52	107.65	-	-	-
	HTG [25]	104.45	104.40	104.59	104.99	105.28	106.83	104.51	-	-
	Ours	**102.05**	**102.07**	**102.34**	**101.93**	**102.21**	**102.10**	**102.43**	**102.28**	**102.42**

The runtime of our method is presented in the form of $a+b$, where a represents the planning time and b represents the graph time-varying time.

Additionally, we deployed MRPP solvers, such as HCA* and WHCA* [9], on the undirected graph in SBS/RS to compare dynamic *roadmaps* with MRPP solvers. CBS [10] was not selected for comparison due to its excessive time consumption. Tests were performed in both simulated systems and real-world applications. In simulations, the methods were evaluated in one-shot and lifelong scenarios. In the one-shot settings, Makespan [8], Execution Optimality (EO), Distance Optimality (DO), Number of Turns (NoT), and Success Rate (SR) are employed to evaluate the performance. Execution Optimality refers to the optimality of time, defined as $EO = \frac{T_a}{T} \geq 1$, where T is the total execution time without collisions, and T_a is the actual execution time. A smaller EO indicates better performance. Similarly, Distance Optimality measures the optimality of distance, defined as $DO = \frac{L_a}{L} \geq 1$, where L is the total execution distance without collisions, and L_a is the actual distance. A smaller DO is also preferable. In the lifelong experiments, Runtime, Throughput, EO, and DO are selected as performance metrics. In real-world applications, the methods were deployed in a production environment, and throughput testing was carried out. All experiments were executed at Intel(R) Core(TM) i9-12900H 2.50GHz.

A. Evaluation of the One-shot Planning Problems

In this part, we evaluated the performance of various methods in one-shot planning under varying shuttle numbers. We set the task number equal to the shuttle number and executed the system. Upon all tasks completed, we recorded various performance metrics. If tasks were not completed within a reasonable deadline, it was deemed a planning failure. For each shuttle number, we conducted 25 repetitions and recorded the average, as shown in Table I. From the table, the following conclusions can be drawn:

a) The MRPP solvers struggle to scale in scenarios with large-scale shuttles. While HCA* [9] achieves optimal Makespan with fewer robots, its computational time is prolonged. Moreover, the optimality of HCA* significantly decreases as the shuttle number increases. With a shuttle number of 80, HCA* and WHCA* both completely fail.

b) The *roadmap*-based methods can operate in large-scale shuttle scenarios. However, due to the predefined *roadmap*, its planning flexibility is compromised, resulting in sub-optimality. Additionally, fixed *roadmap* leads to significant detours, resulting in a higher number of turns.

c) Our proposed solver achieves optimal performance in metrics of execution optimality, distance optimality, and number of turns, with a planning success rate of 100%.

B. Evaluation of the Lifelong Planning Problems

In this part, we set the number of tasks to be ten times the number of shuttles. Upon all tasks completed, various performance metrics were recorded. Failure to complete all tasks within a reasonable time was considered a planning failure. The performance is recorded in Table II:

a) MRPP solvers fail to complete lifelong tasks when the number of shuttles exceeds 40, due to the weak connectivity of SBS/RS, which increases susceptibility to local deadlocks.

b) The *roadmap*-based solvers exhibit optimal real-time performance, yet ensuring optimality poses challenges. Furthermore, circular deadlocks occur at shuttle numbers 80 and 90 for Manual-designed and HTG, respectively.

c) Our proposed solver strikes a balance between optimality and real-time performance, capable of scaling to larger SBS/RSs. This highlights the enhanced flexibility in planning enabled by dynamic *roadmaps*, which improves the algorithmic scalability while ensuring collision-free paths.

