Path Planning and Navigation Technologies for Autonomous Mobile Robots in Dynamic Indoor Environments

Shun Jiang*

Department of Mechanical Engineering, Xi'an Jiaotong University, Xian, China

*Corresponding author: jiangshun@stu.xjtu.edu.cn

Abstract. The widespread application of Autonomous Mobile Robots (AMRs) in dynamic indoor environments has exposed the shortcomings of traditional navigation systems. For example, they cannot solve problems, like moving obstacles, layout changes, and multi-robot collaboration. The development of artificial intelligence and sensor technologies speeds up the changes in AMR navigation systems. This paper contains the development of AMR navigation systems. It analyses the technical challenges brought by dynamic indoor environments, including problems such as interference from mobile obstacles, bottlenecks in location and map updates, and efficiency conflicts in multi-robot collaboration. Additionally, it explores the key technologies used to solve these challenges, including dynamic object detection and removal in dynamic SLAM, multi-sensor fusion, semantic enhancement of dynamic map construction, and distributed task allocation based on a market auction mechanism, conflict search, and spatiotemporal path coordination in multi-robot collaborative navigation. According to these studies, this paper guides the development of more robust and adaptive AMR navigation frameworks to meet the application needs of modern indoor robots.

Keywords: Autonomous Mobile Robots; Dynamic Indoor Environments; Path Planning; Navigation Technology; Simultaneous Localization and Mapping.

1. Introduction

In dynamic indoor environments, the rapid deployment of autonomous mobile robots (AMRs) has exposed critical defects in traditional navigation systems. In these environments, problems such as unpredictable obstacles, frequent changes in layout design, and the demand for multi-robot coordination often result in some bad consequences, such as navigation errors, delayed task completion, or system inefficiencies. For instance, in a busy hospital corridor, a robot could disrupt key medical material transportation if it cannot avoid sudden obstacles. In warehouses, the inefficient multi-robot path allocation might reduce productivity. These challenges emphasize the importance of customized path planning and navigation technologies in dynamic environments [1].

The development of artificial intelligence and sensor technology has sped up the revolution of AMR navigation systems. Compared to traditional automated guided vehicles that depend on static magnetic tracks or pre-determined routes, modern AMRs have achieved a new level that can accomplish real-time environmental perception, adaptive path planning, and collision-free navigation by integrating multi-modal sensors and AI-driven algorithms [2]. Core technologies, including SLAM and dynamic re-planning, can enable AMRs to operate autonomously in changing scenes constantly. For example, SLAM enables robots to update maps in real time. The reinforcement learning algorithms can optimize path planning for sudden obstacles. These innovations can enhance operational reliability and unlock new applications such as factory material handling and medical safety assistance.

This review explores how path planning and navigation technologies solve the internal challenges of dynamic indoor environments. Firstly, the paper analyses the history of AMR navigation systems, emphasizing the transformation from the rule-driven approach to the AI-enhanced solution. Secondly, we analyse the demands of dynamic environments, emphasizing the importance of these technologies, including high-precision orientation, real-time obstacle avoidance, and scalable multi-robot coordination. By comparing the differences between classical algorithms, learning-driven strategies, and hybrid approaches, this study reveals their strengths and limitations in practical scenarios. Finally, we use the cases including intelligent warehousing and industrial automation to show the distance between theory and reality. By summarizing these studies, this paper aims to guide researchers to develop better adaptive navigation frameworks to satisfy the demands of modern indoor robotics.

2. AMR Navigation and Path Planning

2.1. Components of Autonomous Mobile Robot Systems

The autonomous mobile robot system consists of four parts, including perception, localization and mapping, planning and decision-making, and motion control [2].

The perception system captures environmental data and processes this data to extract feature information by using sensors such as LiDAR and visual cameras. The localization and mapping module estimates real-time pose and constructs maps based on SLAM. The planning and decision-making module generates dynamic paths and replans in response to environmental changes by using task instructions and map data. The motion control module converts commands into instructions to accomplish precise trajectory tracking.

