Research on Distributed Multi-Robot Path Planning Based on Optimized Ant Colony Optimization Algorithm

Jinke Tan *

School of International Education, Changchun University of Technology, Changchun, China, 130012

* Corresponding Author Email: jungonglu580@gmail.com

Abstract. With the rapid advancement of robotics, multi-robot systems (MRS) have found extensive applications across various domains, including logistics, environmental monitoring, and post-disaster rescue operations. Effective path planning is crucial for facilitating the cooperative operation of multiple robots. This paper proposes a collaborative path planning method based on Optimized Ant Colony Optimization (OACO) and systematically compares it with traditional Ant Colony Optimization (ACO) and Model Predictive Control (MPC) in the context of single-robot path planning, as well as in multi-robot collision avoidance scenarios. The comparison with ACO reveals that OACO reduces the path length by approximately 10%, decreases the running time by 35%, and lowers the number of iterations by 27%. Additionally, the length variance indicates the stability of OACO in complex environments. When compared to MPC, OACO achieves a path length reduction of about 15% and demonstrates greater suitability for addressing global path planning challenges characterized by high complexity. These findings suggest that OACO exhibits high efficiency and accuracy in multi-robot collaborative path planning, making it well-suited for applications in complex dynamic environments.

Keywords: Distributed Path Planning, Optimized Ant Colony Optimization, Industrial Robots, Cooperative Collision Avoidance.

1. Introduction

As industry rapidly advances, the reliability of robots has led to their widespread adoption, making robotics research a growing area of interest [1]. One fundamental issue in robotic systems is the path planning problem, which has garnered significant attention [2-5]. Solutions for multi-robot path planning can generally be categorized into two types: centralized [6] and distributed methods [7]. In distributed path planning, there is no central control unit; instead, each robot operates on an equal level, allowing for information exchange and real-time autonomous data processing. In the 1990s, Dorigo, Maniezzo, et al. introduced the Ant Colony System [8, 9], which has since become a widely utilized global optimization algorithm. However, at the onset of the 21st century, modern control theory did not yield the anticipated outcomes when applied to industrial process control. Consequently, a new control algorithm, Model Predictive Control (MPC) [10], was proposed and implemented in industrial settings. This paper presents an enhancement of the Optimized Ant Colony Optimization (OACO) algorithm, derived from the Ant Colony Optimization (ACO) framework, and introduces a distributed multi-robot cooperative path planning method based on OACO. In this approach, each robot conducts path planning within a static obstacle environment and employs OACO to determine the optimal collision-free trajectory. Initially, the working environment is modeled utilizing the raster method, and the initial pheromone distribution of the Ant Colony Optimization (ACO) algorithm is determined by the distance of the subsequent optional node to the connecting line established between the starting point and the goal point. This approach is designed to mitigate the potential for blind exploration by the ant colony during the initial search phase. Furthermore, the pheromone update mechanism of the Optimized Ant Colony Optimization (OACO) enhances the accumulation of pheromone in the later stages of the algorithm, thereby facilitating a more rapid convergence of the overall process.

2. Algorithm overview

2.1. Ant Colony Optimization, ACO

Ant Colony Optimization (ACO) is a swarm intelligence optimization algorithm that draws inspiration from the foraging behavior of ants [11-14]. The core concept involves the use of pheromone accumulation and evaporation to direct ants in their search for optimal paths [15-19]. The primary steps of the algorithm are outlined as follows:

(1) The pheromone concentration is updated following each iteration according to the specified formula:

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij} \tag{1}$$

Where *i*,*j* is distributed as the start and end points, $\tau_{ij}(t)$ is the pheromone concentration on edge (*i*,*j*) at time t, ρ is the pheromone evaporation rate, and $\Delta\tau_{ij}$ is the amount of pheromone deposited by ants on the path.

(2) Pathway selection probability:

$$P_{ij}^{k} = \begin{cases} \dfrac{\left[\tau_{ij}(t)\right]^{\alpha} \cdot \left[\eta_{ij}(t)\right]^{\beta}}{\sum_{s \in allow_k} \left[\tau_{is}(t)\right]^{\alpha} \cdot \left[\eta_{is}(t)\right]^{\beta}} & s \in allow_k \\ 0 & s \notin allow_k \end{cases} \tag{2}$$

Where P_{ij} is the probability of selecting edge (i, j), η_{ij} is the heuristic information (typically the inverse of distance), α and β are parameters controlling the influence of pheromone and heuristic information, respectively, s denotes a neighboring node of the current node in the summation process, and $allow_k$ is the set of neighboring nodes that have not yet been visited.

2.2. Optimized Ant Colony Optimization, OACO

The OACO enhances traditional ACO by implementing dynamic pheromone updates and local search mechanisms to improve path planning efficiency and accuracy.

2.2.1. Algorithmic principle

The OACO algorithm simulates the foraging behavior of ants and directs path selection based on pheromone concentration. In contrast to ACO, OACO incorporates several optimization strategies, which are outlined as follows:

(1) Dynamic pheromone update:

OACO algorithm dynamically modifies the pheromone concentration throughout the pheromone update process to more accurately represent the merit of the current path. This mechanism enables the algorithm to implement real-time adjustments in response to environmental changes and variations in path quality, thereby enhancing search efficiency.

