

Research on Navigation Path Planning for An Underground Load Haul Dump

Qi Yulong^{1,*}, Meng Qingyong² and Tian Xu³

¹Jiangsu Xuzhou Construction Machinery Research Institute, Xuzhou 221004, China

²School of mechanical engineering, the university of Science and Technology Beijing, 100083, China

³Department of Mechanical and Mechatronics Engineering, University of Waterloo, Waterloo N2L3G1, Canada

Received 24 May 2015; Accepted 30 November 2015

Abstract

The improved A * algorithm is a method of navigation path planning for articulated underground scrapers. Firstly, an environment model based on a mining Geographic Information System (GIS) map is established, and then combined with improved A * algorithm, the underground global path planning problem of the intelligent Load Haul Dump (LHD) is solved. In this paper, for the articulated structure, the method of expanding nodes by articulation angle is adopted to make expanded nodes meet the trajectory characteristics. In addition, collision threat cost is introduced in the evaluation function to avoid collisions between the LHD and the tunnel walls. As per an analysis of the simulation test to verify the effectiveness of the improved A * algorithm and a comparison with the traditional A * algorithm, the improved A * algorithm can enhance search efficiency. A contrast of multiple sets of test parameters suggests that when the price weighted coefficient of collision is 0.2, the shortest path can be derived to avoid impact. Finally, tracking results indicate that the proposed algorithm for navigation path planning can maintain the tracking error to within 0.2 m in line with the structural characteristics of the scraper in the laboratory environment to realize the path planning of unmanned scrapers and trajectory tracking. Moreover, the algorithm can enhance the safety of scrapers and prevent roadway collisions. The feasibility and practicality of the proposed method is verified in this work.

Keywords: unmanned underground scraper, path planning, A * algorithm

1. Introduction

Mineral resources are the material basis for national economic development and security. The recent rapid development of digital mines has supported mining efficiency and safety considerably, and the automation of LHD is an important aspect of such mines in terms of safe production and yield as well as increased efficiency, among others [1]. Studies have long been conducted on automation and the intelligent technology of underground LHD in the mining industries of developed countries, and several advanced research results have been presented [2].

The traditional navigation of an unmanned LHD relies on successful recognition and successive navigation through each uniquely identified beacon. These beacons are installed in the LHD traveling tunnel. Induction cables are laid on underground roadways in a method proposed by G. Eriksson, wherein LHD is used in navigation by receiving induction cable signals. An unmanned LHD was also developed and applied for the first time [3]. Subsequently, a series of navigation guidance facility methods was established, including the use of reflective tape in roofs, resonant reflection circles, and one-dimensional bar codes [4], [5], [6]. The advantages of such methods are the utilization of a simple algorithm and high reliability; nonetheless, the navigation line often requires LHD to change given the

underground mine environment, tunnel excavation, and transitions. Therefore, boot facilities must frequently be removed and installed; in the process, operating costs are increased and automation degree is decreased. No mature unmanned LHD program has been generated; thus, the “underground intelligent LHD” is expected to be the focus and dominant direction of research on underground vehicle automation to maintain the level of underground mine work safety and mine digitization [7].

A lot of domestic and international studies have been conducted on path planning, and a series of efficient algorithms has been developed, such as artificial potential fields [8], genetic algorithms [9], Ant colony algorithms [10], particle swarm optimization [11], and A * algorithms [12], [13], [14]. All of these algorithms have been widely used in the field of route planning for unmanned aerial vehicles, unmanned vehicle navigation, and robot path planning, among others. The present work studies the navigation path planning problem of an unmanned LHD; in this scenario, the shortest collision-free path along the navigation path can be determined based on the current position and the given target position. In general, current path planning algorithms consider that the actuator can pivot turn or be processed as particle relative to the map, which cannot be achieved in reality since the underground LHD have low slender central hinge, and two-way body structure of, which is adjust to the narrow and limited underground roadway space [15]. If LHD is requested to track the travel planning path, then the algorithm results must match the characteristics of scraper path trajectory. The A * algorithm is therefore proposed for the specific requirements related to the problem of

* E-mail address: qiyulong2002@163.com

unmanned scraper path planning according to the special demands of the issue associated with unmanned LHD path planning.

