Sensors and Materials, Vol. 33, No. 8 (2021) 2679–2691
MYU Tokyo

S & M 2648

Improved Ant Colony Algorithm-based Automated Guided Vehicle Path Planning Research for Sensor-aware Obstacle Avoidance

Rong Liu*

College of Port and Shipping Management, Guangzhou Maritime University, Guangzhou 510725, China

(Received March 23, 2021; accepted June 23, 2021)

Keywords: ant colony algorithm, automated guided vehicle, sensor-aware obstacle avoidance, path planning, logistics management

Automated guided vehicles (AGVs) are the main delivery vehicle for the horizontal transport of containers between the quayside and yard of automated container terminals (ACTs). The coordination of AGVs with the quayside bridge and yard bridge is necessary for loading and unloading operations at the wharf and to improve logistics management efficiency. Toward solving the problem of AGV path planning and sensor-aware obstacle avoidance in a dynamic complex environment for the Internet of Things (IoT), we proposed an improved ant colony algorithm based on an adaptive dynamic parameter adjustment strategy (IACA-ADPA) in this paper. The grid method is first used to construct a motion space model because it is easy to implement, analyze, store, and express, and make the AGV reach its target node safely and smoothly. Then the proposed IACA-ADPA is used for global path planning and for efficient AGV path design and adjustment. Finally, the improved time window adjusts the waiting time of the AGV to avoid local collisions. The simulation results of different scale paradigms show that the IACA-ADPA can effectively avoid road section obstacles and node obstacles, and improve the safety and efficiency of a multi-AGV system.

1. Introduction

In logistics storage spaces, such as intelligent warehouses and automated terminals, multiple automated guided vehicles (AGVs) are often used to transport goods, and issues such as congestion and collision among AGVs inevitably exist. Therefore, the path planning of multiple AGVs is more complicated than single-path planning.(1,2) Multi-AGV path planning is a hot and difficult topic in current research. Many studies have focused on AGV path planning and adjustment, which is a key point in multi-AGV path planning. For example, Smolic-Rocak *et al.* used a time window in the form of a vector, judged the feasibility of candidate paths by inserting an appropriate time window, and performed an overlapping test of the time window in their proposed multi-AGV dynamic path method.(3) Given the deficiency of static path planning, Ren *et al.* updated the time window for tasks with low priority based on the ideal time window for alternative paths to avoid vehicle–vehicle obstacles and collisions.(4)

*Corresponding author: e-mail: 121712170@qq.com

https://doi.org/10.18494/SAM.2021.3396

ISSN 0914-4935 © MYU K.K.
https://myukk.org/

In addition to path design and adjustment, AGVs can also avoid obstacles by waiting (such as by slowing down and stopping). Xu *et al.* realized multi-AGV collision avoidance by adjusting the AGV sequence of corresponding nodes in real time to solve the problem of AGVs being unable to follow a predetermined path planned by the time window in a dynamic unstable environment.(5) Cao *et al.* adopted a two-stage traffic control strategy to avoid obstacles and deadlocks in a multi-AGV system. First, they calculated the moving path of each AGV offline and then controlled the operation of each AGV online through a traffic controller.(6) However, such waiting strategies are only temporary measures to avoid obstacles, and a waiting strategy is not considered in initial path planning. Different waiting strategies have different effects on the overall operating efficiency of AGVs, and the waiting time is also a key problem in the process of multi-AGV operation. Zhou and He predicted collisions by space and time obstacle detection, compared the time overhead of two kinds of collision avoidance based on the wait and changing the path, and chose the implementation requiring less time in their proposed collision avoidance scheduling policy based on the rolling time domain of shuttle cars.(7) However, in their study, an AGV on a node was only allowed to run or stop simultaneously, which limits its application scope. Mehrabad designed a mathematical model including job shop scheduling and an obstacle-free path problem, and proposed a two-stage ant colony algorithm (ACA) while considering the route choice and waiting situation on the node.(8) However, in these studies, it was assumed that the road network grid was a square, whereas the actual situation is more complex and requires further study.

The AGV path planning problem refers to finding an optimal path from a starting point to a termination point in a working environment with obstacles so that the AGV can safely bypass all obstacles without collision and take the shortest path in the movement process.(9,10) At present, much progress has been made in research on the path planning problem under a static environment, which mainly uses one-off global planning to find a safe path from the start to the finish, where the path is optimized by an algorithm such as the genetic algorithm (GA), neural network, or fuzzy algorithm. However, in these methods the search space used for evaluation is large, the algorithm is complex, and the efficiency is low.

