Multi-Robot Cooperative Path Planning: Advances, Challenges, and Future Trends

Wenbo Zhang*

Department of Applied Science, Simon Fraser University, Vancouver, Canada

*Corresponding author:wza122@sfu.ca

Abstract. From automated warehouse robots coordinating shelf movements to drones and ground vehicles navigating disaster zones, multi-robot systems (MRS) are transforming how complex tasks are performed in dynamic environments. Unlike single-robot systems, MRS can execute tasks in parallel, cover large areas efficiently, and adapt to unexpected changes. These advantages make them ideal for applications such as smart manufacturing, autonomous logistics, and emergency response. However, as robot teams scale up, they face growing challenges—task allocation becomes harder, path conflicts increase, and communication delays can undermine real-time coordination. This paper provides a structured review of cooperative path planning approaches for MRS. First, outline the fundamental characteristics of multi-robot coordination and key challenges such as collision avoidance, computational complexity, and dynamic re-planning. Then, categorize the main planning algorithms into three types: graph-based methods, intelligent optimization techniques, and deep learning models. We also compare centralized and distributed planning frameworks in terms of scalability, robustness, and real-world feasibility. Application case studies from warehouse systems, intelligent transportation, and search-and-rescue missions are presented to demonstrate how these planning strategies are implemented in practice. Finally, discuss open challenges such as heterogeneous robot integration, safety assurance, and communication resilience, and highlight future research directions including hybrid architectures and learning-driven coordination. This review aims to support the development of scalable, adaptive, and efficient path planning in multi-robot systems.

Keywords: Multi-robot systems; cooperative planning; decentralized control; path planning; search and rescue.

1. Introduction

Multi-robot systems (MRS) are now used more often in smart factories, warehouse systems, city traffic, and emergency rescue. Compared to single robots, MRS can do many tasks simultaneously, divide work, and deal better with changing or unknown surroundings. This makes them a good choice for complex and fast-moving tasks. But when more robots are added, more problems appear. It becomes harder to give tasks to each robot; they may block each other's paths more often, and working together costs more time and effort [1]. Without good planning, these problems can lead to traffic jams, delays, or even system failure, especially in urgent jobs like rescue or scheduling.

Old path planning methods like A* and Dijkstra were made for one robot in simple settings [2]. They work well in those cases, but not when robots need to move together, deal with real-time changes, or talk to each other. So, researchers have turned to team-based planning methods that help robots plan their paths with space and time in mind. These methods need to avoid crashes, find good paths, solve timing issues, improve how robots move together, and lower the time or data needed. Some useful ideas include setting robot priorities, planning space and time separately, and using group planning methods to make the system faster and stronger.

This paper looks at many planning methods for multi-robot teamwork. Part 2 explains basic ideas like system design, how robots work together, and the main problems in fixed or changing settings. Part 3 groups the main methods into basic graph tools, search-based optimizers, and learning-based models. It also compares how well central control and split control work. Part 4 gives examples from real tasks in shipping, smart traffic, and rescue work, showing how these methods are used. Part 5 looks at what problems are still not solved and where future work could go.

This paper also looks at how ideas in the field have changed. It started with simple rule-following robots and moved to smart systems where each robot makes its own choices. This change came from progress in AI, small computers, and real-time sensors, letting robots work together and learn as they go [3]. Swarm robotics has also brought new ways to build large robot teams that can manage tough, crowded spaces.

This review also points out some real-world problems that are often missed, like different robot types, safety checks, and slow communication. Solving these is important for using these systems outside the lab.

2. Foundational Theories of Multi-Robot Cooperative Path Planning

2.1. Characteristics of Multi-Robot Systems

Multi-robot systems (MRS) differ significantly from single-robot systems in terms of task structure, control architecture, and coordination requirements. In an MRS, tasks must be distributed efficiently across agents while ensuring that their actions align toward a shared objective [4]. This task allocation problem is often tightly coupled with path planning, as the spatial positions and motion trajectories of robots are interdependent and may conflict in both time and space.