These modules interrelate by exchanging information and enable AMR systems to have the capabilities, including environmental perception, autonomous decision-making, and action execution, which enable the system to operate autonomously.

2.2. Classification of AMR Application Scenarios

The application scenarios of autonomous mobile robots (AMR) can be divided into structured and unstructured environments, and this classification influences the design of the robot's perception, planning, and control architecture [2].

In structured environments, the AMR can operate efficiently because the space is regular and static features are prominent. In such environments, robots ensure task execution by using positioning technologies and global path planning algorithms.

In unstructured environments, space is unpredictable, which requires the AMR to have capabilities such as real-time environment reconstruction and adaptive decision-making. The key to solving these technical challenges is to focus on robust perception and dynamic obstacle avoidance, and the AMR system should be combined with edge computing and online learning mechanisms to enhance system adaptability.

In the future, as cross-scenario collaboration demands increase, it is essential to develop hybrid AMR systems with both high precision and strong robustness.

3. Core Technical Challenges in Dynamic Indoor Environments

The dynamic indoor environments impose higher demands on the navigation capabilities of Autonomous Mobile Robots, including perception, planning, and collaboration. This section introduces three core challenges.

3.1. Moving Obstacles and Uncertainty Interference

In dynamic environments, the randomness of moving obstacles and environment layout is a threat to the robustness of AMR path planning. In high-dynamics environments such as warehouses and hospitals, AMRs’ path planning faces the threat of missing trajectory prediction and fixed safety margins because they need to share space with humans, other robots, or transportation equipment. So, the traditional global planning algorithms cannot predict the movement intentions of pedestrians or vehicles because they are all based on static assumptions, such as RRT, A*, and Dijkstra [3-6]. For example, the sudden movement of pedestrians in hospital corridors results in frequent emergency stops. At the same time, sudden changes in the environment aggravate the vulnerability of global path planning [7]. The problem of mismatch between static maps and real-time environments is prominent. Because the SLAM’ s initial map cannot be dynamically updated, the AMR may plan an invalid path that passes through actual obstacle areas. Even more challenging, AMR is unable to distinguish the priority of different obstacles because current navigation systems lack semantic labelling for movable objects, which is a huge threat to global path planning.

3.2. Real-Time Bottlenecks in High-Precision Localization and Map Updating

Dynamic environments impose higher requirements on localization accuracy and map updating efficiency. Existing technologies face significant bottlenecks. The interference from dynamic objects to SLAM systems cannot be ignored: blocking LiDAR feature points by pedestrians may result in inaccurate pose estimation, and although dynamic object filtering algorithms can improve mapping accuracy, at the cost of computing resources [8]. In large-scale environments, the problem is more serious than it is difficult for embedded devices to support the ever-increasing usage of high-resolution; the delay in loop detection leads to the inability to correct positioning errors in time, which may lead to navigation drift [9]. Moreover, the real-time performance has a serious conflict with accuracy in dynamic environments: increasing sensor fusion layers for higher accuracy results in computational load, while lightweight solutions may sacrifice environmental details, resulting in obstacle avoidance failures.

3.3. Collaboration Efficiency Conflicts in Multi-Robot Systems

In multi-AMR collaborative operations, task allocation and path conflicts become a huge problem. The fairness issue of task allocation cannot balance centralized and distributed strategies: greedy algorithms may lead to overload for some AMRs, and auction mechanisms may decrease overall efficiency because of bidding conflicts [10]. Path crossing in narrow passages is more likely to cause system-level deadlocks, such as the mutual blocking phenomenon of four-way shuttle cars at crossroads in warehouse scenarios. The threat of communication delays is also essential, robots may make actual collisions due to outdated information. Furthermore, multi-robot systems in dynamic environments may face scalability challenges: traditional collaborative algorithms experience significant growth in computational complexity when the number of robots exceeds 50, while completely distributed strategies may fall into suboptimal solutions due to limited local information [11].

