IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 6, 2020, accepted November 15, 2020, date of publication November 24, 2020,
date of current version December 9, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3039959

Research on Multi-AGVs Path Planning and Coordination Mechanism

YIYANG LIU[1,2,5,6], ZHENG HOU[2,3,5], YUANYUAN TAN[3], HAOQUN LIU[4],
AND CHUNHE SONG[1,2,5]

[1]State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China
[2]Key Laboratory of Networked Control Systems, Chinese Academy of Sciences, Shenyang 110016, China
[3]Institute of Artificial Intelligence, Shenyang University of Technology, Shenyang 110870, China
[4]In-Store Business Group, Beijing Sankuai Online Technology Company Ltd., Beijing 110190, China
[5]Institutes for Robotics and Intelligent Manufacturing, Chinese Academy of Sciences, Shenyang 1100169, China
[6]Kunshan Intelligent Equipment Research Institute, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 215347, China

Corresponding author: Chunhe Song (songchunhe@sia.cn)

This work was supported in part by the National Nature Science Foundation of China under Grant U1908212, and Grant 6101020101, in part by the Key Project of Natural Science Foundation of China under Grant 61533015, in part by the National Key R&D Program of China under Grant 2018YFB2003203, in part by the Revitalizing Liaoning Outstanding Talents Project under Grant XLYC1907057, in part by the Liaoning Province Education Department Scientific Research Foundation of China under Grant LQGD2019014, and in part by the Liaoning Provincial Natural Science Foundation of China under Grant 2019-ZD-0218.

ABSTRACT Reasonable automatic guided vehicle path planning can shorten the transportation time of materials and improve the production efficiency of the intelligent assembly workshop. Ant colony algorithm is a widely used path planning method, however, it suffers from the shortcomings that being easy to fall into local optimum and low search efficiency. To overcome these shortcomings, first, this paper proposes a step optimization method to improve the search efficiency of the ant colony algorithm, and a path simplification method to avoid getting blindly tortuous paths; Second, to overcome the problem that the ant colony algorithm is easy to fall into the local optimum, this paper proposes an adaptive pheromone volatilization coefficient strategy, which uses different pheromone volatilization coefficients at different stages of the search path; third, for the path conflict problem of multiple automatic guided vehicles, this paper proposes a load balancing strategy to avoid it, which is based on the consideration that, path conflicts are caused by excessive concentration of multiple automatic guided vehicles paths. Extensive simulation results demonstrate the feasibility and efficiency of the proposed methods.

INDEX TERMS Ant colony algorithm, intelligent assembly, multiple automatic guided vehicles, path planning.

I. INTRODUCTION

With the steady progress of the Made in China 2025 and the Industry 4.0 plans, advanced technologies such as digital factories, Industrial Internet of Things, and artificial intelligence have developed rapidly. Automatic guided vehicle (AGV), as the main means of transportation, shows a high level of intelligence and automation in workshop logistics, which greatly improves the flexibility of transporting materials and improves the work efficiency of the intelligent assembly workshop. The AGV has the advantages of stable transportation and accurate dispatch [1]. The tasks in the intelligent assembly workshop include transporting raw materials to processing machine tools, transporting semi-finished products and finished products to warehouses. Planning the optimal path from the starting position to the target position for the AGV that accepts the task is the key to improving the work efficiency and economic benefits of the intelligent assembly workshop [2], [3].

Huo *et al.* [4] conduct research on AGV transporting containers at the terminal, who use a linear mixed integer model to plan the path for the AGV transport container, and take the overall task completion time as the system evaluation index to realize the safe and stable operation of the AGV. Kim *et al.* [5] construct the obstacle force field based on the predicted trajectory of the moving obstacle, and propose an improved one-dimensional virtual force field (1D-VFF) method. The conventional 1D-VFF obstacle force field is replaced by the predicted obstacle force field, which enhance the obstacle avoidance performance of the AGV. In the actual

The associate editor coordinating the review of this manuscript and approving it for publication was Nagarajan Raghavan.

working environment, multiple AGVs are required to perform tasks together to make the workshop more efficient. Zhao *et al.* [6] compare the paths of different AGVs to determine conflict nodes and AGV priorities, so that multiple AGVs can operate in conflict-free coordination on the same map. In addition, during the actual operation, there will be motion delay problems caused by network instability, pedestrian interference, and so on. When the AGV that is performing a task has path conflicts with other AGVs, the time spent on the path becomes uncertain. In response to this problem, Li *et al.* [7] propose a dynamic adjustment based on the AGV conflict probability to reduce the catch-up conflicts.

There are many algorithms for solving the AGV path planning problem, include Dijkstra algorithm [8], A$*$ algorithm [9], artificial potential field method [10], ant colony optimization (ACO) algorithm [11] and so on. The above several algorithms have their own advantages and disadvantages in planning the path.

The ACO algorithm is a heuristic optimization method based on positive feedback mechanism, in which the pheromone concentration on the path has a heuristic effect on searching for the next node [12]. In the initial stage when the ACO algorithm is used to search the path from the starting node to the target node, the pheromone is evenly distributed on the map, the search process is blind and many nodes are traversed, so the search efficiency is low [13]. With the increase of the number of iterations, the probability of subsequent ants choosing the optimal path increases as the concentration of pheromone on the optimal path increases, thereby the ACO algorithm obtain the optimal path through positive feedback based on pheromone. This paper uses the minimum number of iterations after convergence to indicate the convergence speed of the algorithm. The smaller the minimum number of iterations after convergence, the faster the convergence speed of the algorithm [14].

Based on the research of the above algorithms, this paper uses the grid method to build the model, and use the ACO algorithm to search for the shortest path. However, the ACO algorithm has shortcomings such as low search efficiency and easy to fall into the local optimum, so this paper improves the original ACO algorithm.

The contribution of this paper can be summarized as follows:

First, a method is proposed to improve the search efficiency by the ACO algorithm with adaptive search step length to search the path. But the planned path is a blindly tortuous path, so this paper proposes a method of simplifying the path to avoid planning a blindly tortuous path.

Second, the ACO algorithm of adaptive pheromone volatilization coefficient whose pheromone volatilization coefficient decreases as the number of iterations increases is used to plan the shortest path to avoid falling into the local optimum.

Third, when multiple AGVs perform tasks together, in view of the inefficiency of the multi-AGV scheduling system caused by path conflicts of multiple AGVs, this paper proposes a method based on load balancing to avoid path conflicts of multiple AGVs.

The rest of the paper is organized as follows: Section 2 presents related research on path planning. The section 3 introduces the ACO algorithm and the mathematical model. In section 4, this paper improves the search step length of the original ACO algorithm and the simulation of improved algorithm. Followed by section 5 improves the pheromone volatilization coefficient of the original ACO algorithm and the simulation of improved algorithm. Section 6 presents a multi-AGVs path planning method based on load balancing. Section 7 draws the conclusion and its future research direction.

II. RELATED WORKS

In recent years, tremendous research efforts have been made for improving the path planning efficiency. Xiaolin *et al.* [15] use the evaluation function of the A* algorithm to improve the heuristic information of the ACO algorithm and accelerate the convergence speed of the search process. Gao *et al.* [16] redefine the heuristic distance, which not only considers the heuristic distance from the neighbor node of the ant to the target node, but also considers the distance from the ant's current node to the target node. This strategy improves the probability of ants searching for the target node and improves the efficiency of path planning. Du *et al.* [17] use hybrid genetic particle swarm optimization algorithm (GA-PSO) to plan the path, and propose a particle iteration mechanism based on time priority, which makes the algorithm directional search path and accelerates the convergence speed of the algorithm. Chen *et al.* [18] employ an improved pheromone updating strategy which combines the global asynchronous feature and "Elitist Strategy", this method emphasizes the influence of the best ant (the individual with the current shortest solution) by "Elitist Strategy". Therefore, the iteration number of ACO algorithm invokes by chaos-based particle swarm optimization can be reduced reasonably so as to decrease the search time effectively.