Various cooperative strategies—such as behavior-based control, market-based negotiation, and centralized task scheduling—have been proposed to coordinate robot actions while optimizing global team performance. These methods differ in their communication assumptions, scalability, and fault tolerance. The effectiveness of such cooperation is also highly dependent on environmental characteristics. In static environments, where obstacles and agent behaviours remain predictable, planning tasks can often be handled offline or with minimal updates. In contrast, dynamic environments—typical in applications such as urban mobility and disaster response—require robots to operate under uncertainty, detect changes in real-time, and frequently re-plan their paths and roles. This greatly increases the complexity of coordination, especially when communication is intermittent or latency sensitive.

2.2. Core Challenges in Multi-Robot Path Planning

One of the most fundamental challenges in cooperative path planning is collision avoidance. Robots must generate trajectories that not only avoid static obstacles but also actively prevent inter-agent collisions in shared environments. In densely populated or constrained spaces, efficient resolution of path intersections and prevention of deadlocks require predictive, time-parameterized planning or dynamic negotiation protocols.

Another major challenge is the computational complexity involved in planning paths for multiple agents simultaneously. The joint configuration space increases exponentially with the number of robots, making global optimization intractable in real time for large systems. Centralized planning frameworks can yield high-quality solutions but suffer from poor scalability and vulnerability to single points of failure. In contrast, distributed or decentralized planning offers better robustness and computational efficiency but may result in suboptimal coordination.

Real-time responsiveness is another essential requirement, particularly in logistics and manufacturing systems involving high robot density. For example, large-scale commercial implementations such as Amazon's Kiva warehouse robots rely on efficient coordination mechanisms to plan and adjust robot paths in milliseconds [5]. Balancing optimality, responsiveness, and safety remains a central challenge for algorithm designers.

3. Multi-Robot Path Planning Algorithms

3.1. Traditional Path Planning Methods

Basic path planning algorithms like A* and Dijkstra have been used for a long time in single-robot navigation. These methods build a search graph and find the shortest path from the start to the goal without hitting obstacles. A* adds a guess function to guide the search, making it faster while keeping the solution complete and correct under some rules (LaValle, 2006). In multi-robot systems, these methods are often changed to plan one robot at a time or give robots a fixed planning order. However, these changes don't fully handle how robots interact, and they often don't work well when many robots move in the same space or when the environment changes.

To use these traditional methods in multi-robot tasks, researchers often divide the environment into small units. Grid-based methods turn the space into cells and allow easy movement planning. Geometric approaches, like Voronoi diagrams and visibility graphs, can generate smoother and more global paths by using spatial features. However, these methods usually assume static, fully known maps and centralized control. Their performance declines in dynamic or partially observable environments, and their computation load increases rapidly with more robots. Geraerts and Overmars compare these geometric methods and show that although they are effective in static environments, they are not well-suited for online replanning or large multi-agent systems [6].

3.2. Intelligent Optimization-Based Methods

To solve the problems of older methods, many researchers now use smart optimization tools. One of the most used is deep reinforcement learning (DRL), which lets robots learn how to move by trying things in a training environment. For example, Tai showed that DRL can help mobile robots learn to move without maps, and this skill can be used in real-world tasks [7]. DRL methods like Deep Q-Networks (DQN) and Proximal Policy Optimization (PPO) let robots make their own decisions based on sensor data, helping them work in new or changing environments without needing a full map.

DRL has also been used for avoiding crashes between robots, even when they don't share their plans. Chen et al. suggested a DRL method that lets each robot learn how to react to others [8]. These methods are better for large groups and changing tasks, but they also face problems like long training times, hard-to-transfer policies, and limits when used in real life.

Compared to old methods, these smart approaches give robots more freedom and let them work in harder settings. But they often need more computing power and don't always give strong guarantees, which makes them harder to use in places like hospitals or space systems where safety is very important.

