2021 International Conference on Networking Systems of AI (INSAI) | 978-1-6654-0859-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/INSAI54028.2021.00028

An Improved Ant Colony Optimization Algorithm for Multi-Agent Path Planning

Shuai Huang
Academy for Engineering and Technology
Fudan University
Shanghai, China
Engineering Research Center of AI and Robotics
Shanghai, China
shuaihuang20@fudan.edu.cn

Dingkang Yang
Academy for Engineering and Technology
Fudan University
Shanghai, China
Ji Hua Laboratory
Foshan, China
dkyang20@fudan.edu.cn

Chuyi Zhong
Academy for Engineering and Technology
Fudan University
Shanghai, China
Engineering Research Center of AI and Robotics
Shanghai, China
cyzhong20@fudan.edu.cn

Shichao Yan
Academy for Engineering and Technology
Fudan University
Shanghai, China
scyan20@fudan.edu.cn

Lihua Zhang*
Academy for Engineering and Technology
Fudan University
Shanghai, China
Ji Hua Laboratory
Foshan, China
Engineering Research Center of AI and Robotics
Shanghai, China
Jilin Provincial Key Laboratory of Intelligence Science and Engineering
Changchun, China
lihuazhang@fudan.edu.cn

***Abstract*—In recent years, Ant Colony Optimization algorithm has become one of the most widely used heuristic algorithms and has been apply to solve different types of path planning problems. However, there still are some problems in Multi-Agent Path Finding, such as low convergence efficiency, easy to fall into local optimum and vertex conflict. In this paper, we proposed an Improved Ant Colony Optimization algorithm based on parameter optimization and vertex conflict resolution. First of all, we initialize the distribution of pheromones to reduce the blindness of the algorithm in the early stage. Secondly, we introduce an adaptive pheromone intensity and pheromone reduction factor to avoid the algorithm falling into local optimum. On this basis, the algorithm's global search ability and convergence speed are improved by dynamic modification of the evaporation factor and heuristic function. In addition, the strategy of dynamically modifying the influence factor and heuristic function improves the global search ability and convergence speed of the algorithm. To solve vertex conflict in MAPF, we use the design conflict prediction and resolution strategy to effectively avoid vertex conflict and improve the reliability of the multi-agent system. Simulation experiments verify the effectiveness and adaptability of IACO under different complexity environments, and prove that IACO has good convergence speed and path global optimization ability.**

Keywords—ant colony optimization, multi-agent path finding, vertex conflict, path planning

I. Introduction

Recently, with the rapid development of artificial intelligence technology, path optimization has become one of the hot research topics. Path optimization refers to finding an optimal forward path in the workspace and effectively avoiding obstacles according to the evaluation criteria of optimal performance when obstacles are known or unknown, including convergence speed, path length. A* algorithm [1] and Dijkstra algorithm [2] are typical path planning methods. With the continuous development of artificial intelligence optimization algorithms, bionic algorithms such as genetic algorithms [3] and ant colony optimization algorithm [4] have attracted more and more attention from people. However, they also have some defects that need to be improved. As a kind of bionic intelligent algorithm, Ant Colony Optimization algorithm (ACO) has been widely studied by scholars due to its advantages, such as good robustness and robust and intelligent search ability. As mentioned above, it also has problems, such as long search time, slow convergence speed, optimal local solution, and so on [5]. In this paper, we study the multi-agent routing problem based on ant colony optimization algorithm, and perform simulation experiments to verify the performance of our method. The contributions of this paper are summarized as follows:

- We propose an Improved Ant Colony Optimization (IACO) algorithm, which focuses on the multi-agent path finding problems, and improves the efficiency

978-1-6654-0859-2/21/$31.00 ©2021 IEEE
DOI 10.1109/INSAI54028.2021.00028

and reliability of the multi-agent system by parameter optimization and vertex conflict resolution.

