International Journal of Information Technologies and Systems Approach
Volume 17 • Issue 1

Intelligent Logistics Vehicle Path Planning Using Fused Optimization Ant Colony Algorithm With Grid

Liyang Chu, Liaoning Institute of Science and Technology, China*
Haifeng Guo, Liaoning Institute of Science and Technology, China
Qingshi Meng, Shenyang Aerospace University, China

ABSTRACT

Aiming at the problem that the environmental planning path of intelligent logistics vehicles on urban roads and remote mountainous areas cannot fit the actual driving scene well. This study creates the algorithm model that combines an ant colony algorithm with a dynamic window algorithm and a Bessel smoothing strategy. Compared to the traditional colony algorithm with the same parameters, this fusion algorithm makes the path smoother by 72.2% when used on an urban highway. It also follows the right-hand rule for right-turn intersections. When the vehicle's height is determined in a mountain environment, this fusion algorithm reduces the driving's mean square deviation of height by 81.5% and shortens the path distance by 38.7%. The fusion algorithm can plan the target path of intelligent logistics vehicles and has the characteristics of scenarios available, multiple factors coordinated, and driving safety. It has provided certain research value and ideas for the digital transformation of the logistics industry.

KEYWORDS

Algorithm Fusion, Ant Colony Algorithm, Improved Grid Map, Intelligent Logistics

INTRODUCTION

Intelligent logistics vehicles can deliver goods in the transportation system of the intelligent logistics ecology. In large quantities of goods transported, the implementation of land logistics vehicles' distribution efficacy is higher (Lo Storto & Evangelista, 2023). Therefore, the path-planning problem of intelligent logistics vehicles occupies a pivotal position in intelligent logistics ecological transportation. Using path-planning algorithms for start-to-end path optimization of vehicles has also become a hot topic in intelligent logistics. The Dijkstra algorithm (Sundarraj et al., 2023) and the A-Star algorithm (Zhang et al., 2023) are classical and feasible optimal path-planning algorithms. They can find the shortest distance between two points with no obstacles. However, they have partial time complexity and memory consumption. Later, by applying more efficient approximate algorithms,

DOI: 10.4018/IJITSA.342613 *Corresponding Author

such as the ant colony optimization (ACO) algorithms (W. Wang et al., 2023), the genetic algorithm (GA) (Xu & Li, 2023), the annealing simulation algorithm (SA) (Venkateswaran et al., 2022), and so on, the NP (nondeterministic polynomial time) problems with complex paths are solved, which increases the algorithms' real-time resource utilization rates. The ant colony algorithm is a potent technique for conducting global searches and can converge to globally optimal solutions, even in highly complex search spaces. When dealing with a large-scale problem, it can effectively utilize the approximate exact solution approach to achieve an extremely close to optimal solution while staying within reasonable time and computing constraints. Since the ant colony algorithm is designed in a distributed and discrete form and is a simulation of the cooperative behavior of ants, the travel path of each ant in the team represents a potential solution (Yi et al., 2020). Moreover, each ant can influence the path choice of other ants by releasing pheromones. This distributed and discrete computing model is suitable for dealing with multiple traveling salesman problems (MTSP) to find optimal solutions (Changdar et al., 2023). It is precisely because the design of the ant colony algorithm is discrete that it is more suitable to combine the path environments of logistics vehicles with the grid method of spatial environment unitization. This method simulates the smallest moving unit of logistics vehicles in path planning in a discrete environment. The two-dimensional region created by this division is called a grid. The grid method is used as an auxiliary scheme to implement obstacle avoidance so that the research object can bypass obstacles and reach the target as soon as possible (Han, 2021; Mathiyalagan, 2010; Ajeil et al., 2020).

Sometimes, the ant colony algorithm's positive feedback mechanism has the potential to cause issues like slow convergence or falling into local optimization (Dorigo & Socha, 2018). The pheromone and heuristic rules of the ant colony algorithms are redefined in the improved ant colony algorithm proposed by Zhang and Zeng (as cited in Zhao et al., 2016; Zeng et al., 2016). However, the initial path planning will remain confused and fall into the local optimal solution. After that, although the EH-ACO algorithm of Gao et al. (2020) can reduce the probability of falling into local optimization and improve the efficiency of complex grid maps' search, there will still be a distance deviation between the grid maps and the driving paths in the actual environments. The ant colony algorithm also has some limitations in avoiding dynamic and static obstacles in the driving environment with random interventions. It is hard to meet the actual environmental driving demands of logistics vehicles only by relying on the traditional ant colony algorithm combined with a grid map, which lacks natural environmental factors and has the shortcomings of avoiding obstacles.

To counter these issues, only combining the grid maps to simulate the actual scenes and the changing obstacles in the local environments can make the optimal paths more in line with the requirements of logistics vehicles. For example, Che et al. (2020) realized the point-to-point path planning of an underwater autonomous vehicle's seabed environment by optimizing the ant colony algorithm. In addition, M. Li et al. (2023) planned the ship's shipping path in the maritime navigation scene using the APF-ACO algorithm. On the other hand, Jones et al. (2023) expounded on uncrewed aerial vehicle research on path planning, obstacle perception, and obstacle avoidance under the improved ant colony algorithm. However, the research on the path planning of land logistics vehicles in the intelligent logistics ecology system is not rich.

So this paper combines a multi-parameter optimization ant colony algorithm under improved versions of the different scenes' grid maps with the dynamic window algorithm (DWA), adds the Bessel smoothing strategy to optimize the paths, and proposes a fused optimized ant colony algorithm, Opt Ant-DWA (Jiang et al., 2023; F. Li et al., 2022). This new technique, Opt Ant-DWA, can match the improved grid maps, find the globally optimal paths, adapt to changes in the local environments, and easily avoid moving and fixed obstacles in dynamic scenes. The Opt Ant-DWA algorithm balances global and local optimization while simulating the driving process of logistics vehicles and can calculate a better planning path. The algorithm has more general flexibility by managing and coordinating the parameters of the fused optimization ant colony algorithm. It provides a new idea and method for transportation path planning in intelligent logistics.

IMPROVED GRID MAP IN DIFFERENT SCENES

Construction of Grid Map

Map construction is a key step in path planning, providing the algorithm's driving scene information. Commonly used map construction methods include the grid method, the geometry method, the topology method, and so on. Among them, the grid method divides the space into equal, discrete grid areas. The design of the ant colony algorithm is also distributed discretely, so it is easy to process the combination solutions of the ant colony algorithm and grid method on the computer. The grid method has higher adaptability and expansibility than other map construction methods. So, the study establishes height grids that adapt to the improved ACO algorithm, which can be crossed using the vehicle's stride characteristic.

The overall flatness of urban highways is good, so they are usually described as tiled grids. In the section where the logistics vehicle can pass, there will be obstacles that the vehicle cannot cross or touch, and the vehicle will also encounter other cars or obstacles that are randomly involved in relative movement and static. At the same time, the driving path may also include bends or intersections that require specific turning directions. So a 20*20 sampling grid map of the urban highway environment is established in Figure 1. The black grids represent the initially static obstacles, the white grids represent the flat paths, and the brown grid represents the restricted right turn point. The yellow square represents the movable car or obstacle, and the gray grids represent the obstacles with random interventions in the later stage. To simulate the change in the map's heights, the absolute heights from the lowest horizontal reference are used as a reference. The depths in green indicate the heights of the grids. The deeper they are, the higher they are, until the vehicle cannot cross them.

Moreover, creating a grid map model must adhere to several constraints in a mountainous area with more complex terrain and surroundings. Aside from increasing the number of samples to improve the planning path's accuracy, it also requires more environmental details. In addition to describing the common steep slopes, absolute obstacles, and gullies on the map, there is a class of relative obstacles with three-dimensional heights with bridge arches or tunnels so the logistics vehicle can pass by. As shown in Figure 2, a 30x30 mountain sampling grid map is simulated. The layered grid construction method (Yu et al., 2017) can construct non-obstacle grids like bridge arches or tunnels. The red color added here means that the unit grids are the relative obstacle grids with three-dimensional heights. The suitable-height logistics vehicle may choose to pass straight through them. Here, the more profound red grid means the vehicle that can be crossed is short in height.