(2) Local search mechanism:

Upon identifying the initial solution, the OACO algorithm conducts a local search within the vicinity of the current optimal path. This procedure enhances the optimization of the path by investigating neighboring solutions to ascertain the discovery of a superior solution. The incorporation of local search significantly enhances both the accuracy and efficiency of the algorithm.

(3) Multiple heuristic information:

OACO integrates multiple heuristic factors, such as distance and obstacle density, to establish a multidimensional framework for path selection.

(4) Adaptive Parameter Adjustment:

The algorithm adaptively modifies its parameters in response to the prevailing search conditions in order to improve its global search capabilities.

2.2.2. Pheromone update formula

A local search mechanism and a dynamic pheromone update strategy are introduced to improve the global search capability and convergence speed of the algorithm. Its pheromone update formula:

$$\tau_{ij}(t+1)=(1-\rho)\tau_{ij}(t)+\Delta\tau_{ij}+\Delta\tau_{ij}^{local} \tag{3}$$

Where $\tau_{ij}\left(t+1\right)$ denotes the pheromone concentration on $\left(i,j\right)$ at moment t+1, and $\Delta\tau_{ij}^{local}$ represents additional pheromone from local search.

2.3. Model Predictive Control, MPC

Model Predictive Control (MPC) is a control strategy that optimizes control inputs by forecasting the future behavior of a system and adjusting the control inputs to achieve optimal performance [20-23]. MPC is capable of addressing a diverse array of constraints; however, it may exhibit significant computational complexity, particularly in large-scale applications. The primary formulation of MPC is as follows:

$$u^*=\arg\min_u\sum_{t=0}^{N}L(x_t,u_t)+\Phi(x_N) \tag{4}$$

Where u^* denotes the optimal control input, u denotes the control input vector, N denotes the length of the prediction time domain, $L(x_t,u_t)$ is the loss function at time t, and $\Phi(x_N)$ is in predicting the loss of endpoints.

3. Path planning phase

3.1. Description of the problem

The multi-robot path planning problem pertains to a shared environment in which multiple robots are required to accomplish a series of tasks and arrive at designated destinations. This problem is inherently more complex than the single-robot path planning problem, as it necessitates the planning of individual paths for each robot while also accounting for time costs and various factors, including collision detection and collaboration strategies. Consequently, the objective is to minimize the total path length or overall time cost of the multi-robot system [24-26]. In dealing with the multi-robot path planning problem, there are generally two methods: centralized and distributed. Compared to centralized methods, distributed path planning methods offer greater interoperability and independence, making their application scenarios more diverse. Therefore, this paper adopts a distributed path planning approach, dividing the multi-robot collaborative path planning into two stages. First, in the path planning stage, each robot is treated as an individual, and the optimized ant colony optimization method is used to plan the optimal global collision-free path for each robot. Then, the robots interact with each other, and the planning results of each robot are input into the cooperative collision avoidance stage. In the collaborative collision avoidance stage, collision avoidance rules are used to resolve local collisions between robots, thereby obtaining an optimal or suboptimal path combination, ensuring that the robots can work together safely and efficiently in the entire operational environment.

3.2. Environmental modeling

The raster map method effectively manages obstacle information in a straightforward and convenient manner, thereby facilitating computation and implementation. This approach partitions an environmental space into multiple grids of uniform dimensions, with obstacles represented by assigned values. Furthermore, a variety of algorithms can be applied to the raster map, which contributes to the selection of this method [27, 28].

Assuming there are n identical robots (a = 1, 2, ..., n) in a two-dimensional static environment, these robots need to cooperate to complete a transportation task. Each robot has a unique starting point and target point, denoted as and . The robots can communicate with each other through the sensors they carry and can adjust their communication based on the state of the environment. The robots can maintain a constant speed or pause their operation depending on the environmental conditions. To reduce computational complexity, a grid method is used to abstract and discretize the external environment, establishing a motion environment model for the robots. As shown in Figure 1, assume that each grid corresponds to a two-dimensional spatial coordinate. The grid centered at (0.5, 9.5) is numbered 1, the grid centered at (1.5, 9.5) is numbered 2, the grid centered at (0.5, 8.5) is numbered 11, and so on. Each grid is numbered sequentially from left to right and from top to bottom, with white grids representing active space and black grids representing obstacle space. The relationship between the grid number and the coordinate number for each grid is as follows:

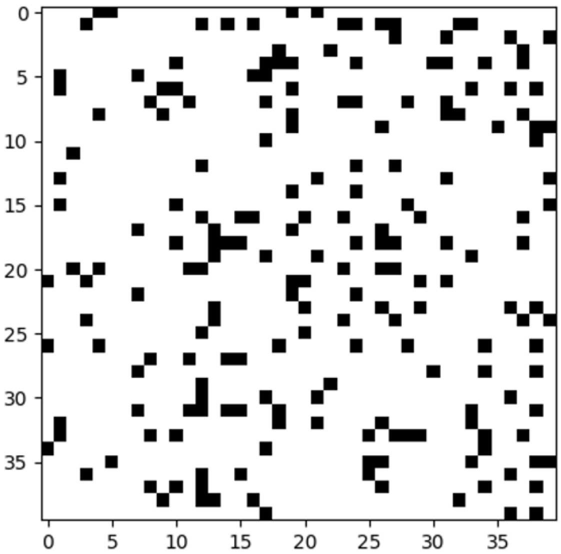

Figure 1. Grid environment

$$\begin{cases} x_a = \mathrm{mod}(a, N) \\ y_a = N + 0.5 - ceil\left(\dfrac{a}{N}\right) \end{cases} \tag{5}$$

Where a is the raster number and N the number of rows and columns of the raster map.