2. Methodology

2.1 A * algorithm

The general form of the evaluation function for the A * algorithm is

$$f(n) = g(n) + h(n) \tag{1}$$

Where $g(n)$ is the actual expense from the starting node S_0 to the current node S_n and $h(n)$ is the estimated cost of the current node S_n to the destination node S_t . Due to the presence of the estimated cost $h(n)$, the A * algorithm is converted into a heuristic search algorithm that can be used to guide the search such that the increase in the number of nodes during the search process is minimized. Therefore, search efficiency is enhanced. The evaluation function $f(n)$ estimates the degree of importance of each node and determines the order in an Open table. $g(n)$ identifies the lateral trend of the search, which influences search efficiency in terms of search completeness [16]. To calculate function $f(n)$, all the pros and cons in the A * algorithm must be weighed to generate a certain proportion between $g(n)$ and $h(n)$.

2.2 Improved A * algorithm

2.2.1 Extended node method in accordance with the track characteristics of an underground LHD

When the A * algorithm is applied to solve general problems such as robot path planning, the child nodes are expanded according to either 8 or 24/48 neighboring nodes without considering the problems of minimal turning radius and track characteristics, among others. In the narrow and restricted underground roadway space, the expansion mode must match the track characteristics of the underground LHD.

In kinematic modeling, the LHD tires presumably roll without sliding sideways during steering. The LHD can be simplified into two segments, as shown in Figure 1. The angle γ between the second half of the LHD center line and the y-axis direction is approximately regulated as the heading angle. Meanwhile, the angle between the first half of the LHD center line and the extended center line of the second half is denoted by the splice angle θ , point A represents the center of the front axle, point B indicates the hinge point, and point C corresponds to the center of the rear axle. During the search, the length of each step is regarded as the length $|\vec{S}|$ of arc \vec{S} of the LHD running track, and the articulation angle θ remains unchanged at each step, that is, each point of the tub circle sits corresponding steering center O at each step.

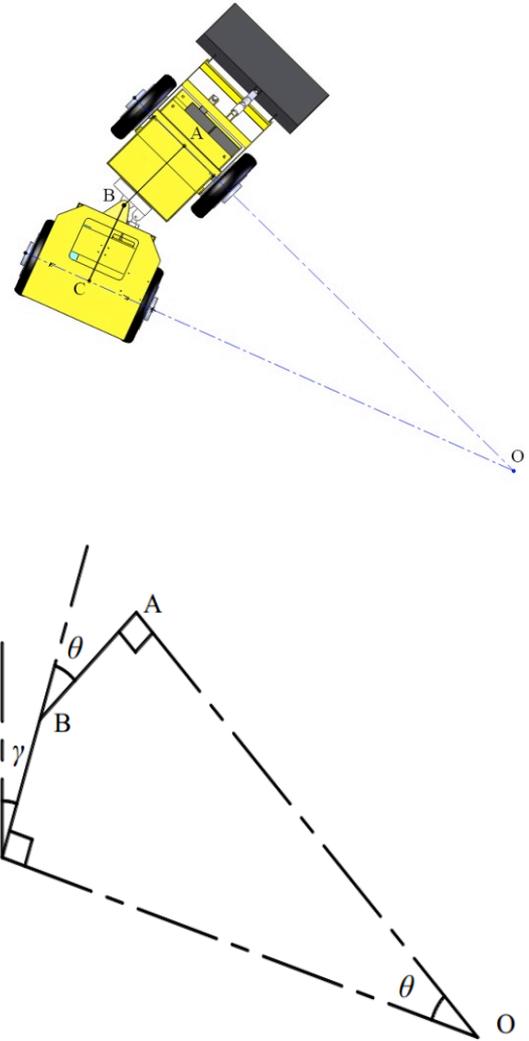


Fig.1. Simplified LHD kinematic model

Considering that the coordinates of the hinge point $B(x_B, y_B)$, articulation angle θ , and heading angle γ are known, the coordinates of the front axle center $A(x_A, y_A)$ and the rear axle center $C(x_C, y_C)$ can be obtained. Then, the self-constraint equations of the underground LHD can be formulated based on the horizontal coordinates.

The angle between l_{AB} and y-axis is $\gamma + \theta$:

$$\Delta(i) = \frac{1}{i} \left(\frac{1}{N} \sum_{j=1}^{N-i} \ln \frac{d_j(i)}{d_j(0)} \right)$$

$$\begin{bmatrix} x_A - x_B \\ y_A - y_B \end{bmatrix} = \begin{bmatrix} \cos(\gamma + \theta) & -\sin(\gamma + \theta) \\ \sin(\gamma + \theta) & \cos(\gamma + \theta) \end{bmatrix} \begin{bmatrix} 0 \\ |l_{AB}| \end{bmatrix} \tag{2}$$