3.4. Coupling Effects and Systemic Risks

These challenges are not isolated, and their coupling effects may cause huge Malfunctions. For example, localization drift may aggravate path planning failures, while multi-robot conflicts increase the computational load, forming a vicious circle. Therefore, AMR navigation in dynamic environments requires a full reconstruction of perception, decision, and control, rather than depending on optimizing a single technology.

4. Key Technologies for AMR Navigation

4.1. Environmental Perception and Dynamic SLAM

The core of dynamic SLAM is to address the challenges of localization and mapping in dynamic environments through multi-modal perception, semantic understanding, and adaptive learning.

4.1.1 Dynamic Object Handling

Dynamic object detection and removal is a key aspect of dynamic SLAM, which is to eliminate the interference of dynamic objects in localization and mapping. In dynamic environments, the feature points of moving objects may cause wrong matches and motion estimates, which will lead to localization drift and map distortion and disrupt the static consistency of the environmental structure. Detecting and removing dynamic features and retaining the static structure to optimize pose and update the map can improve the accuracy of SLAM.

Dynamic object detection methods based on geometric motion consistency distinguish between static and dynamic elements by analysing the motion trajectories of feature points. The core of this approach is to assuming that static feature points are same as the camera's motion model, and dynamic feature points exhibit completely different performance, whose Specific implementations include optical flow tracking and multi-view geometry validation: optical flow tracking compares the differences between feature point motion vectors and the camera's motion model, while multi-view geometry validation through peripolar constraints or triangulation errors to distinguish dynamic feature points. For example, the framework proposed by Xie enhances RGB-D SLAM accuracy in dynamic environments by using optical flow and multi-view geometry constraints [12].

Semantic segmentation methods based on deep learning recognize the categories of dynamic objects by training models and generate pixel-level masks to accurately remove dynamic regions by using instance segmentation networks. For example, the DS-SLAM framework by Yu et al. (2018) achieves a 40% reduction in localization error on the TUM dataset by combining ORB-SLAM2 with SegNet [13].

The joint optimization based on the two methods mentioned above establishes a Two-factor validation mechanism based on geometric motion confidence and semantic probability. For example, Dynamic-SLAM achieves a 95% recall rate in the KITTI dynamic scene by combining geometric motion clustering and semantic segmentation networks, and DynaSLAM II supports long-term trajectory prediction of dynamic objects by using tightly coupled multi-object tracking and optimizing SLAM [14,15].

4.1.2 Dynamic SLAM Based on Multi-Sensor Fusion

The dynamic SLAM based on multi-sensor fusion builds a robust perception system in dynamic environments by aggregating the advantages of multi-modal sensor data, such as LiDAR, cameras, IMUs, and event cameras. The key is utilizing the complementarity between sensors to overcome the inherent limitations of individual modalities.

LiDAR-vision fusion technology improves the robustness of localization and mapping in dynamic environments by using the complementary characteristics of LiDAR and visual sensors. LiDAR generates high-precision point cloud data with accurate depth perception and constructs the environmental geometry, though it is insensitive to changes in illumination. On the other hand, the camera captures relevant information to support feature matching and dynamic object semantic recognition. In technical implementations, LiDAR points can provide deep constraints through monocular visual features to solve the scale ambiguity problem. For example, motion interference regions can be accurately separated by comparing the static background point cloud from LiDAR with dynamic features from vision. A typical example is Huang et al. combined LiDAR and visual point-line features for camera tracking and optimized pose estimation through back-end bundle adjustment, which improves dynamic SLAM accuracy and robustness [16].

