Mobile Robot Path Planning in Dynamic Environments: A Survey

Kuanqi Cai [1,2], Chaoqun Wang [2], Jiyu Cheng[2], Clarence W De Silva[1,3], *Fellow, IEEE* and Max Q.-H. Meng[1, 2], *Fellow, IEEE*

(1. School of Mechanical Engineering and Automation, Harbin Institute of Technology (Shenzhen), Shenzhen 518055;

2. Department of Electronic Engineering, The Chinese University of Hong Kong, Hong Kong SAR 999077;

3. The University of British Columbia, Vancouver, BC, Canada;

Abstract: The environment that the robot operating in is becoming more and more complex, which poses great challenges on robot navigation. This paper gives an overview of the navigation framework for robot running in dense environment. The path planning in the navigation framework of mobile robots is divided into global planning and local planning according to the planning scope and the executability. Robot navigation is a multi-objective problem, which not only needs to complete the given tasks but also needs to simultaneously maintain the social comfort level. Consequently, we focus on the reinforcement learning-based path planning algorithms and analyze the development status, advantages, and disadvantages of the existing algorithms. Besides, path planning in a dynamic environment for robots will be further studied in the future in the areas of the advanced algorithm, hybrid algorithm, multi-robot collaboration, social model, and artificial intelligence algorithm combination.

Keywords: Navigation; Dynamic Environment; Local Planner; Global Planner; Human Motion; Reinforcement Learning;

1 Introduction

With the development of robotic technologies, robots not only operate in simple and repetitive environments, such as factory environment and warehouse, but also it can enter human space. For example, they work in hotels, supermarkets, and even in outdoor dynamic environments as various service robots. For example, MultiModal Mall Entertainment Robot (MuMMER)[1] can provide an entertaining and engaging experience to the public in dynamic environments of an open-air plaza. Peppe[2], a human-like robot developed by Aldebaran used by Bechade to establish and develop a family companion robot for the elderly in the hotel. Atlas robot[3], developed by Boston Dynamics, can replace people to perform special tasks in both indoor and outdoor environments. With the demand for different functions of service robots, robots work in more and more complex environments, such as shopping malls, city streets, hospitals, and train stations. In these dynamic environments, there are not only dynamic obstacles and static obstacles, but also some "special obstacles". The classification of obstacles is shown in Fig. 1. These "special obstacles" are different from the general dynamic obstacles. The dynamic obstacles in this paper refer to obstacles which speed and direction movement do not change with the external environment, for example, the blind people walk along blind roads [4]. The "special obstacles", however, do not only refer to the agent which has pro-active ability. For example, they can take actions to achieve a certain goal and avoid collisions[5], but also many pedestrians. How to coexist harmoniously with people and other agents in this highly dynamic environment? It is one of the key problems for the robot to avoid the walking pedestrians and reach the target smoothly. Therefore, the navigation research of service robots in densely and populated environment has important theoretical value and application significance.

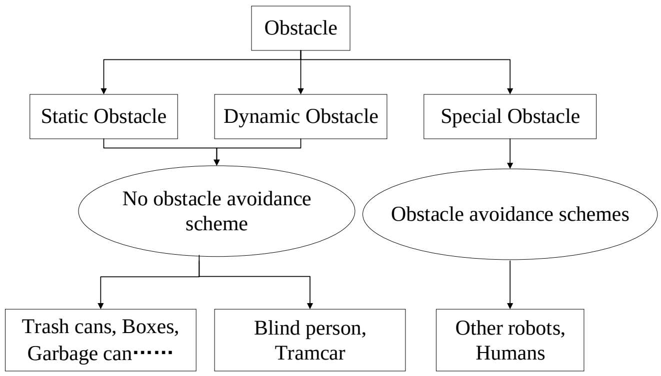

Fig. 1 Classification of Obstacles

In recent years, many scholars have written lots of reviews about robot obstacle avoidance. Kruse et al. [6] wrote a survey about human-aware robot navigation. This article focuses on the characteristics that robots need to display in terms of human perception. The main features that robot concerns includes comfort, naturalness, and sociality. Chik et al.[7] introduced some parts of navigation including global planner and local planner. Through the introduction of four navigation frameworks, different navigation components; and different robot platforms, the author provides solutions for the implementations of service robots. Douthwaite et al.[8] wrote a comparative study of Velocity Obstacle (VO) approaches for multi-agent systems. They also put forward several evaluation scenarios to cope with both sensor uncertainty and increasing difficulty. Mohanan et al.[10] gave a review of the major research in Robot Motion Planning (RMP) in dynamic environments. The author classified RMP in the literature and made a comparative analysis of their performance. Zafar et al.[11] divided the motion planning approaches into classical approach and heuristic approach. The comparative analysis of the two aspects shown that the heuristic approach has higher intelligence. Cheng et al.[12] divided methods into reactive-based method, predictive-based method, model-based method, and learning-based method. These reviews are summarized in Table 1.

Table 1 The Summary of Literature Review

Author	**Year**	**Main content**
Thibault Kruse	2013	Characteristics that robots need to display in terms of human perception
S. F. Chik	2016	Global planner, Local planner, Four navigation frameworks, Different navigation components, Different robot platforms
Douthwaite,J.A	2018	Comparative study of velocity obstacle approaches
M.G.Mohanan	2018	Robot Motion Planning
MN Zafar	2018	Classical approach, Heuristic approach
Jiyu Cheng	2018	Divided method into four-part: reactive based method, predictive based method, model-based method and learning based method

The path planning algorithms towards obstacle avoidance from static obstacles to dynamic obstacles were investigated by experts. This paper analyzes the development prospects of multi-agent collision avoidance in a dynamic environment. It is a survey of the major contributions in the literature for motion planning algorithms which related to social-aware navigation in dynamic environments for the past few decades. The advantages, disadvantages, and evolution of the algorithms are illustrated.

The rest of the paper is organized as follows. In Section II, we introduce the classic framework of navigation. In Section III, we further discuss the classical global planners developed by researchers in recent years. In Section IV we introduce the traditional methods for local planning, which focuses on VO, and highlights the human trajectory prediction model and the use of learning for path planning in dynamic environments. We make a summary at the end of this paper in Section V.

2 Navigation Component

Path planning is one of the key techniques for guiding the robot in dynamic environments. The goal of the path planning is to control the robot from the starting point to the target point, subject to the constraints such that the robot does not touch any obstacle throughout the process.