In the original ACO algorithm, the ants in the search process may fall into the local optimum, there are numerous researches on the ACO algorithm and other algorithms about how to jump out of the local optimum. Hub *et al.* [13] propose the maximum and minimum ant strategy, the pheromone concentration on the map is limited between the maximum pheromone concentration and the minimum pheromone concentration. Wang *et al.* [19] propose a new dual-operator and dual-population ant colony optimization (DODPACO) algorithm, the load operators are adopted to limit the accumulation of the pheromone on the path in the early search path, and adjust the pheromone concentration on the path to avoid falling into the local optimum. Gao *et al.* [16] propose a path merging strategy, which connects different paths from different ants to jump out of the local optimum and obtain the global optimal path. Wang *et al.* [20] introduce a reward and punishment mechanism when updating pheromone, and the improved ACO algorithm is more efficient in searching paths.

For the multi-AGVs path planning, the coupled approach regards the group of robots as a single entity, such that all paths are planned simultaneously in a joint or composite configuration space, and therefore could guarantee completeness, but these solutions do not scale well with large robot teams and they usually cannot be solved in real-time [21]. Das and Jena [22] employ an improved version of particle swarm optimization (IPSO) with evolutionary operators (EOPs) to calculate an optimal collision-free trajectory path for each robot in a known and complex environment. During the path generation process, each robot calculates its next local optimal coordinates in a stepwise manner to avoid path conflicts of multiple AGVs. Fransen *et al.* [23] conduct research on large-scale multi-AGVs systems, and update the weights of road segment length according to the waiting time of the AGV when the paths of multiple AGVs conflict. Zhong *et al.* [24] embed the Dijkstra algorithm into the genetic algorithm to search for the shortest path, and uses the time window to detect the path conflicts of multiple vehicles, and finally obtain the shortest conflict-free paths of multiple AGVs. The decoupled approach first computes separate paths for the individual robots and then employs different strategies to resolve possible conflicts. These solutions are usually fast enough for real-time applications, but they cannot guarantee completeness and the robots might easily get stuck in common deadlock situations. Zhao *et al.* [6] use the A* algorithm to plan the path of each AGV in the workshop, by comparing the paths of different AGVs, the waiting method is adopted to solve the catch-up conflicts and the method of re-planning the paths is used to resolve the opposite conflicts. Guo and Zhu [25] use the Dijkstra algorithm to plan paths for multiple AGVs in sequence, and control the speed before the AGV reaches the conflicting node to reduce the negotiation time and resolve the conflict.

III. THE ESTABLISHMENT OF MATHEMATICAL MODEL OF ACO ALGORITHM

Ants leave pheromone on the path they walked during their foraging. Pheromone volatilizes over time. The concentration of pheromone remaining on the path determines the path the ant will take [26]. The more ants that travel on a certain path, the greater the pheromone concentration on the path [18], [27]. The following ants have a high probability to choose the path with high pheromone concentration, and the optimal path becomes clear [28].

The pheromone concentration between adjacent nodes and the expected heuristic information between adjacent nodes calculated by the heuristic function are used to search path, and the transition probability function $p_{ij}^k(t)$ can be described as follows [29]:

$$p_{ij}^k(t) = \left\{ \begin{array}{c} \dfrac{[\tau_{ij}(t)]^{\propto}\eta_{ij}(t)^{\beta}}{\sum\limits_{j\in allowed_k}[\tau_{ij}(t)]^{\alpha}[\eta_{ij}(t)]^{\beta}}, j \in allowed_k \\ 0, other\ situaions \end{array} \right\} \tag{1}$$

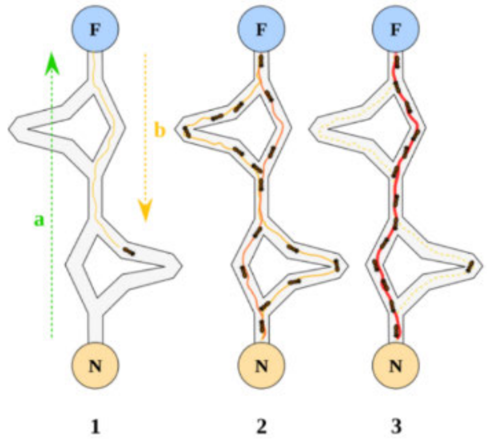


FIGURE 1. **The process in which ants leave pheromone on the path they have traveled, and finally find the optimal path.**

where $\tau_{ij}(t)$ represents the pheromone concentration released by ants on the road segment between node i and node j on the map, $\eta_{ij}(t)$ represents the heuristic information of the path segment between node i and node j on the map, and α represents the inspiration factor of pheromone, β represents the expected heuristic factor, which reflects the strength of the priori and deterministic factors in the process of finding the optimal path, and k represents the set of nodes that are allowed to be visited by ants [30]. The heuristic function $\eta_{ij}(t)$ can be calculated as:

$$\eta_{ij} = \frac{1}{d_{ij}} \tag{2}$$

where d_{ij} represents the distance from node i to node j on the map.

When the ant reaches the target node from the starting node, the pheromone concentration on the path is updated [31], the pheromone concentration $\tau_{ij}(t+1)$ on the road segment between node i and node j after the ant reach the target node can be described by the following equations:

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

$$\Delta\tau_{ij}(t) = \sum\nolimits_{k=1}^{m} \Delta\tau_{ij}^k(t) \tag{4}$$

$$\Delta\tau_{ij}^k(t) = \begin{cases} \dfrac{Q}{L_k}, & \text{The ant } k \text{ passes the node } (i,j) \\ 0, & Others \end{cases} \tag{5}$$

where $\tau_{ij}(t)$ represents the pheromone concentration on the road segment between node i and node j before the algorithm updates the pheromone concentration. ρ represents the pheromone volatilization coefficient, and the value range is $\rho \in (0,1)$, $\Delta\tau_{ij}(t)$ represents the sum of the increased pheromone concentration of all ants passing through the road segment between node i and node j before the algorithm updates the pheromone concentration, and $\Delta\tau_{ij}^k(t)$ represents that before the algorithm update the pheromone concentration, the increased pheromone concentration on the road segment between node i and node j by ant k. Q represents the sum of pheromone released on the map when the ant reaches the target node, m represents the number of ants, and L_k presents the length of the path that the ant travels after reaching the target node.

IV. THE PROPOSED ACO ALGORITHM WITH ADAPTIVE SEARCH STEP LENGTH AND SIMULATION ANALYSIS

A. THE WORKFLOW OF ACO ALGORITHM WITH ADAPTIVE SEARCH STEP LENGTH

When the original ACO algorithm is used to search the optimal path, the search efficiency is low and time -consuming, this paper proposes an ACO algorithm with adaptive search step length. In the ACO algorithm with adaptive search step length, this paper chooses whether to use the 2 times search step length to search path, which improves the search efficiency of the algorithm.

Algorithm 1 Strategy to Adaptive Search Step Length

```
1   Input: The path being planned
2   Output: The path being planned after adopting adaptive
      search step length method
3   length← size (path)
4   $q_0(x_0, y_0) \leftarrow$ path (length-1)
5   $q_1(x_1, y_1) \leftarrow$ path (length)
6   $\Delta x \leftarrow x_1 - x_0$
7   $\Delta y \leftarrow y_1 - y_0$
8   $x_2 \leftarrow x_1 + \Delta x$
9   $y_2 \leftarrow y_1 + \Delta y$
10    if $(x_2 > 0 \&\& x_2 \le MM) \&\& (y_2 > 0$
      $\&\& y_2 \le MM)$ then
11      if $G\,(x_2, y_2) == 0$
12        if $Tau\,(q_1, q_2) \ge Tau\,(q_0, q_1)$
13          path← [path, $q_2$]
14        end if
15      end if
16    end if
    end Algorithm 1
```

As shown in Algorithm 1, according to the last two nodes of the path, through the linear principle, the node $q_2(x_2, y_2)$ is obtained when the path is searched with twice the search step length (**line 3-9**). If the location of node q_2 is within the map range, the algorithm executes the next step, where the size of the map is $MM \times MM$ (**line 10**). If the position of node q_2 on the map is not an obstacle, the algorithm executes the next step. if the pheromone concentration between node q_1 and node q_2 is not less than the pheromone concentration between q_0 and q_1, then node q_2 is added to the path (**line 11-12**).