The specific steps are as follows:

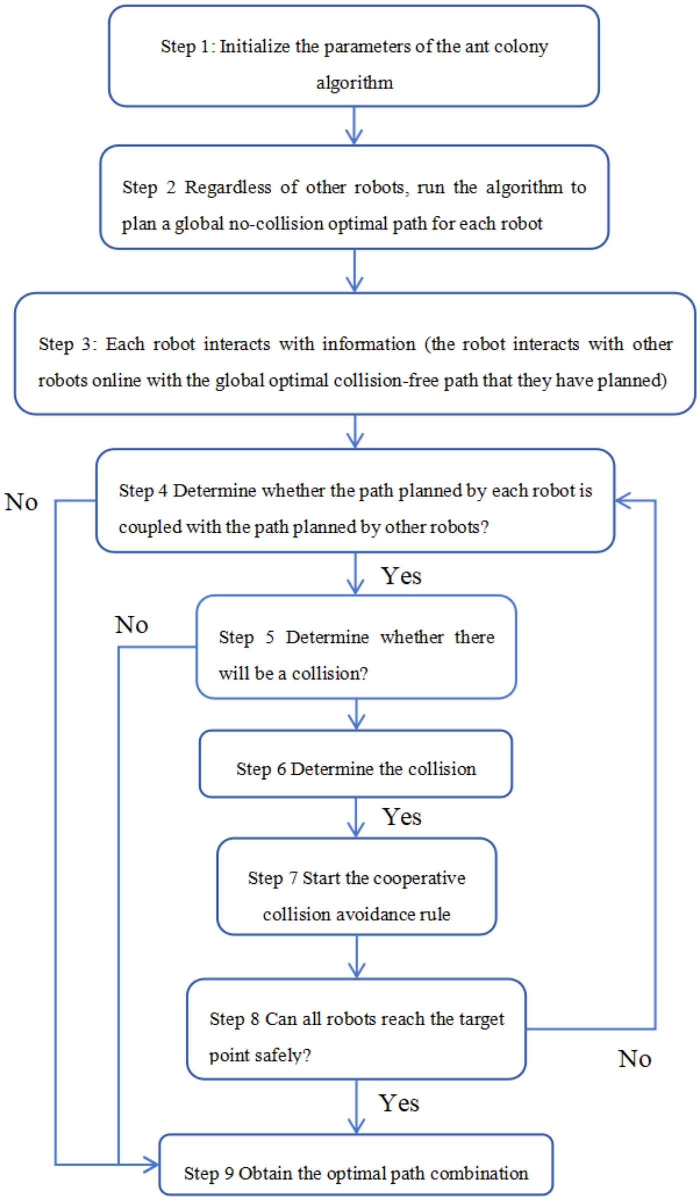

Figure 2. Flowchart of multi-robot path planning steps

3.3. Cooperative collision avoidance phase

Type of path conflict: Based on the map constructed by the raster method, a path with the lowest total cost and no collision is found when there are multiple mobile robots in the working environment, so that each robot can move from its starting point to the end point without collision. In order to achieve this goal, conflicts need to be taken into account when performing path planning, and if two robots do not collide on a path, that path can be considered as a valid solution. Below are a few common definitions of conflicts [29]. Vertex conflict: Robot 1 and Robot 2 occupy the same vertex in the same time step, then a vertex conflict occurs between the two paths, as shown in Figure 3 (a). Swap Conflict: A swap conflict occurs between two paths when Robot 1 and Robot 2 swap positions with each other within a time step, as shown in Figure 3 (b). Following Conflict: A following conflict occurs when robot 0 plans to occupy a vertex occupied by robot 1 in a previous time step, as shown in Figure 3 (c). The figure above shows the many types of collisions (conflicts) that are common for multiple robots in the same operating environment, but does not include all possible forms of conflicts. Only the common forms of conflict are listed in the figure.