Path planning can be divided into two categories based on the utilized environment information [9]. The former is the global planner that the robot knows the global environmental information, and the latter one is the local planner that the robot only knows the information nearby itself.

When the robot plans the path, the robot first inputs the map information, the target position, and the position of the robot itself to the global path planner. An optimal path is obtained by a global path planning algorithm combined with social costs, where social costs include object padding, object occlusion, and hidden zones. Then the robot uses this path as an initial reference to the local planner or an optimization objective. In a dynamic environment. To avoid collisions like robots and humans, robots must choose path planner adaptively. When facing human or non-human obstacles, social compliant path planners make robots live in harmony with people. Therefore, obstacles detection and classification are necessary for robots. When the obstacle is the human or other robots that have obstacle avoidance capabilities, the robots should estimate their trajectories by the trajectory prediction models. After that, the robots put these models into the local planner to generate an optimal path. Finally, the path is transmitted to the robot actuator for the root to execute. The framework of navigation is shown in Fig. 2. The paths planned by the local planner and global planner are shown in Fig. 3.

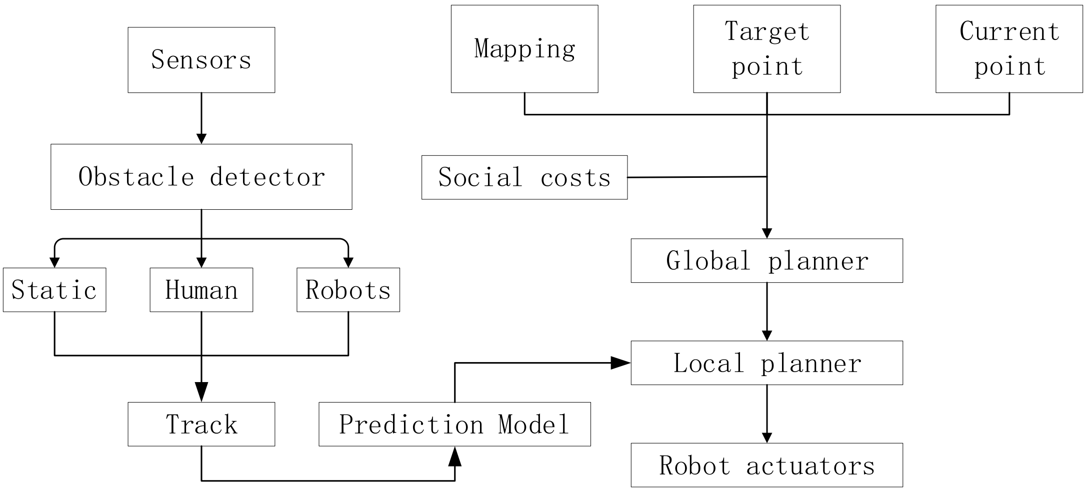

Fig. 2 Navigation Framework

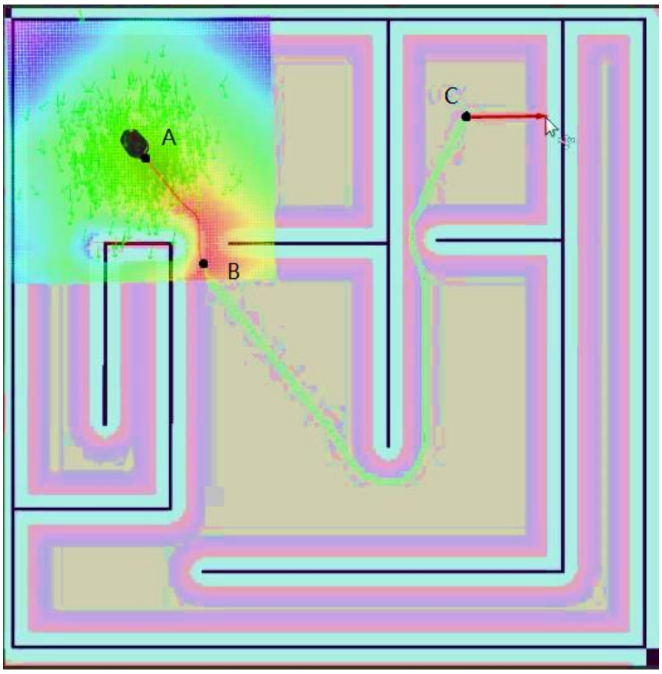

Fig. 3 The paths planned by the local planner and global planner. point A is the origin of the robot, and point C is the target of the robot. The red track is planned by the local planner, while the green path is planned by the global planner.

2.1 Global planner

The global path planning method plans a path for the robot according to the acquired environmental information such as the global map and target point. The accuracy of the path depends on the environment completeness in real-world environments[13]. Robots often cannot travel strictly according to the global planner. Therefore, it is necessary to plan the route that the robot can follow by using the map information obtained in the dynamic environment to cope with the obstacles that may appear at any time near the robot. Researchers have done a lot of research on global path planning. They also put forward many efficient algorithms. These algorithms can be divided into Graph Search Algorithm[14], Random Sampling Algorithm[15], Intelligent Bionic Algorithm[16], etc. The classic algorithms for graph search mainly include the Dijkstra algorithm[17], A* algorithm[18], DFS algorithm[19], BFS algorithm[20] and so on. The Dijkstra algorithm and A* algorithms are widely implemented in the Robot Operating System (ROS)[21] system. These methods are improved through a heuristic estimation, which reduces the number of searching grids and improves the searching efficiency. However, when the environment is complex, the planning efficiency is inevitably low. Random sampling algorithms include

Batch Informed Trees (BIT)[22], Regionally Accelerated Batch Informed Trees (RABIT)[23], Rapidly-exploring Random Tree (RRT)[24], and Risk-based Dual-Tree Rapidly exploring Random Tree (Risk-DTRRT)[25]. These algorithms are more efficient and widely used in dynamic or high-dimensional environments.