Due to the complex situation in which both known and unknown dynamic obstacles exist in the dynamic environment, in this paper, we adopt the promising idea of combining global path planning and local collision avoidance planning. We use the improved ant colony algorithm based on an adaptive dynamic parameter adjustment strategy (IACA-ADPA) for the global path planning of an AGV according to global static information for the first time. On this basis, dynamic collision prediction and information feedback based on a local scroll window are then used to carry out local collision avoidance planning, and a safe collision-free AGV path with a relatively short distance is determined, thus providing a joint global path planning and local collision avoidance planning solution for research on multi-AGV path planning in a dynamic environment.

2. Related Work

2.1 AGV path planning for obstacle avoidance

If an obstacle on an AGV path is not properly considered, it will increase the time and cost of the operation of the AGV, as well as the waiting time at the quayside bridge or yard bridge, and the overall loading time of an automated container terminal (ACT), thus reducing its operational efficiency and greatly increasing the operation cost of the terminal. Therefore, video multi-AGV path planning considering obstacle avoidance is a key issue for ACT operation management as shown in Fig. 1. Research on relevant models and algorithms has provided us with an important theoretical basis for terminal operation and management.

AGV path planning for obstacle avoidance has become the focus of current research. Lyu *et al*. embedded the Dijkstra algorithm with a time window into the GA to search the shortest path and detect the collision of multiple vehicles simultaneously.(11) Fazlollahtabar *et al*. established a mixed-integer programming (MIP) model and proposed a two-stage optimization method (e.g., searching a solution space and finding the optimal solution) to avoid obstacles between AGVs.(12) Nishi and Tanaka proposed a Petri net decomposition method for bidirectional AGV system scheduling and path obstacle avoidance in a dynamic environment. The whole Petri net was decomposed into a task subnet and an AGV subnet, and a method to avoid deadlock was embedded in the AGV subnet.(13) Miyamoto and Inoue established a MIP model and proposed a local random search method to solve the path planning problem in AGV system scheduling and obstacle avoidance with a limited capacity.(14) Both AGV path planning and task scheduling were considered in the previous research, and path obstacles were avoided by adjusting the job sequence of AGV tasks. Usually, however, the order of AGV task execution is determined.

When it is necessary to determine the order of AGV tasks, most studies have focused on the path search method. Maeopolski used a matrix to describe the layout of a transportation system and regarded the AGV as moving from matrix to matrix at a fixed average velocity, thus

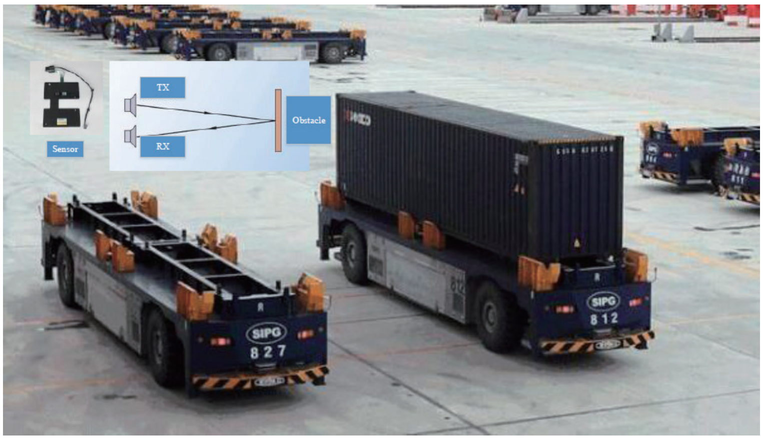

Fig. 1. (Color online) Model of sensor-aware obstacle avoidance.

proposing a new paradigm to prevent AGV collision and deadlock.(15) Antakly *et al.* utilized temporal Petri nets to model the path search and proposed a three-stage heuristic algorithm to avoid collisions with other AGVs by imposing appropriate delays on AGVs.(16) Mohammadi and Shirazi proposed an advanced flexible process model based on simulation to dynamically prevent AGV collision and deadlock through a predictive mechanism.(17) Fanti *et al.* coordinated AGV movements through a regional control and dispersion protocol to avoid collisions.(18) Nishi *et al.* considered path planning for obstacle avoidance under AGV acceleration or deceleration conditions, established a continuous-time model, and proposed a heuristic algorithm based on column generation.(19) Duan *et al.* proposed an operator for finely tuning paths to make path fragments shorter and avoid obstacles, and realized dynamic path planning based on a GA.(20) Ahmed *et al.* proposed a collision prediction method based on vertex attributes and real time location information combined with graph theory, established a MIP model, and proposed an improved particle swarm optimization (PSO) algorithm suitable for optimizing collision avoidance decisions of multi-AGV systems.(21) Hu *et al.* established a MIP model by analyzing the obstacles between sections and nodes and proposed an induced ant colony particle swarm algorithm. In the state transition rules of the algorithm, inducer factors were added to guide AGVs to avoid obstacles.(22) Miao *et al.* proposed a two-stage path planning algorithm based on ant swarm optimization and game theory for multi-agent path planning. To avoid collision between multiple AGVs, a dynamic obstacle avoidance model involving multiple agents was constructed using game theory. (23) Also, the virtual action method was used to solve the selection problem of multi-Nash equilibrium. The above studies avoided path obstacles by designing and adjusting AGV driving routes or taking waiting measures. However, the integration of the ACA and PSO, the path, and the waiting time were simultaneously optimized. However, most of the path planning studies for obstacle avoidance focused on manufacturing systems.