Figure 1. 20x20 Sampling Grid Map of Urban Highway

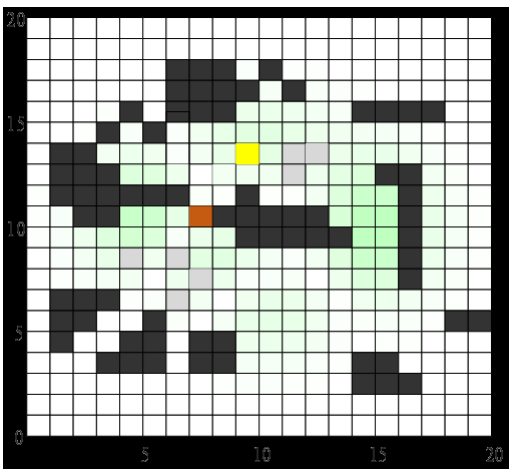
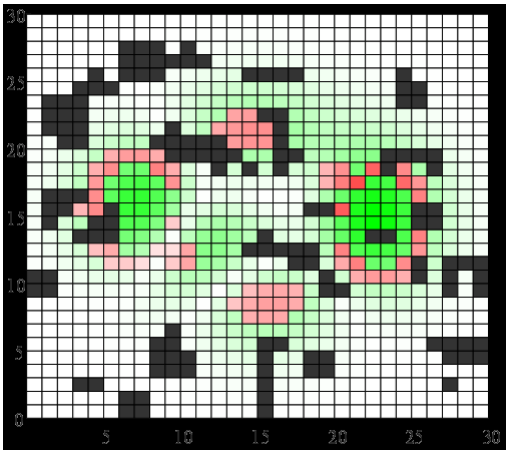


Figure 2. 30x30 Sampling Grid Map of Mountain



Grid Rules Based on the Optimizing Ant Colony Algorithm

During the process of map information storage by the ant colony algorithm, the movements of the research object can be regarded as a gradual accumulation process of transferring from the current grid's center to the next grid's center. The transferable direction vectors will affect the research object's transfer flexibility and the optimal path-planning solution. The most common grid transfer vector is eight (Fu et al., 2022), including four straight and four oblique transfer directions, as shown by the black arrows in Figure 3. It also adds eight new oblique transfer directions, indicated by the red arrows, to improve the accuracy of completing the best path planning. The vehicle should move without touching the obstacle grids, including bumping the vertices of the obstacle grids. Examples of the transfer mesh selection rule are shown in Figure 4.

OPTIMAL ANT COLONY-DWA ALGORITHM

Typical Ant Colony Algorithm

The ant colony algorithm's path-planning behaviors come from a bionic search mode with positive feedback. The team's pheromones and the particular heuristic variables of the current scene determine the likelihood of choosing the next grid as a probabilistic method, as shown in Formula 1.

Figure 3. Grid Vectorable

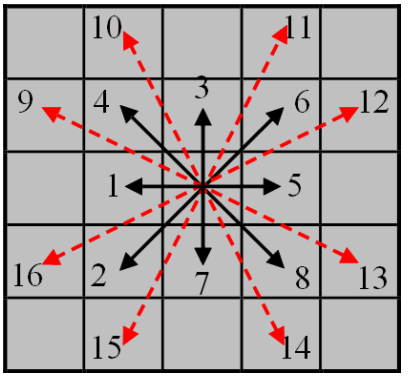
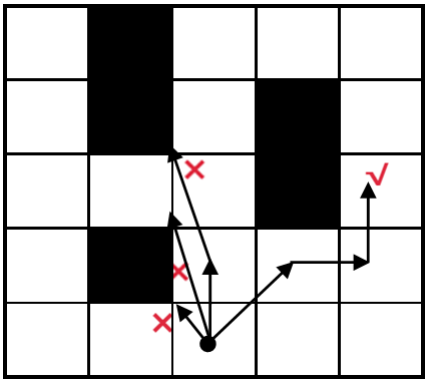


Figure 4. Examples of Travelable Rule



$$P_{i,j}^{m}\left(t\right)=\begin{cases}\dfrac{\left[\left[\tau_{i,j}^{m}\left(t\right)\right]^{\alpha}\left[\dfrac{1}{d}_{i,j}^{m}\left(t\right)\right]^{\beta}\right]}{\sum_{s\subset allowed_{i+1}}\left[\tau_{i,j}^{m}\left(t\right)\right]^{\alpha}\left[\dfrac{1}{d}_{i,j}^{m}\left(t\right)\right]^{\beta}}, j\in allowed_{i+1}\\ 0, j\notin allowed_{i+1}\end{cases} \tag{1}$$

In Formula 1, the current position of the numbered m in the t iteration is represented by i, j represents the next grid number that i may transfer, $d_{i,j}$ represents the length of road section to be transferred, $\tau_{i,j}^{m}\left(t\right)$ represents the pheromone of m on $d_{i,j}$, and $\dfrac{1}{d}_{i,j}^{m}\left(t\right)$ represents the heuristic function of m on the section to be rotated and is inversely proportional to $d_{i,j}$. α and β are the important weights of the pheromone and the heuristic function, respectively, and $s\subset allowed_{i+1}$ is the collection of feasible neighborhood grids.

Setting Initial Pheromone

Suppose each grid is allocated a unified initial pheromone to guide the adjustment of travel direction. In that case, the algorithm will fall into a locally optimal solution, or the convergence time will be excessively long. Combined with the height grid map design in the actual scene, the boundary grids have the smallest pheromone's initial value C. In contrast, for the nonboundary grids, the effect of the obstacle influence function $f\left(i\right)$ and the obstacle height H on the pheromone initial values $\tau_{i,j}^{m}\left(0\right)$ of the grids in different directions to be transferred should be considered together. Therefore, the initial pheromone calculation relationship is established as shown in Formula 2.

$$\begin{cases} \tau_{i,j}^{m}\left(0\right) = C, i \in boundary\ grid \\ \tau_{i,j}^{m}\left(0\right) = C + \left[1 / f\left(i\right)\right)] + \sigma \cdot H / H_{Max}, i \in non - boundary\ grid \\ f\left(i\right) = num\left(C_U\left(allowed_{i+1}\right)\right), i \in non - boundary\ grid \end{cases} \tag{2}$$

Here C is the minimum pheromone initial value, $f\left(i\right)$ is the barrier influence function in different directions to be transferred, H is the actual height of the transferable grid, H_{Max} is the maximum height of the grids, σ is the pheromone height harmonic parameter, and the greater it is, the lower the probability of selecting the high grids as the transfer grids. $allowed_{i+1}$ are the grids that alow transfer, and $num\left(C_U\left(allowed_{i+1}\right)\right)$ is the number of complements of the transferable grids. When i is a nonboundary grid, the value of $num\left(C_U\left(allowed_{i+1}\right)\right)$ is larger, and the current grid's neighborhood obstacle grids have a greater influence on the initial pheromone of the i. At the same time, non-absolute obstacle grids with different height ratios H / H_{Max} have different effects on the initial pheromones of nonboundary grids. According to the actual situations of grids, the initial pheromone values can be allocated differently to guide driving and avoid obstacles quickly and safely, accelerating the algorithm's convergence.

Setting Multi-Parameter Pheromone and the Heuristic Function

In the updating process of the heuristic function and accumulated pheromone in the t+1 iteration of $d_{i,j}$, it is no longer a simple problem that only considers the distance between two points from start to end, but chooses the distance, path tortuosity, and road condition turbulence as the comprehensive parameters that affect the grid transition probability $P_{i,j}^{m}\left(t\right)$. Pheromones have not been accumulated to a specific scale in the initial stage of path planning. The reference of heuristic functions to path selection can play a role in preventing falling into local optimization and error selection. Formula 3 is the heuristic function relationship with comprehensive parameters.

$$\eta_{i,j}^{m}\left(t\right) = l^{m}\left(i,j,q\right) + r_{g,i,j}^{m}\left(t\right) + h^{m}\left(i,j\right) \tag{3}$$

Here, $l^{m}\left(i,j,q\right)$ is the distance factor in the comprehensive parameters, which is the modified Euclidean distance from the center of the current grid i's neighborhood grid j to the center of the final target grid (Ojha et al., 2014; Baydogmus, 2023), and q represents the final target grid, $r_{g,i,j}^{m}\left(t\right)$ is the path tortuosity factor in the comprehensive parameters, which represents the importance of straight travel. $r_{g,i,j}^{m}\left(t\right)$ is assigned different comprehensive factor contribution values based on whether the transfer direction formed by the current grid i and the previous traveling grid g is equal to the transfer direction formed by the current grid i to the next optional traveling grid j. $h^{m}\left(i,j\right)$ is the road bumpiness factor in the comprehensive parameters and is the corrected height value of the current grid i and the neighboring grid j in the height grid map relative to the lowest reference horizontal line.

