Proceedings of CONF-SPML 2026 Symposium: The 2nd Neural Computing and Applications Workshop 2025
DOI: 10.54254/2755-2721/2026.TJ31029

A Survey of Multi-Robot Collaborative Operation Scheduling Algorithms

Yiting Zhao

School of Software, Taiyuan University of Technology, Taiyuan, China
2739137223@qq.com

Abstract. Multi-robot collaborative operations have demonstrated significant potential in complex scenarios such as logistics, exploration, and rescue due to their efficiency and robustness. The core of their effectiveness lies in the collaborative operation scheduling algorithm, whose efficiency directly determines the overall system performance. This paper systematically reviews and synthesizes research in this field, aiming to provide researchers with a clear technical overview and development path. Specifically, it elaborates on the principles and typical methods of three core algorithms: task allocation, path planning, and communication coordination, encompassing approaches ranging from market-based distributed strategies to optimization-theoretic centralized methods. Furthermore, typical application cases such as logistics warehousing, underground exploration, and disaster relief are analyzed to illustrate the practical effectiveness of different algorithms. Finally, the main challenges currently confronting the field are summarized, along with an outlook on future research directions for collaborative operations.

Keywords: Multi-robot collaborative operation, path planning, scheduling application

1. Introduction

Driven by the sustained advancement of Industry 4.0, multi-robot collaborative systems (MRCS) have emerged as a pivotal solution for domains such as intelligent warehousing, leveraging their inherent scale advantages [1]. As the "central hub" orchestrating multi-robot interaction, the collaborative scheduling algorithm serves as the decisive factor in system efficiency. Despite significant progress in task allocation, path planning, and communication coordination, the core challenge remains: how to ensure safe and efficient collaboration in dynamic and unknown environments. While existing approaches have established robust frameworks assessed against metrics such as decision-making efficiency and energy consumption, they are still hindered by issues such as response latency and limited dynamic adaptability. Consequently, this paper presents a systematic review of collaborative scheduling algorithms, by analyzing their technical characteristics and current limitations. Furthermore, it forecasts future research directions aiming to guide the optimization and practical application of multi-robot systems.

DOI: 10.54254/2755-2721/2026.TJ31029

2. Multi-robot cooperative scheduling algorithms

Collaborative scheduling algorithms serve as the core decision-making component of multi-robot systems. Based on their functional objectives, these algorithms fall primarily into three categories: task allocation, path planning, and communication coordination

2.1. Task allocation algorithms

The core challenge of task allocation lies in establishing an optimal mapping between tasks and available robots [2,3]. The selection of a specific algorithm depends heavily on the application contexts and the underlying scheduling architecture. Generally, current mainstream algorithms can be categorized into three distinct types.

Optimization-based centralized approaches [4] leverage a central node to process global state information and utilize rigorous mathematical models to achieve optimal task assignment. Prominent examples include the Hungarian algorithm for bipartite matching and Mixed Integer Linear Programming (MILP) for scenarios involving complex constraints [5]. While these methods offer theoretical guarantees of optimality, they encounter significant scalability limitations. As the problem size increases, computational complexity grows rapidly, resulting in a pronounced performance bottleneck.

In contrast, heuristic-based centralized algorithms offer a more practical alternative. By employing heuristic or metaheuristic search strategies, often simulating natural evolution or swarm intelligence, these methods are capable of obtaining satisfactory near-optimal solutions within a finite time.

Market-based distributed algorithms represent a current research focus. Among these methods, the Consensus-Based Bundle Algorithm (CBBA) [6] is widely utilized for its superior performance. Given its prominence in distributed systems, this section elaborates on its core mechanism: CBBA is a fully distributed algorithm that rapidly converges on a conflict-free task allocation solution through local communication and iterative negotiation. By decomposing global optimization into local decision-making, CBBA demonstrates strong scalability, rendering it highly suitable for dynamic scenarios such as search and rescue missions.

2.2. Path planning algorithm

Once task allocation is finalized, multi-robot path planning emerges as a pivotal stage to ensure efficient system operation. The primary challenge lies in generating trajectories for all robots that are collision-free. Generally, MRPP algorithms fall into coupled and decoupled approaches. Coupled methods treat the entire robot fleet as a single agent within a high-dimensional joint state space. While this guarantees the completeness and optimality of the solution, the computational complexity increases exponentially with the number of robots, making it feasible only for small-scale systems [7].

In contrast, decoupled methods significantly enhance computational efficiency by decomposing the global problem into two stages: individual path planning and subsequent conflict resolution. A prominent example includes Conflict-Based Search (CBS) [8] and its variants. CBS operates by searching in the low-dimensional space of individual robots and imposes constraints only when a specific conflict is detected, thus substantially enhancing planning efficiency.