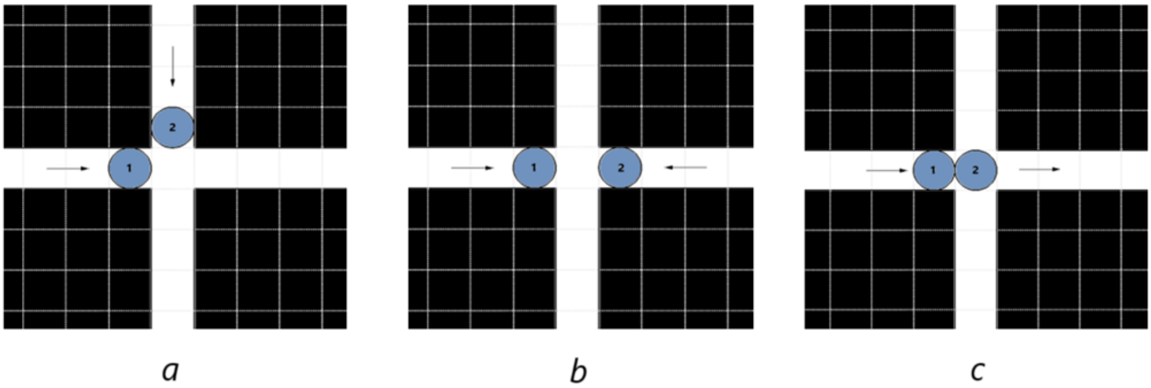

Figure 3. Type of conflict

In practical applications, the task weights assigned to each robot may vary, leading to differing priorities among the robots. This paper employs a priority-based collision avoidance strategy. According to this strategy, if two robots are scheduled to arrive at a path intersection simultaneously, and their subsequent paths do not intersect or overlap while moving in the same direction, the robot with higher priority will proceed along its planned global optimal route, while the robot with lower priority will pause momentarily before reaching the conflict point and then continue along its original path. Conversely, if the two robots arrive at the intersection simultaneously and their paths overlap while moving in the same direction, the robot with lower priority will wait for one step before reaching the conflict point, allowing the higher-priority robot to pass before it resumes its original route. In cases where both robots arrive at the intersection at the same time and their paths overlap while moving in opposite directions, the higher-priority robot will follow its originally planned path, while the lower-priority robot will re-plan a locally optimal path to ensure that all robots reach their designated goal points safely [30].

4. Results

4.1. Single Robot Path Planning Simulation Comparison Experiment

Simulation Case 1: Figures 4, 5, and 6 show the optimal paths and iterative curves planned by ACO, OACO, and MPC, respectively, in a 20×20 raster map with “U” shaped obstacles.