The angle between l_{CB} and y-axis is γ :

$$\Delta(i) = \frac{1}{i} \left(\frac{1}{N} \sum_{j=1}^{N-i} \ln \frac{d_j(i)}{d_j(0)} \right)$$

$$\begin{bmatrix} x_C - x_B \\ y_C - y_B \end{bmatrix} = \begin{bmatrix} \cos \gamma & -\sin \gamma \\ \sin \gamma & \cos \gamma \end{bmatrix} \begin{bmatrix} 0 \\ |l_{CB}| \end{bmatrix} \quad (3)$$

$l_{AB} \perp l_{AO}$ and $l_{CB} \perp l_{CO}$:

$$l_{AB} \cdot l_{AO} = 0 \quad (4)$$

$$l_{CB} \cdot l_{CO} = 0 \quad (5)$$

Thus, articulation angle variation $\theta(t)$ is the sole factor that influences the path of the underground LHD to ensure the initial position and attitude of this LHD if speed is constant. This variation can initiate methods of improving the extension node. The articulation angle does not induce significant change in the searching step time and is limited by the maximum articulation angle θ_{max} ; therefore, the articulation angle θ is not extended directly. The following equation is used to obtain the current articulation angle for the current node S_n according to the integration of one-order differential θ' :

$$\theta_n = \int_0^n \theta' dt \quad (6)$$

where θ_n is the articulation angle of LHD in the search for node S_n .

Considering that the search algorithm implements a discrete process for time, the number of changes in the articulation angle $\Delta\theta$ within a sampling period Δt is determined through:

$$\Delta\theta = \theta' \Delta t \quad (7)$$

The articulation angle θ_n of node S_n can be accumulated through the extent of the change from the angle of node S_0 to node S_n ; this accumulation is written as:

$$\theta_n = \sum_{i=0}^n \theta'_i \Delta t = \sum_{i=0}^n \Delta\theta_i \quad (8)$$

In Figure 2, arc \hat{S} corresponds to the moving trajectory of the articulated point whose length $|\hat{S}|$ denotes the search step. The angle of the LHD turned around point O is calculated with $\alpha = \frac{|\hat{S}|}{|l_{OB}|}$ when the length of the walk is $|\hat{S}|$, based on the arc formula. Then, the coordinates of the corresponding articulated tramcar point $A'B'C'$ (x_{k+1}, y_{k+1}) at time $k+1$ can be obtained according to the coordinates of any articulated tramcar point (x_k, y_k) ABC at the time k turns α around point O .

$$\begin{bmatrix} x_{k+1} - x_O \\ y_{k+1} - y_O \end{bmatrix} = \begin{bmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{bmatrix} \begin{bmatrix} x_k - x_O \\ y_k - y_O \end{bmatrix} \quad (9)$$

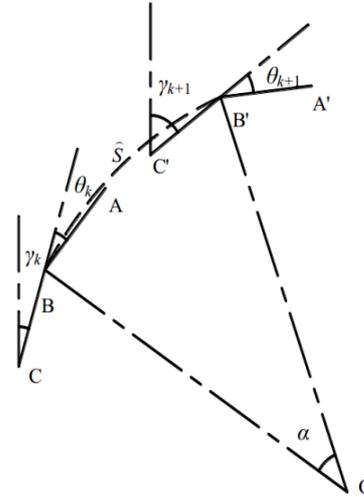


Fig.2. Extension of the nodes

This node expansion method fully accounts for the track characteristics of the articulated tramcar, and the searched path is in line with the trajectory characteristics of this tramcar. When searching step length $|\hat{S}|=2$ m, the distance of the neighbor node sat the same search depth is precisely controlled within 0.2 m. If an expansion mode with eight neighboring nodes is adopted in the underground work environment whose roadway is generally only 4 m wide, then the searching step length is controlled within 0.3 m. Along searching step reduces search depth and increases the number of nodes, which benefits the implementation of the most efficient path search algorithms. The most accurate algorithm can then be determined with reference to the allowable range of accuracy.

2.2.2 Comprehensive evaluation function given the requirements for underground LHD

The navigation path of the underground LHD not only requires the shortest path to match the trajectory path properties of this LHD, but also demands that the underground LHD should not collide with the tunnel wall and maintain a safe distance when traveling along the path. Meanwhile, the evaluation function must be modified to prevent the LHD from moving too closely to the wall since the A* algorithm generates the shortest path.

In Figure 3, the underground LHD in an arrow roadway can be reduced to nine mutual constraint points, eight vertices, and an articulated point of the outer most contour body. Accordingly, the simplified model can be used to determine whether the underground LHD collides or interferes with the wall; that is, if any point on the wall is included in the underground LHD simplified model, the event is considered a collision with the wall. In the meantime, this node is marked as impassable and then deposited in the Close table.