- We improve and optimize the ACO based on the essential parameters affecting the performance of the ACO. Through adaptive and dynamic parameter adjustment mode, the algorithm is prevented from falling into local optimum. We utilize an adaptive and dynamic parameter adjustment strategy to avoid local optimum.
- We propose a conflict resolution strategy in multi-agent path finding (MAPF), which effectively avoids the problem of vertex conflict in path finding and ensures the stability of the system.

II. RELATED WORKS

A. Multi-Agent Path Finding

MAPF is a problem of path finding for multiple agents, which requires that each agent reaches its goal without conflict [6], [7]. It has topical applications in warehouse management [8], airport towing [9], autonomous vehicles, robotics [10] and so on. Nazarahari et al. [11] improved the genetic algorithm and added collision elimination operator to the algorithm to eliminate possible collisions between robots. However, this method is limited to the external environment with fewer obstacles and cannot be apply to the complex environment with more obstacles. Cao et al. [12] proposed a path planning method for multi-robots based on reserved regions, which avoided the problem of highly coupled paths among robots in path planning. Yu et al. [13] set the priority of each robot to minimize the work completion time. However, due to the strong independence of each robot, it is easy to find an optimal local situation by using this method.

Nevertheless, AI researchers in the past years have made substantial progress in finding optimal solutions to a growing number of scenarios and for over a hundred agents [14] - [16]. However, most prior work assumed that (1) time is discretized into time steps, (2) the duration of every action is one time step, and (3) in every time step each agent occupies exactly a single location. These simplifying assumptions limit the applicability of MAPF algorithm in real world.

B. Ant Colony Optimization Algorithm

ACO is essentially a system based on agents that simulate the natural behavior of ants, including mechanisms of cooperation and adaptation. Corne et al. [17] take this kind of system as a new meta heuristic to solve combinatorial optimization problems. The new meta heuristic has been shown to be both robust and versatile and it has been applied successfully to a range of different combinatorial optimization problems. ACO is a bionic optimization algorithm, which obtains the shortest path by simulating the behavior of ants artificially. Because ants are not sensitive to the surrounding environment, changes in the background will not affect the ants quickly to find a new shortest path. For ants, the deciding factor is a special secretion called pheromones, which are secreted by the ants to find the shortest route. Each ant releases pheromone along the path, and the shortest path can be found based on the difference in pheromone concentration along each path.

III. METHODOLOGY

In the traditional ACO, the information heuristic factor, the expected heuristic factor, the pheromone volatility coefficient, and so on are all essential parameters, which are usually fixed. So the primary ACO is easy to fall into local optimum and the iteration speed of ACO is slow in a dynamic environment. Besides, the influence of parameters on the algorithm is different at different periods in MAPF. Therefore, we improve the traditional ACO to overcome the above defects.

A. Improved Initial Pheromone

Uniform distribution of initial pheromone concentration often leads to some problems such as strong search blindness and slow convergence speed in the initial iteration. In previous studies, researchers [18], [19] confirm that differentiated allocation of pheromone initialization is conducive to finding the optimal path more robustly. In order to reduce the misleading of false heuristic information to ants and improve the ability of searching for the best, we initialize the pheromone distribution in the algorithm based on the idea of weak heuristic. The initialization function is defined as follows:

$$\tau_t = \begin{cases} C & t \in R \\ \tau_0 & \text{others} \end{cases} \tag{1}$$

where C is constant and greater than τ_0, and R is the set of feasible favorable node regions for ants.

B. Adaptive Pheromone Intensity

In the early stage of ants exploration, it is necessary to expand the search range to find the better path. At this time, the pheromone intensity must be set to be minor so that the ant can avoid continuing to walk in one path or getting the optimal local solution. However, in the later stage of exploration, the exploration of the path has been completed, and the shorter path has been determined. At this time, it is not necessary to explore in a large range, but to stabilize a short path. Therefore, the value of Q needs to be large enough. In this case, the value of pheromone intensity is no longer constant and needs to change continuously with the increase of the number of iterations. The adaptive pheromone intensity function is as follows:

$$Q_n = \phi * ln(n), \phi \in [0,1] \tag{2}$$

where n is the current iteration number, and ϕ is the regulatory factor.