On the other hand, event cameras trigger events by observing pixel brightness variation, which have microsecond-level time resolution and high dynamic range characteristics. They are good at capturing high-speed motion and transient environmental changes. Due to the direct reflection of scene changes, such as object edge movement, the event cameras can extract motion information through event clustering or optical flow estimation. Compared to traditional cameras with a fixed 30Hz frame rate, which are prone to motion blur, event cameras can update local maps in real-time and respond quickly to environmental changes in dynamic SLAM. The fusion strategy often combines event cameras with IMUs, using the high-frequency data from event cameras and inertial measurements to improve the accuracy of high-speed motion pose estimation, while also compensating for data loss in traditional sensors during high-speed motion. Furthermore, multi-modal fusion with LiDAR or cameras can further enhance the accuracy of dynamic scene perception.

4.1.3 Semantic-Aware Dynamic Mapping

Semantic-enhanced dynamic map construction aims to incorporate semantic information (such as object categories and dynamic states) into the SLAM system, generating dynamic maps that not only describe the geometric structure of the environment but also contain semantic understanding.

The core objective of the semantic SLAM framework is to maintain long-term consistency of the dynamic map by real-time tracking of dynamic objects and continuously updating their positions in the map. Its implementation path involves front-end dynamic detection and back-end tightly coupled optimization: the front-end detects dynamic objects by combining geometric methods and semantic segmentation networks, while the back end treats dynamic objects as independent entities, using multi-object tracking algorithms (e.g., DeepSORT) to continuously update motion trajectories. At the same time, in the SLAM back-end, the camera pose, static map points, and dynamic object poses are optimized synchronously. Static elements participate in global optimization, while dynamic targets are independently updated based on local motion models. A typical example is the DynaSLAM II framework proposed by Bescos et al., which achieves long-term maintenance of dynamic object poses through multi-object tracking and tightly coupled optimization, reducing localization error by 35% in the KITTI dataset [17].

On the other hand, the online learning mechanism automatically generates training data for dynamic objects through self-supervised learning. The technical process first uses the pose and depth map output by SLAM to reconstruct the scene’s 3D geometric structure. Then, through geometric consistency checks, such as reprojection error analysis, dynamic regions (e.g., regions of point clouds with motion inconsistency) are automatically labeled. These dynamic regions are then used as training labels for the deep learning model (e.g., PointNet++) for dynamic object detection. Based on the model’s update results, SLAM pose and map are optimized in a closed loop to improve label quality. A typical example is the DeFlowSLAM method proposed by Ye et al., which uses a dual-flow field representation to fuse static/dynamic pixel information. Combined with a dynamic update module and dense bundle adjustment layer, it optimizes camera poses and depth estimation, achieving self-supervised dynamic SLAM training without manual labelling [18].

4.2. Multi-Robot Collaborative Navigation

The efficiency of multi-robot systems operating in dynamic and complex environments relies heavily on two core technologies: task allocation and path coordination. The main challenges lie in adapting dynamic task assignments to changes in task priorities and robot heterogeneity and resolving spatiotemporal conflicts arising from intersecting paths and kinematic constraints.

4.2.1 Distributed Task Allocation Based on Market Auction Mechanism

The market auction mechanism, based on simulating resource bidding behaviour in a market economy, implements decentralized task-resource matching as its core idea, enabling autonomous task allocation in multi-robot systems. The process is divided into three stages: First, after the task is published, each robot calculates the task execution cost (such as path length and energy consumption) based on its status (e.g., position, battery level, load capacity) and submits a bid [19]. Then, distributed algorithms (such as consensus protocols and greedy strategies) are used to optimize the bidding information globally, ensuring the maximization of the system’s total profit or minimization of total cost, resulting in the final task allocation decision. Finally, the winning robot locks in the task and executes it, while non-winning robots continue to participate in subsequent auction cycles. The key technical features of this mechanism include its decentralized architecture—no central controller is required, and robots only need to communicate with neighbours or within a local network to complete negotiations; dynamic adaptability—supporting real-time task priority adjustment, robot fault recovery, and the insertion of sudden tasks; and heterogeneous compatibility—flexibly modelling robot capability differences (e.g., speed, sensor configuration), adapting to the collaborative needs of complex heterogeneous systems [20].