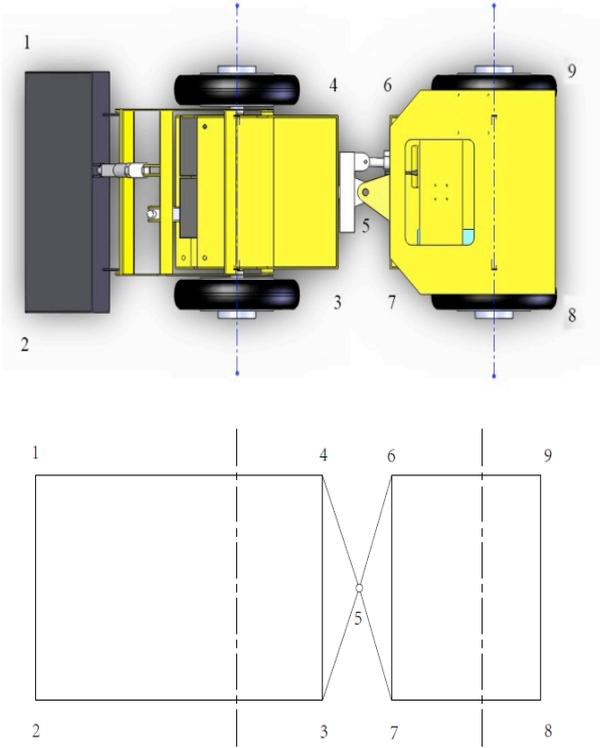


Fig.3. Simplified LHD crash

The actual cost $g(n)$ of the valuation function $f(n)$ in the traditional A* algorithm corresponds to the total distance from the source node S_0 to the node S_n . Nonetheless, the cost of collision threat $g_c(n)$ should be added to the distance already traveled along the path as the actual cost $g_p(n)$. Moreover, the weighting coefficients W_p and W_c are introduced to manage both the path and the proportion costs of the actual collision threat cost as follows:

$$g(n) = W_p g_p(n) + W_c g_c(n) \quad (10)$$

where $g_p(n)$ stands for the distance cost when searching for node S_n . The cumulative path distance between two nodes is expressed as:

$$g_p(n) = g_p(n-1) + d \quad (11)$$

The path cost of each step d is equal to the search step $|\hat{S}|$.

$g_c(n)$ corresponds to the collision cost of node S_n , which is the cost of possible collision with the wall at node S_n . The possibility of collision with the wall and the LHD cost can be quantified by detecting the distance among the nine mutual constraint points and the tunnel wall C_n is defined as a collision threat involving node S_n and the wall within a track. This threat can be induced based on the

minimal distance of the nine outline feature points from the wall $d_{\min} = \min(d_1, L, d_9)$ as follows:

$$C_n = f(d_{\min}) \quad (12)$$

A safe distance d_{safe} is generated when LHD is at node S_n ; if the distance of the nine outline feature points from the wall is greater than this safe distance ($d_{\min} \geq d_{\text{safe}}$), then these points will not collide with the wall ($C_n = 0$). When the distance of the nine outline feature points from the wall is less than the safe distance ($d_{\min} < d_{\text{safe}}$), a risk of colliding with the wall is generated. When the distance decreases, the value of C_n at node S_n should increase monotonically. A quadratic function is used to represent the relationship between minimum distance d_{\min} and collision threat C_n :

$$C_n = \begin{cases} 0 & d_{\min} \geq d_{\text{safe}} \\ (d_{\text{safe}} - d_{\min})^2 & d_{\min} < d_{\text{safe}} \end{cases} \quad (13)$$

All the collision threats $C_i (i = 0, 1, 2, \dots, n)$ from S_0 to S_n along the planned path are accumulated to calculate the collision cost $g_c(n)$:

$$g_c(n) = \sum_{i=0}^n C_i \quad (14)$$

The actual cost $g(n)$ in the improved valuation function consists of the path costs $g_p(n)$ and the collision cost $g_c(n)$. The weighting coefficients W_p and W_c control the proportion of the path and collision threat costs in the actual cost; furthermore, the value is based on tunnel traffic conditions and LHD working conditions. Estimated cost $h(n)$ is represented by the straight distance from the current node to the destination node:

$$h(n) = D(S_n, S_t) \quad (15)$$

3. Results

3.1 Verification of the algorithm through simulation results

The algorithm described in this paper is tested on a PC running on a Windows7 x64-bit operating system (processor: AMD-6300-Six Core-3.50 GHz; RAM: 8.00 GB), and the programming environment is realized with Visual Studio 2010 VC++ (V100). A 50 m × 35 m virtual view map of the underground mine haulage tunnel plan is utilized as the map; roadway width is 4 m, LHD length is 6m, maximum width is 2m, the front axle from the hinge point is 1.5 m, and the rear axle is 2 m away from the hinge point. In Figure 4, the southwest border point of the map is regarded as the origin of the coordinates; the starting point is

set to (6, 2), and the end point is set to (48, 25). The points are represented in the map by a □ symbol.

