

Article

A Path Planning Method for Underground Intelligent Vehicles Based on an Improved RRT* Algorithm

Hao Wang^{1,2}, Guoqing Li^{1,2,*}, Jie Hou^{1,2}, Lianyun Chen^{1,3} and Nailian Hu^{1,2}

¹ College of Civil and Resource Engineering, University of Science and Technology Beijing, Beijing 100083, China; haowang@xs.ustb.edu.cn (H.W.); houjie@ustb.edu.cn (J.H.); chenlianyun@sd-gold.com (L.C.); hnl@ustb.edu.cn (N.H.)

² Key Laboratory of Efficiency Mining and Safety of Metal Mines, Ministry of Education, Beijing 100083, China

³ Shandong Gold Group Co., Ltd., Jinan 250014, China

* Correspondence: qqlee@ustb.edu.cn

Abstract: Path planning is one of the key technologies for unmanned driving of underground intelligent vehicles. Due to the complexity of the drift environment and the vehicle structure, some improvements should be made to adapt to underground mining conditions. This paper proposes a path planning method based on an improved RRT* (Rapidly-Exploring Random Tree Star) algorithm for solving the problem of path planning for underground intelligent vehicles based on articulated structure and drift environment conditions. The kinematics of underground intelligent vehicles are realized by vectorized map and dynamic constraints. The RRT* algorithm is selected for improvement, including dynamic step size, steering angle constraints, and optimal tree reconnection. The simulation case study proves the effectiveness of the algorithm, with a lower path length, lower node count, and 100% steering angle efficiency.

Keywords: underground intelligent vehicles; path planning; RRT* algorithm; articulated vehicles; unmanned driving



Citation: Wang, H.; Li, G.; Hou, J.; Chen, L.; Hu, N. A Path Planning Method for Underground Intelligent Vehicles Based on an Improved RRT* Algorithm. *Electronics* **2022**, *11*, 294. <https://doi.org/10.3390/electronics11030294>

Academic Editor: Daniel Gutiérrez Reina

Received: 13 December 2021

Accepted: 14 January 2022

Published: 18 January 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In recent years, major mining groups have increased their investment in intelligent mining, and the mining industry is gradually entering the era of being remote, smart, and unmanned [1–5]. Intelligent vehicles are the most important pieces of equipment for intelligent mining with unmanned driving. Path planning is one of the key technologies for autonomous driving of intelligent unmanned vehicles. A reasonable path planning algorithm helps vehicles optimize the running trajectory, avoid obstacles according to the environment, and realize safe and efficient driving. The intelligent vehicles include drilling rigs, charging jumbo, load-haul-dump (LHD), trucks, scaling jumbo, and bolting jumbo, etc., the goal of which is to achieve intelligent mining processes by autonomous positioning, autonomous navigation, autonomous driving, and autonomous operation. These underground intelligent vehicles are shown in Figure 1.

The path planning of underground intelligent vehicles is one of the branches of research of unmanned ground vehicles (UGV) and unmanned aerial vehicles (UAV). With the advancement of technology, they have been widely used in the fields of logistics, transportation, disaster relief, etc., [6,7]. The research into UGV automatic driving in underground mining can be traced back to the 1960s [8,9]. The USA, Canada, Sweden, etc., have researched the remote control of vehicles, but due to the limitations of communications and sensors, the application progress was slow. With the technological revolution, such as the Internet of Things (IoT) and machine learning, unmanned mining operation has become a research hotspot in the mining field again. The European Union (EU) initiated the “Robominers” project to develop bionic robots for underground mining operations in harsh environments [10]. Rio Tinto launched the “Mine of the Future” program, which

aims to remote control more than 10 mines in Pilbara from Perth to realize unmanned mining operations [11]. The Swedish Mining Automation Group (SMAG) also proposed a plan to lead the automation upgrading of the mining industry [12]. The main research interest in this paper is the path planning of underground intelligent vehicles. Based on the known environmental map, starting point, and target, we use the path planning algorithm to obtain an appropriate path that accords with mining operation and vehicle kinematics. More generally, we research global path planning algorithms.

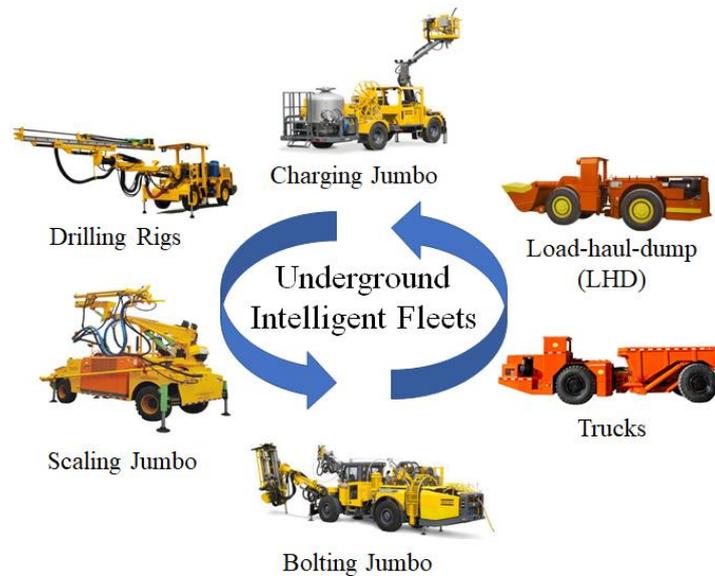


Figure 1. Underground intelligent vehicles.