3.3. Centralized vs. Distributed Path Planning

How planning is done—centrally or separately—matters a lot in multi-robot systems. Centralized methods plan all robot paths together using full system knowledge. This gives better results and solves conflicts well. But it also needs a lot of computing and doesn't scale well. For example, Van Den Berg et al. came up with a method that creates safe, ordered paths over time. It works for a small group, but not when there are many robots or when communication is limited.

Distributed planning lets each robot decide its path based on what it sees or gets from nearby robots. This works better when communication breaks down or the team is big. But robots still need rules to avoid crashes and stay on task. Methods like reciprocal velocity obstacles (RVO) or path negotiation help, but they don't always work well when tasks are tightly connected or in crowded spaces.

To get the best of both, hybrid systems are being studied more. These systems let robots plan their moves but also connect to a central planner now and then or follow shared maps. For example, Honig, Ayanian, and Sukhatme built a system for quadrotor teams that mixes global planning using MILP with local control [9]. This helps with both teamwork and flexibility when conditions change. These systems work well in real-world tasks where robot types and environments are different.

4. Application Case Studies of Multi-Robot Cooperative Path Planning

4.1. Intelligent Warehousing and Logistics

In large warehouses, using many robots to work together has become key to keeping operations running smoothly. One well-known case is Amazon's use of Kiva robots. These robots move on their own to bring storage racks to human workers. They work in a well-organized space, where planning paths, avoiding crashes, and quickly changing tasks are all very important for keeping things moving fast.

In these systems, centralized or mixed planning methods are often used. A central system gives robots their jobs and planned paths, making sure they don't run into each other or block the way. But when there are hundreds or thousands of robots, small traffic jams and delays still happen. This means the system needs to quickly update plans and keep strong communication.

Wurman, D'Andrea, and Mountz explained how the Kiva system is built [5]. They showed how it handles robot movement, tracks inventory, and grows to handle more robots. Their design uses both pre-made plans and real-time updates, showing how robot teamwork can make warehouses much more efficient and lower both space and labour costs.

4.2. Intelligent Transportation and Autonomous Driving

In the context of smart cities and autonomous driving, cooperative path planning among multiple vehicles is essential for traffic safety, congestion mitigation, and energy efficiency. Multi-robot path planning here takes the form of multi-vehicle trajectory coordination, often under the umbrella of vehicle-to-everything (V2X) communication frameworks. These systems allow autonomous vehicles to share information about their trajectories, intentions, and environment to achieve globally optimized traffic flows.

Decentralized planning is widely adopted in this context due to the dynamic nature of road environments and the need for real-time responsiveness. However, achieving coordinated behaviours, such as merging, intersection crossing, or platoon formation, requires the integration of predictive models, priority rules, and message-based negotiation. Real-time performance, latency constraints, and partial observability further complicate the deployment of path planning algorithms in real traffic systems.

Talebpour and Mahmassani analyse the impact of cooperative driving strategies on traffic dynamics, showing that V2V and V2I coordination improves lane usage, reduces delays, and increases throughput, particularly in high-density scenarios [10]. Their findings support the integration of cooperative planning into connected vehicle infrastructure.

4.3. Search and Rescue in Disaster Response

Search and rescue (SAR) missions are some of the hardest situations for multi-robot teams. These missions usually take place in fast-changing, partly unknown, and dangerous places like earthquake ruins, wildfire areas, or flooded cities. In these settings, flying and ground robots need to work together to explore, find survivors, understand the environment, and help human teams.

A common and useful setup is to use UAVs and UGVs together. UAVs fly above to scout and build maps, while UGVs move on the ground through broken or tight spaces. This mix of air and ground robots makes SAR work more flexible and gives a better view of the area. But it also brings new problems, like how to split up tasks, stay connected when networks are weak, and change plans quickly during the mission.

Queralta reviewed how multi-robot SAR systems work together. They focused on how planning, teamwork, and sensing can be linked closely [11]. The paper pointed out the value of active perception and shared control. Most systems today use a mix of centralized tools, like putting maps together and giving out tasks, and decentralized choices made by each robot on the fly. The authors also stressed that strong communication is key, especially when GPS doesn't work or when signals are weak. Even then, robots must keep sharing important data.