2.2 IACA

The ACA was first proposed by the Italian scholars Dorigo and Gambardella to achieve the global optimal solution by simulating the intelligent behavior of an ant colony and cooperative food-seeking.(24) Also, the ACA was characterized by a distributed calculation, heuristic search, and information-based positive feedback, and had strong robustness and other advantages. Over the past decade, a large number of studies on the ACA have been performed by scholars at home and abroad, and it has been widely applied to solve travel agent, routing, workpiece sequencing, vehicle transportation scheduling, graph coloring, and AGV path planning problems.

3. Grid Method-based Problem Formulation

Most moving environment trajectories of AGVs are modeled as a grid map, topology map, or feature map, in which the grid method is widely used in AGV models because it is easy to implement, store, analyze, and express in the modeling process. Therefore, we adopt the grid method to model the path planning of multiple AGVs.

The working environment of multiple AGVs is assumed as follows: the working interval of each AGV is defined as a finite region on a two-dimensional plane, denoted as $\Re$, where stationary obstacles with a constant size are uniformly distributed, and the AGV moves among grids. More specifically, 0 corresponds to white grids, 1 corresponds to black grids, i.e., obstacle grids are not feasible, and each grid is an $l \times l$ square. The detailed grid model is shown in Fig. 2, in which the grids are numbered from left to right and from top to bottom with serial numbers 1, 2, 3,..., m, ..., M_u to distinguish the location of each grid. From left to right is the positive direction of the X-axis, and from bottom to top is the positive direction of the Y-axis. The grid in row i and column j is denoted as $G(i, j)$, the ordinal number is m, and the relation between grid number m and coordinate $G(i, j)$ is expressed as

$$\begin{cases} i = \mathrm{mod}[(m-1) \div M_u] + 1, \\ j = 10 - floor[(m-1) \div M_u], \end{cases} \tag{1}$$

where the function *floor* defines rounding down, the function mod is used to calculate the remainder, and M_u is the total number of grids.

According to the research scope specified in Sect. 1, the proposed overall path-planning algorithm consists of two stages, i.e., global path planning and local collision avoidance planning, as shown in Fig. 3. Based on the static obstacle information provided by the global perception module, the IACA-ADPA is adopted to determine an unexplored path. The initial global optimization path of unknown dynamic obstacles is also considered. Then, the AGV follows the globally optimized path and moves on the basis of the sensing results. The motion information of the dynamic obstacle in the scroll window is detected. According to the predicted motion trajectory of the dynamic obstacle, it is judged whether the AGV will collide with it, and then a corresponding collision avoidance strategy is adopted to reach the destination safely and ensure better path planning.

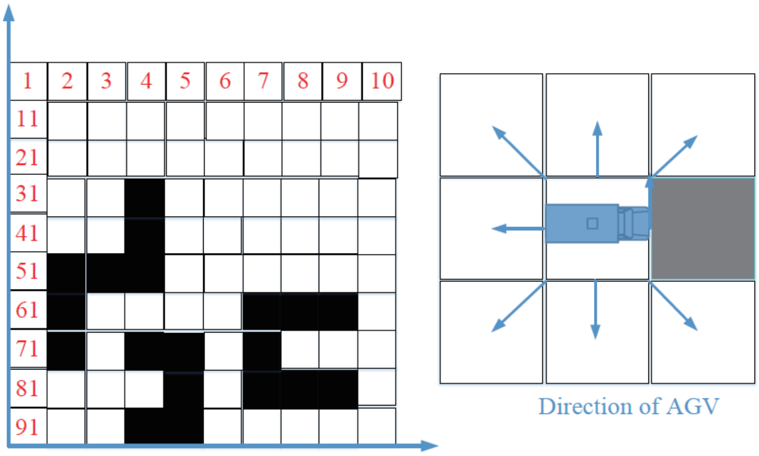

Fig. 2. (Color online) Environment model of the grid method.