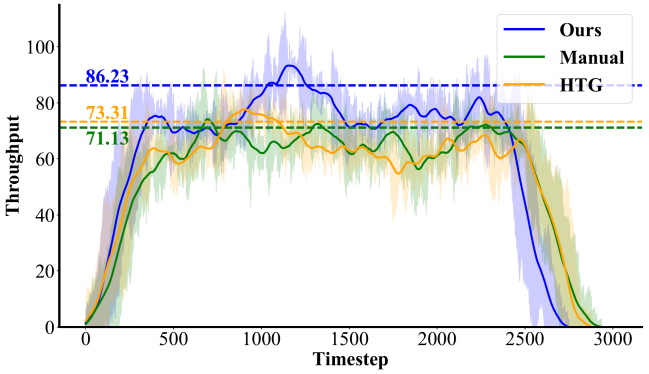

Fig. 3: Comparison of throughput between Manual (in Green), HTG (in Orange), and Ours (in Blue). The solid and shaded lines represent raw and smoothed throughput, while the dashed lines indicate the peak throughput.

VI. Real-World Applications

To validate the practicality of our method in real-world systems, we deployed it within an actual instance of SBS/RS. The system frequently executes outbound and inbound tasks dispatched by the upper-level WMS. The primary metric is the throughput of transport tasks. To this end, we assigned 600 transport tasks in a practical setting with 60 shuttles and 6000 storage locations. After execution, we recorded the throughput achieved for all tasks. Figure 3 shows the throughput of various methods during system operation. It is evident that our method consistently exhibits the highest throughput throughout almost the entire task execution process. Particularly noteworthy is that our method increases the peak throughput from 73.31 to 86.23, representing an improvement of nearly 20% in execution efficiency compared to the traditional algorithms. Moreover, our method demonstrates a swifter task completion compared to alternative methods, accelerating the execution speed of transport tasks and reducing the time costs in logistics transport.

VII. Conclusion And Discussion

We propose an MRPP method for SBS/RSs based on a hierarchically dynamic graph. Leveraging the topological structure, we extracted a sector-level graph and introduced a time-varying mechanism, enabling a scalable path-planning framework for up to 100 shuttles. Our method significantly enhances planning capacity and task execution efficiency. Future work will focus on addressing challenges like storage location selection and shuttle execution failures in SBS/RSs.

References

[1] Z. Liu, W. Chen, J. Lu, H. Wang, and J. Wang, "Formation control of mobile robots using distributed controller with sampled-data and communication delays," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 6, pp. 2125–2132, 2016.

[2] T. Lerher, M. Borovinsek, M. Ficko, and I. Palcic, "Parametric study of throughput performance in sbs/rs based on simulation," *International journal of simulation modelling*, vol. 16, no. 1, pp. 96–107, 2017.

[3] M. Küçükyaşar, B. Y. Ekren, and T. Lerher, "Cost and performance comparison for tier-captive and tier-to-tier sbs/rs warehouse configurations," *International transactions in operational research*, vol. 28, no. 4, pp. 1847–1863, 2021.

[4] J. Mao, J. Cheng, X. Li, H. Zhao, and C. Lin, "Modelling analysis of a four-way shuttle-based storage and retrieval system on the basis of operation strategy," *Applied Sciences*, vol. 13, no. 5, p. 3306, 2023.

[5] M. Küçükyaşar, B. Y. Ekren, and T. Lerher, "Cost and performance comparison for tier-captive and tier-to-tier sbs/rs warehouse configurations," *International transactions in operational research*, vol. 28, no. 4, pp. 1847–1863, 2021.

[6] Z. Wang and Y. Zhou, "Order sorting optimization of four-way shuttle based storage and retrieval system," in *2022 4th International Conference on Machine Learning, Big Data and Business Intelligence*, pp. 94–97, IEEE, 2022.

[7] J. Song, M. Yang, and X. Zhou, "Scheduling optimization of automated storage and retrieval system based on four-way shuttles," in *2020 IEEE International Conference on Mechatronics and Automation*, pp. 524–529, IEEE, 2020.

[8] Z. Liu, H. Wang, H. Wei, M. Liu, and Y.-H. Liu, "Prediction, planning, and coordination of thousand-warehousing-robot networks with motion and communication uncertainties," *IEEE Transactions on Automation Science and Engineering*, pp. 1705–1717, 2020.