The ant-cycle model is a practical global information fusion model that can traverse all nodes in the solution and update the global pheromones after one cycle (Tan et al, 2023; Scianna, 2024). The corresponding pheromone update formula is shown in Formulas 4, 5, and 6. The single-step path planned in a single iteration is affected by the elements of distance, path tortuosity, and road bumpiness.

$$\tau_{i,j}\left(t+1\right) = \rho\tau_{i,j}\left(t\right) + \sum_{m=1}^{M}\Delta\tau_{i,j}^{m}\left(t\right) \tag{4}$$

$$\Delta\tau_{i,j}^{m}\left(t\right)=\begin{cases}Q/S_{m}\left(t\right)\\0,Others\end{cases} \tag{5}$$

$$S_{m}\left(t\right)=aL_{m}\left(t\right)+bR_{m}\left(t\right)+cH_{m}\left(t\right) \tag{6}$$

ρ depicts the pheromone volatilization residue, which can avoid the local optimal solution in a restricted fashion. M denotes the total number of iterations, Q is the pheromone constant, and $S_{m}\left(t\right)$ is the total inverse proportional coefficient of pheromones. $L_{m}\left(t\right)$ represents the path length of this plan, $R_{m}\left(t\right)$ represents the total number of turns required to reach the target grid, and $H_{m}\left(t\right)$ represents the height- mean square deviation of all grids related to this plan. a,b,c are the adjustment coefficients of each factor. Too high or too low a pheromone will cause the algorithm to enter the incorrect local optimization, so the upper and lower limits of the pheromone are set, as shown in Formula 7.

$$\tau_{i,j}\left(t\right)=\begin{cases}\tau_{i,j}\left(t\right),\tau_{i,j}\left(t\right)\in\left(\tau_{min},\tau_{max}\right)\\\tau_{min},\tau_{i,j}\left(t\right)\le\tau_{min}\\\tau_{max},\tau_{i,j}\left(t\right)\ge\tau_{max}\end{cases} \tag{7}$$

τ_{min},τ_{max} represent the minimum and maximum pheromone values assigned based on the actual driving scenes.

Setting Dead-End Alert and Right-Hand Turning Principle

When judging the stored grids' traveling distance and direction information, it is also necessary to consider whether the current grid is a dead-end grid. First, it is essential to clarify what a dead-end grid is. The neighborhood grids corresponding to all directions in the existing grid's 16 vector directions are either an absolute obstacle grid that cannot travel or will be rubbed against these grids while traveling. Then, the present grid is defined as a dead-end grid. When the former grid is determined to be a dead-end grid, the grid is not included in the path computation and storage of the optimal path-planning scheme. The grid becomes a pseudo-obstacle, like the obstacle grid, which can no longer be passed like the non-obstacle grid. At this point, the current iteration interrupts, and all subsequent iterations are broadcast. After this, walking into the dead-end grid again is forbidden to improve calculation speed and optimize the path-planning scheme. Formula 8 is a record of the traveling distance of the logistics vehicle at every step.

$$D\left(i,j\right)=\begin{cases}l,G\left(i\right)=0\cap j\in int\left[1,9\right)\cap mod\left(j,2\right)=1\cap G\left(\hat{i}\right)=0\\\sqrt{2}l,G\left(i\right)=0\cap j\in int\left[1,9\right)\cap mod\left(j,2\right)=0\cap G\left(\acute{i}\right)+G\left(i^{*}\right)=0\\\sqrt{5}l,G\left(i\right)=0\cap j\in int\left[9,16\right]\cap G\left(\hat{i}\right)+G\left(\acute{i}\right)=0\end{cases} \tag{8}$$

Set the unit length of the grid as l, $G\left(i\right)=0$ as the current non-obstacle grid, and when turning in the eight-vector directions $j \in int\left[1,9\right)$, use $mod\left(j,2\right)$ to take the cofunction to judge the parity of grid numbers and use $G\left(\hat{i}\right)$ and $G\left(\acute{i}\right)$ to represent the straight grid and oblique grid of the traveling direction, respectively. $G\left(i^{*}\right)$ are two adjacent vertical grids of oblique grid $G\left(\acute{i}\right)$. When the steering grid is in the newly added eight-vector directions $j \in int\left[9,16\right]$, it is necessary to ensure that both $G\left(\hat{i}\right)$ and $G\left(\acute{i}\right)$ in the direction $G\left(i\right)$ travel are not absolute obstacles.

The ant colony algorithm's right-hand turning principle is set based on the driving rule stipulated by urban traffic. When facing intersections and bends, the vehicle must turn right in actual urban road scenes. The grid on the right side of the current grid is selected as the next planning grid point. This setting ensures smooth and efficient traffic conditions and is more suitable for driving traffic rules on urban roads. It should be noted here that the principle of right-hand turning is not absolute. For example, when the target grid is adjacent to the current grid, adopting the right-hand turning principle may cause the logistics vehicle to fail to reach the target grid as soon as possible. By judging whether the grid to be transferred is the target grid and giving the target grid absolute priority, self-locking in the steering process can be avoided.

Fusing the Optimizing Ant Colony Algorithm

The driving environment for logistics vehicles is complicated and constantly changing. When dynamic or random obstacles appear, the DWA can adjust the speed and turn in real-time according to the current state and performance limitations of the vehicle to avoid all kinds of obstacles. However, in the trajectory planning of the DWA, the trajectory prediction is on the X-Y axis, so the DWA alone cannot adapt to the path planning on the improved grid with height in the actual driving scenes of the logistics vehicle. Using the optimized ant colony algorithm combined with the DWA and the Bessel curve can make up for the driving restriction problem of the height-grid map model and optimize the path with the various obstacles.

When dynamic or random obstacles appear, the DWA can adjust the speed and angular velocity, which are calculated based on the kinematic characteristics of the logistics vehicle. Then, according to the current state and performance limitations of the vehicle, these obstacles can be avoided in real-time, as shown in Formula 9, which represents the prediction model of logistics vehicle trajectory.

$$\begin{cases} \Delta x = v \cdot \Delta t \cdot cos\theta \\ \Delta y = v \cdot \Delta t \cdot sin\theta \\ x_{t+1} = x_t + \Delta x \\ y_{t+1} = y_t + \Delta y \\ \theta_{t+1} = \theta_t + \omega \cdot \Delta t \end{cases} \tag{9}$$

In Formula 9, Δx and Δy are the position changes passing through Δt on the grid map, x_{t+1}, y_{t+1} are the new node positions after Δt, and θ_{t+1} is the direction angle of the research object relative to the target grid after Δt time. When combined with the above-mentioned trajectory prediction model, the new node information after the next period Δt can be obtained. The speed is limited by the movement conditions of the logistics vehicle itself, and the setting of the logistics

vehicle itself will limit a practical speed window. According to the actual situation, it is necessary to combine the actual linear velocity, angular velocity, acceleration, and angular acceleration of the logistics vehicle engine to predict the V_d of the logistics vehicle's acceleration and deceleration window, as shown in Formula 10.

$$V_d = \left\{\left(v,w\right) \mid v \in \left[v_t - v_a\Delta t, v_t + v_a\Delta t, w \in \left[w_t - w_a\Delta t, w_t + w_a\Delta t\right]\right\}\right. \tag{10}$$

v_t and w_t are the current speed and angular speed, and the maximum linear acceleration or deceleration v_a and the maximum angular acceleration or deceleration w_a limit the acceleration and deceleration window V_d of the logistics vehicle.

