A Demonstration of Continuous Lifelong Conflict-Aware AGV Routing with Kinematic Constraints

Ruizhong Wu
DSA Thrust, HKUST(GZ)
rwu601@connect.hkust-gz.edu.cn

Tianqi Zhang
DSA Thrust, HKUST(GZ)
tzhang657@connect.hkust-gz.edu.cn

Yunjie Huang
DSA Thrust, HKUST(GZ)
yhuang863@connect.hkust-gz.edu.cn

Lei Li
DSA & INTR Thrust, HKUST(GZ)
thorli@ust.hk

ABSTRACT

Automated Guided Vehicles (AGVs) are the backbone of modern e-commerce warehouses, but the existing multi-agent pathfinding (MAPF) solutions fail to simultaneously satisfy the practical requirements of continuous space and time, lifelong and dynamic tasks, kinematic movement constraints, online responsiveness, and large-scale scalability in real-world deployments. This demo presents a fully functional lifelong conflict-aware AGV routing system built on our novel routing algorithm OHSMD optimized for warehouse scenarios. The system supports customizable warehouse map configuration, AGV kinematic parameter tuning, lifelong task scheduling, conflict-free path planning, interactive visualization, and physical validation using Sony's Toio™ Core Cube robots. Through this demo, the audience can intuitively experience the system's performance and verify the practical availability of our routing algorithms in industrial warehouse scenarios.

PVLDB Reference Format:
Ruizhong Wu, Tianqi Zhang, Yunjie Huang, and Lei Li. A Demonstration of Continuous Lifelong Conflict-Aware AGV Routing with Kinematic Constraints. PVLDB, 19(12): 4622 - 4625, 2026.
doi:10.14778/3827998.3828081

1 INTRODUCTION

The exponential growth of e-commerce has driven explosive demand for high-throughput automated warehouses, where hundreds of AGVs transport goods between storage shelves and packing stations. To achieve high operational throughput, AGV fleets require efficient and collision-free routing planning. However, the existing MAPF solutions like CBS [1, 3, 4] have critical limitations when applied to real-world scenarios: 1) They rely on discrete time steps, which mismatch the *continuous* nature of physical AGV movement, so the actual running time is longer; 2) Most are designed for one-shot batch tasks, not *lifelong* dynamic task streams in 24/7 warehouse operations, so the system experience periodic halts; 3) They ignore *kinematic constraints* (acceleration, deceleration, rotation) of physical AGVs, leading to unforeseen collisions and suboptimal performance; 4) They fail to balance online responsiveness and scalability for large AGV fleets and complex warehouse networks.

To address these gaps, we proposed a novel conflict-aware routing algorithm OHSMD (One-Hop Search with Motion Decomposition) [5]. Different from the discrete MAPF solutions in the robotic field, OHSMD originates from the time-dependent routing [2] in the spatial data management: by replacing the static time-dependent functions that provide the future travel time with the dynamic conflict-aware functions that record the AGV's future locations, the above four requirements could be satisfied naturally for the first time. Specifically, 1) OHSMD computes the travel time and distance precisely with the actual acceleration/deceleration, so it is naturally continuous and aligns with practice; 2) The planned routes serve as the future conflict information from the network's perspective, so the conflicts can be avoided beforehand; 3) The route-by-route scheduling aligns with the streaming tasks, so the system runs without a halt, and tasks can be rescheduled flexibly; 4) The action space pruning improves the computational efficiency so it can scales to larger networks with larger fleets.

In this demonstration, we implement a physical AGV routing system based on our OHSMD algorithm [5] using Sony Toio™ Core Cube robots. The demo provides an end-to-end solution from warehouse configuration, task scheduling, path planning, to physical robot control. Compared with existing MAPF demos that only focus on simulation and fixed scenarios, our system fully supports continuous time modeling, kinematic constraints, and lifelong dynamic tasks, with interactive user operations and physical validation.