C. Pheromone Reduction Factor

Pheromone updating can be divided into two processes: real-time updating and global updating. When the ant updates the global pheromone at the end of an iteration, the pheromone value on the shortest path found will increase. Based on the average pheromone value of all paths after the end of each round, we introduce the pheromone reduction factor, which is defined as:

$$\begin{gathered} v = \frac{L_b}{L_a} \\ L_a = \sum_{i=1}^{k} L^i / n \end{gathered} \tag{3}$$

where L_b denotes the shortest path found in one search pass, and L_a denotes the average path length of ants in one search pass. The shorter the search path is, the smaller the pheromone reduction factor is, the more pheromones are retained, and vice versa. When an ant complete path search, the global pheromone need to be updated. The update rules are as follows:

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + (1-\upsilon)\nabla\tau_{ij}(t) \tag{4}$$

where $\tau_{ij}(t)$ denotes the pheromone in (i, j) at time t, and ρ, υ are the weight parameters. The path pheromone updating method with a reduction factor can accelerate the algorithm's convergence speed and significantly increase the probability of choosing the optimal path.

D. Optimized Heuristic Function

Due to the insignificant positive feedback, ants easily fall into local optimum in a grid environment. By using the square of the sum of the distance between the current node and the next node and the distance between the next node and the target node to optimize the heuristic function, we improve the heuristic function in iterative search. This function enables ants to obtain the guiding direction at the initial stage of search and increase the influence of the target node on the next node. The improved heuristic function is defined as:

$$\begin{gathered} d_{jk} = \sqrt{(x_j - x_k)^2 + (y_j - y_k)^2} \\ \eta_{ij}^* = \frac{1}{(d_{ij}+d_{jk})^2} \end{gathered} \tag{5}$$

where d_{jk} denotes the Euclidean distance between node J and target point K, and η_{ij}^* denotes the influence coefficient. By changing the traditional d_{ij} to $(d_{ij} + d_{jk})^2$, the purpose of the search is enhanced, and the probability of falling into the locally optimal solution is reduced.

E. Dynamic Evaporation Factor

The traditional evaporation factor of ant colony optimization is constant in algorithm running and cannot adapt to the dynamic changing environment. Therefore , we adopt an adaptive dynamic evaporation factor, which is as follows:

$$\rho_d(t+1) = \begin{cases} \frac{9n}{N}\rho(t) & \frac{9n}{N}\rho(t) \le \rho_{max} \\ \rho_{max} & \text{others} \end{cases} \tag{6}$$

where ρ_d denotes the improved pheromone evaporation factor ,and N denotes the maximum iteration number.

F. Vertex conflict problem

MAPF system is gradually replacing Single-Agent Path Finding (SAPF) system. Compared with the single agent, all agents can coordinate and cooperate to improve work efficiency when multi-agent works in a cluster [7], [20]. However, if the agent arrives at the same node at the same time, path conflict and system deadlock will occur in the multi-agent cluster system. To solve the problem of vertex conflict in raster map path-finding in this paper, we proposed a conflict prediction and resolution strategy.

1) The Conflict Check : Multi-agent system *A* plans the initial path group $L_c = [L_{c_1}, L_{c_2}, \cdots, L_{c_n}]$ denotes for each agent, and each path is a set of node coordinates. If the intersection of the initial path L_{c_i} of A_i and the initial path node set of other agents is empty, then agent A_i can move safely according to the initial path. Suppose the intersection of the coordinate location of the path and the coordinate set of the initial path of other agents is not empty. In that case, there is an intersection point between the initial path and the initial path of other agents and it is necessary to perform safety judgment. The judgment method is as follows:

$$\left|\frac{d_{S_i W_{i,j}(k)}}{v_i} - \frac{d_{S_j W_{i,j}(k)}}{v_j}\right| \ge \Delta T$$