In the face of new movable obstacles and random stationary obstacles, it is necessary to introduce obstacle safety radius $R\left(v,w\right)$ as shown in Formula 11. In the DWA, obstacle safety radius measures the collision possibility between the vehicle and obstacles.

$$\begin{cases} v \le \sqrt{R\left(v,w\right)v_a} \\ w \le \sqrt{R\left(v,w\right)w_a} \end{cases} \tag{11}$$

When the stochastic dynamic or static obstacles are located in the obstacle safety radius area $R\left(v,w\right)$ of the vehicle, it indicates that there is a potential collision risk. At this time, the algorithm will take corresponding obstacle avoidance measures and adjust v and w to avoid collisions with obstacles.

Following the establishment of the speed sampling window $G\left(v,w\right)$, the nodes to be moved can be evaluated and screened using the evaluation function $G\left(v,w\right)$, and the final planning path scheme can be obtained through multiple predictions. Formula 12 depicts the evaluation function $G\left(v,w\right)$.

$$G\left(v,w\right) = \sigma[w_1 heading\left(v,\omega\right) + w_2 obdist\left(v,\omega\right) + w_3 velocity\left(v,\omega\right)] \tag{12}$$

$heading\left(v,\omega\right)$ is used to calculate the direction angle deviation of the current node towards the target node, $obdist\left(v,\omega\right)$ is an obstacle avoidance function, which makes the node more inclined to choose the trajectory far away from obstacles, and $velocity\left(v,\omega\right)$ is used to evaluate the linear speed of logistics vehicle and encourage it to pass quickly. w_1, w_2 and w_3 are the weights of the three sub-evaluation functions. σ represents normalization treatment, which is used to avoid the imbalance of the evaluation system caused by the excessive influence of a certain evaluation factor (Ji et al., 2021). Following that, the inverse interpolation smoothing of the node sequence for the path of the Opt Ant-DWA algorithm is performed to obtain a more accurate and gentle final optimal planning path.

The third-order Bessel curve has good smoothness, adaptability, and simplicity, which can deal with the sharp points in the algorithm's path, improve the final path's smoothness, and shorten the driving path (Z. Wang et al., 2018). When the improved ant colony algorithm, the DWA, and the Bessel curve are used together, the vehicle no longer drives in the fixed center-line direction of the grids. This fixes the grid map model's problem with restrictions on driving. The mathematical expression for the Bessel curve is Formula 13.

$$C\left(x\right)=\sum_{i=0}^{n}P_{i}B_{i,k}\left(x\right),x\in\left[0,1\right] \tag{13}$$

$C\left(x\right)$ is the motion control point on the bessel curve, P_i is the position point, k is the order, and $B_{i,k}\left(x\right)$ is the bessel parameter. The cubic Bessel curve has k=3, and the mathematical expression of $B_{i,k}\left(x\right)$ in the Bessel curve is as Formula 14 shows.

$$B_{i,3}\left(x\right)=\frac{1}{k!}\sum_{j=0}^{k-i}\left(-1\right)^{j}C_{k+1}^{j}\left(x+k-i-j\right)^{k} \tag{14}$$

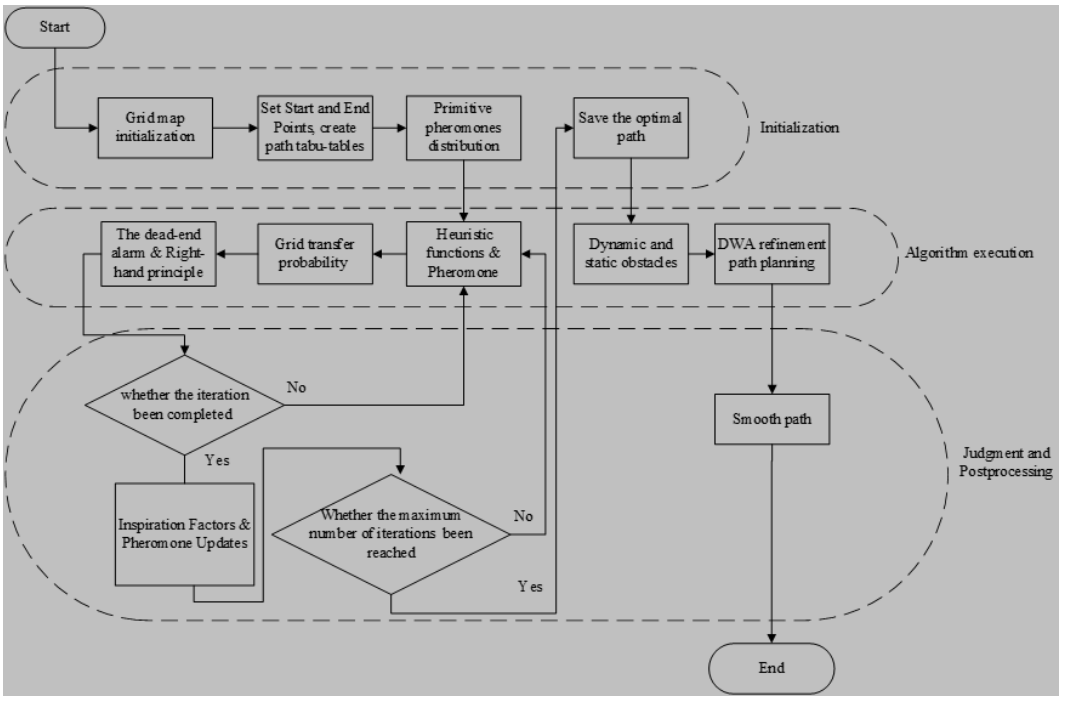
Here $i=0,1,2,$ and $C_{k+1}^{j}=\frac{\left(n+1\right)!}{j!\left(n+1-j\right)!}$ is related to statistics.

Figure 5 shows the algorithm flow chart of the fused optimization ant colony algorithm, the Opt Ant-DWA algorithm.

Urban Highway Path-Planning Simulation

As the most critical parameters in the transition probability of the ant colony algorithm α and β, if α is large and β is small, the accumulation of pheromones in the initial stage of driving is insufficient, and there is no global reference. The path shown in the plan will be very chaotic. But when β is too large and α is too small, the local pheromones will be given too much attention, and the algorithm

Figure 5. Flowchart of the Opt Ant-DWA Algorithm



will easily fall into nonconvergent local optimization. The trend of the best parameter combination of the pheromone accumulation weight α and the heuristic factor weight β changes when the dead-end alert grids in the driving path and the inverse ratio coefficient $S_m(t)$ of comprehensive factors do, too. This can be seen in Figure 6. The figure shows that when $\alpha = 1$ and $\beta = 7$, the combination of pheromone and heuristic function weight is a harmonious optimal solution suitable for the current urban highway environment.

In testing the performance of the Opt Ant-DWA algorithm under the urban highway environment grid, the logistics vehicle needs to have high smoothness when facing relatively flat urban roads. The logistics vehicle should drive straight on flat roads as much as possible to reduce the possibility of turning other than the specified turning points. Therefore, increasing the weight ratio of path tortuosity in comprehensive influencing factors will reduce unnecessary turnings. In the smoothing function $r^m_{g,i,j}(t)$, $\mu\%$ represents the straight-ahead importance of the smoothing function $r^m_{g,i,j}(t)$, which can positively impact the value of the smoothing functions. At the same time, in the face of forced right-hand turning points, the driving path of the logistics vehicle needs to be reasonably planned on the premise of meeting the corresponding traffic regulations so that driving can not only avoid dynamic and static obstacles safely but also take into account the actual driving environment and the movement characteristics of the logistics vehicle themselves.

The optimal driving scheme is determined by setting the internal parameters of the optimized ant colony algorithm. Tables 1 and 2 show the typical initial parameters and smoothness comparison parameters under the urban road environment grid map.

Setting these parameters can test the generation of optimal planning paths under different path smoothness in logistics vehicles. Moreover, it can verify the correct turn principle and the dead-end alert. As shown in Figure 7 (a), (b), and (c), the actual routes of the map, the optimal path length of each iteration, and the turning times of each iteration are indicated, respectively. Among them, black represents the relevant results of the traditional ant colony algorithm, and blue and red represent the relevant test results under the weights of $\mu_1\%$ and $\mu_2\%$, respectively.