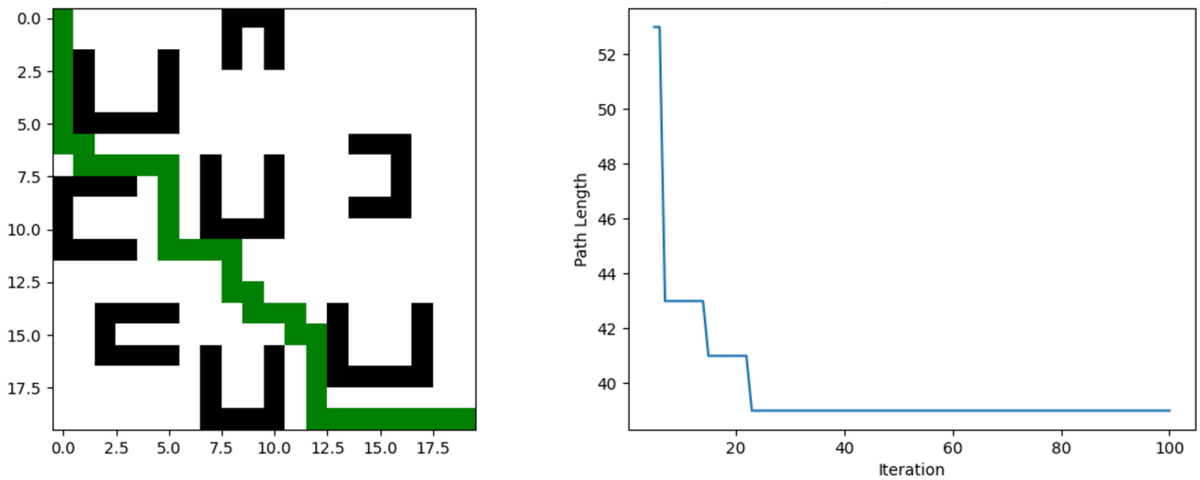

Figure 4. ACO path planning and iteration count curves in 'U' grid graphs

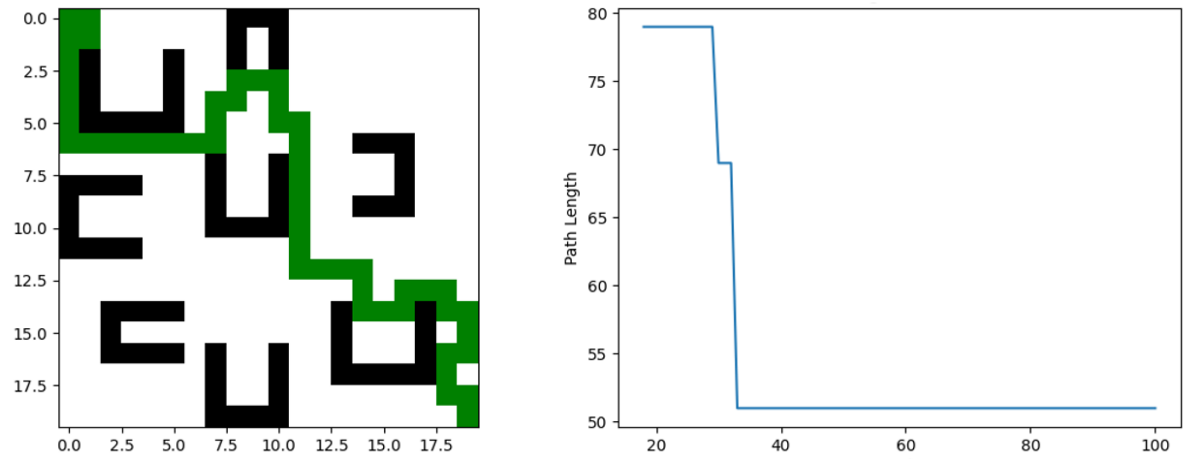

Figure 5. OACO path planning and iteration count curves in 'U' grid graphs

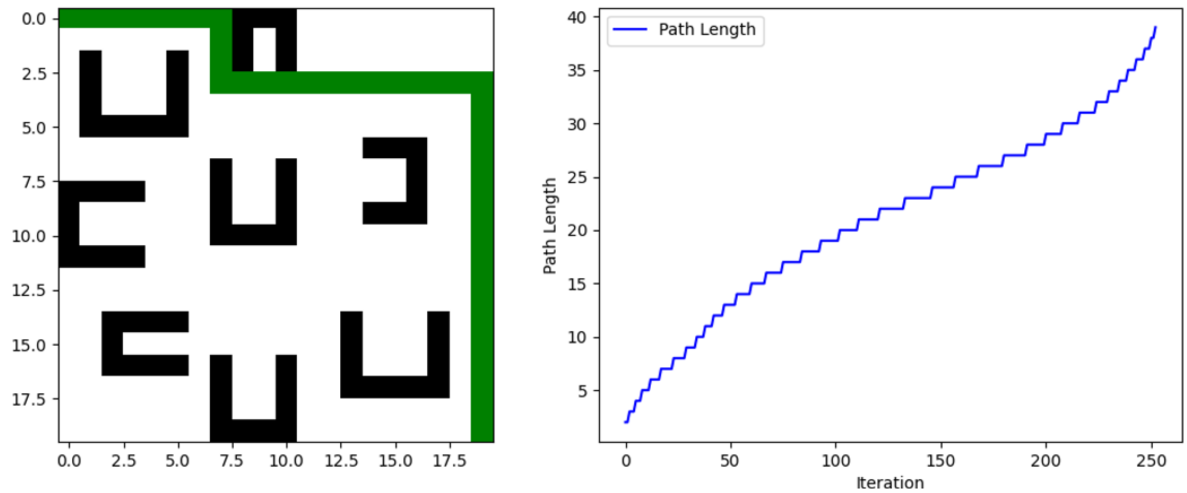

Figure 6. MPC path planning and iteration count curves in 'U' grid graphs

From the comparison of Figs. 4, 5, and 6, OACO appears to have long paths and a high number of iterations, while MPC is stuck in the case of local optimization.

Simulation Case 2: In order to verify the effectiveness of the OACO proposed in this paper and the quality of solving the path planning problem, a 40×40 raster environment is now randomly generated by python software, which requires that the obstacles in the raster should have a large complexity so as to obviously show the excellence of OACO and the prevalence of obstacles in the distribution of the map, so that the obstacles in the raster map are evenly full spread all over the map.

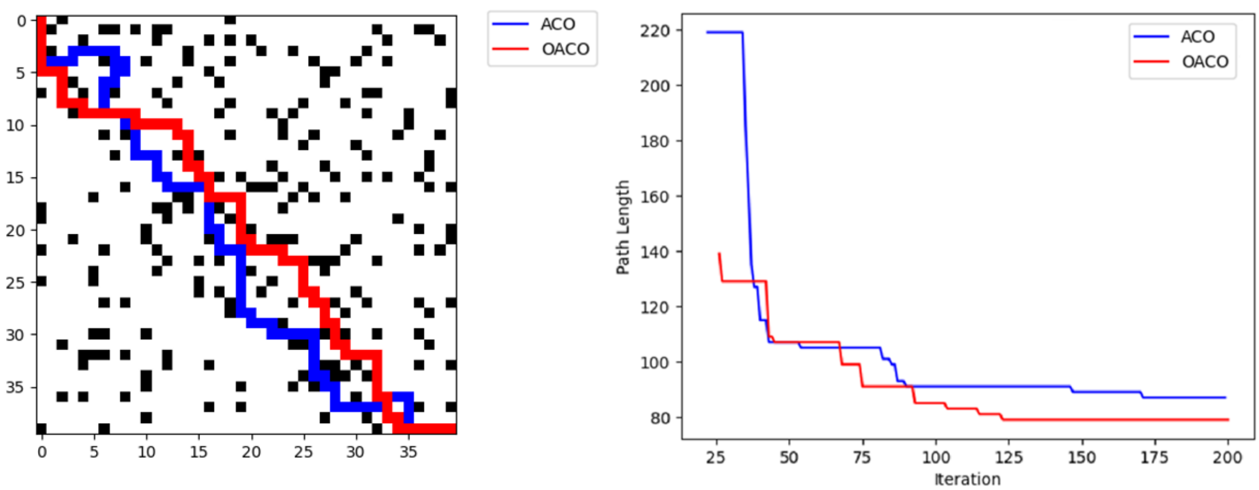

Figure 7. Path Planning and Iteration Count Curves for ACO and OACO in Complex Grid Maps