For dynamic or unknown environments, distributed algorithms are widely adopted, such as Optimal Reciprocal Collision Avoidance (ORCA) [9] and spatiotemporal constraint-based planning

DOI: 10.54254/2755-2721/2026.TJ31029

[10]. In these approaches, agents adjust their velocities in real-time based on local sensing, effectively avoiding dynamic obstacles. A comparative analysis of mainstream path planning algorithms is presented in Table 1 [11].

Table 1. Comparison of mainstream path planning algorithms

algorithm	Security	Path length	time consuming
Artificial potential field method	Easily trapped in local minima	Non-shortest path	Good real-time performance
D' algorithm	High	Can find the shortest path	better
DWA Algorithm	Real-time obstacle avoidance	Non-shortest path	better
Bug Algorithm	Go around the edge of the obstacle	Possibly not the shortest path	better
Diikstra algorithm	High efficiency, suitable for static environments.	Can find the shortest path	Large graphs are inefficient
A' algorithm	High	Can find the shortest path	Heuristic function influence
RRT algorithm	Random sampling to avoid collisions	Shortest path not guaranteed	The search process takes time.
Particle Swarm Optimization	High, swarm intelligence avoids	Possibly not the shortest path	better
Genetic Algorithm	Iterative optimization to avoid collisions	possible to find the shortest path	It takes a long time

3. In-depth analysis of centralized and distributed scheduling

The scheduling architecture defines the distribution of information and decision-making authority within the system. As the fundamental design choice in multi-robot systems, it establishes the system's operational capabilities, directly determining performance boundaries, scalability, robustness, and execution efficiency.

3.1. Centralized scheduling

Centralized scheduling architectures depend on a powerful central server to serve as the core decision-maker [12]. The central controller continuously collects status updates and sensory data from all agents to synthesize a globally consistent world model. Leveraging this holistic view, the system executes global planning algorithms to generate optimal solutions, ultimately disseminating precise control directives to each robot.

The primary advantage of centralized architectures is their ability to achieve global optimization. With a comprehensive world model, the central node can orchestrate highly coupled tasks and optimize global objectives—such as minimizing total fleet energy consumption—far more effectively than local strategies. Conversely, this approach introduces critical challenges, including exponential increase in computational load, communication bottlenecks, and vulnerability to Single Points of Failure (SPOF). Therefore, it is most suitable for static, structured environments with a fixed number of robots, as exemplified by automated production lines and intelligent warehousing [13].

DOI: 10.54254/2755-2721/2026.TJ31029

3.2. Distributed scheduling

In stark contrast to centralized scheduling, distributed scheduling decentralizes decision-making authority to individual robots in the system. Each robot is an autonomous decision-making unit that makes decisions autonomously based on its own local perception and limited communication with neighboring robots. The overall coordination of the system (such as the bidding rules in CBBA) emerges from pre-defined local interaction rules. The core advantage of this design lies in its excellent scalability and robustness. When the system scales up, it only brings an increase in local computation and communication load. The failure of a single robot can be "degraded" through task reassignment rather than system collapse. The cost lies in the difficulty of ensuring global optimality. Additionally, it must address inherent challenges in distributed systems, including decision conflicts, deadlock prevention, and consistency maintenance [14]. Therefore, distributed scheduling is suitable for large-scale robot clusters, dynamic unknown environments (such as field exploration [15]) and communication-constrained scenarios.

4. In-depth analysis of application cases of multi-robot cooperative operation scheduling algorithms

4.1. Logistics sector

Logistics warehousing (exemplified by the Amazon Kiva system) represents the most successful commercial application of centralized scheduling. This scenario is characterized by a highly structured environment and clear, deterministic tasks. The primary challenge is to maximize global efficiency while strictly ensuring collision-free operations. This has led to the development of a hybrid strategy that integrates "global path planning" with "local dynamic obstacle avoidance."

In this architecture, the central server maintains real-time data on the location and inventory status of all Automated Guided Vehicles (AGVs), utilizing operations research techniques for high-level task allocation. At the path planning level, the server employs an improved A* algorithm to generate globally optimal paths.