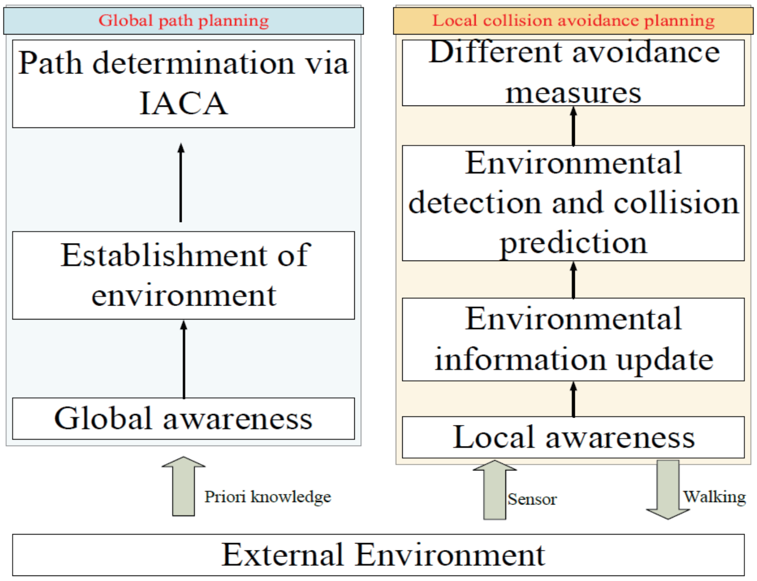
Global path planning
Path determination via IACA
Establishment of environment
Global awareness
Local collision avoidance planning
Different avoidance measures
Environmental detection and collision prediction
Environmental information update
Local awareness
Priori knowledge
Sensor
Walking
External Environment

Fig. 3. (Color online) Two-stage solution based on IACA-ADPA.

4. Global Path Planning based on IACA-ADPA

Ants leave behind substances called pheromones in their path and sense them as they move, which instruct them of the direction in which they need to go. Therefore, the collective behavior of an ant colony composed of a large number of ants shows a positive feedback phenomenon: the shorter a path is, the more ants have passed along the path, the greater the pheromone strength left behind, and the greater the probability that subsequent arrivals will choose the path. It is through this information exchange that individual ants choose the shortest path and search for food. An ACA is an optimization algorithm that simulates the behavior of an ant colony.

4.1 ACA-based mathematical model

During the movement, ant k will determine the next direction in which to proceed according to the pheromone concentration along each path. Here, we denote the set of ants as $K(k \in K)$ and the scale of the grid diagram as $M_u * M_u$. At time t, ant k will transfer from position i to position j, and the corresponding transfer probability is defined as

$$p_{ij}^k(t) = \begin{cases} \dfrac{\gamma_{ij}^{\alpha}(t) * \eta_{ij}^{\beta}(t)}{\sum\limits_{j \in A_i} \gamma_{ij}^{\alpha}(t) * \eta_{ij}^{\beta}(t)}; & \text{if} \quad j \in A_i, \\ 0; & \text{otherwise,} \end{cases} \tag{2}$$

where $\gamma_{ij}(t)$ is the residual pheromone concentration at time t for path $\langle i, j \rangle$, $\eta_{ij}(t)$ is the local heuristic function of visibility (defined as $1/d_{ij}$), α and β respectively express the influence weights of $\tau_{ij}(t)$ and $\eta_{ij}(t)$ on the overall metastasis probability, and A_i represents the feasible region node (obtained after removing the obstacle node and the set of taboo points).

The pheromone will evaporate over time, with ς representing the evaporated remnant of the pheromone. After Δt time units, the ant colony will complete a circular movement. At this time, the pheromone concentration on each path will be adjusted according to the following global adjustment criterion:

$$\gamma_{ij}(t+\Delta t) = \varsigma \cdot \gamma_{ij}(t) + \sum_{k=1}^{m} \Delta\gamma_{ij}^{k}, \tag{3}$$

where $\Delta\gamma_{ij}^{k}$ defines the pheromone concentration on path $\langle i, j \rangle$ for the kth ant in this circulation. In this paper, the following ant cycle calculation model based on global information is adopted to update information:

$$\Delta\gamma_{ij}^{k}(t) = \begin{cases} Q/L_k; & \text{if ant } k \text{ across the path } \langle i, j \rangle \text{in this circulation,} \\ 0; & \text{otherwise,} \end{cases} \tag{4}$$

where Q is a constant that defines the pheromone concentration and L_k is the path length of ant k in this circulation. More specifically, the traditional ACA (TACA) involves initialization, construction of the state transition probability, modification of the Tabu table, updating of pheromones, and the iteration cycle.