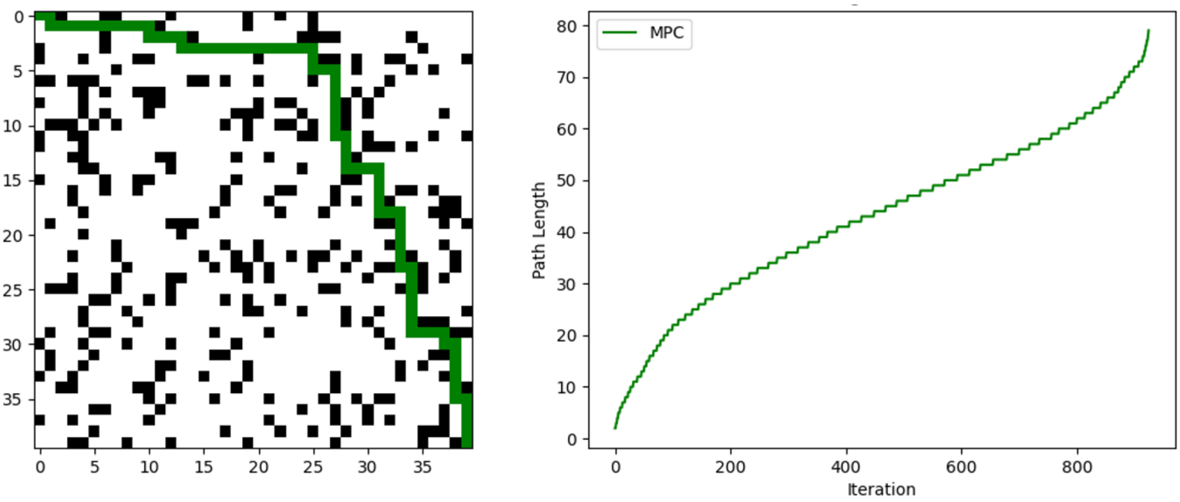

Figure 8. Path Planning and Iteration Count Curves for MPC in Complex Grid Maps

It is clear from Figure 7 that the paths planned by OACO are superior to ACO, and it can be seen from the iteration curves that OACO is more convergent than ACO, with faster iterations and shorter path lengths. Figure 8 shows the paths and iterative curves planned by MPC on a raster map (a visual comparison on the same map is not possible since the logic of ACO and MPC are different).

This simulation experiment by adjusting the system parameters and environmental complexity of the algorithm, it is obviously that OACO outperforms ACO both in terms of the length of the planned paths and the number of iterations, and MPC still falls into a local optimum.

Ten raster maps were randomly generated so that the above ACO, OACO and MPC find the optimal solution in the maps, respectively, with the average length of the paths and the average number of iterations as shown in the following table.

Table 1. Comparison of ACO, OACO and MPC in terms of path length, number of iterations, run time and length variance

Environment	Algorithm	Mean path length	Mean number of iterations	Mean running time	Path length variance
40×40 grid environment	ACO	86.7	157.1	12.45	10.7
	OACO	79	116.7	9.82	0.27
	MPC	91.2	99	0.7	8.77

4.2. Multi-robot cooperative path planning simulation experiment

Simulation Case 3: In order to verify the effectiveness of the OACO proposed in this paper, a 25×25 grid environment is selected in this paper, and a multi-robot cooperative path planning simulation experiment is carried out using the OACO. As shown in Figure 9, it is assumed that in this environment, the task weights of robots R1, R2 and R3 are 1, 1.5 and 2, respectively, the corresponding starting points of the four robots are S1, S2 and S3, and the goal points are G1, G2 and G3, respectively.

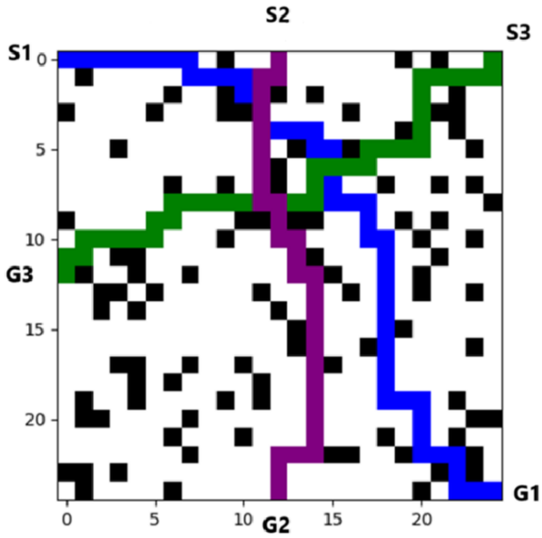

Figure 9. Multi-robot path planning using OACO

4.3. Analysis of experimental results

In the analysis of Simulation Case 1, it is essential to meticulously evaluate the algorithm parameters, design objectives, and the complexity of the specific application scenario when comparing the performance of OACO and ACO. It is possible to observe instances where the path length and the number of iterations for OACO exceed those of ACO. Such occurrences may be attributed to several factors:

(1) Setting Parameters for Algorithms

Parameter sensitivity is a critical factor influencing the performance of the OACO and ACO algorithms, as their efficacy may be significantly affected by the specific parameter settings, such as the importance of pheromones, the significance of heuristic functions, and the volatility coefficient, among others. Inadequate tuning of the parameters in OACO can result in a deterioration of the algorithm's overall performance. Furthermore, the exploration versus exploitation trade-off presents another challenge; while OACO incorporates a greater degree of randomness, which can enhance the diversity of path selections, it may also result in the formation of longer paths. An excessive degree of randomness may hinder the algorithm's ability to converge on an optimal solution.