However, the path obtained by the ACO algorithm with adaptive search step length is a blindly tortuous path. Therefore, in each iteration, after each ant reaches the target node, the method of simplifying path can avoid getting blindly tortuous paths, at the same time this paper uses this method to reduce the length of the path searched by each ant in each iteration. As shown in Fig. 3, there are three kinds of curved paths in the path of the AGV that need to be simplified, and different methods are used to simplify different turning paths.

When the pheromone concentration on the path is updated, this paper can know from Eq. (3), Eq. (4), and Eq. (5) that the smaller the path length, the more the pheromone concentration on the path increases, which further accelerates the convergence of the algorithm. Therefore, the strategy to adaptive search step length and the method to simplify the path of each ant in each iteration can improve the search efficiency of the original ACO algorithm [32].

Algorithm 2 Method to Simplify the Path of Each Ant in Each Iteration

```
1   Input: The path of each ant in each iteration
2   Output: The path of each ant in each iteration after
    simplifying path
3   $count \leftarrow$ size (path)
4   for $i \leftarrow 1tocc$-2
5     $q_{11}(x_{11}, y_{11}) \leftarrow$ path $(i)$
6     $q_{22}(x_{22}, y_{22}) \leftarrow$ path $(i+1)$
7     $q_{33}(x_{33}, y_{33}) \leftarrow$ path $(i+2)$
8     Calculate the slope based on the coordinates, and
      then calculate the angle
9     if the angle is 90 degrees by Fig.2(a) then
10      path $(i+1)$←0
11     end if
12     if the angle is 45 degrees by Fig.2(b) then
13      path $(i+1)$←0;
14     end if
15     if the angle is 90 degrees by Fig.2(c) then
16      path$(i+1)$←sub2ind(size(G),
          $(x_{11}+x_{33})/2, (y_{11}+y_{33})/2)$
17      end if
18      Delete the nodes numbered 0 in the path
19    end for
  end Algorithm 2
```

As shown in Algorithm 2, in the path obtained by each ant in each iteration, three consecutive nodes are extracted in sequence, and the slope of the straight line is calculated by the coordinates of the nodes, and the turning angle is further calculated (**line 3-8**). According to different turning angles, different path simplification strategies are adopted as shown in Fig. 3(a), Fig. 3(b), and Fig. 3(c) (**line 9-17**). The size of the simulation map is $MM \times MM$, for example, when $MM = 20$, there are 400 nodes on the map, and each node is numbered, and each number corresponds to the coordinates of a node. Sub2ind is a function of MATLAB, which is used to convert the subscript of an element in the array to the corresponding index value of the element in the array. G is a matrix, used to represent a map. size (G) represents an array storing the number of rows and columns of the matrix. Sub2ind(size(G), $(x_{11}+x_{33})/2, (y_{11}+y_{33})/2)$ refers to the number of the node with coordinate $((x_{11}+x_{33})/2, (y_{11}+y_{33})/2)$ on the map (**line 16**). After all path nodes are traversed, inappropriate path nodes in the path are deleted, that is, the nodes

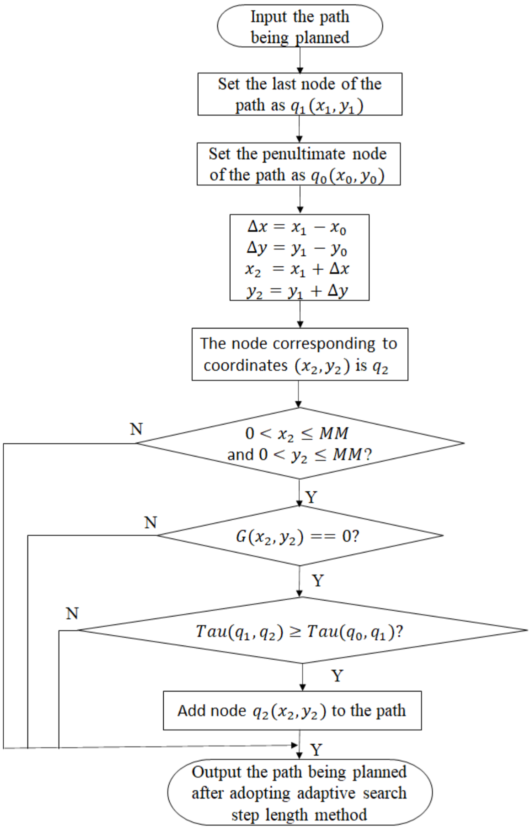

Input the path being planned
Set the last node of the path as $q_1(x_1, y_1)$
Set the penultimate node of the path as $q_0(x_0, y_0)$
$\Delta x = x_1 - x_0$
$\Delta y = y_1 - y_0$
$x_2 = x_1 + \Delta x$
$y_2 = y_1 + \Delta y$
The node corresponding to coordinates (x_2, y_2) is q_2
N
$0 < x_2 \leq MM$ and $0 < y_2 \leq MM$?
Y
N
$G(x_2, y_2) == 0$?
Y
N
$Tau(q_1, q_2) \geq Tau(q_0, q_1)$?
Y
Add node $q_2(x_2, y_2)$ to the path
Y
Output the path being planned after adopting adaptive search step length method

FIGURE 2. Flow chart of the strategy to adaptive search step length.

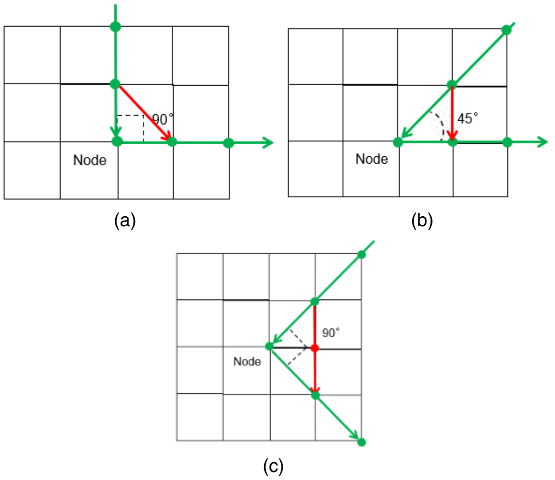


FIGURE 3. The path of the AGV when turning. (a) Simplified the path when the turning angle is 90°. (b) Simplified the path when the turning angle is 45°. (d) Simplified the path when the turning angle is 90°.

numbered 0 in the path are deleted, so as to obtain the simplified path (**line 9-18**).

The specific implementation steps are as follows:

The first step is to set the initial values of the parameters.

The second step is to select the next node q_1 visited by the ant according to the transition probability function $p_{ij}^{k}(t)$.

The third step is to record the node q_1, and update the path length.

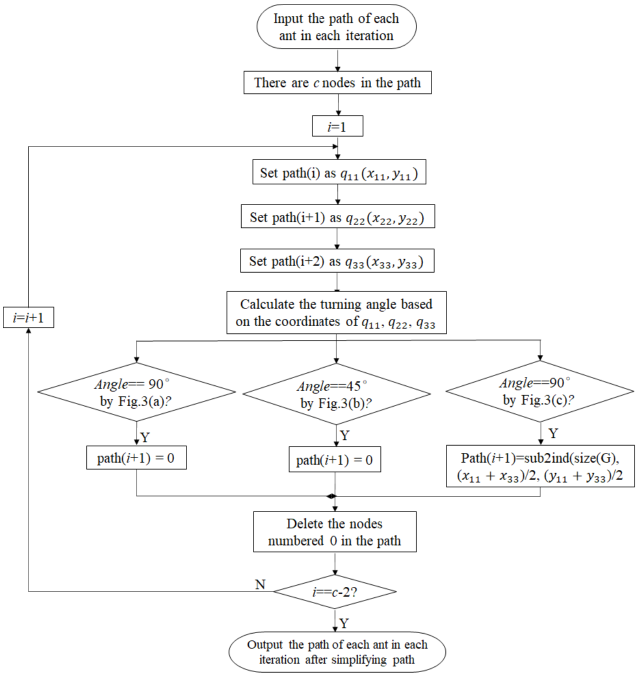

Input the path of each ant in each iteration
There are *c* nodes in the path
i=1
Set path(i) as $q_{11}(x_{11}, y_{11})$
Set path(i+1) as $q_{22}(x_{22}, y_{22})$
Set path(i+2) as $q_{33}(x_{33}, y_{33})$
Calculate the turning angle based on the coordinates of q_{11}, q_{22}, q_{33}
i=*i*+1
Angle== 90° by Fig.3(a)?
Angle==45° by Fig.3(b)?
Angle==90° by Fig.3(c)?
Y
Y
Y
path(*i*+1) = 0
path(*i*+1) = 0
Path(*i*+1)=sub2ind(size(G), $(x_{11}+x_{33})/2$, $(y_{11}+y_{33})/2$
Delete the nodes numbered 0 in the path
N
i==*c*-2?
Y
Output the path of each ant in each iteration after simplifying path

FIGURE 4. Flow chart of the method to simplify the path of each ant in each iteration.