4.2 Proposed IACA-ADPA process

Although the TACA faces disadvantages such as low convergence speed, being easily trapped at a local optimal solution, and search stagnation, it has significant advantages (e.g., an excellent distributed mechanism, robustness, and easy combination with other algorithms) in achieving the global optimal solution. To improve the effectiveness and practicability of the existing ACA approach, it is vital to find the optimal path when ants avoid obstacles and fall into concave obstacles, as addressed by the IACA-ADPA approach proposed in this paper. The proposed approach has two sub-approaches as follows. One novel sub-approach is that the dynamic parameter adaptive adjustment strategy is adopted to improve α and β, instead of using fixed values in the TACA solution.(25) After each generation of ant optimization, the dynamic parameters α and β are further updated to expand the global search space to avoid falling into a local optimum. The other sub-approach is to update the adaptive adjustment volatility coefficient ς accordingly after several iterations. When the number of iterations exceeds a certain value and the optimal path remains unchanged, the volatility coefficient is updated to increase the global search range so that the optimal solution can be achieved in the proposed IACA-ADPA approach.

(1) Adjusting the dynamic parameters α and β

Acquiring exact environmental information improves the performance of the existing ACA. Therefore, it is crucial to find how to extract the environmental information. More specifically, the dynamic parameters α and β will directly affect the randomness of the ant's path. Therefore, the ant colony will fall into a local optimum when α is too large or too small. On the other hand, the larger the value of β, the greater the probability the ants will choose a path closer to the target. Therefore, the minimum values of α and β are set in the initial stage, where the global search capability for the AGVs is increased to improve the efficiency of the global optimal solution. When the number of iterations reaches a given constant Θ, the dynamic parameters α and β are updated using Eqs. (5) and (6), respectively, which improves and even accelerates the convergence performance of the proposed IACA-ADPA. In Eqs. (5) and (6), $\alpha_{\min} = 1$ and $\beta_{\min} = 4$ are the initial values of α and β, I_N is the current iteration number, as well as $\alpha_{\max} = 4$ and $\beta_{\max} = 9$ are selected as the maximum and minimum values of α and β, respectively.

$$\alpha = \begin{cases} \alpha_{\max} - \dfrac{\Theta(\alpha_{\max} - \alpha_{\min})}{I_N}; & I_N \ge \Theta, \\ \alpha_{\min}; & I_N \le \Theta, \end{cases} \tag{5}$$

$$\beta = \begin{cases} \beta_{\max} - \dfrac{\Theta(\beta_{\max} - \beta_{\min})}{I_N}; & I_N \ge \Theta, \\ \beta_{\min}; & I_N \le \Theta, \end{cases} \tag{6}$$

(2) Adjusting the dynamic parameter ς

The pheromone volatile element ς plays a crucial role in global path planning for the proposed IACA-ADPA approach. When ς is large, the former search path is more likely to be selected again, and the positive feedback of information plays a dominant role. However, considering the above factors at the initial stage of the iterations, a large ς helps to improve the positive feedback of the pheromone to make up for the lower convergence speed of the dynamic parameters α and β. When the total number of iterations reaches a certain value Θ, Eq. (7) is adopted to reduce ς and make up for the shortcoming of the global search. The initial value of $\varsigma_0 = 0.9$ is set. The detailed process of the proposed IACA-ADPA approach is shown in Algorithm 1.

$$\varsigma = \begin{cases} \left(1 - \sqrt{\dfrac{\Theta}{I_N}}\right) \cdot \varsigma_0; & I_N > \Theta, \\ \varsigma_0; & I_N \le \Theta, \end{cases} \tag{7}$$

Algorithm 1
Proposed IACA-ADPA.

```
Initialization: Construct the environment model, generate the grid diagram with the known obstacles, and set the grid sequence number, starting point, and ending point.
Select the next node according to the transition state probability using Eq. (2).
Move from the current node to the next node using Eq. (3).
Update $\alpha$, $\beta$, and $\varsigma$ [Eqs. (5)–(7)].
Storage path: Store the search path and path length of the current ant $k$;
    Pheromone update: If the ant is trapped in a concave obstacle and a deadlock occurs, the amount of pheromone on the path is set to zero. When a concave obstacle is encountered but deadlock does not occur, the amount of pheromone is gradually reduced. Otherwise, the global amount of pheromone is updated using Eqs. (3) and (4).
Termination condition: When the number of iterations is greater than the given number of iterations $I_N$ (max), the program ends and the optimal solution is achieved.
```