Building upon this, combinatorial auction methods allow buyers to bid on sets of items collectively, with the technical implementation using a hierarchical task processing strategy: first, all potential combination bids are aggregated, and the complexity of solving the WDP (Winner Determination Problem) is reduced using mixed integer programming, heuristic algorithms, or parallel block processing, while local optimization is used to improve efficiency[21]. The combination design must balance flexibility with computational feasibility—predefined high-value bundles can simplify auction logic, and machine learning can dynamically discover high-frequency complementary combinations from historical data, avoiding a combinatorial explosion through logical constraints. When faced with budget constraints from buyers, the system implements bi-directional optimization: the buyer side selects the optimal strategy based on dynamic programming or marginal utility, while the auction side verifies budget compliance through constraint programming, introducing the VCG mechanism to suppress malicious overspending, and uses delayed allocation and robust optimization to handle bidding uncertainty [22]. These auctions rely on solvers like CPLEX/Gurobi and game equilibrium of mechanism design, such as the framework proposed by Wen and Zhao (2021), which uses dynamic task combinations and budget-constrained social welfare maximization strategies, combined with hierarchical task decomposition to address NP-hard problems, significantly reducing execution costs in dynamic environments [23].

Deep auction, as an emerging paradigm, optimizes task allocation by integrating deep reinforcement learning with market mechanisms: deep neural networks dynamically adjust robot bids by predicting the potential value of tasks in real-time, and shared strategy coordination is achieved through multi-agent collaborative modelling to avoid resource conflicts. Combining priority experience replay technology, the sample priority is adjusted by temporal difference errors, accelerating strategy convergence. Its technical implementation builds state spaces (robot state and task attribute encoding vectors) and action spaces (accept/reject task) using Markov decision processes, optimizing multi-agent collaboration with utilitarian objectives. Robots share Q-values to prioritize tasks with the highest global benefits, avoiding resource duplication issues caused by independent bids in traditional auctions. This method has been validated in an e-commerce RMFS scenario, reducing task allocation time by 14.91%-29.15% compared to traditional mechanisms, demonstrating the technical advantages of deep auction in dynamic and complex environments [24].

4.2.2 Conflict-Based Search (CBS) and Spatiotemporal Path Coordination

In multi-robot collaborative task allocation, Conflict-Based Search (CBS) and spatiotemporal path coordination techniques are core methods for solving dynamic path planning and resource conflicts, particularly suitable for high-density, high-dynamic warehouse logistics scenarios (such as e-commerce RMFS). CBS, as a hierarchical optimization algorithm for multi-agent path planning (MAPF), achieves global coordination through dynamic detection and resolution of spatiotemporal conflicts. The core process defines three types of spatiotemporal conflicts: vertex conflicts (multiple robots occupying the same location simultaneously), edge conflicts (path intersections due to robots moving towards each other), and resource conflicts (competing for shared devices). These conflicts are resolved layer by layer by generating a constraint tree, where each node represents a set of path constraints. Branch constraints are added iteratively to optimize the path solution, with the underlying A or Dijkstra algorithm used to generate locally optimal paths for each robot that satisfy the current constraints. The CBS algorithm is highly scalable and can be integrated with technologies such as discontinuity-bounded A* algorithms, hybrid A algorithms, Kino dynamic conflict-based search (K-CBS), Traveling Salesman Problem algorithms, and genetic algorithms, to enhance efficiency and adaptability [25-29].

The complementary spatiotemporal path coordination mechanism addresses the issue of path overlap caused by the neglect of temporal sequencing in traditional methods by introducing a time dimension. It transforms the planning problem into a spatiotemporal domain search. This is done by modelling the search space with a spatiotemporal grid graph, where each node contains spatial coordinates and timestamps [30]. Agents are allowed to pass through a node only when it is in the "free" state, thus avoiding vertex and edge conflicts. This mechanism models conflict conditions (such as preventing multiple agents from occupying the same spatiotemporal node simultaneously), achieving a balance between path safety and system throughput in dynamic scenarios.