(2) Algorithm Design

The exploration capability and convergence speed of the OACO algorithm are critical factors in its performance. OACO is specifically designed to enhance the exploration of the solution space, which may result in a trade-off with convergence speed. An excessively strong exploration capability may lead to longer solution paths and an increased number of iterations. Additionally, the pheromone updating mechanism is a crucial component of various algorithms, and significant differences may exist in how these mechanisms operate and influence path selection. If the pheromone updating mechanism employed by OACO is not sufficiently effective, it may result in suboptimal path selection compared to that of ACO algorithms.

(3) Initialization and Diversity of Population

The quality of initial solutions in the OACO algorithm may be inferior to that of the ACO algorithm, resulting in the generation of suboptimal paths during the early iterations. Furthermore, with respect to population diversity, OACO may exhibit reduced diversity compared to ACO in certain instances. This diminished diversity can lead to a greater number of paths being explored within the same solution space, ultimately resulting in longer paths.

(4) Environmental Complexity

The complexity of the environment can significantly impact the performance of optimization algorithms. In certain intricate environments, the stochastic nature of the OACO algorithm may lead to the selection of suboptimal paths. In contrast, the ACO algorithm may more rapidly identify superior paths.

In the analysis of Simulation Case 2, as presented in Table 1, it is evident that OACO optimizes about 10% optimization in path length, 35% in runtime, and 27% in the number of iterations. When compared to ACO. Compared to MPC, OACO exhibits a 15% enhancement in path length during global planning; however, it does not match MPC in terms of the number of iterations and runtime are not as good as MPC. Comparing with Table 1 above, although the run response time and the number of iterations of MPC are faster than that of OACO, the path planning on the raster map falls into a local optimum, which indicates that OACO is more suitable for solving the global path planning problem. In addition, OACO can update the paths by re-iterating when facing dynamic environments (e.g., obstacle changes or goal point movement), does not require explicit model descriptions, and can deal with nonlinear, non-convex, and large uncertainty problems with wider adaptability. In contrast, MPC requires an explicit mathematical model of the system, especially in dynamic environments, where the model needs to be updated in real time, with higher computational complexity, making it difficult to adapt quickly to changes in the environment. The computational complexity of MPC grows rapidly with the increase of the prediction time window and the control variables, and its optimization problems usually require the solution of a continuous optimization problem, which makes it difficult to carry out large-scale parallelization, and thus MPC is not suitable for the study of multi robot interaction parallel planning.

In the analysis of Simulation Case 3, as illustrated in Figure 9, it is evident that the trajectories planned by robots R1 and R2, as well as the trajectory planned by robot R3, exhibit intersections within the designated numbered grid. Upon further examination, it can be concluded that these robots will not reach the intersection point simultaneously. Furthermore, it is observed that the remaining robots will also not arrive at the path intersection concurrently. Consequently, the likelihood of collisions between the robots is effectively mitigated, thereby ensuring that the planned paths are both safe and reliable.

Overall, the OACO algorithm demonstrates superior performance compared to the other algorithms presented in this paper regarding global path planning in both static and dynamic environments. This is particularly evident in high-complexity environments, where OACO exhibits notable responsiveness and interoperability.

5. Conclusion

Comprehensive comparative analyses indicate that the OACO algorithm demonstrates effective performance in multi-robot collaborative path planning, particularly in operations characterized by high complexity and demanding workloads. The algorithm is associated with shorter path lengths, accelerated convergence rates, and enhanced stability. Future research endeavors should focus on further investigating the application and refinement of the algorithm in more intricate environments, particularly in contexts involving dynamic obstacles and fluctuating task requirements. Nonetheless, the computational complexity of the algorithm when addressing ultra-large-scale problems warrants careful consideration. Furthermore, the algorithm may not achieve optimal performance in highly dynamic environments or tasks characterized by significant uncertainty, especially in application scenarios that impose stringent real-time requirements. Consequently, it may be necessary to integrate the algorithm with other techniques to improve its adaptability.

References

[1] ZHOU X, WANG X, XIE Z, et al. Online obstacle avoidance path planning and application for arc welding robot [J]. Robotics and Computer-Integrated Manufacturing, 2022, 78: 102413.

[2] ZHANG X, XUE L, ZHANG Z, et al. A Novel Application System of Assessing the Pronunciation Differences Between Chinese Children and Adults [J]. The Open Biomedical Engineering Journal, 2016, 10: 91 - 100.

[3] GUO X, JI M, ZHAO Z, et al. Global path planning and multi-objective path control for unmanned surface vehicle based on modified particle swarm optimization (PSO) algorithm [J]. Ocean Engineering, 2020, 216: 107693.

[4] MIAO C, CHEN G, YAN C, et al. Path planning optimization of indoor mobile robot based on adaptive ant colony algorithm [J]. Computers & Industrial Engineering, 2021, 156: 107230.

[5] WU L, HUANG X, CUI J, et al. Modified adaptive ant colony optimization algorithm and its application for solving path planning of mobile robot[J]. Expert Systems with Applications, 2023, 215: 119410.