The intelligent bionic algorithm is an intelligent algorithm for simulating the evolutionary and biomimetic insects' behaviors, where the researchers mainly using Genetic Algorithm (GA)[26], Ant Colony Algorithm (ACO)[27], Artificial Bee Colony Algorithm (ABC)[28], and Particle Swarm Optimization Algorithm (PSO) [29]. To speed up the calculation as much as possible and solve the shortcoming of locally optimal, Wang et al. [30] aimed to solve the shortest collision-free path planning problem of welding robot, so they raised the Optimization of the Genetic Algorithm-Particle Swarm Optimization Algorithm (OGA-PSO). Liu et al.[31] combined the artificial potential field and geometric local optimization method to search for the globally optimal path. Mac et al.[32] proposed particle swarm optimization called constrained multi-objective particle swarm optimization with an accelerated update methodology which provides optimal global robot paths in terms of the path length and smoothness. Diagrammatic representation of diverse research approaches the global planner are shown in Fig. 4.

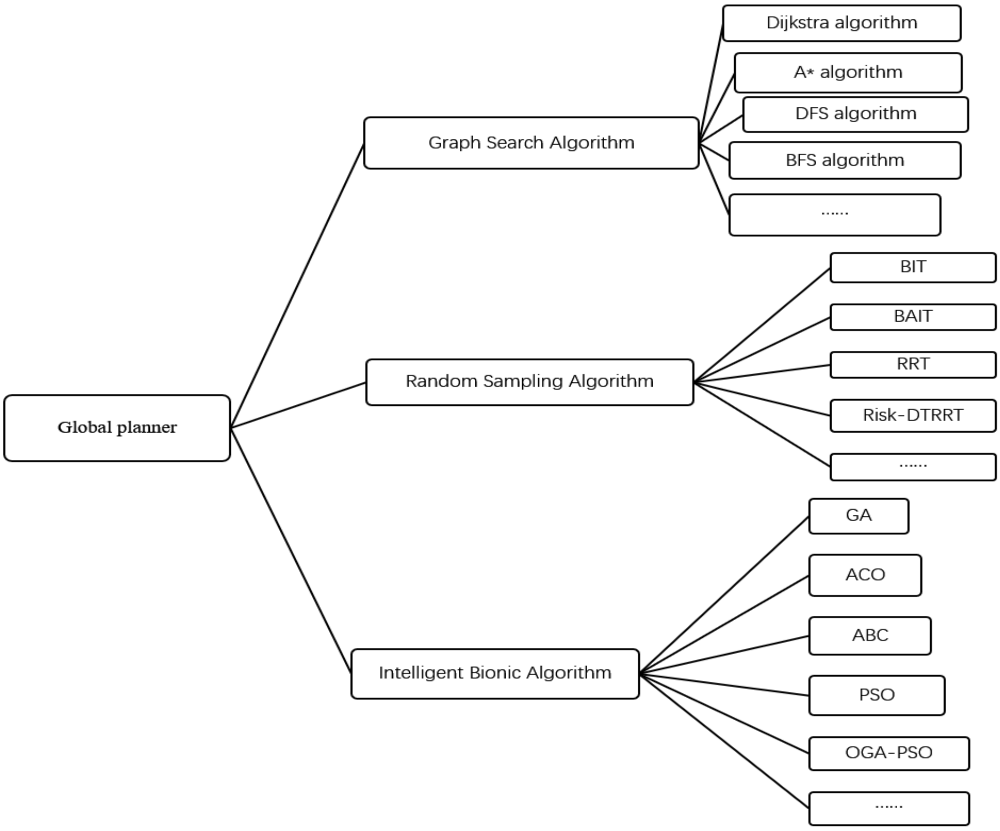

Fig. 4 Diagrammatic representation of diverse research approaches in global planner

2.2 Local planner

The local planner focuses on the current local environment information of the robot so that the robot has good collision avoidance capability. Local planners are wildly used because their information acquired by the sensor system is changing in real-time when the environment changes. Compared with the global planning method, the local planning method is more efficient and practical. The disadvantage is that sometimes the local planner may be trapped into the local minimum.

There are many classical algorithms for obtaining an optimized path in local planner, such as Artificial Potential Field method[33], Fuzzy Logic Algorithm[34], Simulated Annealing Algorithm[35], a hybrid method combined with genetic algorithm[36], particle algorithm[37], and other algorithms to avoid the disadvantage that the solution is locally optimal.

However, the aforementioned methods do not consider the velocity of the obstacle. With an increasingly complex and crowded environment, it was unreliable for inter-agents to communication with each other. The agent should generate a trajectory that is as optimized as possible to reach the goal while avoiding other agents and obstacles in the environment at the same time. To avoid the other agents in the multi-agent systems, each agent navigates independently without explicit communication with other agents[38]. P.Fiorini[39] put forward the concept of the Velocity Obstacle (VO), which defines a geometric region as a constraint that the speed of the agent falls into which will cause a collision at the next time step. However, oscillations will happen when the agents are on the collision course. Since both sides of the robot are offset by the current speed for too many obstacles, the original speed will not lead to collide after the speed is updated. Is there a way to reduce the offset to the current speed and to ensure the avoidance of the obstacle? Van den Berg et al.[38] introduced the reciprocal velocity obstacle (RVO) to solve this problem. They regard the new velocity profile as the average velocity of agent current velocity and velocity that lies outside of the other agent's VO. It suggests a useful way to smoothly and safely plan a path in multiple-agent navigation amongst each other without oscillations. But RVO also has a disadvantage that it can't find agreement on which side to pass each other, which is called "reciprocal dance". To solve this problem, Jamie Snape[40] extended RVO to the Hybrid Reciprocal Velocity Obstacle (HRVO). Not only do they use this approach for the navigation of multi-robot, but they also consider the kinematics and sensor uncertainty of the robot. But if there are many robots or people in the scenario, the apexes of the VOs are close to the origin in the velocity space[41]. It will make the robot trapped in an area. This problem can be solved by the truncation approach. Robots do not collide at the defined timesteps after truncation. A sample figure about Vos is shown in the Fig. 5. The r_A, v_B are the speed of the robot, P_A and P_B are the poses of the two robots, r_A and r_B are the radii of the two robots, and the gray areas in (b), (c), (d) are velocity obstacle with different methods.