[9] D. Silver, "Cooperative pathfinding," in *Proceedings of the aaai conference on artificial intelligence and interactive digital entertainment*, vol. 1, pp. 117–122, 2005.

[10] G. Sharon, R. Stern, A. Felner, and N. R. Sturtevant, "Conflict-based search for optimal multi-agent pathfinding," *Artificial Intelligence*, vol. 219, pp. 40–66, 2015.

[11] Z. Liu, H. Wei, H. Wang, H. Li, and H. Wang, "Integrated task allocation and path coordination for large-scale robot networks with uncertainties," *IEEE Transactions on Automation Science and Engineering*, vol. 19, no. 4, pp. 2750–2761, 2021.

[12] X. Han, S. Chen, X. Xiong, Q. Liu, S. Zhou, H. Zhang, and Z. Liu, "Traffic flow learning enhanced large-scale multi-robot cooperative path planning under uncertainties," in *2024 IEEE International Conference on Robotics and Automation*, pp. 16581–16587, IEEE, 2024.

[13] X. Xiong, X. Han, Z. Liu, and H. Wang, "Exhaustiveness does not necessarily mean better: Selective task planning for multi-robot systems," in *2023 IEEE International Conference on Robotics and Biomimetics*, pp. 1–6, IEEE, 2023.

[14] B. Wang, Z. Liu, Q. Li, and A. Prorok, "Mobile robot path planning in dynamic environments through globally guided reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6932–6939, 2020.

[15] Z. Liu, S. Zhou, H. Wang, Y. Shen, H. Li, and Y.-H. Liu, "A hierarchical framework for coordinating large-scale robot networks," in *2019 International Conference on Robotics and Automation*, pp. 6672–6677, IEEE, 2019.

[16] X. Zhao, R. Zhang, N. Zhang, Y. Wang, M. Jin, and S. Mou, "Analysis of the shuttle-based storage and retrieval system," *IEEE Access*, vol. 8, pp. 146154–146165, 2020.

[17] W. He and R. Tian, "An improved a* algorithm for four-way shuttle vehicle path planning," in *2023 5th International Conference on Intelligent Control, Measurement and Signal Processing*, pp. 1161–1165, IEEE, 2023.

[18] X. Zhao, R. Zhang, N. Zhang, Y. Wang, M. Jin, and S. Mou, "Analysis of the shuttle-based storage and retrieval system," *IEEE Access*, vol. 8, pp. 146154–146165, 2020.

[19] R. Luna and K. E. Bekris, "Efficient and complete centralized multi-robot path planning," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3268–3275, IEEE, 2011.

[20] O. Salzman and R. Stern, "Research challenges and opportunities in multi-agent path finding and multi-agent pickup and delivery problems," in *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*, pp. 1711–1715, 2020.

[21] M. Peasgood, C. M. Clark, and J. McPhee, "A complete and scalable strategy for coordinating multiple robots within roadmaps," *IEEE Transactions on Robotics*, vol. 24, no. 2, pp. 283–292, 2008.

[22] H. Ma, J. Li, T. Kumar, and S. Koenig, "Lifelong multi-agent path finding for online pickup and delivery tasks," *arXiv preprint arXiv:1705.10868*, 2017.

[23] R. Stern, N. Sturtevant, A. Felner, S. Koenig, H. Ma, T. Walker, J. Li, D. Atzmon, L. Cohen, T. Kumar, *et al.*, "Multi-agent pathfinding: Definitions, variants, and benchmarks," in *Proceedings of the International Symposium on Combinatorial Search*, vol. 10, pp. 151–158, 2019.

[24] J. Mao, J. Cheng, X. Li, and B. Cao, "Research on scheduling optimization of four-way shuttle-based storage and retrieval systems," *Scientific Reports*, vol. 13, no. 1, p. 3999, 2023.

[25] J. Hopcroft and R. Tarjan, "Algorithm 447: efficient algorithms for graph manipulation," *Communications of the ACM*, vol. 16, no. 6, pp. 372–378, 1973.