5. Local Path Planning based on Scroll Window

When motorized obstacles occur in the grid environment labeled as $M_u * M_u$, it is assumed that AGVs spend some time Δt to move from the original grid i to the next target grid j, simulate the scroll window accurately under the moving path, and predict the moving trajectory of the AGVs and dynamic obstacles in the next scroll window. Therefore, the above problem description is converted to predict whether the trajectory intersects with an obstacle to judge the state of the collision. The method used to solve the problem is presented as follows. First, if there is no collision between the AGV and the obstacle in time unit Δt, the AGV will continue to move to the next target grid according to the original planned route. Then, if the two entities are predicted to undergo a side collision, the AGV will wait for time Δt before reaching the predicted collision point and then proceed along the original planned route. Next, if the two entities are predicted to collide head-on, the white grid for the moving path will be planned by regarding the grid at the predicted collision point as the local target point, then using the proposed PSO approach for local path planning.(26)

Under a more complicated grid environment, some local target points cannot be reached, as shown in Fig. 4. When the AGV moves to point A, the scroll window proceeds according to the

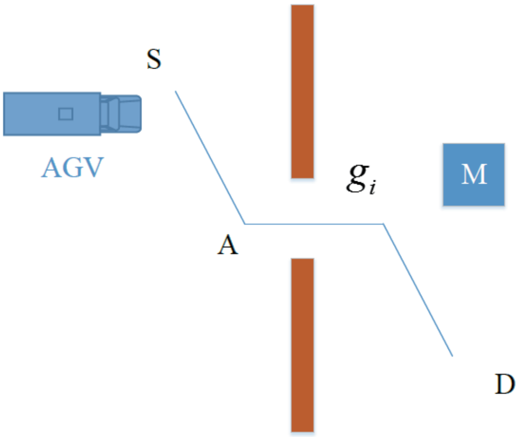

Fig. 4. (Color online) Case of local target points that are not reachable.

original path and detects a dynamic obstacle at point g_i. Therefore, the general method cannot be used to plan the path when the local target is unreachable. To solve this problem, we propose the following two solutions. One sub-solution is that the AGV no longer looks for a local sub-goal node in the dynamic obstacle environment. Instead, the grid where the collision point is predicted by the scroll window is temporarily set as a static obstacle. The other sub-solution is to plan a new global path to replace the original path.

6. Simulation Experiment and Analysis of Results

6.1 Performance comparison of proposed joint path planning

To verify the effectiveness and feasibility of the proposed IACA-ADPA approach, a simulation experiment is carried out, and both the TACA(25) and the IACA(26) are regarded as baselines for comparison with the IACA-ADPA approach. MATLAB 2015b software is used as an experimental testbed in the Windows 7 operating system. The basic experimental parameters used to verify this algorithm are K = 100 ants, a maximum number of iterations of I_N(max) = 150, a pheromone intensity of $\rho = 1$, and Θ = 10, where Θ is based on the maximum number of iterations, the dynamically adjusted pheromone impact weight, heuristic weights, and ACA search characteristics. The proposed algorithm can find the shortest path of AGVs in a concave-type obstacle in the environment, and the optimal solution converges after 40 generations, as shown in Fig. 5. In contrast, the proposed joint path planning algorithm with a concave-type obstacle only needs 20 generations to converge to the optimal solution, thus proving the feasibility and effectiveness of the algorithm.

To further evaluate the effectiveness of the proposed joint path planning algorithm for the AGV path planning, the working environment of the AGV is selected to be the same as that in Ref. 26, i.e., a 20 × 20 grid unit, and the proposed IACA-ADPA is compared with the TACA and the IACA. The simulation results are shown in Fig. 6. Although the TACA can search for the

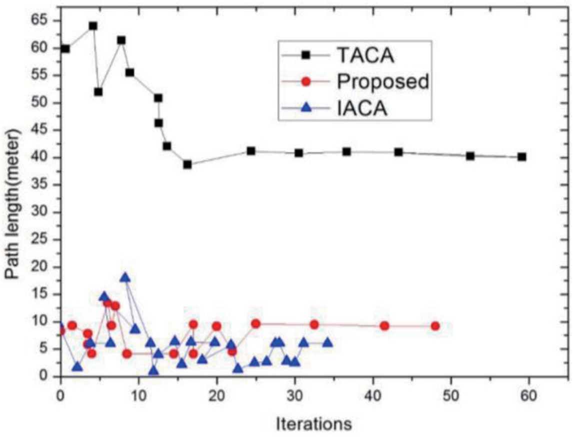

Fig. 5. (Color online) Effect of iteration number on path length for different algorithms.