The new velocity of agents must be chosen outside the region. In order to find an optimal velocity, there are several ways to calculate the new velocity. In this paper, three commonly used methods will be introduced. Berg et al.[42] introduced a method named Optimal Reciprocal Collision Avoidance (ORCA). In this way, agents compute half-planes of collision-free velocities for each other agent by themselves, and then the optimal velocity region can be defined by solving a linear program. The agents select a velocity profile that is closest to optimal velocity, with which the robot will travel at the next moment. Another method to compute the collision-free velocity is the ClearPath. This algorithm was introduced by Guy et al.[43]. ClearPath is a robust algorithm that is faster than prior VO based approaches for collision avoidance among multiple agents. There are two ways for ClearPath to determine the collision-free velocity. In the first scheme, velocity is chosen at the intersection of two boundary lines of arbitrary velocity obstacles, while in the second method, the velocity is determined by the projection of the preferred velocity onto the nearest of each velocity obstacle[44]. Daniel et al.[42] introduced a Collision Avoidance with Localization Uncertainty (CALU) method that used Optimal Reciprocal Collision Avoidance (ORCA) and non-holonomic robots Optimal Reciprocal Collision Avoidance (NH-ORCA) [43] and combined with Adaptive Monte-Carlo Localization (AMCL) [44] to alleviate the need for prefer sensing.

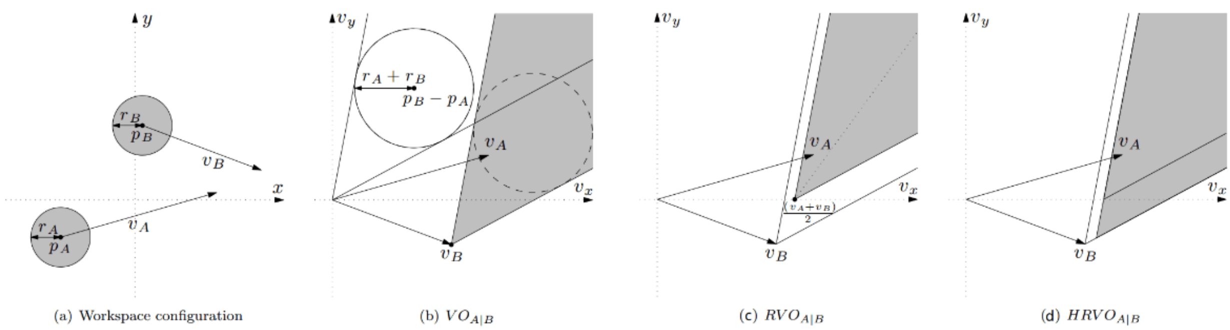

Fig. 5 A sample figure about VOs. Figure courtesy[42]

VO based algorithms require perfect information about the shape, velocities, and position of agents. But in the complex environment, inaccurate localization and message passing delays occur frequently. This method can solve this problem effectively. But it was useless under the two conditions: a) The robot chassis cannot be approximated as a disk b) The pose belief distribution of AMCL is elongated along with one axis such as in the long corridor. Claes[48] introduced Collision Avoidance under Bounded Localization Uncertainty (COCALU) to solve this problem. The main difference is that he changes the shape of the particle cloud instead of using a circumscribed circle. The summary of classical algorithms based on velocity obstacle is shown in Fig. 6.

Oftentimes, human are regards as the obstacles for robot navigation. Therefore, Nishitani et al.[49] used grid-based X-Y-T space path planning to avoid obstacles and humans. This method considers the human's orientation and personal space, but the efficiency depends on the defined grid size. Kollmitz et al.[50] proposed a method about layered social cost map for navigation in a complex environment. This method predicted human trajectories use a social cost function as an extension of A* algorithm 错误!未找到引用源。, but it does not concern about obstacle motion.

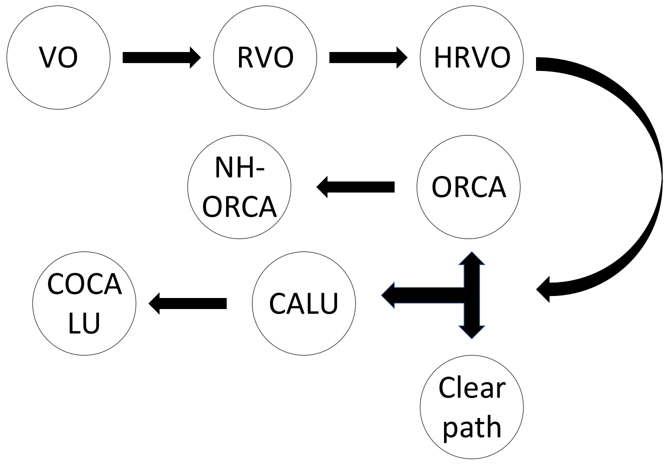

Fig6. Summary of Classical Algorithms Based on Velocity Obstacle

Table 2 Some classic algorithms on the local planner

Local planners	Advantages	Disadvantages
Artificial Potential Field method	The scheme has high efficiency and it can solve the problem of a local minimum in the traditional algorithm	There is a trap area and the robot will oscillate when it passes through the narrow passage.
Fuzzy Logic Algorithm	It reduces the dependence on environment information and has the advantages of good robustness and effectiveness	Fuzzy rules are often predetermined by people's experience, so they are unable to learn and have poor flexibility
Simulated Annealing Algorithm	Simple description, flexible use, high efficiency, less initial conditions	Slow convergence and high randomness
VOs	It considers the obstacle velocity	The complex relationships between societies are not considered
X-Y-T Space	It considers the human's directional area and personal space	Efficient was depends on defined grid size
Time-dependent A*	It can predict human trajectories	It doesn't concern about obstacle motion

3 Application of Reinforcement Learning Algorithm in Path Planning

In a static environment, if the robot can obtain global knowledge of the operating environments, then the global planning method can be used for the robot path planning. However, the difficulty of the algorithm increases with the increase of the complexity and uncertainty of the environments, especially in the more complex dynamic environments. If there is no precise environmental model, the efficiency of robot navigation will be reduced. In this case, the use of information from the sensor will benefit a lot for local path planning. Several methods described in prior have their own advantages under different conditions, but there are also have some problems such as injecting local minimum, slow algorithm operation, and large computer storage. For these problems, scholars have proposed many methods to improve real-time and the speed of robot response. The neural network can be used to predict trajectories, but the acquisition and learning of samples are very difficult in complex environments. Also, with the development of intelligent systems, the path planning algorithm of robots with autonomous learning ability is a useful tool to solve the above problems. Reinforcement learning is a special and environment-friendly machine learning method that uses environmental feedback as input.