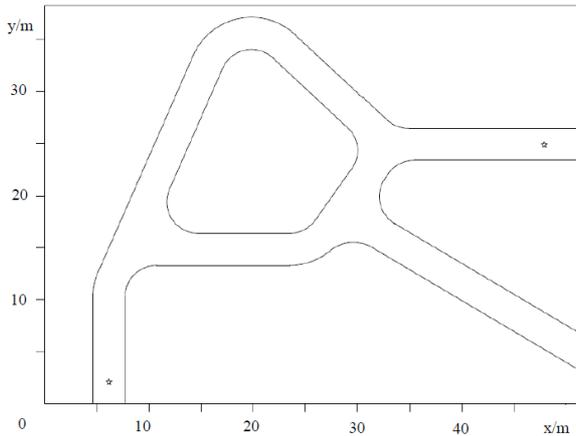


Fig.4. Test map for algorithm performance and the settings of the start and end points

3.1.1 Comparison of the performance of the traditional and the improved A* algorithms

Without considering the effect of collision threat, the extended node method is implemented according to the LHD track characteristics. The search step is set to 1.5 m, the articulation angle variation $\Delta\theta = 6^\circ$, $W_p = 1$, and $W_c = 0$.

Figure 5 illustrates the search path result. According to the extension method of the adjacent eight-node ordinary A* algorithm, the grid size is 0.3 m × 0.3 m; the search path result is shown in Figure 6.

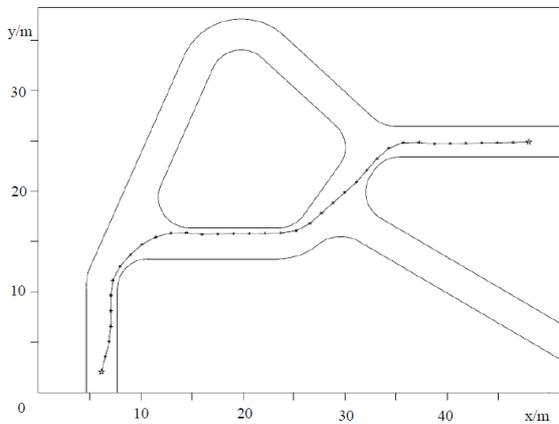


Fig.5. Path result obtained using the extension method that matches the LHD track

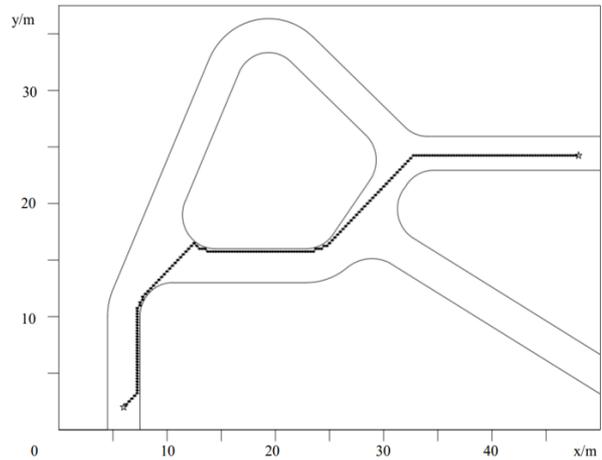


Fig.6. Path result generated with the method of adjusting eight adjacent nodes

In Table 1, the number of expanded nodes in the improved algorithm declined significantly in comparison with that observed under the adjacent eight-node expansion mode. This outcome is in line with the expansion mode of the LHD track characteristics in the search path.