Figure 6. β / α **Curve**

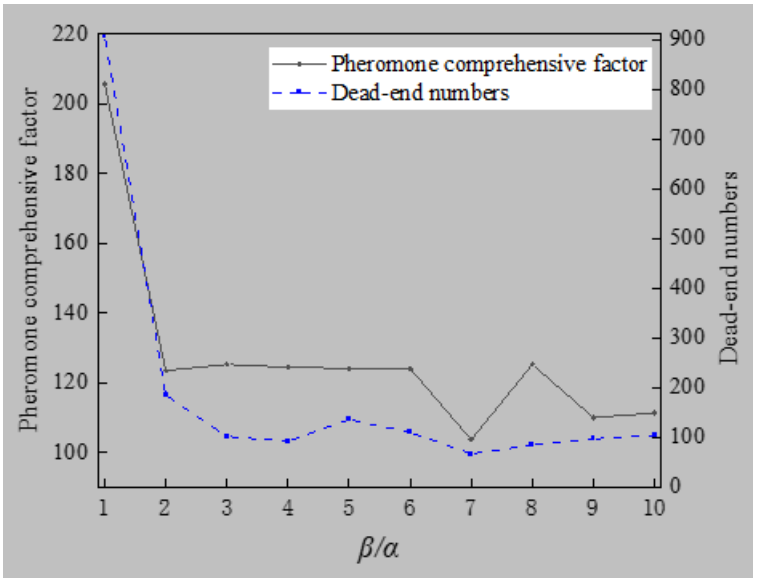


Table 1. Parameter Table of Opt Ant-DWA Algorithm

Parameter	Value	Parameter	Value
Iterative global times M / (time)	30	Maximum linear velocity v_{max} /(m/s)	2.0
Number of logistics vehicles *W*/(*unit*)	50	Maximum angular velocity w_{max} /(rad/s)	20
Minimum pheromone initial value *C*	20	Maximum linear acceleration v_a /(m/s^2)	0.2
Pheromone constant Q	100	Maximum angular acceleration w_a /(rad/s^2)	50
Volatilization residue of pheromone ρ	0.3	Linear velocity change rate v_r /(m/s)	0.01
Pheromone height harmonic parameter σ	0.125	Angular velocity change rate w_r /(rad/s)	1
Maximum pheromone τ_{max}	40	Unit time Δt /(s)	0.1
Minimum pheromone τ_{min}	10	Obstacle safety distance Δd/(m)	0.6

Table 2. Parameter Table of Reducing Turning Times

Parameter	Value
Adjustment coefficient of path length a	1
Adjustment coefficient of turning times b	100
Adjustment coefficient of mean square deviation of driving height c	1
Weight of straight-ahead $\mu_1\%$	75
Weight of straight-ahead $\mu_2\%$	50

The curve's stationary values' turning points in Figure 7 are selected as the stable points for iteration times. The above data summary in Table 3 is as follows.

The most intuitive comparison between the traditional ant colony algorithm's planning path in Figure 7(a) and the optimal iterative paths with two different straight-ahead weights reveals that conventional path planning does not take into account the combination of path planning and the actual driving scene in terms of the number of turns, collisions with obstacles, and choice of driving direction at the right turn. According to the examination of Figure 7 (b), the path planning with a smooth weight of $\mu_1\% = 75\%$ in the optimized ant colony algorithm section is an extreme case simulation with a high weight. Compared to the path with a weight of $\mu_2\% = 50\%$, the path will avoid driving as much as possible in multi-turns. However, this high smoothness will come at the sacrifice of a particular path length. As seen in Figure 7 (c) and Table 3, after iterative stability, the classic ant colony algorithm turns 18 times with a path length of 22.4. The lengths with weights of $\mu_1\% = 75\%$ and $\mu_2\% = 50\%$

Figure 7. 20x20 Urban Highway Path Planning (a) Optimal Path Planning (b) Optimal Path Length (c) Number of Turns

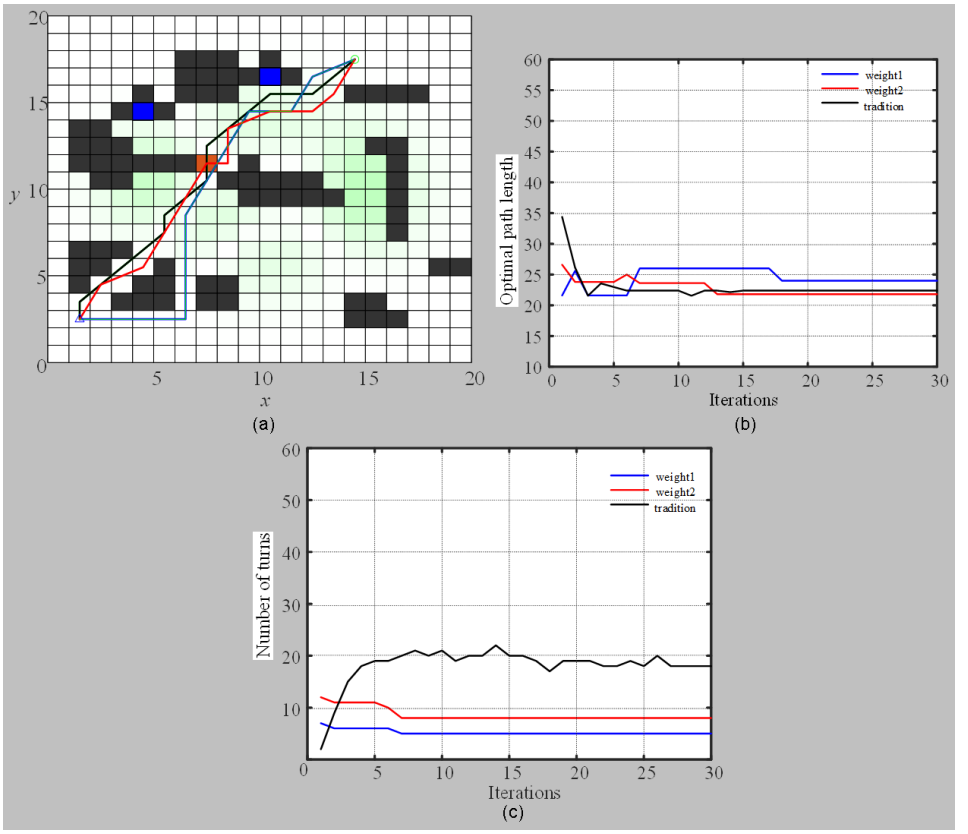


Table 3. Simulation Results of 20x20 Map

Related data	Traditional ACO	μ_1	μ_2
Optimal path length	22.4	24	21.8
Number of turns	18	8	5
Comprehensive index	-	50.8	43.0
Iterative running times	30	30	30
Iterative stabilization times	18	7	7
Program running time/(s)	1.87	1.77	1.27