Reinforcement learning is the constant interaction between the robot and the environments. It acquires knowledge in the process of action-evaluation-action, and it can improve the motion planning performance to facilitate the robot operation continuously. The robot faced with people and other robots will produce a reasonable solution. The environments referred in the reinforcement learning are not only in the dynamic environments such as supermarkets and shopping malls but also to static, dynamic, and special obstacles appearing in the environments. Reinforcement learning is trying every action and then making a judgment based on that. It mainly relies on the feedback information of the environments to evaluate the behavior. Then it guides the robot future actions according to the evaluation. In the whole process, the robot strengthens its excellent behavior and weakens its improper behavior. Robots get better action strategies to adapt to the environment through heuristics. The application process of reinforcement learning in robot navigation is as follows. When the robot faces different obstacles, it will not be told how to go. Thus, the robot will try to move forward. The environment changes as the robot chooses where to move. This change will be evaluated according to the robot behavior, and the robot will get the strengthened signal (reward and punishment mechanism) from the environment. The reward and punishment feedback to the robot and the robot calculates the cumulative reward and punishment according to the following Eq. (1)

$$V^*(s_t) = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \ldots = \sum_{i=0}^{\infty} \gamma^i r_{t+i} , \qquad (1)$$

where s_t is the internal state of the robot, γ is the learning discount rate, r_t is the reward and punishment situation, V^* is the number of cumulative reward and punishment.

The optimal strategy is π*, the formula is (2) and (3)

$$\pi^* = \arg\max V^{\pi}(s)(\forall s) \qquad (2)$$

$$s_t \subset S , \qquad (3)$$

where S is the set of agent state space.

With a right trajectory, the environment will give the robot a positive value. The trend of adopting this behavior strategy will be strengthened when the robot encounters the same situation. Conversely, if a behavior strategy leads to negative reward value, the tendency of the robot to take this action will be weakened. However, robots are often rewarded for approaching their destinations without colliding or interfering with other agents. The current state of the signal and environments will affect robots in selecting the next action. The principle of choice is to increase the value of positive reward. The selected action not only affects the immediate reward value but also affects the state of the next moment and the final enhanced value. The purpose of reinforcement learning is to find an optimal strategy that maximizes the cumulative reward value. The robot obtains the value from the starting point to the goal point. The process of reinforcement learning is shown in Fig. 7.

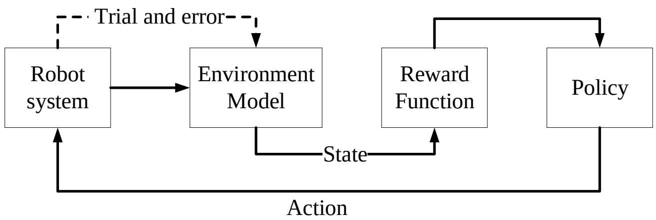

Fig. 7 The process of reinforcement learning

Reinforcement learning has the following characteristics: 1. Trial-and-error search and delayed return are the two most prominent features of the reinforcement learning algorithm. 2. Adaptable, that is, the robot can continuously use the feedback information in the environment to improve its performance. 3. Feedback, that is, the robot can directly obtain the state action rules from the experience. 4. The robot only needs to learn according to the enhanced signal, and these enhanced signals can be obtained from the built-in reinforcement mechanism of the robot.

The classical reinforcement learning algorithms in mobile robot path planning include Q-learning algorithm[51], SARSA algorithm[52], R-learning algorithm, etc. Q-learning is the most common one in the reinforcement learning algorithms. The algorithm forms a reward value for the robot state and motion through the feedback that the robot obtains from the environment. The

Q value of the correct action is increasing, while the Q value of the wrong action is decreasing. Then the Q value-based method makes the action of the robot tend to the optimal behavior after the Q value is filtered. However, the Q-learning algorithm has some disadvantages: 1. The memory space is large. 2. Long learning time. 3. The convergence speed is slow. In order to tackle the problems of Q-learning, Peng [53] proposed the Q(λ) algorithm, which uses the idea of backtracking. The subsequent data can be fed back in time so that it has a certain memory and the robot cannot only predict the next behavior but also can control its own behavior. The Q value corresponding to the wrong behavior is gradually forgotten in the process of updating. In recent years, the environment that mobile robots operating in has become more and more complex. Researchers are constantly improving the intensive learning algorithms. Inverse Reinforcement Learning contains a reward function for learning the Markov [54] decision process. Some researchers applied IRL (Inverse Reinforcement Learning) to collaborative navigation[55]. In order to interact with humans in a harmonious way, these robots need to understand and follow the rules. Therefore, Kretzschmar[56] proposed an approach to model the cooperative navigation behavior of humans. This method is based on a mixture distribution. It can not only capture the discrete navigation decisions but also can capture the natural variance of human trajectories. Chen et al.[57] proposed a decentralized multi-agent collision avoidance algorithm based on deep reinforcement learning, which transfers online computing (for predictive interaction mode) to offline learning effectively. It is better than the ORCA algorithm [58]. But the method of this model may lead to an oscillating path. Therefore, the author also proposed SA-CALDRL[59] to solve the randomness of human behavior. Chen proposed a time-effective navigation policy to respect common social norms. The method can achieve the low-speed autonomous navigation of the robot vehicle in densely and populated environment. But this method does not consider the relationship between pedestrians. Everett [60] extended the SA-CALDRL algorithm by introducing the LSTM strategy in the algorithm. The advantage of the method is that it does not need to assume the specific behavioral model. In the future, it can try to clarify the running direction of the robot in a simple way. In addition, based on reinforcement learning, Ciou et al.[61] proposed a compound reinforcement learning framework. Under this framework, the robot learns appropriate social navigation through sensor input and rewards updates according to people's feedback. The experiments show that CRL (Composite Reinforcement Learning) system can learn how to navigate in the environment safely, but this method has a greater need for the prior knowledge. A multi-scenario multi-stage training framework learning optimal strategy was proposed by Long et al.[62]. Such a strategy can be well extended to new scenarios that do not appear in the training phase. The applications of reinforcement learning in dynamic path planning in recent years are shown in Table 3.

Table 3 The application of reinforcement learning in dynamic path planning in recent years

Method	Advantage	Disadvantage
IRL	Build a human model in different environments, collaborative navigation.	Need a lot of calculation, feature selection determines performance.
Socially compliant mobile robot Navigation method of inverse reinforcement	Capture the discrete navigation decisions, capture the natural variance of human trajectories.	Highly dependent on the demonstration information, data collection procedure cost too much time.
Decentralized Non-communicating Multiagent Collision Avoidance with Deep Reinforcement Learning	High real-time performance and high path quality	May lead an oscillating path.
SA-CADRL	Solve the randomness of human behavior, with respect to common social norms.	Does not consider the relationship between pedestrians.
Motion Planning Among Dynamic, Decision-Making Agents with Deep	It does not need to assume that other robots follow any specific	Clarify the running direction of the robot in a complex way.