[6] JOSE K, PRATIHAR D K. Task allocation and collision-free path planning of centralized multi-robots' system for industrial plant inspection using heuristic methods [J]. Robotics and Autonomous Systems, 2016, 80: 34 - 42.

[7] JIANG X, ZENG X, XIE L, et al. Variance-reduced reshuffling gradient descent for nonconvex optimization: Centralized and distributed algorithms [J]. Automatica, 2025, 171: 111954.

[8] ROMET P, GECHTER F, AGLZIM E, et al. An environment-based approach for the ant colony convergence [J]. Procedia Computer Science, 2020, 177: 488 - 495.

[9] BLUM C. Ant colony optimization: A bibliometric review [J]. Physics of Life Reviews, 2024, 51: 87 - 95.

[10] MORATO M M, FELIX M S. Data Science and Model Predictive Control: A survey of recent advances on data-driven MPC algorithms [J]. Journal of Process Control, 2024, 144: 103327.

[11] LIU C, WU L, XIAO W, et al. An improved heuristic mechanism ant colony optimization algorithm for solving path planning [J]. Knowledge-Based Systems, 2023, 271: 110540.

[12] GUAN X, LI G. Optimization of Cold Chain Logistics Vehicle Transportation and Distribution Model Based on Improved Ant Colony Algorithm [J]. Procedia Computer Science, 2023, 228: 974 - 982.

[13] BAYRAM O B, OZCAN A. Determining optimal paths of virtual links in Avionics Full-Duplex Switched Ethernet networks using modified ant colony optimization algorithm [J]. Expert Systems with Applications, 2023, 229: 120433.

[14] ZHANG D, LUO R, YIN Y, et al. multi-objective path planning for mobile robot in nuclear accident environment based on improved ant colony optimization with modified A*[J]. Nuclear Engineering and Technology, 2023, 55 (5): 1838 - 1854.

[15] COMERT S E, YAZGAN H R. A new approach based on hybrid ant colony optimization-artificial bee colony algorithm for multi-objective electric vehicle routing problems [J]. Engineering Applications of Artificial Intelligence, 2023, 123: 106375.

[16] LIU D, HU X, JIANG Q. Design and optimization of logistics distribution route based on improved ant colony algorithm [J]. Optik, 2023, 273: 170405.

[17] EBID A M, ABDEL-KADER M Y, MAHDI I M, et al. Ant Colony Optimization based algorithm to determine the optimum route for overhead power transmission lines [J]. Ain Shams Engineering Journal, 2024, 15 (1): 102344.

[18] HOU Y, GUO X, HAN H, et al. Knowledge-driven ant colony optimization algorithm for vehicle routing problem in instant delivery peak period [J]. Applied Soft Computing, 2023, 145: 110551.

[19] SCIANNA M. The AddACO: A bio-inspired modified version of the ant colony optimization algorithm to solve travel salesman problems [J]. Mathematics and Computers in Simulation, 2024, 218: 357 - 382.

[20] BERNARDINO L F, SKOGESTAD S. Optimal switching of MPC cost function for changing active constraints [J]. Journal of Process Control, 2024, 142: 103298.

[21] YOU Y, YANG Z, ZHUO H, et al. Research on ground mobile robot trajectory tracking control based on MPC and ANFIS [J]. Control Engineering Practice, 2024, 152: 106040.

[22] MALLICK S, AIRALDI F, DABIRI A, et al. multi-agent reinforcement learning via distributed MPC as a function approximator [J]. Automatica, 2024, 167: 111803.

[23] GUPTA S, SAINI R S T, GANESH H S. Hierarchical MPC for a dynamic process system employing parametric global optimization strategy [J]. Digital Chemical Engineering, 2023, 9: 100120.

[24] ABBAS M N, LISTON P, LEE B, et al. CESDQL: Communicative experience-sharing deep Q-learning for scalability in multi-robot collaboration with sparse reward [J]. Knowledge-Based Systems, 2024, 306: 112714.

[25] LI Z, SHI N, ZHAO L, et al. Deep reinforcement learning path planning and task allocation for multi-robot collaboration [J]. Alexandria Engineering Journal, 2024, 109: 408 - 423.

[26] TIAN B, KAUL H, JANARDHANAN M. Balancing heterogeneous assembly line with multi-skilled human-robot collaboration via Adaptive cooperative co-evolutionary algorithm [J]. Swarm and Evolutionary Computation, 2024, 91: 101762.

[27] FU B, CHEN Y, QUAN Y, et al. Bidirectional artificial potential field-based ant colony optimization for robot path planning [J]. Robotics and Autonomous Systems, 2025, 183: 104834.

[28] ZHANG Y, WANG J, WANG Y, et al. Intelligent planning of fire evacuation routes in buildings based on improved adaptive ant colony algorithm [J]. Computers & Industrial Engineering, 2024, 194: 110335.

[29] DAS S, MISHRA S K. A Machine Learning approach for collision avoidance and path planning of mobile robot under dense and cluttered environments [J]. Computers and Electrical Engineering, 2022, 103: 108376.

[30] LAZAROWSKA A. A comparison of the Firefly Algorithm and the Ant Colony Optimization for ship collision avoidance [J]. Procedia Computer Science, 2023, 225: 2037 - 2046.