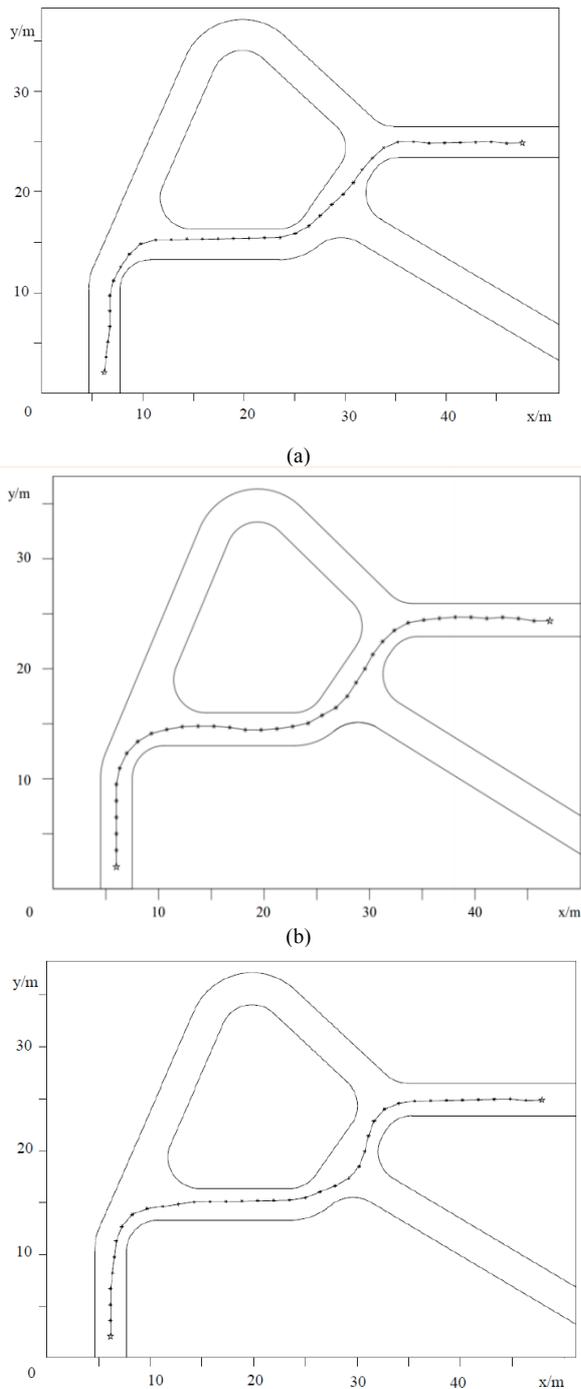
Table 1. Comparison between two different extension methods

Expansion mode	Number of nodes in the Open list	Number of nodes in the Close list
Adjacent eight-node	422	255
Improved algorithm	174	65

As Table 1 shows, the search path which is consistent with the expansion mode of the LHD track characteristics is a group of end-to-end continuous fold lines with the change in angle, and the LHD can travel along the path with minimal errors. Since the angle at the steering area searched by the expansion mode of the eight adjacent nodes has changed significantly, this angle is clearly unsuitable for the LHD traveling along the path.

3.1.2 Effect of collision threats on the track

The search path shown in Fig. 5 matches the LHD trajectory characteristics; however, many nodes are generated near the wall, and a risk of collision with the wall is induced when the LHD is driven along the path. In Figure 7, the search path results are produced under the following conditions: $W_p = 0.9$, $W_c = 0.1$; $W_p = 0.8$, $W_c = 0.2$; $W_p = 0.7$, $W_c = 0.3$, and a search step = 2 m.



(c)
Fig.7. Influence of different collision threat parameters on the path
 (a) $W_p = 0.9, W_c = 0.1$; (b) $W_p = 0.8, W_c = 0.2$; (c) $W_p = 0.7, W_c = 0.3$

Given different collision threat parameters, the path results change significantly. When the distance cost weighting coefficient W_p decreases and collision cost weighting coefficient W_c increases, the path gradually moves to the center of the roadway to avoid colliding with the wall. Table 2 shows the algorithm performance evaluations under each parameter path; when W_p decreases and W_c increases, the number of nodes decreases in both the Open and the Close lists. This outcome matches the theory that a collision threat can eliminate the node which is too close to the wall to accelerate a search. The value generated when $W_p = 0.9$ and $W_c = 0.1$ is higher than that obtained when $W_p = 1$ and $W_c = 0$ due to such a threat. C_n is a quadratic function of the distance to the wall(Formula 13); when the path is close to the wall, the collision threat of the adjacent nodes varies significantly, which make the valuation function $f(n)$ of the update nodes change significantly as well, and be easier to re-search from the root of the tree search. Meanwhile, the collision cost $g_c(n)$ of the path decreases and path length increases when W_p decreases and W_c increases, which confirms that when the proportion of the collision cost $g_c(n)$ effect on the valuation functions $f(n)$ increases, the algorithm is inclined to sacrifice the shortest path and seek a difficult path to avoid collision with the wall. To sum up, when the collision threat parameters are collectively valued at $W_p = 0.8$ and $W_c = 0.2$, few nodes are expanded; and the result is not only consistent with the LHD track characteristics but also the shortest path to avoiding collision with the wall under the given premise.