Reinforcement Learning	behavioral model.	
CRL	Safely learn how to navigate in the environment.	A greater need for prior knowledge.
Towards Optimally Decentralized Multi-Robot Collision Avoidance via Deep Reinforcement Learning	Extended to new scenarios that do not appear in the whole training.	Unable to understand the world at the agent level

With the increasing demand for intelligent application of mobile robots, the reinforcement learning based path planner has become a hot topic in current research. This method maps the sensor-aware environmental state to the actuator action and responds quickly to changes in the external environment. Finally, it can achieve autonomous path planning [63].

4. Prospects for path planner in a dynamic environment

With the wide application of robots, the path planning algorithm is required to have the ability to respond quickly to complex dynamic environments. The flourishing development of path planning technology has already obtained a lot of research achievements. But in the specific path planning algorithm design, each algorithm has its inadequacies and limitations. There are many theories and methods that need constant improvement. In addition to the study of new path planning algorithms in future path planning. The path planner will also be developed from the following aspects. First, the improvement of existing path planning algorithms. Path planner is used in a wide range of applications and it needs to be improved for different dynamic environments such as highway, piazza, and supermarket. Therefore, in the process of practical application, any kind of algorithm will face many difficulties. The main problem was caused by its limitations. Different improvements in different environments can quickly and effectively improve the performance of the algorithm while solving practical problems. Therefore, targeted improvement in specific application environments can quickly and effectively improve the performance of the algorithm while solving practical problems. Secondly, the combination of multiple algorithms. It is almost impossible for any single path planning algorithm to solve the path planning problem in a dynamic environment when facing new problems of interdisciplinary. The complementary advantages of path planning algorithms provide the possibility to solve this problem. For example, the artificial intelligence method, the new mathematical method, and the bionic algorithm are combined to solve the problem. Thirdly, the future path planning method will be developed towards multi-robot cooperation and human-computer interaction. Fourthly, in recent years, brain science and brain-like intelligence have become the hot topic in robotic navigation community. Artificial intelligence algorithms based on brain-like intelligence are widely used in robotic path planning. Brain-like intelligence has a strong learning ability and is widely adopted in complex and unknown environments. It helps plan a path through imitation and interaction with the environment. It can develop independently and greatly enhance the intelligent level of path planning. The prospects were shown in Fig. 8.

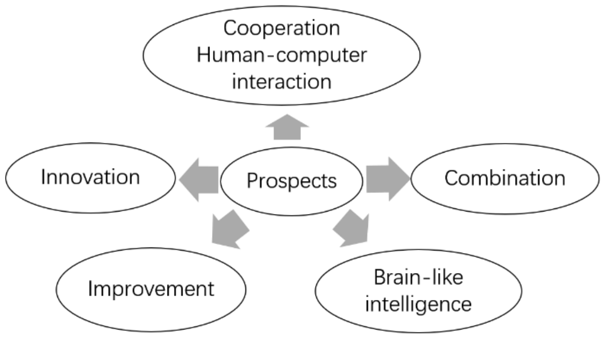

Fig. 8 Prospects for path planner in dynamic environment

4. Conclusion

In the process of environmental exploration, because of the complexity and dynamics of the environment, mobile robots will encounter a variety of obstacles. How to identify and avoid these obstacles? How to get to the target position as soon as possible and live in harmony with people and other robots in this dynamic system. These are the basic problems of robot navigation. It is an important indicator to measure the performance of mobile robot path planning. This paper first analyzes the framework of navigation in a dynamic environment and introduces the global planner and local planner. Then we study the characteristics of common path planning algorithms and analyzes and compares their application scopes. What is more, this paper summarizes the commonly used path planning methods based on reinforcement learning and it also introduces the application algorithms. Finally, this paper points out the path planning algorithms in a dynamic environment and the research direction. With the development of path planning techniques, robots will be to integrate into human life quickly soon.

References

[1] Foster, M. E., Alami, R., Gestranius, O., Lemon, O., Niemelä, M., Odobez, J. M., & Pandey, A. K. (2016, November). The MuMMER project: Engaging human-robot interaction in real-world public spaces. In International Conference on Social Robotics (pp. 753-763). Springer, Cham.

[2] Tanaka, F., Isshiki, K., Takahashi, F., Uekusa, M., Sei, R., & Hayashi, K. (2015, November). Pepper learns together with children: Development of an educational application. In 2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids) (pp. 270-275). IEEE.

[3] Feng, S., Whitman, E., Xinjilefu, X., & Atkeson, C. G. (2014, November). Optimization based full body control for the atlas robot. In 2014 IEEE-RAS International Conference on Humanoid Robots (pp. 120-127). IEEE.

[4] Wang, C., Liu, W., & Meng, M. Q. H. (2015, August). Obstacle avoidance for quadrotor using improved method based on optical flow. In 2015 IEEE International Conference on Information and Automation (pp. 1674-1679). IEEE.

[5] Douthwaite, J. A., Zhao, S., & Mihaylova, L. S. (2018, September). A Comparative Study of Velocity Obstacle Approaches for Multi-Agent Systems. In 2018 UKACC 12th International Conference on Control (CONTROL) (pp. 289-294). IEEE.

[6] Kruse, T., Pandey, A. K., Alami, R., & Kirsch, A. (2013). Human-aware robot navigation: A survey. Robotics and Autonomous Systems, 61(12), 1726-1743.

[7] Chik, S. F., Yeong, C. F., Su, E. L. M., Lim, T. Y., Subramaniam, Y., & Chin, P. J. H. (2016). A review of social-aware navigation frameworks for service robot in dynamic human environments. Journal of Telecommunication, Electronic and Computer Engineering (JTEC), 8(11), 41-50.

[8] Douthwaite, J. A., Zhao, S., & Mihaylova, L. S. (2018, September). A Comparative Study of Velocity Obstacle Approaches for Multi-Agent Systems. In 2018 UKACC 12th International Conference on Control (CONTROL) (pp. 289-294). IEEE.