Chen Haixia et al. [16] noted that the traditional A* algorithm suffers from issues such as redundant nodes and sharp turning angles. By introducing optimization criteria to smooth path nodes, both the operational efficiency and kinematic adaptability of the AGVs are significantly improved. However, global planning alone cannot cope with sudden contingencies, such as human intrusion or falling goods. In such instances, the Artificial Potential Field (APF) method, improved by Cheng Pan et al. [17], demonstrates its value. The traditional APF method is prone to being trapped in local minima. The improved algorithm ensures that the robot can reliably escape these minima by incorporating a gravitational influence factor and intelligent optimization algorithms. The success of logistics scenarios ultimately relies on this synthesis of global optimization and local robustness within a centralized framework.

4.2. Exploration and inspection scenarios

In the field of exploration, multi-robot systems face three primary challenges: environmental dynamism, map uncertainty, and communication limitations [18]. Consequently, system robustness and autonomous safety are prioritized over global optimality. Underground coal mine inspection serves as a typical "semi-structured" application scenario: while the roadway topology is known, the

DOI: 10.54254/2755-2721/2026.TJ31029

environment is fraught with unknown dynamic hazards such as falling rocks and water accumulation [19].

These characteristics render purely centralized approaches (vulnerable to communication failures) or purely distributed ones (suffering from low global efficiency) suboptimal. The best practice for this scenario is a layered hybrid strategy.

At the global planning level, the system leverages the known map and an improved Ant Colony Optimization (ACO) algorithm to perform centralized offline planning, generating a macro-optimal inspection path.

At the local execution level, robots switch to a distributed autonomous mode, utilizing onboard sensors for real-time perception. Upon detecting an unmapped obstacle, the robot immediately activates a reactive mode—such as the optimized Artificial Potential Field (APF) method—to achieve autonomous avoidance, before smoothly realigning with the predetermined path once safety is ensured.

This integrated strategy of "centralized global planning + distributed local obstacle avoidance" effectively balances overall efficiency with local safety in exploration tasks. The path planning process is illustrated in Figure 1.

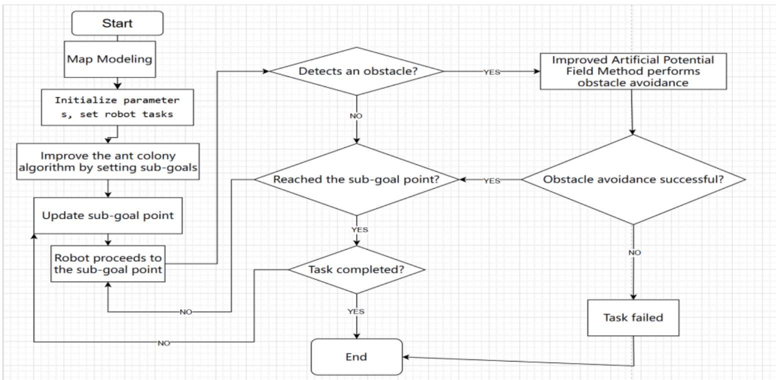

Figure 1. Path planning flowchart

4.3. Rescue field

Disaster relief is the most extreme scenario faced by multi-robot collaboration, characterized by unknown environments, high dynamism, extremely limited communication, and high timeliness requirements. In such an environment, any centralized scheduling that relies on accurate global information and stable communication will fail.

The system typically adopts a “layered hybrid” architecture: a high-level command is formed by human commanders [20] and unmanned aerial vehicles (UAVs) [21] in the rear command center. UAVs provide a macro view from high altitude and construct a rough map of the disaster area. Based on this information, the command center decomposes high-level tasks, for example, “buildings have collapsed in areas A, B, and C, which have the highest priority”, and assigns these macro tasks to different ground robot formations. This is a centralized decision-making process of human-machine collaboration. Ground robots (UGVs) are responsible for the bottom-level execution. After receiving general instructions, they execute tasks in a completely distributed autonomous manner. They need to autonomously perform local environment exploration (SLAM), dynamic path planning (such as D* or RRT), and target recognition. Task allocation is also implemented in a distributed manner. For example, using the CBBA algorithm, the robot will “bid” in real time based on its own state (such as distance and ability) and newly discovered tasks (such as

DOI: 10.54254/2755-2721/2026.TJ31029

detecting heat sources) to achieve local dynamic self-organization. This architecture maximizes the robustness and response speed of the system, allowing the robot to prioritize the core missions of “searching” and “discovering” even without a global optimal solution.

5. Conclusion

This paper provides a systematic review of multi-robot collaborative task scheduling algorithms. Starting with the three core issues of task allocation, path planning, and communication coordination, it delves into the essential differences and applicable scenarios of centralized and distributed scheduling architectures. Through in-depth analysis of three typical application cases—logistics, exploration, and rescue—it demonstrates the inevitability of the strong coupling between scheduling strategies and application scenarios: scheduling algorithms are evolving towards higher intelligence, stronger robustness, and enhanced distributed autonomy.