- Building upon the OHSMD algorithm, we develop an end-to-end physical AGV scheduling and demonstration system, which simultaneously satisfies continuous, lifelong, conflict-free, and kinematic moving requirements.
- This demonstration is a physical and interactive validation tool for users to test the warehouse designs, task assignments, MAPF algorithms, and lifelong ability flexibly. Conference attendees are welcome to play with it using their own design.
- This demonstration also shows the superiority of OHSMD in practice in terms of higher system throughput, more accurate routing, and smoother lifelong ability.

2 SYSTEM OVERVIEW

As illustrated in Figure 1, our AGV system has four components: the *system configuration* takes the delivery task and warehouse design as input, the *lifelong task scheduler* assigns tasks to AGVs on the fly to keep the system operating continuously, the *routing engine*

Proceedings of the VLDB Endowment, Vol. 19, No. 12 ISSN 2150-8097.
doi:10.14778/3827998.3828081

* Lei Li is the corresponding author.

plans the conflict-free routes with detailed kinematic movements, the *virtual simulation* verifies the validness of the schedules and the *physical execution* directs the AGVs to operate in practice.

2.1 AGV System Configuration Interface

This module serves as the primary user interface for system setup that consists of three sub-components:

2.1.1 *Warehouse Map*. We model the warehouse as an industrial standard *directed grid network* $G(V, E)$. Here, V is the set of vertices representing the center of grids with three special types: *pickup locations* ($S \subseteq V$) for goods loading, *drop-off locations* ($D \subseteq V$) for goods unloading, and *parking buffers* ($P \subseteq V$) for idle AGV parking. $E \subseteq V \times V$ is the set of directed edges, restricting AGV movement to adjacent grids horizontally or vertically. This network can be modified flexibly through the editor: users can define network size ($m \times n$ grids), set directed or undirected edges between adjacent grids, set the special vertices, and place obstacles (*e.g.,* shelves, pillars) by removing vertices. It should be noted that the network design can be virtually reflected in the physical environment.

2.1.2 *AGV Fleet*. Users can also specify the fleet size (the number of AGVs $|A|$) and their physical parameters, including maximum velocity (v_{max}), constant acceleration and deceleration, and angular velocity for 0°/90°/180°/270° turns ($w_0, w_{90}, w_{180}, w_{270}$). The default parameters are calibrated to match the physical constraints of the Sony Toio™ Core Cube for seamless hardware validation.

2.1.3 *Task Editor*. A *task* is a triplet $\tau = (s, d, t)$, where $s \in S$ is the pickup location, $d \in D$ is the drop-off location, and t is the pickup time. $T = \tau_i$ is the task set at hand. These tasks can be added and edited on the fly as long as they have not been processed.

2.2 Lifelong Task Scheduler

This task scheduler manages the full life cycle of tasks and AGVs with the following two sub-modules:

2.2.1 *AGV Task Assignment*. When a new task $\tau = (s, d, t)$ arrives at time t, the module selects an AGV by estimating the earliest arrival time at the pickup location s. The estimation considers both the AGV's current state (position, speed, heading direction) and the Manhattan distance to s divided by v_{max}.

2.2.2 *Parking Location Allocation*. After an AGV completes delivery at d, the module allocates a parking position using our flexible buffer strategy [5]. It first identifies the column with the smallest horizontal distance from d that has unassigned parking spaces, then selects the topmost available buffer location in that column. This strategy leverages the upward-moving nature of AGVs towards pickup locations, reducing detours and improving throughput.

2.3 Conflict-Aware Routing Engine

This is the core of the system, implementing our proposed OHSMD algorithms [5]. We first present the overall procedure.