[9] Wang, C., Meng, L., She, S., Mitchell, I. M., Li, T., Tung, F., ... & de Silva, C. W. (2017, September). Autonomous mobile robot navigation in uneven and unstructured indoor environments. In 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 109-116). IEEE.

[10] Mohanan, M. G., & Salgoankar, A. (2018). A survey of robotic motion planning in dynamic environments. Robotics and Autonomous Systems, 100, 171-185.

[11] Zafar, M. N., & Mohanta, J. C. (2018). Methodology for Path Planning and Optimization of Mobile Robots: A Review. Procedia computer science, 133, 141-152.

[12] J. Cheng, H. Cheng, M. Q.-H. Meng, H. Zhang, Autonomous Navigation by Mobile Robots in Human Environments: A Survey, in: 2018 IEEE International Conference on Robotics and Biomimetics (ROBIO), IEEE, 2018, pp.

[13] Wang, C., & Meng, M. Q. H. (2016, June). Variant step size RRT: An efficient path planner for UAV in complex environments. In 2016 IEEE International Conference on Real-time Computing and Robotics (RCAR) (pp. 555-560). IEEE.

[14] Chen, P. C., & Hwang, Y. K. (1998). SANDROS: a dynamic graph search algorithm for motion planning. IEEE Transactions on Robotics and Automation, 14(3), 390-403.

[15] Karaman, S., & Frazzoli, E. (2011). Sampling-based algorithms for optimal motion planning. The international journal of robotics research, 30(7), 846-894.

[16] Chen, T. (2009, April). A simulative bionic intelligent optimization algorithm: artificial searching swarm algorithm and its performance analysis. In 2009 International Joint Conference on Computational Sciences and Optimization (Vol. 2, pp. 864-866). IEEE.

[17] Broumi, S., Bakal, A., Talea, M., Smarandache, F., & Vladareanu, L. (2016, November). Applying Dijkstra algorithm for solving neutrosophic shortest path problem. In 2016 International Conference on Advanced Mechatronic Systems (ICAMechS) (pp. 412-416). IEEE.

[18] Duchoň, F., Babinec, A., Kajan, M., Beňo, P., Florek, M., Fico, T., & Jurišica, L. (2014). Path planning with modified a star algorithm for a mobile robot.

Procedia Engineering, 96, 59-69.

[19] Guo, M., Johansson, K. H., & Dimarogonas, D. V. (2013, May). Revising motion planning under linear temporal logic specifications in partially known workspaces. In 2013 IEEE International Conference on Robotics and Automation (pp. 5025-5032). IEEE.

[20] Yu, J., & LaValle, S. M. (2013, May). Planning optimal paths for multiple robots on graphs. In 2013 IEEE International Conference on Robotics and Automation (pp. 3612-3617). IEEE.

[21] Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., ... & Ng, A. Y. (2009, May). ROS: an open-source Robot Operating System. In ICRA workshop on open source software (Vol. 3, No. 3.2, p. 5).

[22] Gammell, J. D., Srinivasa, S. S., & Barfoot, T. D. (2015, May). Batch informed trees (BIT*): Sampling-based optimal planning via the heuristically guided search of implicit random geometric graphs. In 2015 IEEE International Conference on Robotics and Automation (ICRA) (pp. 3067-3074). IEEE.

[23] Choudhury, S., Gammell, J. D., Barfoot, T. D., Srinivasa, S. S., & Scherer, S. (2016, May). Regionally accelerated batch informed trees (rabit*): A framework to integrate local information into optimal path planning. In 2016 IEEE International Conference on Robotics and Automation (ICRA) (pp. 4207-4214). IEEE.

[24] Bry, A., & Roy, N. (2011, May). Rapidly-exploring random belief trees for motion planning under uncertainty. In 2011 IEEE International Conference on Robotics and Automation (pp. 723-730). IEEE.

[25] Chi, W., Wang, C., Wang, J., & Meng, M. Q. H. (2018). Risk-DTRRT-Based Optimal Motion Planning Algorithm for Mobile Robots. IEEE Transactions on Automation Science and Engineering.

[26] Hu, Y., & Yang, S. X. (2004, April). A knowledge based genetic algorithm for path planning of a mobile robot. In IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 (Vol. 5, pp. 4350-4355). IEEE.

[27] Wang, J., Wang, N., & Jiang, H. (2015, October). Robot global path planning based on improved ant colony algorithm. In 5th International Conference on Advanced Design and Manufacturing Engineering. Atlantis Press.

[28] Liu, H., Xu, B., Lu, D., & Zhang, G. (2018). A path planning approach for crowd evacuation in buildings based on improved artificial bee colony algorithm. Applied Soft Computing, 68, 360-376.

[29] Tharwat, A., Elhoseny, M., Hassanien, A. E., Gabel, T., & Kumar, A. (2018). Intelligent Bézier curve-based path planning model using Chaotic Particle Swarm Optimization algorithm. Cluster Computing, 1-22.

[30] Wang, X., Shi, Y., Ding, D., & Gu, X. (2016). Double global optimum genetic algorithm–particle swarm optimization-based welding robot path planning. Engineering Optimization, 48(2), 299-316.

[31] Liu, J., Yang, J., Liu, H., Tian, X., & Gao, M. (2017). An improved ant colony algorithm for robot path planning. Soft Computing, 21(19), 5829-5839.

[32] Mac, T. T., Copot, C., Tran, D. T., & De Keyser, R. (2017). A hierarchical global path planning approach for mobile robots based on multi-objective particle swarm optimization. Applied Soft Computing, 59, 68-76.

[33] Le Gouguec, A., Kemeny, A., Berthoz, A., & Merienne, F. (2017). Artificial Potential Field Simulation Framework for Semi-Autonomous Car Conception.

[34] Bakdi, A., Hentout, A., Boutami, H., Maoudj, A., Hachour, O., & Bouzouia, B. (2017). Optimal path planning and execution for mobile robots using genetic algorithm and adaptive fuzzy-logic control. Robotics and Autonomous Systems, 89, 95-109.

[35] Turker, T., Sahingoz, O. K., & Yilmaz, G. (2015, June). 2D path planning for UAVs in radar threatening environment using simulated annealing algorithm. In 2015 International Conference on Unmanned Aircraft Systems (ICUAS) (pp. 56-61). IEEE.