The fourth step is to update the path being planned with a strategy to adaptive search step length and update the path length.

The fifth step is to repeatedly update the path with a method to simplify the path of each ant in each iteration. If the path length does not change, the next step is executed.

The sixth step is to repeat the second to the fifth step until all the ants have planned the path.

The seventh step is to update the pheromone on the path after all the ants have reached the target node.

The eighth step is to repeat the second to seventh step until the number of algorithm iterations is met.

B. SIMULATION OF ACO ALGORITHM WITH ADAPTIVE SEARCH STEP LENGTH

This paper uses the grid method to establish the AGV working environment map, and set the task starting node (3,18) and the target node (17,3). The size of the map is 20 × 20, the obstacles are randomly distributed with a probability of 30%, there are no obstacles at the location of the starting node and the location of the target node. The number of iterations of the algorithm is 140, There are 50 ants in each iteration. After 140 iterations of the algorithm, the algorithm stops searching the path and outputs the simulation results. Pheromone volatilization coefficient is 0.4, the inspiration factor of pheromone is 1, and the expected heuristic factor is 7, and use the original ACO algorithm, the ACO algorithm with simplified path, and the ACO algorithm with adaptive search step length and simplified path to plan the AGV path as shown in Fig.5.

In 15 simulations, the locations of obstacles on each simulation map is different from the locations of obstacles on other simulation maps. The original ACO algorithm, the ACO algorithm with simplified path, and the ACO algorithm with adaptive search step length and simplified path are used to conduct simulations and analyze the simulation results. As shown in Table 1, in the simulation, this paper takes the minimum number of iterations after the convergence of the algorithm as the evaluation index to verify that the ACO algorithm with adaptive search step length and simplified path can effectively improve the search efficiency.

TABLE 1. Simulation of the ACO algorithm with adaptive search step length.

No.	CACO	CACOS	CACOIS	RPC1	RPNCT2
1	17	12	9	47.06%	25.00%
2	23	12	9	60.87%	25.00%
3	21	12	13	38.09%	-8.33%
4	22	20	13	40.91%	35.00%
5	21	12	10	52.38%	16.67%
6	22	17	10	54.55%	41.18%
7	20	12	11	45.00%	8.33%
8	17	14	10	41.18%	28.57%
9	24	12	11	54.17%	8.33%
10	14	13	10	28.57%	23.08%
11	15	9	11	26.67%	-22.22%
12	16	11	10	37.50%	9.09%
13	14	11	9	35.71%	18.18%
14	16	11	15	6.25%	-36.36%
15	29	13	12	58.62%	7.69%
Average value	19.4	12.73	10.87	43.97%	14.61%

Notes: CACO: the minimum number of iterations after the convergence of the original ACO algorithm; CACOS: the minimum number of iterations after the convergence of the ACO algorithm with simplified path; CACOIS: the minimum number of iterations after the convergence of the ACO with adaptive search step length and simplified path; RPC1: compared with the original ACO algorithm, the reduced percentage of the minimum number of iterations after convergence; RPNCT2: compared with the ACO algorithm with simplified path, the reduced percentage of the minimum number of iterations after convergence.

As shown in Table 1, compared with the original ACO algorithm, the ACO algorithm with adaptive search step length and simplified path reduces the minimum number of iterations after convergence by 15 times, and the average minimum number of iterations after convergence is reduced by 43.97%. Compared with the ACO algorithm with simplified path, the ACO algorithm with adaptive search step length and simplified path reduces the minimum number of iterations after convergence by 12 times, and the average minimum number of iterations after convergences is reduced by 14.61%. There are 3 times that the minimum number of iterations after the convergence of the ACO algorithm with adaptive search step length and simplified nodes is larger than that of the ACO algorithm with simplified path.

In the initial stage of the search path using the ACO algorithm with adaptive search step length and simplified path, the pheromone distribution is relatively balanced. In most cases, the condition that 2 times the search step length is used to search the path is satisfied. The path planned by the ACO algorithm with adaptive search step length and simplified path is a blindly tortuous path, and the minimum number of iterations after the convergence of the ACO algorithm with adaptive search step length and simplified path is larger than that of the ACO algorithm with simplified path, which occurred 3 times out of 15 trials. In general, the ACO algorithm with adaptive search step length and simplified path improve the search efficiency compared with the original ACO algorithm, and the ACO algorithm with adaptive search step length and simplified path also improve the search efficiency compared with the ACO algorithm with simplified path.

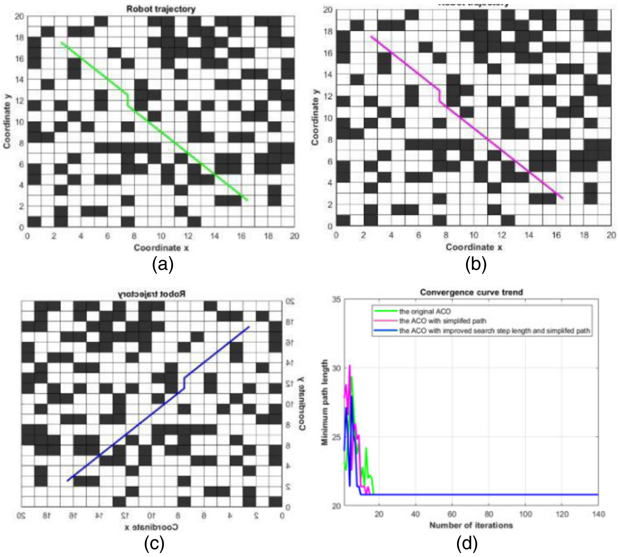


FIGURE 5. Simulation results of ACO algorithm with adaptive search step length and simplified path. (a) The path of the original ACO algorithm. (b) The path of the original ACO algorithm with simplified path. (c) The path of the original ACO algorithm with adaptive search step length and simplified path. (d) The convergence curve of the ACO algorithm with adaptive search step length and simplified path.

Because obstacles are randomly distributed on the simulation map with a probability of 30%, and in 15 simulations, the map used in each simulation is different, the conclusion of the simulations is universal that the ACO algorithm with adaptive search step length and simplified path can effectively improve the search efficiency.

V. THE PROPOSED ACO ALGORITHM WITH ADAPTIVE PHEROMONE VOLATILIZATION COEFFICIENT AND SIMULATION ANALYSIS

A. THE WORKFLOW OF THE ACO ALGORITHM WITH ADAPTIVE PHEROMONE VOLATILIAZATION COEFFICIENT

In the process of the ACO algorithm with improved pheromone volatilization coefficient being used to search for the shortest path, if the pheromone volatilization coefficient takes a larger value, the pheromone has less guiding effect on the ants, which increases the randomness when the ACO algorithm is used to search for the path and prevents the ACO algorithm from falling into the local optimum [33]; if the pheromone volatilization coefficient takes a smaller value, the guiding effect of the pheromone on the ants increases, which is not conducive to the global divergence search path, and the algorithm is easy to fall into local optimum. The appropriate pheromone volatilization coefficient is very important to search the shortest path by the ACO algorithm [34]. Therefore, this paper proposes an ACO algorithm with adaptive pheromone volatilization coefficient to search for the shortest path. The pheromone volatilization coefficient can be calculated as:

$$\rho = (\rho_{max} - \rho_{\min}) \left[\frac{2K_t}{K} - \left(\frac{K_t}{K} \right)^2 \right] \tag{6}$$

where K represents the total number of search iterations, K_t represents that the number of iterations is t, ρ_{max} represents the largest pheromone volatilization coefficient, $\rho_{max} = 0.9$, ρ_{min} represents the smallest pheromone volatilization coefficient, $\rho_{min} = 0.1$.