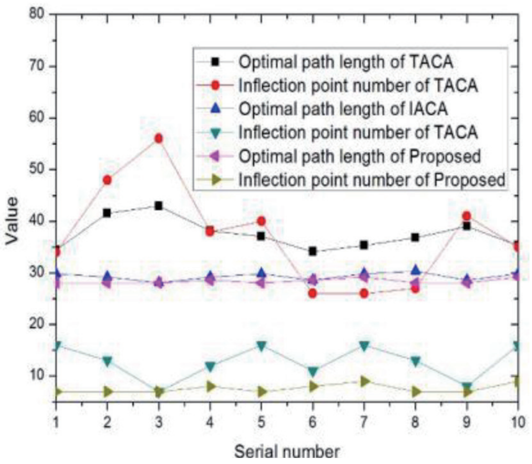

Fig. 6. (Color online) Comparison of different algorithms.

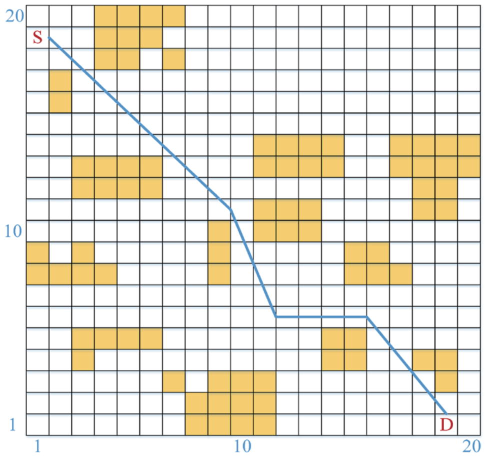

Fig. 7. (Color online) Global path planning of proposed AGV.

optimal path, the proposed IACA-ADPA is relatively stable and can find the optimal path after 25 iterations, with a higher success rate in searching for the optimal value and a smaller number of inflection points.(26) Therefore, the proposed joint path planning algorithm is feasible for planning the path of mobile AGVs by incorporating global path planning and local path planning.

6.2 Feasibility of joint IACA and scroll window

The feasibility of joint IACA and the scroll window is shown. Here, the operating environment of the simplified AGV is a 20 × 20 grid unit, and the starting grid and ending grid coordinates are initialized to (1, 19) and (19, 1), respectively. The parameter settings are the same as those of the IACA-ADPA approach in the static environment. In this experiment, it is assumed that the moving speed of the dynamic obstacle is consistent with the speed of the AGV. First, in the AGV operating environment, dynamic obstacles are ignored and static obstacles are studied. The IACA-ADPA approach first plans a globally optimal path to avoid all static obstacles and then performs local path planning based on the scroll window, as shown in Fig. 7. The AGV moves forward according to the planned global optimal path and performs local collision prediction and collision avoidance.

7. Conclusions

Although the TACA has achieved good results in path planning, it still has problems such as a low convergence speed and easily becoming trapped at local optima in a complex environment. Toward this end, an IACA-ADPA is proposed to improve the performance of the TACA and IACA. More specifically, first, adding such a strategy in an environment with static obstacles in the grid resolves the shortcoming that the TACA is prone to falling into local optima, which leads to the stagnation of the algorithm; second, in a static environment with dynamic obstacles, an IACA based on scroll window is proposed, which first obtains the static global optimal path using the IACA-ADPA, then carries out the local rolling prediction and collision avoidance by using the principle of a scroll window according to the planned global optimal path, and finally obtains an optimal collision-free path with good adaptability to an environment with dynamic obstacles. In the future, AGV dynamic path planning, multi-AGV collaboration, and how to reduce the energy consumption of AGVs need to be further studied with the help of the joint path planning algorithm.

Acknowledgments

This work was supported by Guangzhou Maritime University Innovation and Strength Project in 2017-Logistics Engineering Key Major Construction.

References

1 R. Stetter: Sensors **20** (2020) 4154. https://doi.org/10.3390/s20154154

2 W. Zou, Q. Pan, and M. F. Tasgetiren: IEEE Access **8** (2020) 35063. https://doi.org/10.1109/ACCESS.2020.2973336

3 N. Smolic-Rocak, S. Bogdan, Z. Kovacic, and T. Petrovic: IEEE Trans. Autom. Sci. Eng. **7** (2010) 151. https://doi.org/10.1109/TASE.2009.2016350

4 Z. Ren, J. Lai, Z. Wu, and S. Xie: Neurocomputing **443** (2021) 329. https://doi.org/10.1016/j.neucom.2021.02.034

5 J. Xu and K. S. Park: Microsyst. Technol. **26** (2020) 3533. https://doi.org/10.1007/s00542-020-04948-w

6 X. Cao, X. Gao, X. Zeng, Y. Ma, Y. Gao, W. Baeyens, Y. Jia, J. Liu, C. Wu, and S. Su: Chemosphere **272** (2021) 129891. https://doi.org/10.1016/j.chemosphere.2021.129891