where $d_{S_i W_{i,j}(k)}$ denotes the distance from the starting node S_i to the intersection node $W_{i,j}(k)$ along the initial path of A_i. This means that when the two move along the initial path, they will not reach intersection point $W_{i,j}$ within the safe time, and there will be no path conflict. The intersection node is called pseudo-conflict node $M_{i,j}$. If the formula is not satisfied, path conflict will occur, and path coordination is required. The set of intersection nodes of the initial path between agent A_i, and other agents is $W_i = W_{i,1} \cup W_{i,2} \cup \cdots \cup W_{i,n}$, and the safety judgment is made for the intersection nodes in W_i successively. After the judgment is completed, the pseudo-conflict node-set M_i without path conflict is removed from W_i to obtain the path conflict node-set $Z_i = Z_{i,1} \cup Z_{i,2} \cup \cdots \cup Z_{i,n}$ between agent A_i and other agents.

2) Conflict Resolution: Assume a conflict node $Z_{i,j}(k)$ between agent A_i and agent A_j. Then compare the time required for them to reach the conflict node. We utilizes the principle of "first come first, second come coordination" to determine the agent which needs path coordination. If they reach the conflicting node simultaneously, an agent will be selected for path coordination randomly. The evaluation criteria is as follows:

$$\left|\frac{d_{S_i M_i(k)}}{v_i} + \Delta T - \frac{d_{S_j M_i(k)}}{v_j}\right| \ge \Delta T$$

This indicates that when agent A_i pauses ΔT at the start node, just before moving to the conflict node. When agent A_j drives away from the peak of the conflict , agent A_i moves on the original path. A_i adopts this strategy to resolve conflicts without generating new conflict nodes.

After optimizing the basic ACO and adding the conflict resolution strategy, the overall flow chart of the improved algorithm is shown in Fig.1.

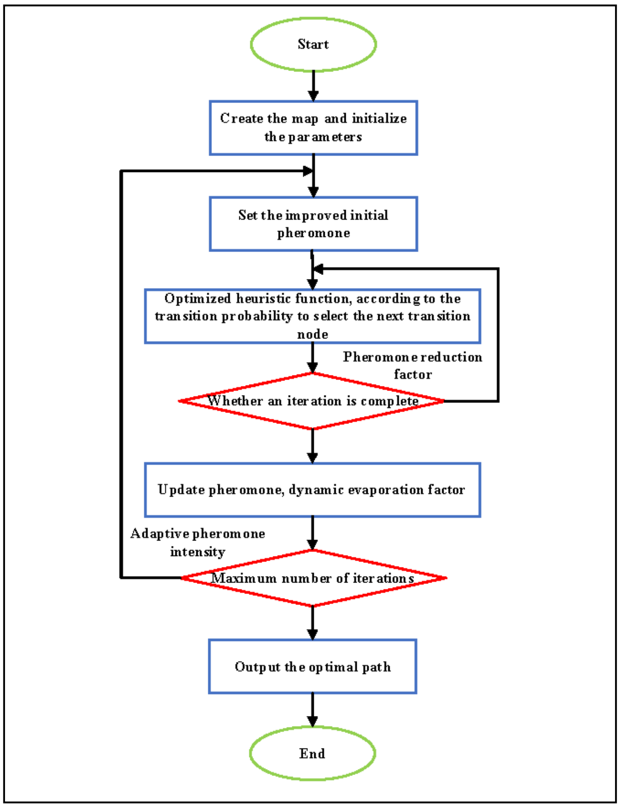

Fig. 1. IMPROVED ALGORITHM FLOW CHART

IV. EXPERIMENTS AND DISCUSSIONS

A. Quantitative Experiments

According to the test requirements of ACO (A) and IACO (I) algorithm in different simulation maps, the number of ants, initial pheromone, pheromone intensity, evaporation factor, and other relevant parameters are selected as shown in Table I. The pheromone intensity and pheromone reduction factor of IACO varies with the number of iterations adaptively.