2.3.1 *Routing Procedures*. After receiving an assignment, the routing engine plans the route based on the following procedures: **1) State Initialization**: It first receives the selected AGV's current state (position, speed, heading direction) and the task's pickup location s, dropoff location d, and allocated parking position p. The path is planned in three segments: current position $\rightarrow s$, $s \rightarrow d$, and $d \rightarrow p$, with waiting time t_{wait} enforced at s. **2) Conflict-Aware

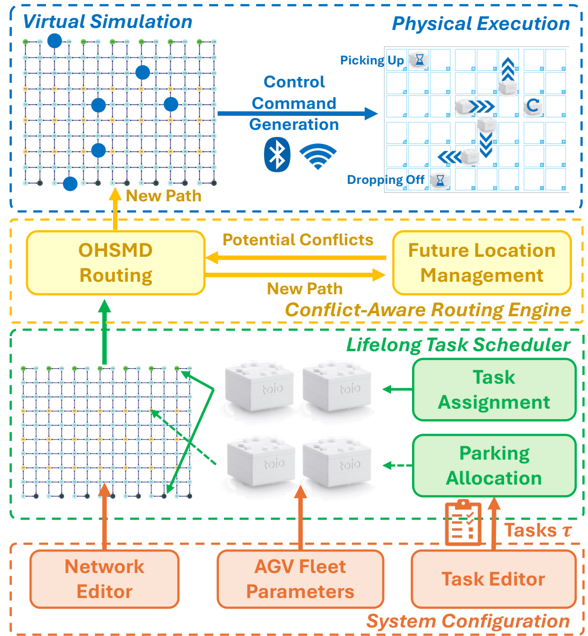

Figure 1: System Overview

Path Search: Then it executes the OHSMD algorithm (or other baselines) to find the fastest conflict-free path. All algorithms use our time-interval-based conflict detection mechanism, which maintains an Occupied Interval Sequence (OIS) for each grid, recording time intervals during which it is occupied by scheduled AGVs. The search expands states in a priority queue sorted by earliest arrival time, using a precomputed heuristic function that estimates the fastest travel time to the destination under kinematic constraints without conflicts. **3) Path Validation & OIS Update**: Once a valid path is found, the engine converts it into a sequence of Occupation Intervals (OIs) for all traversed grids and updates the global OIS to reserve these time intervals for the AGV, ensuring subsequent path searches respect this AGV's schedule.

2.3.2 *Conflict Detection and Management*. This part is the core of conflict avoidance, which is analogous to the time-dependent function but is highly dynamic. We first model the AGV kinematic movement and then convert it to the grid occupation. **AGV Kinematic Model**: AGVs follow a constant acceleration model with a maximum speed limit ($v_{\max}$) and a fixed angular velocity for rotation. An AGV must come to a complete stop before changing its heading direction. Each AGV has two kinematic states: *static*, with zero speed, in which it can perform staying, turning, or moving **actions**; and *moving*, with non-zero speed, in which it can only execute moving **actions** along its current heading direction. **Path, Occupation Interval, and Conflict**: An *AGV path* is a sequence of states $p = \langle (v_0, ac_0, t_0), \ldots, (v_k, ac_k, t_k) \rangle$, where (v_i, ac_i, t_i) denotes the AGV taking action ac_i at vertex v_i at time t_i. We define the *Occupation Interval (OI)* of a grid v_i by an AGV a_j as $O_i^j = [t^s, t^e]$, where a grid is considered occupied by an AGV if there exists spatial overlap between them, as illustrated in Figure 2. Specifically, t^s denotes the earliest time at which a_j starts to overlap with v_i, and t^e denotes the latest time at which the overlap ends. Two paths *conflict* if there exists a grid v such that their OIs on v overlap, and this conservative definition guarantees collision-free movement in continuous physical space. By managing and searching the OIS of the corresponding grids during pathfinding, the conflicts can be detected and avoided beforehand.