7 B. Zhou and Z. He: Comput. Ind. Eng. **140** (2020) 106268. https://doi.org/10.1016/j.cie.2020.106268

8 M. S. Mehrabad, S. D. Arani, F. Evazabadian, and V. Mahmoodian: Comput. Ind. Eng. **86** (2015) 2. https://doi.org/10.1016/j.cie.2015.01.003

9 M. Zhong, Y. Yang, Y. Dessouky, and O. Postolache: Comput. Ind. Eng. **142** (2020) 106371. https://doi.org/10.1016/j.cie.2020.106371

10 J. S. Choi, S. Y. Jeong, B. G. Choi, S.-T. Ryu, C. T. Rim, and Y.-S. Kim: IEEE Trans. Power Electron. **35** (2020) 7863. https://doi.org/10.1109/TPEL.2019.2962194

11 X. Lyu, Y. Song, C. He, Q. Lei, and W. Guo: IEEE Access **7** (2019) 74909. https://doi.org/10.1109/ACCESS.2019.2919109

12 H. Fazlollahtabar, M. S.-Mehrabad, and E. Masehian: Ind. Rob. **42** (2015) 252. https://doi.org/10.1108/IR-12-2014-0437

13 T. Nishi and Y. Tanaka: IEEE Trans. Syst. Man Cybern. Part A Syst. Humans **42** (2012) 1230. https://doi.org/10.1109/TSMCA.2012.2183353

14 T. Miyamoto and K. Inoue: Comput. Ind. Eng. **91** (2016) 1. https://doi.org/10.1016/j.cie.2015.10.017

15 W. Maeopolski: Comput. Ind. Eng. **126** (2018) 472. https://doi.org/10.1016/j.cie.2018.10.002

16 D. Antakly, J. J. Loiseau, and R. Abbou: IFAC-PapersOnLine **50** (2017) 11169. https://doi.org/10.1016/j.ifacol.2017.08.1239

17 E. K. Mohammadi and B. Shirazi: ISA Trans. **96** (2020) 228. https://doi.org/10.1016/j.isatra.2019.06.024

18 M. P. Fanti, A. M. Mangini, G. Pedroncelli, and W. Ukovich: Control Eng. Pract. **70** (2018) 86. https://doi.org/10.1016/j.conengprac.2017.10.001

19 T. Nishi, S. Matsushita, T. Hisano, and M. Morikawa: J. Adv. Mech. Des. Syst. Manuf. **8** (2014) JAMDSM0067. https://doi.org/10.1299/jamdsm.2014jamdsm0067

20 C. Duan, J. Feng, and H. Chang: IEEE Access **9** (2021) 49844. https://doi.org/10.1109/ACCESS.2021.3068972

21 G. Ahmed, T. Sheltami, M. Deriche, and A. Yasar: Ad Hoc Networks **118** (2021) 102519. https://doi.org/10.1016/j.adhoc.2021.102519

22 L. Hu, W. Naeem, E. Rajabally, G. Watson, T. Mills, Z. Bhuiyan, C. Raeburn, I. Salter, and C. Pekcan: IEEE Trans. Intell. Transp. Syst. **21** (2020) 1167. https://doi.org/10.1109/TITS.2019.2902927

23 C. Miao, G. Chen, C. Yan, and Y. Wu: Comput. Ind. Eng. **156** (2021) 107230. https://doi.org/10.1016/j.cie.2021.107230

24 M. Dorigo and L. M. Gambardella: IEEE Trans. Evol. Comput. **1** (1997) 53. https://doi.org/10.1109/4235.585892

25 B. Zhang, D. Chen, and W. Zhao: Comput. Chem. Eng. **29** (2005) 2078. https://doi.org/10.1016/j.compchemeng.2005.05.020

26 L. Wu, X. Tian, H. Wang, Q. Liu, and W. Xiao: Assembly Autom. **39** (2019) 45. https://doi.org/10.1108/AA-02-2018-022.

About the Authors

Rong Liu received her B.S., M.S., and Ph.D. degrees from Huzhong Agricultural University, Guangdong University of Technology, and Jinan University, China, in 2003, 2006, and 2011, respectively. She was a visiting scholar at Sun Yat-Sen University, China, from 2013 to 2014 and at California State University, Chico, USA, from 2018 to 2019. She then joined the College of Port and Shipping Management, Guangzhou Maritime University, where she is currently an associate professor. Her research interests include logistics system engineering and business administration. (121712170@qq.com)