The variation trend of pheromone volatilization coefficient is shown in Fig. 6. The pheromone volatilization coefficient gradually decreases as the number of search iterations increases.

B. SIMULATION OF THE ACO ALGORITHM WITH ADAPTIVE PHEROMONE VOLATILIZATION COEFFICIENT

For the simulation of the ACO algorithm with adaptive pheromone volatilization coefficient, the simulation parameters are the same as those in Part B of section 4, but the pheromone volatilization coefficient of the ACO algorithm with adaptive the pheromone volatilization coefficient is obtained by Eq. (6), the pheromone volatilization coefficient gradually decreases with increasing iterations. The pheromone volatilization coefficient of the original ACO algorithm is 0.4. This paper designs the simulation of the ACO algorithm with adaptive pheromone volatilization coefficient, which uses the original ACO algorithm and the ACO algorithm with adaptive pheromone volatilization coefficient to plan the paths separately, and analyze the simulation results.

As shown in Table 2, there are 15 sets of simulation results. Each set of data corresponds to a map. On the same map, the original ACO algorithm and the improved ACO algorithm with adaptive pheromone volatilization coefficient are simulated.

According to the data in Table 2, compared with the original ACO algorithm, the ACO algorithm with adaptive pheromone volatilization coefficient reduces the minimum number of iterations after the convergence of the original ACO algorithm 15 times out of 15 simulations, and reduces the minimum number of iterations after convergence by 59.25% on average.

After 15 simulations, the simulations of the improved ACO algorithm with adaptive pheromone volatilization coefficient conclude that compared with the original ACO algorithm, the ACO algorithm with adaptive pheromone volatilization coefficient improves the search efficiency.

TABLE 2. Simulation of the improved ACO algorithm with adaptive pheromone volatiliazation coefficient.

No.	MNCTACO	MNCTACOIPVC	RPMNCT
1	18	7	61.11%
2	19	7	63.16%
3	18	7	61.11%
4	24	9	62.50%
5	16	12	25.00%
6	14	5	64.29%
7	18	6	66.67%
8	23	8	65.22%
9	15	8	46.67%
10	17	8	52.94%
11	20	6	70.00%
12	14	5	64.295
13	17	7	58.82%
14	20	7	65.00%
15	22	10	54.55%
Average value	18.33	7.47	59.25%

Notes: MNCTACO: the minimum number of iterations after the convergence of the original ACO; MNCTACOIPVC: the minimum number of iterations after the convergence of the ACO with adaptive pheromone volatilization coefficient; RPMNCT: compared with the original ACO, the reduced percentage of the minimum number of iterations after convergence.

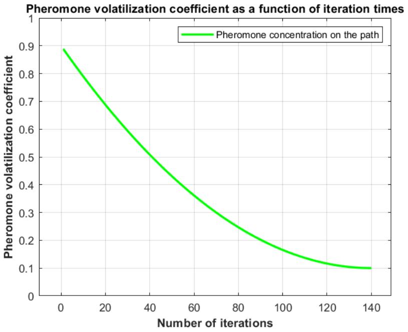
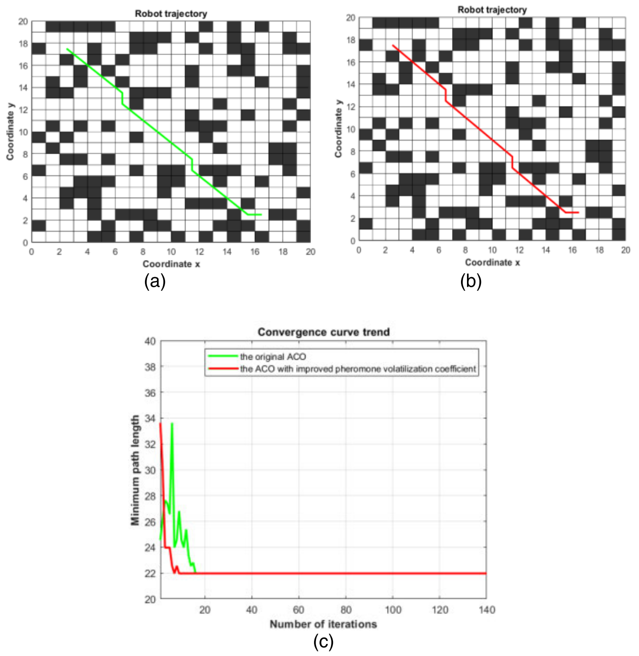


FIGURE 6. Pheromone volatilization coefficient decreases as the number of search iterations increases.

Because obstacles are randomly distributed on the simulation map with a probability of 30%, and 15 different simulation maps are used in 15 simulations, the conclusion of the simulation is universal that the ACO algorithm with adaptive pheromone volatilization coefficient can effectively improve the search efficiency.

VI. THE PROPOSED MULTI-AGVS PATH PLANNING METHOD BASED ON LOAD BALANCING

The section IV and the section V of this paper use the improved ACO algorithm to plan a single AGV path, the improved ACO algorithm improves the search efficiency of the original ACO algorithm. However, a single AGV perform tasks is far from meeting the requirements of the workshop to quickly improve work efficiency, so this paper needs multiple AGVs to perform tasks together. When multiple AGVs perform tasks together, the paths of multiple AGVs are more complicated and conflicts will inevitably occur, which will seriously affect the work efficiency of the workshop. Planning paths for multiple AGVs not only requires the shortest total path length, but also avoids path conflicts of multiple AGVs, which increases the difficulty of planning paths



(a)

(b)

(c)

FIGURE 7. Simulation results of ACO algorithm with adaptive pheromone volatilization coefficient. (a) The path of the original ACO algorithm. (b) The path of the ACO algorithm with adaptive pheromone volatilization coefficient. (c) The convergence curve of the ACO algorithm with adaptive pheromone volatilization coefficient.

for multiple AGVs, so this paper proposes a multi-AGVs coordination method based on load balancing to avoid path conflicts of multiple AGVs. This paper designs a simulation based on load balancing strategy, which uses an improved ACO algorithm to plan the paths for multiple AGVs in the section VI.

A. MULTI-AGVS COORDINATION METHOD BASED ON LOAD BALANCING

According to the established data association graph, this paper can obtain the information of the multi-AGVs scheduling system, so as to formulate the specific steps of the load balancing. Data association diagram as shown in Fig. 8.

This paper proposes a method of load balancing to avoid path conflicts of multiple AGVs. The specific implementation steps can be described as follows:

Step 1: Establish an electronic map of the multi-AGVs scheduling system;

Step 2: The idle AGV obtains tasks from the task list according to the task priority, and clarifies the starting node and target node of the task of the AGV;

Step 3: On the map, this paper sets up the road segment record points to record the number of AGV passes of each segment and the map information near the road segment record points;

Step 4: This paper establishes the penalty coefficient function and penalty term function of the road segment to update the length information of the road segment. The specific expression is:

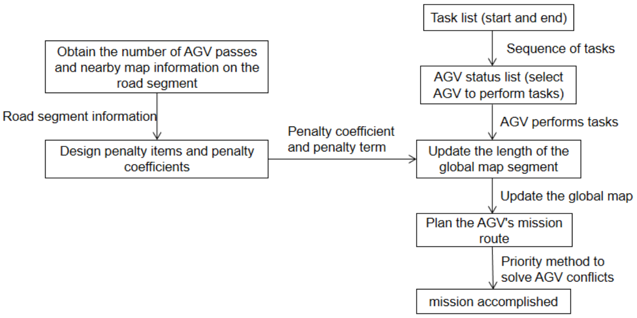

Task list (start and end)
Sequence of tasks
Obtain the number of AGV passes and nearby map information on the road segment
AGV status list (select AGV to perform tasks)
AGV performs tasks
Road segment information
Penalty coefficient and penalty term
Design penalty items and penalty coefficients
Update the length of the global map segment
Update the global map
Plan the AGV's mission route
Priority method to solve AGV conflicts
mission accomplished

FIGURE 8. Data association diagram of load balancing method.