Table 2. Comparison of algorithm performance givendifferent collision threat parameters

Parameters	Numbe of nodes in the Ope: list	Number of nodes in the Close list	Collision cost $g_c(n)$	Path length/m
$W_p = 0.9, W_c = 0.1$	510	216	13.05	74.0
$W_p = 0.8, W_c = 0.2$	130	42	1.12	74.3
$W_p = 0.7, W_c = 0.3$	123	41	0	74.6

3.2 Tracking experiment conducted on an unmanned LHD in a laboratory environment

To prove the feasibility of the improved A * algorithm, a path planning and trajectory tracking experiment was conducted on an unmanned LHD in a laboratory corridor (Figure 8). Figure 9 illustrates the map of this corridor; the starting point is set at the west side of the corridor(1.1,11.6) and the end point is set at the elevator(9.1,7.5) to establish

the travel conditions of the unmanned LHD scraper from the transport roadway to the mining roadway.



Fig.8. Test for unmannedLHD path planning and tracking

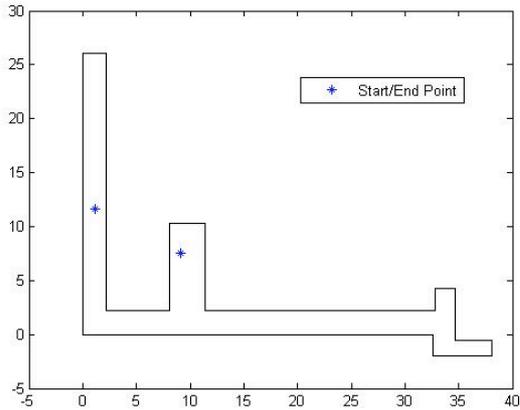


Fig.9. Passageway map and the settings of the start and end points

During the experiment, the location information of the machine model during operation was recorded through the vehicle positioning system [17]. Fig.10 illustrates the contrast between path planning by the improved A* algorithm and the actual trajectory; the unmanned LHD successfully transitions from the starting point to the end point by following the path calculated by this algorithm. Fig.11 depicts the surveillance process of unmanned LHD tracking errors during operation, which is maintained in the range of 0.2 m in accordance with the target path.