2.3.3 *OHSMD Routing Algorithm*. We first introduce the General Conflict-Aware Search (GCAS) that lays the groundwork for OHSMD: it adopts a Dijkstra-like search strategy to guarantee the fastest conflict-free path. The search state is defined as a tuple $(et, lt, v, \text{speed}, HD)$, where et and lt are the earliest and latest valid entry time to the grid vertex v, speed is the AGV's speed when arriving at v, and HD is the AGV's heading direction. The algorithm maintains a priority queue sorted by et, expands all valid conflict-free actions (staying, turning, acceleration, uniform motion, deceleration) for each state, and terminates when the destination is reached with zero speed. While GCAS fully models continuous time and kinematic constraints to ensure path optimality and collision avoidance, it suffers from an exponentially growing search space and prohibitive computational overhead for large-scale scenarios. The Multi-Hop Conflict-Aware Search (MHCAS) improves efficiency by merging multi-step movements in the same direction into a single "multi-hop" action, eliminating intermediate moving states. MHCAS matches real-world AGV control patterns (where AGVs receive commands to move k grids instead of single steps) and reduces the search space, but still generates a large number of states by exploring all possible multi-hop neighbors.

To address the limitations of GCAS and MHCAS, OHSMD combines the completeness of GCAS with the efficiency of MHCAS, achieving three orders of magnitude speedup while maintaining near-optimal path quality. The key insight behind OHSMD is that while real AGVs are controlled via multi-hop commands, their underlying motion follows deterministic kinematic laws that can be decomposed into constrained single-hop sub-actions. By enforcing these motion constraints during the search, we can drastically reduce the state space without losing path optimality. OHSMD decomposes any k-hop AGV command into a sequence of constrained single-hop sub-actions, following the physical laws of motion: 1) The AGV can only accelerate (A_{ma}) if it has not reached $v_{\max}$; 2) Uniform motion (A_{mu}) is only allowed when the AGV is at $v_{\max}$; 3) If the previous sub-action was a deceleration (A_{md}), the next sub-action can only be A_{md} (until the AGV stops); 4) To stop at a target grid, the AGV must decelerate to speed 0 exactly when reaching the grid center. These constraints eliminate invalid action combinations, drastically pruning the search space.

2.4 Visualization & Physical Execution

This layer provides two forms of result validation.

2.4.1 *Virtual Simulation*. It displays the grid network with color-coded zones (green for pickup, blue for drop-off, yellow for parking buffers). AGVs are visualized as circles with arrows indicating heading direction, and their movements are rendered in real time to visualize the future schedules.

2.4.2 *Control Command Generation and Physical AGV Execution*. This layer converts the planned path (sequence of grid positions and time intervals) into executable control instructions for AGVs. For physical robots, it converts the path into low-level motion commands (forward, turn, stop) with precise timing control, and communicates with physical AGVs via Bluetooth. For Sony Toio™ Core Cube AGVs, it converts commands into Bluetooth Low Energy (BLE) packets, including motor speed profiles, rotation angles, and timing synchronization signals. The module implements a feedback control loop that adjusts command timing based on real-time position data from physical AGVs to compensate for execution errors. The actual execution is monitored through a camera.

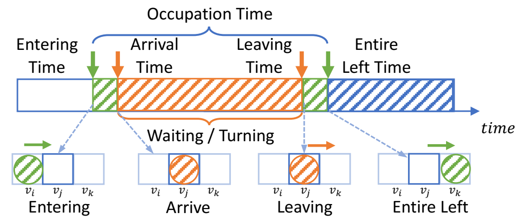

Figure 2: Grid Occupation Example

3 DEMONSTRATION SCENARIOS

This demonstration aims to provide a physical environment to evaluate the actual performance of the MAPF algorithms in practice. It should be noted that our OHSMD has been deployed for testing in industry with hundreds of AGVs with specialized WiFi connections, while this demonstration can only control six robots due to the space and Bluetooth's connected device limitation.

3.1 User Interface

The user interface is illustrated in Figure 3, and the user can interact with it through three steps: 1) On the left system configuration panels, users can edit the map topology, change the AGV parameters, and add or delete tasks; 2) On the right control panel, user can select the MAPF algorithms, monitor the algorithm and AGV running status with live command logs, and also control the AGVs directly through the bottom right panel by directing each AGV to its destinations; 3) The middle panels are the visualization, with the upper one showing the virtual simulation, and the lower one monitoring the physical execution through a live camera feed.