(1) When there is an AGV driving through a certain road segment within the preset time range, the multi-AGVs scheduling system obtains the total number of road segments and the number of impassable road segments within the preset range through the road segment record points, where the road segment within the preset range refers to the passable road segments connected to the road segment i. The penalty coefficient function is described as Eq. (7):

$$\alpha_i = e^{\frac{m_i}{M_i}} \tag{7}$$

where α_i represents the penalty coefficient of the road segment i, M_i represents the total number of road segments within the preset range of the road segment i, and m_i represents the number of impassable road segments within the preset range of the road segment i.

(2) When a certain road segment is driven by an AGV within the preset time range, the multi-AGVs scheduling system records the number of AGV passes in a certain period of time through the road segment i, and the penalty term function of the road segment i is calculated in Eq. (8):

$$P_i = e^{\frac{n_i}{N}} \tag{8}$$

where P_i represents the penalty term function of the road segment i, n_i represents the number of times the road segment i passed by the AGV, and N represents the total number of all road segments passed by the AGV.

(3) The length of the road segment i is calculated in Eq. (9):

$$L_i = \alpha_i P_i \tag{9}$$

where L_i represents the length of the road segment i after updating the global road segment length, α_i represents the penalty coefficient of the road segment i, and P_i represents the penalty term of the road segment i.

Step 5: After updating the global road segment length, the improved ACO algorithm is used to plan the path of the AGV. If there is still a path conflict, on the conflict road segment or conflict node, the AGV with the higher priority will pass first, and the AGV with the lower priority will wait or re-plan the path.

B. VERIFICATION CASE

In the actual intelligent assembly workshop, multiple AGVs are required to perform multiple tasks. This paper uses the

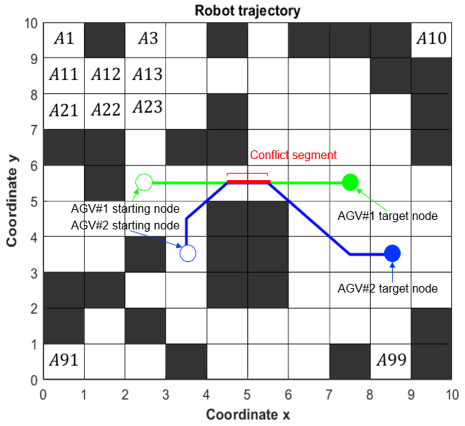


FIGURE 9. The conflict paths of two AGVs.

TABLE 3. AGV path and its corresponding time.

AGV number	AGV path
AGV#1	Path A43-A44-A45-A46-A47-A48
	Corresponding time 0T-1T-2T-3T-4T-5T
AGV#2	Path A64-A54-A45-A46-A57-A68-A69
	Corresponding time 0T-1T-2.4T-3.4T-4.8T-6.2T-7.2T

grid method to establish a map model, the simulation parameters of the simulation are the same as those in Part B of section 4 and Part B of section 5, but the size of the map is 10 × 10, and there are 100 grids in total. The side length of each grid is 1meter. The time of the AGV to travel 1meter is 1T, where T is the time unit, and the AGV ignores the time it takes to turn when performing tasks. As shown in Fig. 9, the starting node of AGV#1 is A43 and the target node of AGV#1 is A48, the starting node of AGV#2 is A64 and the target node of AGV#2 is A69. The preset time range in this paper is 2T.

This paper synthesizes the improvement of ACO algorithm in section IV and section V, and uses improved ACO algorithm to plan the paths for multiple AGVs in sequence. Multiple AGVs may pass the same node or the same road segment at the same time, so there are path conflicts of multiple AGVs. This paper plans the paths for AGV#1 and AGV#2 in sequence and calculates the time for two AGVs to pass on each road segment, as shown in Table 3. As shown in Fig. 10, a path conflict between the two AGVs is indicated by a time window [35]. This paper first plans the path of AGV#1, in order to avoid the path conflict between the two AGVs, this paper proposes a load balancing strategy, which is used to plan the path of AGV#2 to avoid the path conflict caused by the two AGVs passing through the same node or the same road segment at the same time period.

The road segments of AGV#1 are A43-A44, A44-A45, A45-A46, A46-A47, A47-A48, and number the road segments of AGV#1 as 1, 2, 3, 4, 5, 6.

This paper uses the method of load balancing to update the global road segment length.

For the road segment 1, within the preset range of road segment 1 are A33-A43, A52-A43, A53-A43, A54-A43,

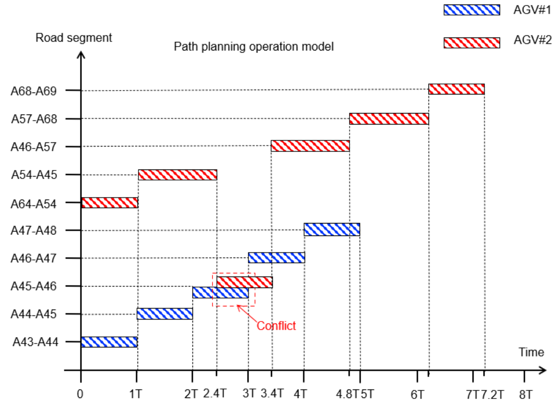


FIGURE 10. Time window of path conflicts between the two AGVs.

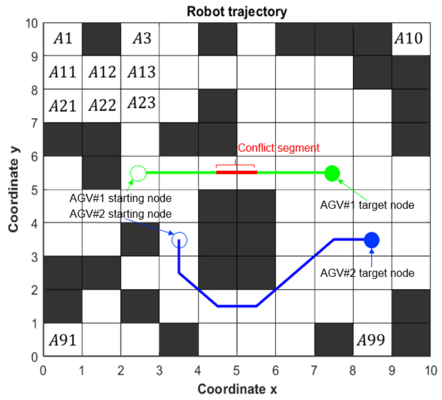


FIGURE 11. The path of the two AGVs after updating the global road segment length using the method of load balancing.

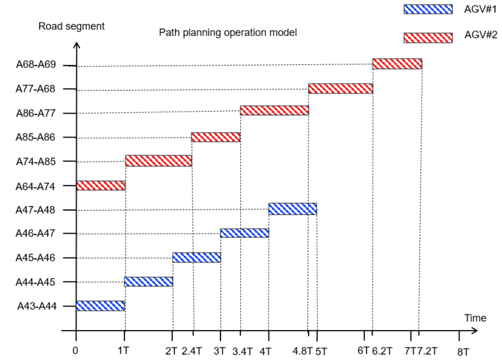


FIGURE 12. Time window model of two AGVs without path conflict.

A43-A44, A44-A53, A44-A54, A44-A45. There are no road segments that conflict with the path of AGV#1 on the map, that is, there is no impassable road segment within the preset range. The total number of road segments within the preset range of road segment 1 is $M_1 = 8$, and the number of impassable road segments is $m_1 = 0$, the penalty coefficient of road segment 1 is obtained by Eq. (7), that is $\propto_1 = 1.0000$. The time for AGV#1 to enter road segment 1 is 0T, the preset time range is 2T. In the time range from 0T to 2T, the number of times that the AGV#1 pass the road segment 1 is

TABLE 4. The global road segment length.

Road segment	α	P	L(m)
A43-A44	1.0000	1.6487	1.6487
A44-A45	1.0000	1.6487	1.6487
A45-A46	1.0000	1.6487	1.6487
A46-A47	1.0000	1.6487	1.6487
A47-A48	1.0000	2.7183	2.7183

Notes: α presents the penalty coefficient of the road segment; P presents the penalty term function of the road segment; L *presents the* length of the road segment i after updating the global road segment length.

TABLE 5. Improved AGV path and its corresponding time.