These points show that SAR robots today are not just machines that move smart, connected parts of a larger system that can learn and react as things change.

5. Conclusion

This paper gives a full review of path planning methods used in multi-robot systems (MRS), focusing on basic theories, types of algorithms, and real-life uses. It starts by pointing out the limits of single-robot planning and explains the special problems that come with using many robots together, such as giving out tasks, avoiding crashes, growing the system, and reacting in real time.

The review looks at three main kinds of planning methods: traditional ones like A* and Dijkstra, smart optimization tools like deep reinforcement learning, and system designs that compare centralized and distributed planning. Classic methods work well in simple and fixed spaces and give clear results, but they often don't scale well when things get crowded or change quickly. Smarter methods help robots act more freely and adjust better, but are harder to train, explain, and keep safe. Centralized systems make global plans but can be slow and need strong communication. Distributed systems are easier to scale and can handle failures, but they are not always well-coordinated.

The paper also looks at how these planning systems are used in places like warehouses, traffic systems, and rescue missions. These examples show why flexible and strong planning is needed in changing, hard-to-see environments.

Looking ahead, some key directions stand out. First, hybrid systems that mix central control with robot-level planning may offer a good balance. Second, smart learning methods should work better across tasks, train faster, and provide more safety. Third, future systems need to work better with different kinds of robots, bad sensors, and weak networks. Last, ideas from other fields—like game theory, human-robot teamwork, and edge computing—will be useful in building real, reliable multi-robot systems.

As robot teams grow larger and tasks become harder, smart path planning will stay key to making these systems more independent, efficient, and ready for real challenges.

References

[1] Y. U. Cao, A. S. Fukunaga, A. B. Kahng, and F. Meng.. Cooperative mobile robotics: antecedents and directions, in Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots, Aug. 1995, pp. 226–234 vol.1.

[2] S. M. LaValle, Planning Algorithms. Cambridge University Press, 2006.

[3] E. Şahin. Swarm Robotics: From Sources of Inspiration to Domains of Application, in Swarm Robotics, 2005, pp. 10–20.

[4] L. E. Parker. ALLIANCE: an architecture for fault tolerant multirobot cooperation. IEEE Trans. Robot. Autom., Apr. 1998, vol. 14, no. 2, pp. 220–240.

[5] P. R. Wurman, R. D'Andrea, and M. Mountz. Coordinating Hundreds of Cooperative, Autonomous Vehicles in Warehouses. AI Mag., Mar. 2008, vol. 29, no. 1, Art. no. 1.

[6] R. Geraerts and M. H. Overmars, "A Comparative Study of Probabilistic Roadmap Planners," in Algorithmic Foundations of Robotics V, J.-D. Boissonnat, J. Burdick, K. Goldberg, and S. Hutchinson, Eds., Berlin, Heidelberg: Springer, 2004, pp. 43–57.

[7] L. Tai, G. Paolo, and M. Liu. Virtual-to-real deep reinforcement learning: Continuous control of mobile robots for mapless navigation, 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Sep. 2017, pp. 31–36.

[8] Y. F. Chen, M. Liu, M. Everett, and J. P. How. Decentralized Non-communicating Multiagent Collision Avoidance with Deep Reinforcement Learning, Sep. 28, 2016.

[9] W. Hönig, J. A. Preiss, T. K. S. Kumar, G. S. Sukhatme, and N. Ayanian. Trajectory Planning for Quadrotor Swarms, IEEE Trans. Robot., Aug. 2018, vol. 34, no. 4, pp. 856–869.

[10] A. Talebpour and H. S. Mahmassani. Influence of connected and autonomous vehicles on traffic flow stability and throughput, Transp. Res. Part C Emerg. Technol., Oct. 2016, vol. 71, pp. 143–163.

[11] J. P. Queralta et al.. Collaborative Multi-Robot Search and Rescue: Planning, Coordination, Perception, and Active Vision, 2020, IEEE Access, vol. 8, pp. 191617–191643.