3.2 Demonstration Plans

We provide four case studies to showcase the flexibility of our demo system and also the superiority of our OHSMD algorithms, while it can also serve as a validation tool for future MAPF research.

3.2.1 *MAPF Algorithm Comparison*. The audience can toggle among three MAPF algorithms and compare their behavior physically: The CBS-S is the step-based execution that follows the discrete schedules strictly to avoid conflict, where each time the AGVs move to their neighboring grids. It is used to demonstrate the slowness of CBS. CBS-C is an adapted variant that merges consecutively executed motion commands in the same direction into a continuous action. It is used to demonstrate the potential conflict of CBS.

The audience first selects a routing algorithm and executes the same task set. The system visualizes AGV movements on both the virtual simulator and physical robot platform, while the dashboard reports runtime, makespan, and task results. This side-by-side comparison shows that CBS-S leads to stop-and-go movement and lower throughput due to its discrete execution model, while CBS-C carries collision risks when discrete schedules are applied to continuous physical space. In contrast, OHSMD natively integrates continuous kinematic constraints into routing, achieving both collision-free

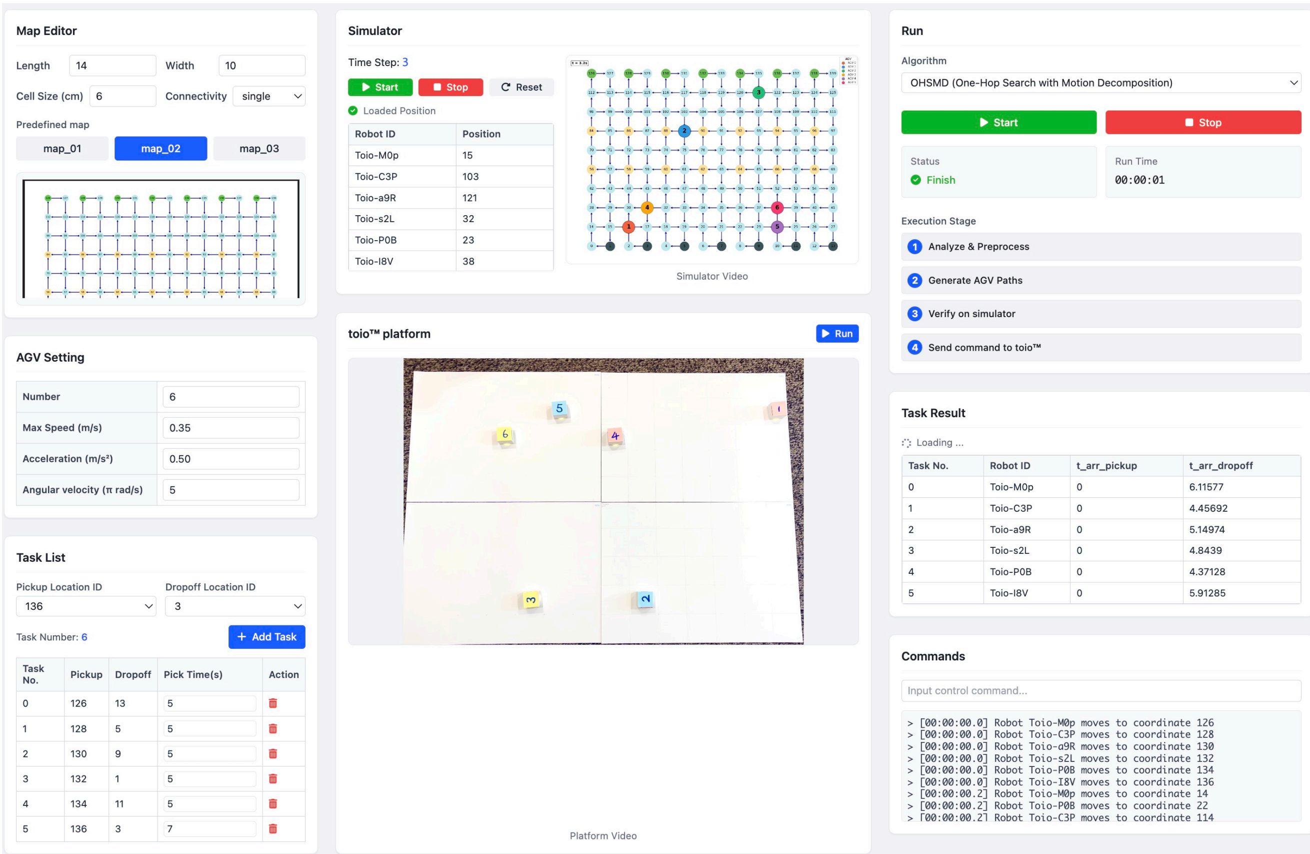

Figure 3: AGV Scheduling System Interface

execution and higher throughput, which demonstrates the importance of continuous routing with kinematic constraints in practice.

3.2.2 ***Interactive Warehouse Design***. The audience can alter the network topology or set up obstacles to test their warehouse design. We also provide two default layouts: 1) Upper-Lower layout, where the pickups are all on one side, and the drop-offs are all on the opposite side. It is widely used for transporting goods from one area to another; 2) Centralized Shelves, where the pickup locations are in the middle and drop-offs are along the surrounding boundaries. It is widely used to deliver goods from shelves. Attendees are encouraged to modify the network topology, add or remove obstacles, and re-run the routing algorithms. The updated routes are visualized, allowing attendees to observe how warehouse topology influences the performance of different MAPF algorithms and warehouse.