AGV number	AGV path
AGV#1	Path: A43-A44-A45-A46-A47-A48 Corresponding time: 0T-1T-2T-3T-4T-5T
AGV#2	Path: A64-A74-A85-A86-A77-A68-A69 Corresponding time: 0T-1T-2.4T -3.4T-4.8T-6.2T-7.2T

$n_1 = 1$, and the road segments passed by the AGV#1 are A43-A44, A44-A45, then the number of road segments passed by AGV#1 is $N = 2$, the value of the penalty function of road segment 1 is obtained by Eq. (8), that is $p_1 = 1.6487$. After the load balancing strategy is used to update the length of the road segment 1, the length of the road segment 1 is obtained by Eq. (9), that is $L_1 = 1.6487$ meters.

As shown in Table 4, the global road segment length is updated by the method of load balancing.

After the method of load balancing is used to update the global road segment length, this paper uses an improved ACO algorithm to plan the path of AGV#2. It can be seen from Table 5 that the specific road segments that the AGV#1 and the AGV#2 perform tasks and the corresponding time information to reach these road segments. This paper marks the time of the two AGVs to pass each segment in the time window model, as shown in Fig. 12, the path conflict between the two AGVs is avoided [36]. This simulation verified that the load balancing strategy can effectively avoid path conflicts of multiple AGVs.

VII. CONCLUSION

This paper uses ACO algorithm to plan the path and improves the shortcomings of the ACO algorithm. Aiming at the shortcomings of the low search efficiency of ACO algorithm, this paper proposes a strategy to improve the search step length to improve search efficiency, and use the method of simplifying path to avoid getting blindly tortuous paths. Aiming at the shortcoming of ACO algorithm that it is easy to fall into local optimum, this paper proposes a method of adaptive pheromone volatilization coefficient. On the one hand, this method is used to avoid ACO algorithm from falling into local optimum, and on the other hand, this method is used to speed up the convergence speed of the ACO algorithm. This paper proposes a multi-AGVs path planning method based on load balancing, which avoids path conflicts of multiple AGVs and further improves the work efficiency of the assembly workshop.

This paper only conducts in-depth research on ACO algorithm, so this paper can combine ACO algorithm with other algorithms to improve the performance of ACO algorithm. This paper has less research on the resolution methods of path conflicts of multiple AGVs, so the methods of conflict resolution can be further studied.

REFERENCES

[1] X. Zhou, "Research on AGV path planning method based on improved BAS algorithm," *J. Phys., Conf. Ser.*, vol. 1533, no. 2, pp. 14–16, 2020, doi: 10.1088/1742-6596/1533/2/022010.

[2] C. Lu, X. Chen, M. Miao, P. Gao, and G. Zhao, "Optimized control system for cross-workshop multi-platform common line mixed flow production, has BDC vehicle body vertical base controlled by vehicle discharging control module to transmit white vehicle body to corresponding general assembly workshops," CHN Patent CN 21 1180 638-U, Aug. 4, 2020.

[3] N. Wan, Q. Chen, Y. Liu, Q. Wu, D. Fu, and F. Liu, "Production workshop data collection system based on the IOT, has environmental information collection vehicle as vehicle body, and drive component arranged inside vehicle body, and environmental monitoring sensor connected to drive assembly," CHN Patent CN 111 491 269-A, Aug. 4, 2020.

[4] K.-G. Huo, Y.-Q. Zhang, and Z.-H. Hu, "Research on scheduling problem of multi-load AGV at automated container terminal," (in Chinese), *J. Dalian Univ. Technol.*, vol. 56, no. 3, pp. 244–251, 2016, doi: 10.7511/dllgxb201603004.

[5] C. Y. Kim, Y. H. Kim, and W.-S. Ra, "Modified 1D virtual force field approach to moving obstacle avoidance for autonomous ground vehicles," *J. Electr. Eng. Technol.*, vol. 14, no. 3, pp. 1367–1374, May 2019, doi: 10.1007/s42835-019-00127-8.

[6] X. F. Zhao, H. Z. Liu, S. X. Lin, and Y. K. Chen, "Design and implementation of a multiple AGV scheduling algorithm for a job-shop," *Int. J. Simul. Model.*, vol. 19, no. 1, pp. 134–145, Mar. 2020, doi: 10.2507/ijsimm19-1-co2.

[7] J.-J. Li, B.-W. Xu, O. Postolache, Y.-S. Yang, and H.-F. Wu, "Impact analysis of travel time uncertainty on AGV catch-up conflict and the associated dynamic adjustment," *Math. Problems Eng.*, vol. 2018, pp. 1–11, Dec. 2018, doi: 10.1155/2018/4037695.

[8] M. Zhong, Y. Yang, S. Sun, Y. Zhou, O. Postolache, and Y.-E. Ge, "Priority-based speed control strategy for automated guided vehicle path planning in automated container terminals," *Trans. Inst. Meas. Control*, vol. 42, no. 16, pp. 3079–3090, Dec. 2020, doi: 10.1177/0142331220940110.

[9] Y. Zhang, L. L. Li, H. C. Lin, Z. W. Ma, and J. Zhao, "Development of path planning approach using improved a-star algorithm in AGV system," *J. Internet Technol.*, vol. 20, no. 3, pp. 915–924, 2019, doi: 10.3966/160792642019052003023.

[10] B. Yu, N. Xie, B. Zheng, and D. Chen, "Methodology and decentralised control of modularised changeable conveyor logistics system," *Int. J. Comput. Integr. Manuf.*, vol. 32, no. 8, pp. 739–749, Jul. 2019, doi: 10.1080/0951192x.2019.1636406.

[11] Y. Qi and Y. Ni, "Improved ant colony algorithm based parking AGV path planning method involves calculating parking AGV running time based on two methods, and applying improved ant colony algorithm to plan parking AGV to obtain optimal path in time," CHN Patent CN 1112 890 07-A, Jun. 16, 2020.

[12] S. Chen, L. Zheng, X. Lu, Z. Liu, Y. Gao, and C. Cao, "Dispatch control system used for controlling laser automated guided vehicle (AGV) cart, has wireless communication base station that connects dispatch system with laser AGV cart and dispatch system with automatic charging pile," CHN Patent CN 111 367 294-A, Jul. 3, 2020.

[13] H. Ali, D. Gong, M. Wang, and X. Dai, "Path planning of mobile robot with improved ant colony algorithm and MDP to produce smooth trajectory in grid-based environment," *Frontiers Neurorobotics*, vol. 14, p. 15, Jul. 2020, doi: 10.3389/fnbot.2020.00044.

[14] T. Mai and C. Sichuan, "The path planning of agricultural AGV in potato ridge cultivation," *Ann. Adv. Agricult. Sci.*, vol. 3, no. 2, May 2019, doi: 10.22606/as.2019.32003.

[15] X. Dai, S. Long, Z. Zhang, and D. Gong, "Mobile robot path planning based on ant colony algorithm with a heuristic method," *Frontiers Neurorobot.*, vol. 13, p. 15, Apr. 2019, doi: 10.3389/fnbot.2019.00015.

[16] W. Gao, Q. Tang, B. Ye, Y. Yang, and J. Yao, "An enhanced heuristic ant colony optimization for mobile robot path planning," *Soft Comput.*, vol. 24, no. 8, pp. 6139–6150, Apr. 2020, doi: 10.1007/s00500-020-04749-3.

[17] L.-Z. Du, S. Ke, Z. Wang, J. Tao, L. Yu, and H. Li, "Research on multi-load AGV path planning of weaving workshop based on time priority," *Math. Biosci. Eng.*, vol. 16, no. 4, pp. 2277–2292, 2019, doi: 10.3934/mbe.2019113.

[18] P. Chen, Q. Li, C. Zhang, J. Cui, and H. Zhou, "Hybrid chaos-based particle swarm optimization-ant colony optimization algorithm with asynchronous pheromone updating strategy for path planning of landfill inspection robots," *Int. J. Adv. Robotic Syst.*, vol. 16, no. 4, Jun. 2019, Art. no. 172988141985908, doi: 10.1177/1729881419859083.