Despite significant progress in multi-robot collaboration, numerous challenges remain. First, scheduling algorithms for heterogeneous collaboration are still immature. Second, the robustness of these algorithms in highly adversarial or extreme dynamic environments needs improvement. Finally, the lack of cross-platform standardization hinders interoperability between robots from different manufacturers. Looking ahead, two key trends deserve close attention: first, the deep integration of artificial intelligence, utilizing deep reinforcement learning to train robots to autonomously learn optimal collaborative strategies in unknown environments; and second, human-robot collaboration, incorporating humans as the highest command node into the scheduling loop to achieve complementary human and machine strengths, which is particularly crucial in rescue and other scenarios.

References

[1] Parker, L. E. (2008). Multiple mobile robot systems. In Springer Handbook of Robotics (pp. 921-942). Springer.
[2] Gerkey, B. P., & Matarić, M. J. (2004). A formal analysis and taxonomy of task allocation in multi-robot systems. The International Journal of Robotics Research, 23(9), 939-954.
[3] Korsah, G. A., Stentz, A., & Dias, M. B. (2013). A comprehensive taxonomy for multi-robot task allocation. The International Journal of Robotics Research, 32(12), 1495-1512.
[4] Dias, M. B., Zlot, R., Kalra, N., & Stentz, A. (2006). Market-based coordination: A survey and analysis. Proceedings of the IEEE, 94(7), 1257-1275.
[5] Jones, E. R., Dias, M. B., & Stentz, A. T. (2011). Time-extended multi-robot coordination for domains with intra-path constraints. Robotics and Autonomous Systems, 59(5), 326-337.
[6] Choi, H. L., Brunet, L., & How, J. P. (2009). Consensus-based decentralized auctions for robust task allocation. IEEE Transactions on Robotics, 25(4), 912-926.
[7] LaValle, S. M. (2006). Planning algorithms. Cambridge university press.
[8] Sharon, G., Stern, R., Felner, A., & Sturtevant, N. R. (2015). Conflict-based search for optimal multi-agent pathfinding. Artificial Intelligence, 219, 40-66.
[9] Van den Berg, J., Lin, M., & Manocha, D. (2008). Reciprocal velocity obstacles for real-time multi-agent navigation. In 2008 IEEE International Conference on Robotics and Automation (pp. 1928-1935).
[10] Ma, H., Li, J., Lamont, G. B., & Wang, H. (2017). Multi-robot path planning with temporal-spatial constraints in dynamic environments. IEEE Transactions on Automation Science and Engineering, 15(3), 1369-1380.
[11] Liu Tianbao, Liu Mei. (2025). A review of research on path planning algorithms for mobile robots. Journal of Guangdong University of Petrochemical Technology, 35(4), 62-68.
[12] Arsie, A., Savla, K., & Frazzoli, E. (2009). Efficient algorithms for optimal multi-robot/task assignment. In 2009 IEEE International Conference on Robotics and Automation (pp. 433-438).
[13] Wurman, P. R., D'Andrea, R., & Mountz, M. (2008). Coordinating hundreds of autonomous robots in distribution centers. AI magazine, 29(1), 9-9.

[14] Olfati-Saber, R., Fax, J. A., & Murray, R. M. (2007). Consensus and cooperation in networked multi-agent systems. Proceedings of the IEEE, 95(1), 215-233.
[15] Stentz, A. (2000). The focused D* algorithm for real-time replanning. In IJCAI (Vol. 16, pp. 1652-1659).
[16] Chen, Haixia, & Wu, Fuyun. (2025). Research on path planning for warehouse logistics robots based on improved A* algorithm. Wireless Interconnection Technology, 18, 39-42.
[17] Cheng, Pan, & Geng, Hua. (2025). Path planning for logistics handling robots based on artificial potential field method. Mechanical Engineering and Automation, 3, 36-39.
[18] Yan, Z., Jouandeau, N., & Ali, A. C. (2013). A survey and analysis of multi-robot coordination. International Journal of Advanced Robotic Systems, 10(12), 399.
[19] Yang, Y. (2025). Research on Improvement of Inspection Path Algorithm for Underground Coal Mine Robot. Coal Mine Machinery, 46(7), 221-225.
[20] Murphy, R. R. (2004). Human-robot interaction in rescue robotics. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 34(2), 138-153.
[21] Michael, N., Mellinger, D., Lindsey, Q., & Kumar, V. (2011). The GRASP multiple micro-UAV testbed. IEEE Robotics & Automation Magazine, 18(3), 56-65.