Table II shows the algorithm performance comparison results of ACO and IACO in different maps containing different agents. According to the test in four kinds of simulation maps, the IACO can find shorter optimal paths in the same map environment. The optical path lengths from large to small are 4, 4, 7, 17, respectively, according to the map scale. In particular, in the Small map, due to the simplicity of the environment, the optimal path lengths obtained by the two algorithms are equal. Secondly, the running speed of the IACO algorithm is faster, reaching 0.11, 0.5, 5.05 and 267.8, respectively. Meanwhile, in terms of the number of iterations to find the optimal path, IACO is significantly less than the ACO, which is 2, 5, 14, 36. This fully shows that the optimization and improvement of ACO in this paper effectively improves the algorithm's convergence ability and search speed.

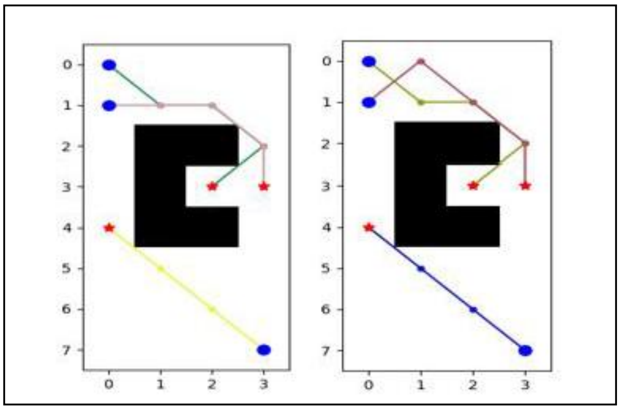

Fig. 2. PATH FINDING RESULTS OF SSMALL MAP AGENTS

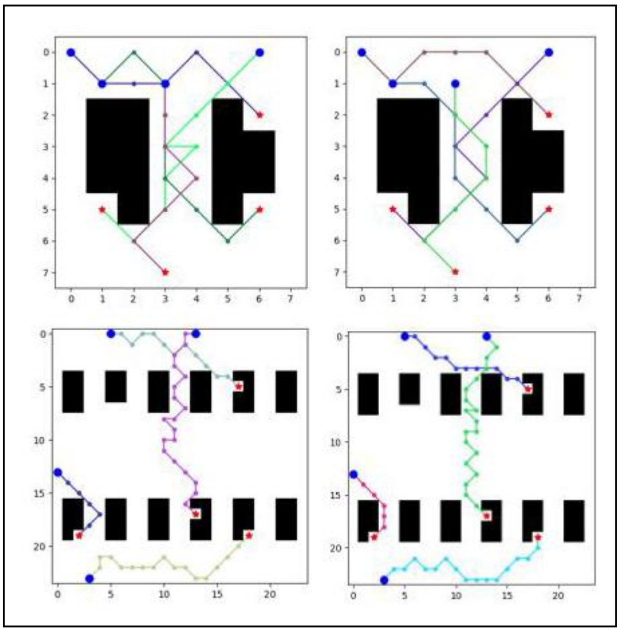

Fig. 3. PATH FINDING RESULTS OF SSMALL MAP AGENTS

TABLE I. ALGORITHM PARAMETER SELECTION

Maps	Agents	Optimal Path Length		Time(s)		Iterations	
		ACO	*IACO*	*ACO*	*IACO*	*ACO*	*IACO*
SMALL	2	4	4	0.296	0.11	5	2
SSMALL	3	5	4	1.046	0.5	10	5
MIDDLE	4	8	7	11.721	5.05	20	14
BIG	4	22	17	514.531	267.8	52	36