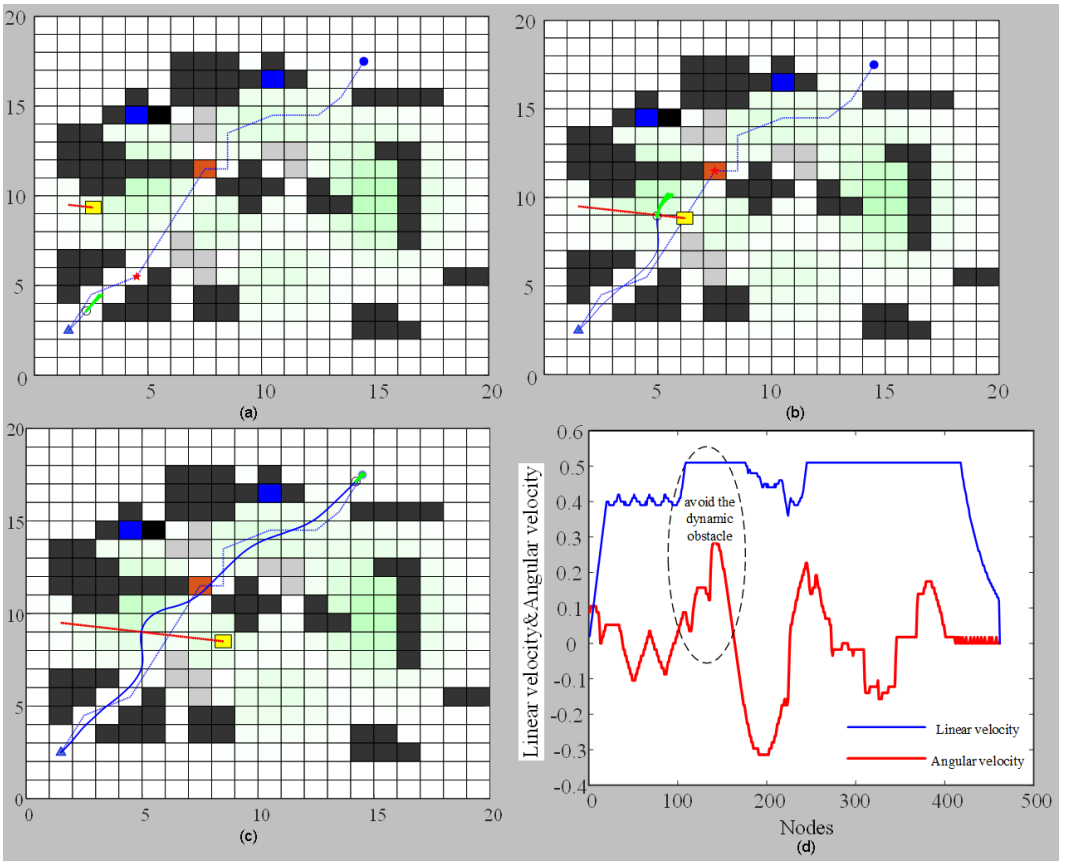
under the 16 vector transfer rule are 24 and 21.8, while the turning times are 5 and 8, respectively. Under $\mu_2\% = 50\%$, the path smoothness can improve by 72.2% when compared to the conventional ant colony algorithm, and it reduces the planned path distance. With the dead-end alert taking effect, grids are transformed into two blue pseudo-obstacle grids successfully that cannot pass through and broadcast to all the next iterations. The running time is also reduced by 32.1%.

Choose the smoothing weight as $\mu_2\% = 50\%$. The path has ideal smoothness with a lower inverse proportional comprehensive factor of 43.0, which represents a higher path-planning score and quality. The Opt Ant-DWA algorithm fuses the optimized ant colony algorithm, the DWA, and the third-order Bessel function. Figure 8 (a), (b), (c), and (d) shows a typical intersecting scene of a dynamic obstacle and a logistics vehicle. This is done to see if the fused Opt Ant-DWA algorithm can effectively avoid dynamic and static obstacles in the later stage. In Figure 8 (a), (b), and (c), the yellow squares represent a relatively moving vehicle to be added here as a simulated dynamic obstacle, and the grey grids represent several new static obstacles that are randomly added.

Take the crossing scene of two cars in traffic as an example. The dynamic change charts in Figure 8 (a), (b), and (c) show that the Opt Ant-DWA algorithm can reasonably avoid obstacles beyond the safety obstacle distance when there are both moving and still obstacles in the scene. Figure 8 (d) shows how a logistics vehicle's line and angular velocity change over time when avoiding different obstacles. When the angular velocity changes from negative to positive, the logistics vehicle starts to avoid the moving obstacle.

It has been found that the Opt Ant-DWA algorithm can improve the solution for the best path, which results in a shorter path length and smoother turning angles. The height grid maps calculate the optimal path solutions, which can work efficiently on urban highways.

Figure 8. Opt Ant-DWA Algorithm Obstacle Avoidance (a) Initial State (b) Obstacles Avoidance State (c) Termination State (d) Linear Velocity and Angular Velocity



Mountain Environment Path-Planning Simulation

Verify and test the Opt Ant-DWA algorithm's performance using the sampled raster map of the mountain environment 30x30 in Chapter 2.1. Before the test, because the scene has been changed, the weight ratio β / α needs to be readjusted, and the corresponding trend change of β / α is shown in Figure 9. It can be seen from Figure 9 that, by changing $\alpha = 1$ and $\beta = 6$, the combination of pheromone and heuristic function weight is suitable for the mountain grid map scene.

When the driving environment of logistics vehicles shifts from urban highways to mountainous areas with more complicated environmental details, how to reduce the path bumps and pass through non-obstacle grids with three-dimensional height, such as tunnels and bridges, which are common in mountainous areas, should be given greater weight. Table 3 shows the distribution of the heights of logistics vehicles with two different heights and the weights of comprehensive influencing factors.

On a 30x30 multi-layer height grid map, tests are done to see how two logistics vehicles handle path bumps, the dead-end alert, and passing the three-dimensional non-obstacles with go-throughs such as bridges and tunnels. The simulation results are shown in Figure 10 (a), (b), and (c). Among them, black represents the relevant results of the traditional ant colony algorithm, and blue and red represent the relevant test results under the weights of H_{v1} and H_{v2}, respectively.

The curve's stationary values' turning points in Figure 10 are selected as the stable points for iteration times. The above data summary table in Table 5 is as follows.

Through Figure 10 (a) and (b), it is found that when the weight of height mean square deviation is greater than the weight of path length and path smoothness, the planning path of the optimized ant colony algorithm will choose the grid with lower height on the height grid map as the transfer choice, which is in line with the driving strategy in the mountain environment.

In addition to verifying the dead-end alert, Figure 10 (a) also shows that when logistics vehicles with different heights pass through tunnels or bridges, due to the height conditions, compared with large cars with a height of $H_{v1} = 1.45$, the vehicle with a height of $H_{v2} = 0.5$ can successfully pass through five three-dimensional non-obstacle grids. Figure 10 (c) also shows that the improved ant

Figure 9. β / α **Curve**

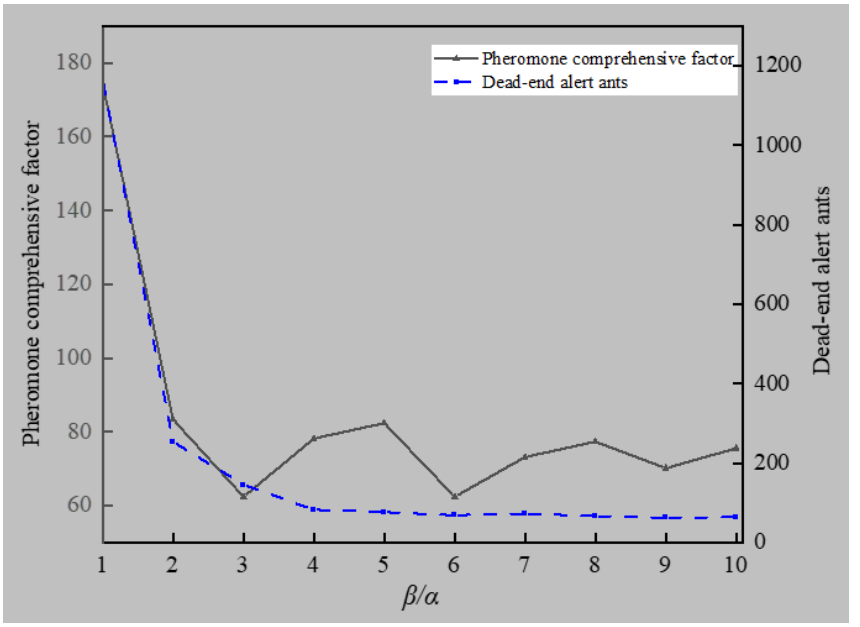
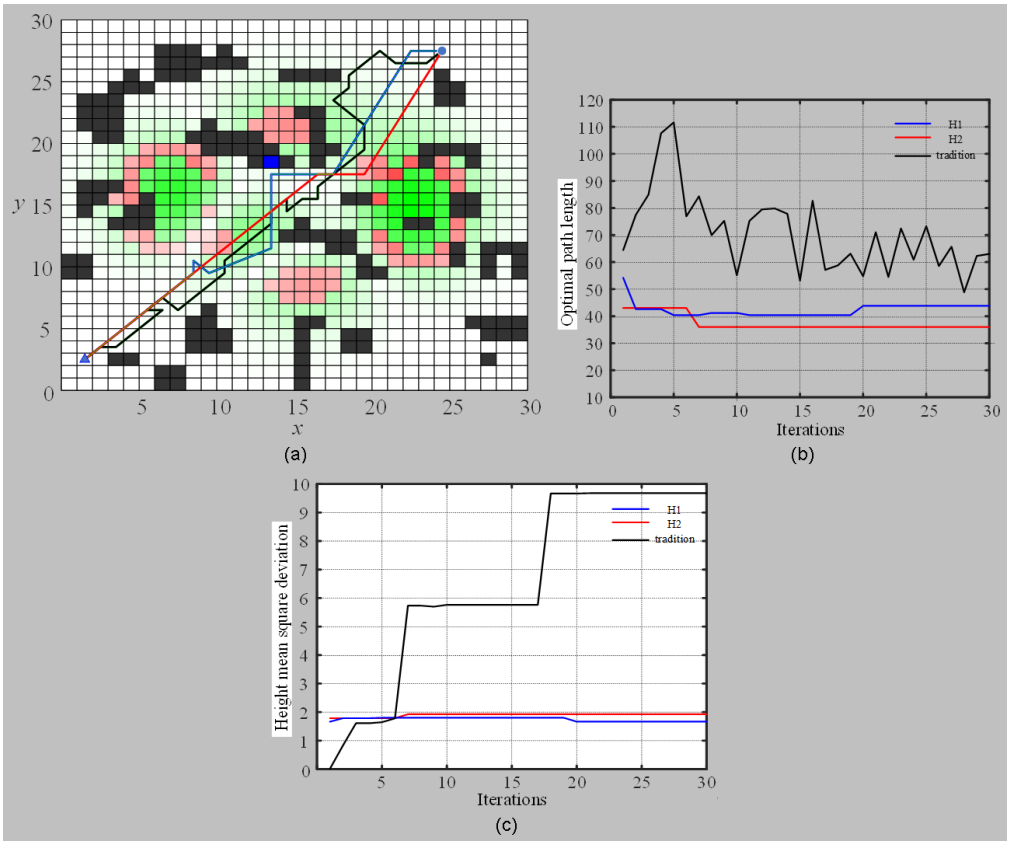