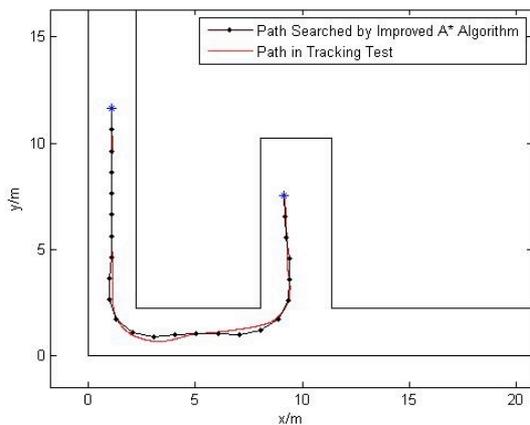


Fig.10. Comparison between the path searched by the improved

A* algorithm and the final test path

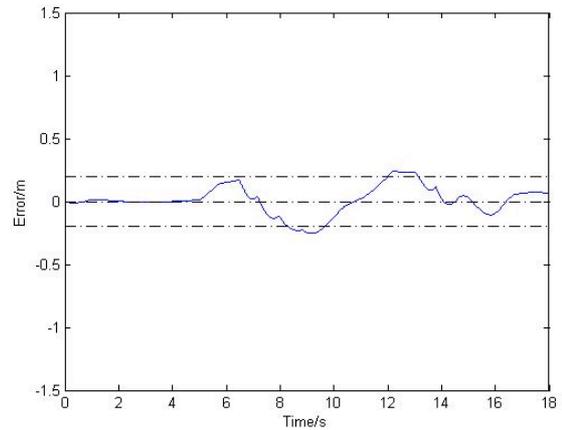


Fig.11. Tracking error detected during the tracking test

4. Conclusions

In this paper, an expanded node method is proposed according to specific requirement for the path planning problem related to the unmanned underground LHD and based on the improved A* algorithm. This method can match the LHD trajectory characteristics, and the threat cost is generated as a result of collisions with tunnel walls in the valuation function. The simulation results show that the path planned by the improved A* algorithm can not only meet the requirements of the unmanned scraper path but can also significantly reduce the number of search nodes and enhance search efficiency in comparison with the traditional A* algorithm. Unlike the simulated collision threat parameters, parameters ($W_p = 0.8$ and $W_c = 0.2$) that are suitable for unmanned scraper path planning are obtained; thus, the algorithm can determine the shortest path under the premise of avoiding collisions with the wall. During the model car experiment conducted in the laboratory corridor environment, the tracking error of the unmanned LHD in an operation path is maintained at less than 0.2 m; this result confirms the applicability of the improved A* algorithm proposed in this paper to unmanned LHD path planning.

Acknowledgements

This work was supported by the National Key Technology Support Program of China under the project 2011AA060404.

References

1. Feng, S., Kai, Z., and Hongshu, G., "A Study on Underground Loader Tracking Track Calculation Model". *Nonferrous Metals(Mining Section)*, 62(6) ,2010,pp.66-69.
2. Zhizhuo, L., Kai, Z., and Hongshu, G., "Global Path Planning of Intelligent Load-Haul-Dump Based on Improved Ant Colony Algorithm". *Nonferrous Metals (Mining Section)*, 65(2), 2013, pp.6-10.

3. Eriksson, G., Kitok, A., "Automatic loading and dumping using vehicle guidance in a Swedish mine". *Proceedings of the First International Symposium on Mine Mechanization and Automation*, Golden, CO, 1991, pp.40-46.
4. Chenmeng, W., Guan, L., Mingtao, J., "An Overview of Autonomous Navigation Techniques and Development Trend for Underground LHD". *China Safety Science Journal*, 23(3), 2013, pp.130-134.
5. Hannu, M., "Overview of LHD navigation without artificial beacons". *Journal of Robotics and Autonomous Systems*, 36(1), 1999, pp.21-35.
6. Marshall, J., Barfoot, T., and Larsson, J., "Autonomous Underground Trammig for Center-Articulated Vehicles". *Journal of Field Robotics*, 25(6-7), 2008, pp.400-421.
7. Mengxiang, G., "The Development of Technology on Underground Loader and Underground Automobile Automation". *Modern Mining*, 12(12), 2009, pp.1-6.
8. Jianying, Z., Zhiping, Z., and Dun, L., "A path planning method for mobile robot based on artificial potential field". *Journal of Harbin Institute of Technology*, 38(8), 2006, pp.1306-1309.
9. Qing, L., Sijiang, X., and Xinhai, T., "A self-adaptive genetic algorithm for the shortest path planning of vehicles and its comparison with Dijkstra and A * algorithms". *Journal of University of Science and Technology Beijing*, 28(11), 2006, pp.1082-1086.
10. Enxiu, S., Minmin, C., and Jun, L., "Research on Method of Global Path-planning for Mobile Robot Based on Ant-colony Algorithm". *Transactions of the Chinese Society for Agricultural Machinery*, 12(6), 2014, pp.53-57.
11. Qing, L., Yinmei, X., and Dezheng, Z., "Global path planning method for mobile robots based on the particle swarm algorithm". *Journal of University of Science and Technology Beijing*, 32(3), 2010, pp.397-402.
12. Qingxuan, J., Gang, C., and Hanxu, S., "Path Planning for Space Manipulator to Avoid Obstacle Based on A * Algorithm". *Journal of Mechanical Engineering*, 46(13), 2010, pp.109-115.
13. Giuseppe, C., Marcello, F., and Giacomo, L., "A network flow based heuristic approach for optimizing AGV". *Journal of Intelligent Manufacturing*, 24(2), 2013, pp.405-419.
14. Dianjun, W., "Indoor mobile-robot path planning based on an improved A * algorithm". *Tsinghua Univ (Sci & Tech)*, 52(8), 2012, pp.1085-1089.
15. Feng, S., Hongshu, G., and Kai, Z., "Analysis of Driving Track for LHD on Typical Path". *Mining & Metallurgy*, 18(2), 2009, pp.67-70.
16. Ji, L., Xiuxia, S., "A Route Planning's Method for Unmanned Aerial Vehicles Based on Improved A-Star Algorithm". *Introducing Journal of China Ordnance*, 29(7), 2008, pp.788-792.
17. Yu, M., Li, L., and Fei, M., "Vision-based Self-localization Method for Underground Mining Vehicle". *Transactions of the Chinese Society for Agricultural Machinery*, 43(1), 2012, pp.22-27.