[19] Y. Wang, T. Han, X. Jiang, Y. Yan, and H. Liu, "Path planning of pattern transfer based on dual-operator and a dual-population ant colony algorithm for digital mask projection lithography," *Entropy*, vol. 22, no. 3, p. 295, Mar. 2020, doi: 10.3390/e22030295.

[20] X. Wang, H. Shi, and C. Zhang, "Path planning for intelligent parking system based on improved ant colony optimization," *IEEE Access*, vol. 8, pp. 65267–65273, 2020, doi: 10.1109/access.2020.2984802.

[21] N. Zagradjanin, D. Pamucar, and K. Jovanovic, "Cloud-based multi-robot path planning in complex and crowded environment with multi-criteria decision making using full consistency method," *Symmetry*, vol. 11, no. 10, p. 1241, Oct. 2019, doi: 10.3390/sym11101241.

[22] P. K. Das and P. K. Jena, "Multi-robot path planning using improved particle swarm optimization algorithm through novel evolutionary operators," *Appl. Soft Comput.*, vol. 92, Jul. 2020, Art. no. 106312, doi: 10.1016/j.asoc.2020.106312.

[23] K. J. C. Fransen, J. A. W. M. van Eekelen, A. Pogromsky, M. A. A. Boon, and I. J. B. F. Adan, "A dynamic path planning approach for dense, large, grid-based automated guided vehicle systems," *Comput. Oper. Res.*, vol. 123, Nov. 2020, Art. no. 105046, doi: 10.1016/j.cor.2020.105046.

[24] M. Zhong, Y. Yang, Y. Dessouky, and O. Postolache, "Multi-AGV scheduling for conflict-free path planning in automated container terminals," *Comput. Ind. Eng.*, vol. 142, Apr. 2020, Art. no. 106371, doi: 10.1016/j.cie.2020.106371.

[25] K. Guo and J. Zhu, "Multi-agent AGV conflict free path planning based on improved speed control method in automated terminals," *Int. Core J. Eng.*, vol. 6, no. 9, pp. 386–398, Sep. 2020, doi: 10.6919/ICJE.202009_6(9).0049.

[26] X. Yu, W.-N. Chen, X.-M. Hu, T. Gu, H. Yuan, Y. Zhou, and J. Zhang, "Path planning in multiple-AUV systems for difficult target traveling missions: A hybrid Metaheuristic approach," *IEEE Trans. Cognit. Develop. Syst.*, vol. 12, no. 3, pp. 561–574, Sep. 2020, doi: 10.1109/tcds.2019.2944945.

[27] Q. Song, Q. Zhao, S. Wang, Q. Liu, and X. Chen, "Dynamic path planning for unmanned vehicles based on fuzzy logic and improved ant colony optimization," *IEEE Access*, vol. 8, pp. 62107–62115, 2020, doi: 10.1109/access.2020.2984695.

[28] M. Vafaei, A. Khademzadeh, and M. A. Pourmina, "QoS-aware multi-path video streaming for urban VANETs using ACO algorithm," *Telecommun. Syst.*, vol. 75, no. 1, pp. 79–96, Sep. 2020, doi: 10.1007/s11235-020-00677-7.

[29] Z. Huang, X. Zhai, H. Wang, H. Zhou, H. Zhao, and M. Feng, "On the 3D track planning for electric power inspection based on the improved ant colony optimization and a algorithm," *Math. Problems Eng.*, vol. 2020, pp. 1–11, Jun. 2020, doi: 10.1155/2020/8295362.

[30] Q. Yuan and C.-S. Han, "Research on robot path planning based on smooth a algorithm for different grid scale obstacle environment," *J. Comput. Theor.*, vol. 13, no. 8, pp. 5312–5321, 2016, doi: 10.1166/jctn.2016.5419.

[31] J. Yuan, H. Wang, C. Lin, D. Liu, and D. Yu, "A novel GRU-RNN network model for dynamic path planning of mobile robot," *IEEE Access*, vol. 7, pp. 15140–15151, 2019, doi: 10.1109/access.2019.2894626.

[32] Y. Liu, P. Zeng, Z. Wang, H. Bai, and Z. Li, "Method for planning automated guided vehicle path based on improved ant colony optimization algorithm and hot and cold principle, involves searching optimal value of convergence and number of convergence iterations by ant colony algorithm," CHN Patent CN 111 353 621-A, Jun. 30, 2020.

[33] Q. Luo, H. Wang, Y. Zheng, and J. He, "Research on path planning of mobile robot based on improved ant colony algorithm," *Neural Comput. Appl.*, vol. 32, no. 6, pp. 1555–1566, Mar. 2020, doi: 10.1007/s00521-019-04172-2.

[34] X. Wang, H. Shi, and C. Zhang, "Path planning for intelligent parking system based on improved ant colony optimization," *IEEE Access*, vol. 8, pp. 65267–65273, 2020, doi: 10.1109/access.2020.2984802.

[35] W. Xie, Q. Yang, Y. Lian, W. Chen, H. Zheng, H. Rao, K. Wang, Y. Huang, D. Lin, G. Qin, Y. Bie, H. Wen, and X. Deng, "Global vision based storage-navigated intelligent vehicle scheduling method, involves performing route planning process by using time window algorithm, obtaining planned path to finish predetermined transportation task," CHN Patent CN 110 780 671-A, Feb. 11, 2020.

[36] S. Chen, L. Zheng, X. Lu, Z. Liu, Y. Gao, and C. Cao, "Dispatch control system used for controlling laser automated guided vehicle (AGV) cart, has wireless communication base station that connects dispatch system with laser AGV cart and dispatch system with automatic charging pile," CHN Patent CN 111 367 294-A, Jul. 3, 2020.

YIYANG LIU was born in Shenyang, Liaoning, China, in 1980. He received the B.S. and M.A. degrees in automation, control theory and control engineering from Northeast University, in 2003 and 2006, respectively, and the Ph.D. degree in pattern recognition and intelligent system from the Shenyang Institute of Automation, Chinese Academy of Sciences, in 2010.

He is currently a Professor and a Principle Investigator with the Shenyang Institute of Automation, Chinese Academy of Sciences. As a Project Leader, he has presided over a number of national natural science foundation projects, ministry of industry and information technology's intelligent manufacturing special projects, and industrial internet special projects. His main research interests include cyber physical systems, complex system modeling and simulation, intelligent optimization, and intelligent control.

Dr. Liu is a member of the Edge Computing Committee, Chinese Association of Automation. He is also an Expert in advanced manufacturing technology with the National Technology Prediction, Ministry of Science and Technology, and a Peer-Reviewed Expert with the National Natural Science Foundation of China.

ZHENG HOU was born in Sanmenxia, Henan, China, in 1995. He received the bachelor's degree from the College of Information Engineering, Shenyang University, Shenyang, in 2018. He is currently pursuing the master's degree with the Shenyang Institute of Automation, Chinese Academy of Sciences, and the Shenyang University of Technology. His research was in path planning of multiple automatic guide vehicles. His main research interests include path planning for multiple automatic guide vehicles, artificial intelligence algorithms, and conflict resolution.

YUANYUAN TAN received the B.S. degree in information and computing science from Bohai University, Jinzhou, China, in 2007, and the M.S. and Ph.D. degrees from the College of Information Science and Engineering, Northeastern University, Shenyang, China, in 2009 and 2013, respectively. She is currently an Associate Professor with the School of Information Science and Engineering, Shenyang University of Technology. She has published over several journal and conference proceeding papers in research areas. Her research interests include scheduling and production plan in steelmaking, continuous casting and hot rolling, and intelligent optimization algorithm.

HAOQUN LIU was born in Zhoukou, Henan, China, in 1992. He received the bachelor's degree from Shenyang University, in 2017, and the master's degree from Northeastern University. He was with the Joint Training Program of the Shenyang Institute of Automation and Northeastern University, from 2017 to 2020. His major research interests include AGV path planning, collision avoidance, and so on.

CHUNHE SONG received the Ph.D. degree in computer science from Northeastern University, Shenyang, China, in 2012. He is currently a Professor with the Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang. His research interests include artificial intelligence and edge computing.