Table 4. Parameter Variation Table of Reducing Height Climbing

Parameter	Value
Adjustment coefficient of path length a	1
Adjustment coefficient of turning times b	5
Adjustment coefficient of mean square deviation of driving height c	15
Vehicle's height H_{v1}	1.45
Vehicle's height H_{v2}	0.5

Figure 10. 30x30 Mountain Environment Path Planning (a) Optimal Path Planning (b) Optimal Path Length (c) Height-Mean Square Deviation



colony algorithm with a larger height variance weight will choose a flatter path as the best driving path to ensure the reliability of logistics vehicles in hilly areas. In Table 5, take H_{v1} as an example, this higher weight effectively lowers the height-mean square deviation by 81.5% compared to the traditional ant colony algorithm, and the H_{v2} planning path can also shorten the path distance by

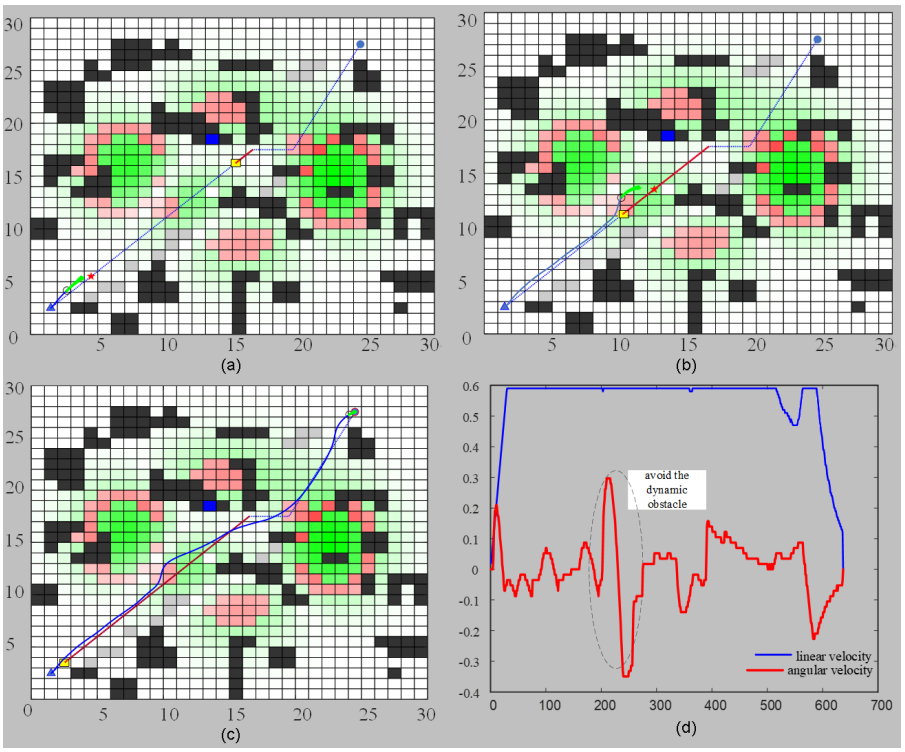
Table 5. Simulation Results of 30x30 Map

Related data	Traditional ACO	H_{v1}	H_{v2}
Optimal path length	60.2	43.4	36.9
Height-Mean square deviation	9.67	1.78	1.93
Comprehensive index	-	73.4	45.8
Iterative running times	30	30	30
Iterative stabilization times	18	20	7
Program running time/(s)	3.37	1.68	1.53

38.7%. Then, choose the logistics vehicle with a height of H_{v2}, fuse the algorithm, and build an encounter scene in addition to the typical traffic scene intersecting. The logistics vehicle's starting position coordinates are (1.5, 2.5), and the dynamic obstacle's starting and end coordinates are (15.5, 16.5), (2.5, 3.5) for the encounter.

Figure 11 (a), (b), and (c) shows changes in dynamic, and (d) shows the change in angular velocity. These indicate that the logistics vehicle under the mountain grid map can cross the tunnel or arch bridge on the map without hitting the moving obstacle in the direction of the encounter.

Figure 11. Opt Ant-DWA Algorithm Dynamic Obstacle Avoidance (a) Initial State (b) Obstacles Avoidance State (c) Termination State (d) Linear Velocity and Angular Velocity



Through the above tests, it has been found that different parameters or different height vehicles in the algorithm will make the algorithm plan different paths to adapt to the environment. The fusion algorithm can align with the essential driving characteristics of logistics vehicles. Vehicles can also show better driving behavior.

TECHNOLOGY PROSPECT

The Opt Ant-DWA algorithm may face the challenge of computing resources and time when dealing with large-scale problems (Yan, 2023; Mao et al., 2022.). Deploying the Opt Ant-DWA algorithm on the distributed platform and using multiple computing nodes to generate data, process data, and save data in parallel can significantly improve the processing ability and efficiency of the algorithm in large-scale optimization problems. When a computing node fails, another can also take over its tasks to enhance the algorithm's robustness.

CONCLUSION

The problem with that the multi-environment planning path of intelligent logistics vehicles on urban roads and remote mountainous areas cannot fit the actual vehicle driving scene well, this study established the corresponding height grid maps. This paper simulates and tests vehicle path planning in static and dynamic mixed-obstacle environments using the Opt Ant-DWA algorithm. The fusion algorithm can effectively plan the target path of intelligent logistics vehicles with high fitness and adjustable path parameters. This study has promoted the development of an intelligent logistics ecosystem by improving the transportation efficiency of intelligent logistics vehicles, reducing costs, reducing energy consumption, and improving traffic safety. It has also provided certain research value and research ideas for the future digital transformation technology of the logistics industry. Finally, how to deepen the combination of theoretical research and experimental testing and how to build a path-planning algorithm combined with parallel distributed platform technology are also a new research challenge and research direction ripe for future investigation.

CONFLICTS OF INTEREST

We wish to confirm that there are no known conflicts of interest associated with this publication and there has been no significant financial support for this work that could have influenced its outcome.