B. Conflict resolution

To stimulate the necessity of the proposed vertex conflict resolution strategy, this paper takes the SSMALL map as an example to conduct experiments only for the conflict problem. Combined with Fig.2 and Table III, agents A_1 and A_2 generate two conflict points (1, 2) and (2, 3) during path finding. In ACO , the path of the A_1 series for [(0, 0), (1, 1), (1, 2), (2, 3), (3, 2)], the path of the A_2 sequence for [(1, 0), (1, 1), (1, 2), (2, 3), (3, 3)]. After the conflict resolution strategy is improved, when the vertex conflict between A_1 and A_2 occurs, the first step is to meditate according to the sequence . The A_1 into the path of the sequence [(0, 0), (1, 1), (1, 2), (2, 3), (3, 2), A_2 in order to avoid conflict, path sequence into [(1, 0). (0, 1), (0, 1), (1, 2), (2, 3), (3, 3)]. By using A_2 to pause once, vertex conflict is effectively avoided.

TABLE II. PERFORMANCE COMPARISON OF ACO AND IACO IN DIFFERENT MAPS

Maps	Number of Ants		Initial Pheromone		Maximum Pheromone Intensity		Evaporation Factor	
	A	*I*	*A*	*I*	*A*	*I*	*A*	*I*
SMALL	20	20	50	[20,50]	3	3	0.5	[0.16-0.68]
SSMALL	30	30	50	[20,50]	3	3	0.5	[0.16-0.68]
MIDDLE	50	50	100	[50,100]	5	5	0.3	[0.2-0.54]
BIG	100	100	200	[100,200]	8	8	0.3	[0.25-0.48]

TABLE III. CONFLICT PROBLEM ANALYSIS

Agents	Start	End	Time(s)		Conflict Point
			ACO	*IACO*	
A_1	(0,0)	(3,2)	1.045	1.048	(1,2)
A_2	(1,0)	(3,3)	1.045	1.048	(2,3)
A_3	(7,3)	(4,0)	1.045	1.048	

We selected representative MIDDLE and BIG maps for analysis. As shown in Fig.3, the circle mark indicates the starting point, and the asterisk mark indicates the endpoint. Both algorithms can help each agent to find the shortest moving path smoothly. With the increase of traditional ACO iterations, ants will frequently enter a deadlock state to adopt a backsliding strategy. This leads to more vertex collisions between agents, and the resulting feasible paths are significantly longer than the improved algorithm. The proposed resolution strategy can effectively prevent ants from falling into local optimum. This also means that the vertex collisions can be avoided. In conclusion, optimizing the initial pheromone and using adaptive pheromone intensity can effectively adjust the influence of the poor ant on the path selection next and prevent the misleading effect of the non-optimal pheromone on the ants. Meanwhile, the pheromone concentration difference which generated by the introduced pheromone reduction factor can effectively help the ants break away from local optimum. As the algorithm iteration goes on, the dynamic pheromone evaporation factor ensures the randomness of the ant colony and improves the algorithm's global search ability.

V. CONCLUSION

In this paper, we use the ACO algorithm to study MAPF problem and proposed IACO to optimize for the defects of traditional ACO. By improving the initial pheromone and proposing an adaptive pheromone intensity scheme, we prevent the algorithm from falling into local optimum. Besides, we also introduce pheromone reduction factor and improved evaporation factor to enhance the algorithm's global search ability and convergence speed. The new heuristic function strengthens the positive feedback effect of the ant. At the same time, based on the problem of vertex conflict in MAPF, we proposes a new strategy of conflict avoidance and resolution, which can effectively detect and resolve the conflict between agents to ensure the reliability of a multi-agent cluster system. Simulation experiments shows that our method improves the efficiency by at least 50% and that the number of iterations is reduces over 30%, which prove the effectiveness and feasibility of IACO. In the future work, we will further explore the application of heuristic algorithm in multi-agent path planning.

ACKNOWLEDGMENT

This work was supported by Shanghai Municipal Science and Technology Major Project 2021SHZDZX0103 and National Natural Science Foundation of China under Grant 82090052.

REFERENCES

[1] S. Bayili and F. Polat, "Limited-damage a*: A path search algorithm that considers damage as a feasibility criterion," Knowledge-Based Systems, vol. 24, no. 4, pp. 501–512, 2011.