[36] Paulo, P., Branco, F., de Brito, J., & Silva, A. (2016). BuildingsLife–The use of genetic algorithms for maintenance plan optimization. Journal of cleaner production, 121, 84-98.

[37] Petrović, M., Vuković, N., Mitić, M., & Miljković, Z. (2016). Integration of process planning and scheduling using chaotic particle swarm optimization algorithm. Expert systems with Applications, 64, 569-588.

[38] Van den Berg, J., Lin, M., & Manocha, D. (2008, May). Reciprocal velocity obstacles for real-time multi-agent navigation. In 2008 IEEE International

Conference on Robotics and Automation (pp. 1928-1935). IEEE.

[39] Fiorini, P., & Shiller, Z. (1998). Motion planning in dynamic environments using velocity obstacles. The International Journal of Robotics Research, 17(7), 760-772.

[40] Snape, J., Van Den Berg, J., Guy, S. J., & Manocha, D. (2009, October). Independent navigation of multiple mobile robots with hybrid reciprocal velocity obstacles. In 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems (pp. 5917-5922). IEEE.

[41] Claes, D., Hennes, D., Tuyls, K., & Meeussen, W. (2012, October). Collision avoidance under bounded localization uncertainty. In 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (pp. 1192-1198). IEEE.

[42] Alonso-Mora, J., Breitenmoser, A., Rufli, M., Beardsley, P., & Siegwart, R. (2013). Optimal reciprocal collision avoidance for multiple non-holonomic robots. In Distributed Autonomous Robotic Systems (pp. 203-216). Springer, Berlin, Heidelberg.

[43] Guy, S. J., Chhugani, J., Kim, C., Satish, N., Lin, M., Manocha, D., & Dubey, P. (2009, August). Clearpath: highly parallel collision avoidance for multi-agent simulation. In Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation (pp. 177-187). ACM.

[44] Van Den Berg, J., Snape, J., Guy, S. J., & Manocha, D. (2011, May). Reciprocal collision avoidance with acceleration-velocity obstacles. In 2011 IEEE International Conference on Robotics and Automation (pp. 3475-3482). IEEE.

[45] Hennes, D., Claes, D., Meeussen, W., & Tuyls, K. (2012, June). Multi-robot collision avoidance with localization uncertainty. In Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 1 (pp. 147-154). International Foundation for Autonomous Agents and Multiagent Systems.

[46] Alonso-Mora, J., Breitenmoser, A., Rufli, M., Beardsley, P., & Siegwart, R. (2013). Optimal reciprocal collision avoidance for multiple non-holonomic robots. In Distributed Autonomous Robotic Systems (pp. 203-216). Springer, Berlin, Heidelberg.

[47] Fox, D. (2002). KLD-sampling: Adaptive particle filters. In Advances in neural information processing systems (pp. 713-720).

[48] Bloembergen, D., Tuyls, K., Hennes, D., & Kaisers, M. (2015). Evolutionary dynamics of multi-agent learning: A survey. Journal of Artificial Intelligence Research, 53, 659-697.

[49] Nishitani, I., Matsumura, T., Ozawa, M., Yorozu, A., & Takahashi, M. (2015). Human-centered X–Y–T space path planning for mobile robot in dynamic environments. Robotics and Autonomous Systems, 66, 18-26.

[50] Kollmitz, M., Hsiao, K., Gaa, J., & Burgard, W. (2015, September). Time dependent planning on a layered social cost map for human-aware robot navigation. In 2015 European Conference on Mobile Robots (ECMR) (pp. 1-6). IEEE.

[51] Van Hasselt, H., Guez, A., & Silver, D. (2016, March). Deep reinforcement learning with double q-learning. In Thirtieth AAAI Conference on Artificial Intelligence.

[52] Xu, D., Fang, Y., Zhang, Z., & Meng, Y. (2017, December). Path Planning Method Combining Depth Learning and Sarsa Algorithm. In 2017 10th International Symposium on Computational Intelligence and Design (ISCID) (Vol. 2, pp. 77-82). IEEE.

[53] Peng, J., & Williams, R. J. (1994). Incremental multi-step Q-learning. In Machine Learning Proceedings 1994 (pp. 226-232). Morgan Kaufmann.

[54] Wang, C., Li, T., Meng, M. Q. H., & De Silva, C. (2018, May). Efficient Mobile Robot Exploration with Gaussian Markov Random Fields in 3D Environments. In 2018 IEEE International Conference on Robotics and Automation (ICRA) (pp. 5015-5021). IEEE.

[55] Vasquez, D., Okal, B., & Arras, K. O. (2014, September). Inverse reinforcement learning algorithms and features for robot navigation in crowds: an experimental comparison. In 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (pp. 1341-1346). IEEE.

[56] Kuderer, M. (2015). Socially Compliant Mobile Robot Navigation (Doctoral dissertation, Verlag nicht ermittelbar).

[57] Chen, Y. F., Liu, M., Everett, M., & How, J. P. (2017, May). Decentralized non-communicating multiagent collision avoidance with deep reinforcement learning. In 2017 IEEE International Conference on Robotics

and Automation (ICRA) (pp. 285-292). IEEE.
[58] Kerfs, J. N. (2017). Models for Pedestrian Trajectory Prediction and Navigation in Dynamic Environments.
[59] Chen, Y. F., Everett, M., Liu, M., & How, J. P. (2017, September). Socially aware motion planning with deep reinforcement learning. In 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 1343-1350). IEEE.
[60] Everett, M., Chen, Y. F., & How, J. P. (2018, October). Motion planning among dynamic, decision-making agents with deep reinforcement learning. In 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 3052-3059). IEEE.
[61] Ciou, P. H., Hsiao, Y. T., Wu, Z. Z., Tseng, S. H., & Fu, L. C. (2018, October). Composite Reinforcement Learning for Social Robot Navigation. In 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 2553-2558). IEEE.
[62] Long, P., Fanl, T., Liao, X., Liu, W., Zhang, H., & Pan, J. (2018, May). Towards optimally decentralized multi-robot collision avoidance via deep reinforcement learning. In 2018 IEEE International Conference on Robotics and Automation (ICRA) (pp. 6252-6259). IEEE.
[63] Wang, C., Cheng, J., Wang, J., Li, X., & Meng, M. Q. H. (2018). Efficient Object Search with Belief Road Map Using Mobile Robot. IEEE Robotics and Automation Letters, 3(4), 3081-3088.