FUNDING STATEMENT

This research was supported by the Basic Scientific Research Project of the Liaoning Provincial Department of Education (JYTMS20231799).

REFERENCES

Ajeil, F. H., Ibraheem, I. K., Azar, A. T., & Humaidi, A. J. (2020). Grid-based mobile robot path planning using aging-based ant colony optimization algorithm in static and dynamic environments. *Sensors (Basel)*, *20*(7), 1880. doi:10.3390/s20071880 PMID:32231091

Baydogmus, G. K. (2023). Solution for TSP/mTSP with an improved parallel clustering and elitist ACO. *Computer Science and Information Systems*, *20*(1), 195–214. doi:10.2298/CSIS220820053B

Changdar, C., Mondal, M., Giri, P. K., Nandi, U., & Pal, R. K. (2023). A two-phase ant colony optimization based approach for single depot multiple travelling salesman problem in type-2 fuzzy environment. *Artificial Intelligence Review*, *56*(2), 965–993. doi:10.1007/s10462-022-10190-9

Che, G., Liu, L., & Yu, Z. (2020). An improved ant colony optimization algorithm based on particle swarm optimization algorithm for path planning of autonomous underwater vehicle. *Journal of Ambient Intelligence and Humanized Computing*, *11*(8), 3349–3354. doi:10.1007/s12652-019-01531-8

Dorigo, M., & Socha, K. (2018). *An introduction to ant colony optimization.* Chapman and Hall/CRC eBooks.

Fu, L., Chen, H., & Gong, W. (2022). Path planning of underwater vehicle based on improved ant colony algorithm. *Zidonghua Yu Yibiao*, (4), 46–50.

Gao, W., Tang, Q., Ye, B., Yang, Y., & Yao, J. (2020). An enhanced heuristic ant colony optimization for mobile robot path planning. *Soft Computing*, *24*(8), 6139–6150. doi:10.1007/s00500-020-04749-3

Han, G. L. (2021, October 5). Automatic parking path planning based on ant colony optimization and the grid method. *Journal of Sensors*, *2021*, 1–10. doi:10.1155/2021/8592558

Ji, X., Feng, S., Han, Q., Yin, H., & Yu, S. C. (2021). Improvement and fusion of A* algorithm and dynamic window approach considering complex environmental information. *Arabian Journal for Science and Engineering*, *46*(8), 7445–7459. doi:10.1007/s13369-021-05445-6

Jiang, Z., Wang, W., Sun, W., & Da, L. (2023). Path planning method for mobile robot based on a hybrid algorithm. *Journal of Intelligent & Robotic Systems*, *109*(3), 47. doi:10.1007/s10846-023-01985-1

Jones, M., Djahel, S., & Welsh, K. (2023). Path-planning for unmanned aerial vehicles with environment complexity considerations: A survey. *ACM Computing Surveys*, *55*(11), 1–39. doi:10.1145/3570723

Li, F. F., Du, Y., & Jia, K. J. (2022). Path planning and smoothing of mobile robot based on improved artificial fish swarm algorithm. *Scientific Reports*, *12*(1), 659. doi:10.1038/s41598-021-04506-y PMID:35027589

Li, M., Li, B., Qi, Z., Li, J., & Wu, J. (2023). Optimized APF-ACO algorithm for ship collision avoidance and path planning. *Journal of Marine Science and Engineering*, *11*(6), 1177. doi:10.3390/jmse11061177

Lo Storto, C., & Evangelista, P. (2023). Infrastructure efficiency, logistics quality and environmental impact of land logistics systems in the EU: A DEA-based dynamic mapping. *Research in Transportation Business & Management*, *46*, 100814. doi:10.1016/j.rtbm.2022.100814

Mao, J., Zhao, Y., Yang, S., Li, R. Y. M., & Abbas, J. (2022). Intelligent transformation and customer concentration. [JOEUC]. *Journal of Organizational and End User Computing*, *35*(2), 1–15. doi:10.4018/JOEUC.333470

Mathiyalagan, P., Suriya, S., & Sivanandam, S. N. (2010). Modified ant colony algorithm for grid scheduling. *International Journal on Computer Science and Engineering*, *2*(02), 132–139.

Ojha, V. K., Abraham, A., & Snášel, V. (2014). ACO for continuous function optimization: A performance analysis. In *2014 14th international conference on intelligent systems design and applications. IEEE*, 145-150. doi:10.1109/ISDA.2014.7066253

Scianna, M. (2024). The AddACO: A bio-inspired modified version of the ant colony optimization algorithm to solve travel salesman problems. *Mathematics and Computers in Simulation*, *218*, 357–382. doi:10.1016/j.matcom.2023.12.003

Sundarraj, S., Reddy, R. V. K., Babu, B. M., Lokesh, G. H., Flammini, F., & Natarajan, R. (2023). Route planning for an autonomous robotic vehicle employing a weight-controlled particle swarm-optimized Dijkstra algorithm. *IEEE Access : Practical Innovations, Open Solutions*, *11*, 92433–92442. doi:10.1109/ACCESS.2023.3302698

Tan, Y., Ouyang, J., Zhang, Z., Lao, Y., & Wen, P. (2023). Path planning for spot welding robots based on improved ant colony algorithm. *Robotica*, *41*(3), 926–938. doi:10.1017/S026357472200114X

Venkateswaran, C., Ramachandran, M., Ramu, K., Prasanth, V., & Mathivanan, G. (2022). Application of simulated annealing in various fields. *Materials and Its Characterization*, *1*(1), 1–8. doi:10.46632/mc/1/1/1

Wang, W., Yin, X., Wang, S., Wang, J., & Wen, G. (2023). Robot path planning method combining enhanced APF and improved ACO algorithm for power emergency maintenance. [IJITSA]. *International Journal of Information Technologies and Systems Approach*, *16*(3), 1–17. doi:10.4018/IJITSA.326552

Wang, Z., Wang, G. M., & Yao, C. (2018). Robot path planning based on TGSA and three-order bezier curve. *Revue D'intelligence Artificielle, 32*(SI), 41.

Xu, X., & Li, X. (2023). Construction of building an energy saving optimization model based on genetic algorithm. [IJITSA]. *International Journal of Information Technologies and Systems Approach*, *16*(3), 1–15. doi:10.4018/IJITSA.328758

Yan, L. (2023). A low-complexity channel estimation in internet of vehicles in intelligent transportation systems for 5G communication. [JOEUC]. *Journal of Organizational and End User Computing*, *35*(1), 1–21. doi:10.4018/JOEUC.326759

Yi, N., Xu, J., Yan, L., & Huang, L. (2020). Task optimization and scheduling of distributed cyber–physical system based on improved ant colony algorithm. *Future Generation Computer Systems*, *109*, 134–148. doi:10.1016/j.future.2020.03.051

Yu, X., Gao, X., Zhang, J., Zheng, B., & Vivien, K. (2017). Path planning of mobile robot based on hierarchical grid map. *Navigation and Control*, (2), 30–36.

Zeng, M. R., Xi, L., & Xiao, A. M. (2016). The free step length ant colony algorithm in mobile robot path planning. *Advanced Robotics*, *30*(23), 1509–1514. doi:10.1080/01691864.2016.1240627

Zhang, Z., Jiang, J., Wu, J., & Zhu, X. (2023). Efficient and optimal penetration path planning for stealth unmanned aerial vehicle using minimal radar cross-section tactics and modified A-Star algorithm. *ISA Transactions*, *134*, 42–57. doi:10.1016/j.isatra.2022.07.032 PMID:36058717

Zhao, J., Cheng, D., & Hao, C. (2016). An improved ant colony algorithm for solving the path planning problem of the omnidirectional mobile vehicle. *Mathematical Problems in Engineering*, *2016*, 2016. doi:10.1155/2016/7672839

Liyang Chu, postgraduate of Yanshan University. Work in Liaoning Institute of Science and Technology. Her research field include automation control and composite materials application.

Haifeng Guo, associate professor of Liaoning Institute of Science and Technology. Doctor of University of Chinese Academy of Science. His research field include industrial intelligent fault diagnosis.

Qingshi Meng, professor of Shenyang Aerospace University. Doctor of University of South Australia. His research field include research on manufacturing of composite materials for aviation.