[2] S. X. Wang, "The improved dijkstra's shortest path algorithm and its application," Procedia Engineering, vol. 29, no. 1, pp. 1186–1190, 2012.

[3] R. Asthana, "Evolutionary algorithms and neural networks," Soft Com-puting and Intelligent Systems, pp. 111–136, 2000.

[4] M. Dorigo, M. Birattari, and T. St ¨ utzle, "Ant colony optimization," IEEE Computational Intelligence Magazine, vol. 1, no. 4, pp. 28–39, 2006.

[5] C. Blum, "Ant colony optimization: Introduction and recent trends," Physics of Life Reviews, vol. 2, no. 4, pp. 353–373, 2005.

[6] A. Andreychuk, K. Yakovlev, D. Atzmon, and R. Stern, "Multi-agent pathfinding (mapf) with continuous time," Twenty-Eighth International Joint Conference on Artificial Intelligence IJCAI-19, 2019.

[7] R. Stern, N. Sturtevant, A. Felner, S. Koenig, and R. Bartak, "Multi-agent pathfinding: Definitions, variants, and benchmarks," 2019.

[8] Wurman, Peter, R., D'Andrea, Raffaello, Mountz, and Mick, "Coordi-nating hundreds of cooperative, autonomous vehicles in warehouses." AI Magazine, vol. 29, no. 1, pp. 9–19, 2008.

[9] R. Morris, C. S. Pasareanu, K. Luckow, W. Malik, and S. Koenig, "Planning, scheduling and monitoring for airport surface operations," in AAAI-16 Workshop on Planning for Hybrid Systems, 2016.

[10] "Cobots: Robust symbiotic autonomous mobile service robots," AAAI Press, 2015.

[11] M. Nazarahari, E. Khanmirza, and S. Doostie, "Multi-objective multi-robot path planning in continuous environment using an enhanced genetic algorithm," Expert Systems with Applications, vol. 115, no. JAN., pp. 106–120, 2018.

[12] Q. Cao, X. Huang, X. Zhu, and F. Zou, "Distributed multi-robot path planning based on reserved area," Huazhong Keji Daxue Xuebao (Ziran Kexue Ban)/Journal of Huazhong University of Science and Technology (Natural Science Edition), vol. 46, no. 12, pp. 71–76, 2018.

[13] "Reliability oriented multi-agvs online scheduling and path planning problem of automated sorting warehouse system," IOP Conference Series Materials Science and Engineering, vol. 1043, no. 2, p. 022035, 2021.

[14] Sturtevant, Nathan, R., Felner, Ariel, Sharon, Guni, Stern, and Roni, "Conflict-based search for optimal multi-agent pathfinding," Artificial Intelligence An International Journal, 2015.

[15] J. Li, A. Felner, E. Boyarski, H. Ma, and S. Koenig, "Improved heuristics for multi-agent path finding with conflict-based search," in Twenty-Eighth International Joint Conference on Artificial Intelligence IJCAI-19, 2019.

[16] R. Bartak, N. F. Zhou, R. Stern, E. Boyarski, and P. Surynek, "Modeling and solving the multi-agent pathfinding problem in picat," in 2017 IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI), 2017.

[17] D. Corne, M. Dorigo, and F. Glover, "New ideas in optimization, chapter 2: The ant colony optimization meta-heuristic," 1999.

[18] V. Maniezzo, L. M. Gambardella, and F. D. Luigi, "Ant colony opti-mization," Springer Berlin Heidelberg, 2004.

[19] "A survey on parallel ant colony optimization," Applied Soft Computing, vol. 11, no. 8, pp. 5181-5197, 2011.
[20] M. M. Khorshid, R. C. Holte, and N. R. Sturtevant, "A polynomial-time algorithm for non-optimal multi-agent pathfinding," in Proceedings of the Fourth Annual Symposium on Combinatorial Search, SOCS 2011, Castell de Cardona, Barcelona, Spain, July 15.16, 2011, 2013.