3.2.3 ***Task Assignment and Lifelong Operation***. The audience can experience the influence of the task assignment and the distribution of pickup and drop-off locations on the system throughput. Meanwhile, the audience can also add new tasks continuously to experience the flexibility of OHSMD's lifelong operation, compared with CBS's batch processing. The audience will observe that batch-based CBS approaches require periodic full fleet planning, which causes temporary operation halts and limits long-term throughput. In contrast, OHSMD incrementally schedules new tasks, enabling uninterrupted lifelong warehouse operations.

3.2.4 ***Large-Scale Simulation***. While only 6 physical robots are present due to Bluetooth connection, the audience can run pure virtual stress tests with the simulator using larger networks and AGV fleets to test the algorithm performance. The audience can increase the warehouse size and fleet size to investigate how routing latency and throughput change under increasing workload. This scenario validates the scalability of OHSMD and allows users to explore the trade-off between system scale and routing efficiency.

ACKNOWLEDGMENTS

This work was supported by National Key R&D Program of China Grant #2024YFA1012700, NSFC #62572414, and Guangdong-Hong Kong Technology Innovation Joint Funding Scheme #2024A0505040012.

REFERENCES

[1] Jiaoyang Li, Wheeler Ruml, and Sven Koenig. 2021. Eecbs: A bounded-suboptimal search for multi-agent path finding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 12353–12362.

[2] Lei Li, Wen Hua, Xingzhong Du, and Xiaofang Zhou. 2017. Minimal on-road time route scheduling on time-dependent graphs. *Proceedings of the VLDB Endowment* 10, 11 (2017), 1274–1285.

[3] Hang Ma, Jiaoyang Li, TK Satish Kumar, and Sven Koenig. 2017. Lifelong Multi-Agent Path Finding for Online Pickup and Delivery Tasks. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*. 837–845.

[4] Guni Sharon, Roni Stern, Ariel Felner, and Nathan R Sturtevant. 2015. Conflict-based search for optimal multi-agent pathfinding. *Artificial Intelligence* 219 (2015), 40–66.

[5] Ruizhong Wu, Mengxuan Zhang, Shuxin Wang, Frodo Kin Sun Chan, Yan Nei Law, and Lei Li. 2025. Continuous Lifelong Conflict-Aware AGV Routing with Kinematic Constraints. *Proceedings of the VLDB Endowment* 18, 7 (2025), 2254–2267.