5. Conclusion

In dynamic indoor environments, the navigation of AMRs faces many challenges, and traditional navigation systems cannot satisfy these demands. With the development of artificial intelligence and sensor technologies, AMR navigation systems have realized significant innovations.

AMR systems contain four parts: perception, localization and mapping, path planning and decision-making, and motion control. Their application environment can be divided into structured and unstructured environments. The main challenges brought by dynamic indoor environments include: interference in path planning caused by moving obstacles and changes in the environment layout which can cause traditional algorithms difficult to adapt; real-time bottlenecks in high-precision localization and map updates which will be more serious under the influence of dynamic object interference, memory consumption and high computational load; low efficiency in multi-robot collaboration including unfair task allocation, path conflicts, communication delays and poor system scalability. These challenges are interrelated and threaten the overall system.

To solve these challenges, various key technologies have emerged. Dynamic SLAM technology has improved the accuracy of localization and mapping by using dynamic object detection and removal, multi-sensor fusion, and semantic enhancement of dynamic map construction. Multi-robot collaborative navigation has achieved efficient collaboration by using distributed task allocation based on a market auction mechanism, conflict search, and spatiotemporal path coordination techniques.

Future research should concentrate on improving navigation performance in dynamic environments, such as developing more efficient dynamic object recognition and tracking algorithms to enhance adaptability to complex scenarios; optimizing multi-robot collaborative algorithms to reduce computational complexity; and exploring new sensor fusion methods to obtain more comprehensive and accurate environmental information. Additionally, it is important to combine theoretical research with practical deployment to speed up the application of AMR technologies in industries such as intelligent logistics and medical services, to better meet the demands of modern society for intelligent technology.

References

[1] Alatise M B, Hancke G P. A review on challenges of autonomous mobile robot and sensor fusion methods[J]. IEEE Access, 2020, 8: 39830-39846.

[2] Siegwart R, Nourbakhsh I R, Scaramuzza D. Introduction to autonomous mobile robots[M]. MIT Press, 2011.

[3] Shan E, Dai B, Song J, et al. A dynamic RRT path planning algorithm based on B-spline[C].2009 Second International Symposium on Computational Intelligence and Design. IEEE, 2009, 2: 25-29.

[4] Kuffner J J, LaValle S M. RRT-connect: An efficient approach to single-query path planning[C]. Proceedings 2000 ICRA. Millennium conference. IEEE international conference on robotics and automation. Symposia proceedings (Cat. No. 00CH37065). IEEE, 2000, 2: 995-1001.

[5] Stentz A. Optimal and efficient path planning for partially-known environments[C]. Proceedings of the 1994 IEEE international conference on Robotics and Automation. IEEE, 1994: 3310-3317.

[6] Javaid A. Understanding Dijkstra's algorithm[J]. Available at SSRN 2340905, 2013.

[7] Floreano D, Godjevac J, Martinoli A, et al. Design, control, and applications of autonomous mobile robots [M]. Advances in intelligent autonomous systems. Dordrecht: Springer Netherlands, 1999: 159-186

[8] Xiao L, Wang J, Qiu X, et al. Dynamic-SLAM: Semantic monocular visual localization and mapping based on deep learning in the dynamic environment[J]. Robotics and Autonomous Systems, 2019, 117: 1-16.

[9] Cummins M, Newman P. FAB-MAP: Probabilistic localization and mapping in the space of appearance[J]. The International Journal of Robotics Research, 2008, 27(6): 647-665.

[10] Bai X, Fielbaum A, Kronmüller M, et al. Group-based distributed auction algorithms for multi-robot task assignment[J]. IEEE Transactions on Automation Science and Engineering, 2022, 20(2): 1292-1303.

[11] Bertsekas D P. Auction algorithms for network flow problems: A tutorial introduction[J]. Computational optimization and applications, 1992, 1: 7-66.

[12] Xie W, Liu P X, Zheng M. Moving object segmentation and detection for robust RGBD-SLAM in dynamic environments[J]. IEEE Transactions on Instrumentation and Measurement, 2020, 70: 1-8.

[13] Yu C, Liu Z, Liu X J, et al. DS-SLAM: A semantic visual SLAM towards dynamic environments[C]. 2018 IEEE/RSJ international conference on intelligent robots and systems (IROS). IEEE, 2018: 1168-1174.

[14] Henein M, Zhang J, Mahony R, et al. Dynamic SLAM: The need for speed[C].2020 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2020: 2123-2129.

[15] Bescos B, Campos C, Tardós J D, et al. DynaSLAM II: Tightly-coupled multi-object tracking and SLAM[J]. IEEE robotics and automation letters, 2021, 6(3): 5191-5198.

[16] Huang S S, Ma Z Y, Mu T J, et al. Lidar-monocular visual odometry using point and line features[C].2020 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2020: 1091-1097.

[17] Bescos B, Campos C, Tardós J D, et al. DynaSLAM II: Tightly-coupled multi-object tracking and SLAM[J]. IEEE Robotics and Automation Letters, 2021, 6(3): 5191-5198.

[18] Ye W, Yu X, Lan X, et al. DeFlowSLAM: Self-supervised scene motion decomposition for dynamic dense SLAM[J]. arXiv preprint arXiv:2207.08794, 2022, 2.

[19] Bertsekas D P. The auction algorithm for assignment and other network flow problems: A tutorial[J]. Interfaces, 1990, 20(4): 133-149.

[20] Liu L, Shell D A. Optimal Market-based Multi-Robot Task Allocation via Strategic Pricing[C].Robotics: Science and Systems. 2013, 9(1): 33-40.

[21] Lagoudakis M G, et al. Auction-Based Multi-Robot Routing[C].Robotics: Science and Systems. 2005, 5: 343-350.

[22] Otte M, et al. Auctions for multi-robot task allocation in communication-limited environments[J]. Autonomous Robots, 2020, 44: 547-584.

[23] Wen X, Zhao Z G. Multi-robot task allocation based on combinatorial auction[C].2021 9th International Conference on Control, Mechatronics and Automation (ICCMA). IEEE, 2021: 27-32.

[24] Yuan R, Dou J, Li J, et al. Multi-robot task allocation in e-commerce RMFS based on deep reinforcement learning[J]. Mathematical Biosciences and Engineering: MBE, 2023, 20(2): 1903-1918.

[25] Moldagalieva A, Ortiz-Haro J, Toussaint M, et al. db-cbs: Discontinuity-bounded conflict-based search for multi-robot kinodynamic motion planning[C].2024 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2024: 14569-14575.

[26] Bai Y, Kotpalliwar S, Kanellakis C, et al. Multi-agent Path Planning Based on Conflict-Based Search (CBS) Variations for Heterogeneous Robots[J]. Journal of Intelligent & Robotic Systems, 2025, 111(1): 26.

[27] Kottinger J, Almagor S, Lahijanian M. Conflict-based search for multi-robot motion planning with kinodynamic constraints[C].2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2022: 13494-13499.

[28] Ren Z, Rathinam S, Choset H. Conflict-Based Steiner Search for Multi-Agent Combinatorial Path Finding[C].Proceedings of Robotics: Science and Systems. 2022.

[29] Liang Y, Zhang Z, Luo Y, et al. Multi-robot Task Allocation and Path Planning Method Based on Improved Genetic Algorithm and Conflict-Based Search[C].China Intelligent Robotics Annual Conference. Singapore: Springer Nature Singapore, 2024: 272-285.

[30] Hönig W, Kumar T K, Cohen L, et al. Multi-agent path finding with kinematic constraints[C].Proceedings of the International Conference on Automated Planning and Scheduling. 2016, 26: 477-485.