In addition to the characteristics of common UGVs, the control of underground vehicles has strong industry specificity, which leads to more complicated path planning. First, the mechanical structure of the underground vehicles is more complicated, which is different from the common four-wheeled UGVs on the ground and UAVs in the air [13]. Thus, the underground vehicles are more difficult to control from kinematics and need a defined path. Second, compared to roads on the ground, the underground space is narrow and curved, with many irregular surfaces. Path planning for underground vehicles needs to focus more on passing narrow points and turns. Finally, there is no GPS underground, and the communication is worse than that on the ground. The path is required to be relatively simple, which reduces the control commands. Above all, the path planning method for UAVs or UGVs will not totally accord with that of underground vehicles. Therefore, it is necessary to upgrade the existing path planning method to adapt to underground intelligent vehicles.

RRT* is a sampling-based algorithm with probabilistic and complete resolution, high speed, and smooth results. For the 2D finite space of underground vehicles, it has a higher probability to create a path through narrow points and turns, which is closer to the underground requirements. Therefore, the RRT* algorithm was selected as the basic algorithm in this paper. With the aim of intelligent mining operation, by considering the kinematics of the intelligent vehicles and the drift environment, three improvements are proposed, including dynamic step size, steering angle constraints, and optimal tree reconnection. The algorithm improves the effectiveness of obstacle avoidance and shortens the distance while ensuring efficiency, which provides a feasible path planning method for intelligent vehicles.

Overall, this paper proposes a path planning method based on an improved RRT* algorithm to solve the problem of path planning for underground intelligent vehicles under articulated structures and drift environment conditions. Fully considering the environmental and equipment characteristics of underground mines is also an important feature.

The remainder of this paper is organized as follows. In Section 2, the related works are reviewed and the necessary preliminaries of intelligent mining are presented. In Section 3, the constraints of intelligent mining are formulated, including the drift environment formulation and the kinematics of vehicles. In Section 4, the process of the classic RRT* algorithm is analyzed and three improvement measures are proposed to adapt to underground intelligent vehicles. In Section 5, the case study by simulation method is presented, and the results are discussed. In Section 6, the paper is concluded.

2. Related Works

2.1. Underground Intelligent Vehicles

Autonomous vehicle driving is one of the key technologies of intelligent mining, and its main sensors and operating modes are shown in Figure 2. The intelligent vehicles collect their states and environmental information by laser lidar, inertial measurement unit (IMU), camera, RFID, and other sensors and calculate their current position and state by using edge computing. Then, they interact with cloud computing through wireless communication to obtain driving paths and complete the current driving process.

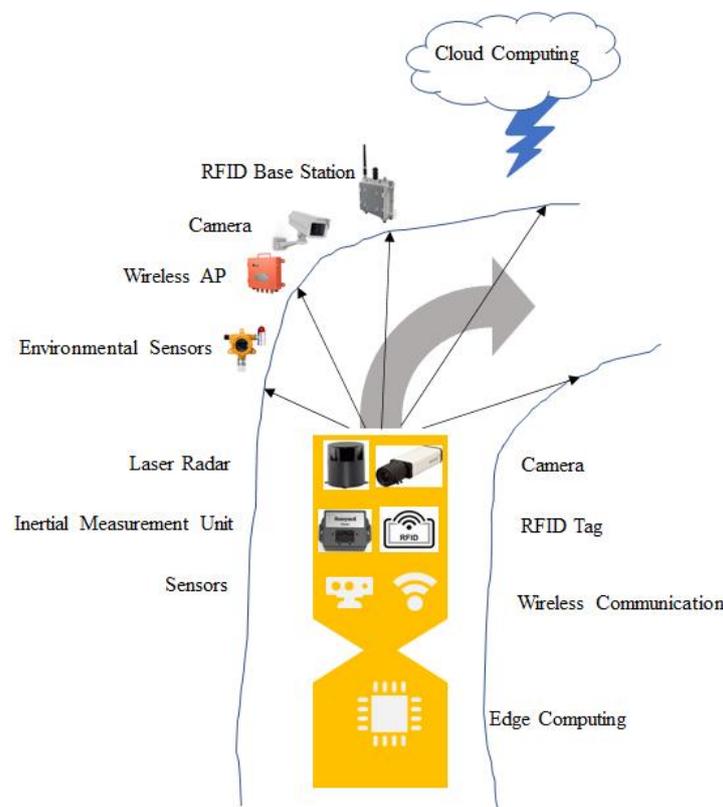


Figure 2. The main sensors and operating modes of autonomous driving vehicles.

Intelligent vehicles for underground mining can be divided into either an integral type or an articulated type according to their structure. The integral type has the advantage of a simple structure but has the disadvantage of often having insufficient power. It is mainly used in pick-up trucks, small LHDs, and other small vehicles. The articulated type has the advantages of a small steering radius and sufficient power and is more suitable for the narrow environment of underground mining [14]. Therefore, it is widely used in heavy equipment such as underground large LHDs, trucks, and jumbos. Articulated vehicles are more suitable in underground mines [13]. However, articulated vehicles have more complex structures than four-wheeled cars. For these reasons, articulated vehicles were selected as the main research object of this paper.

Large vehicles have a higher production capacity, but increasing the size of the drift increases the development cost. The size of the drift is often selected to meet the minimum specifications for vehicles, which put forward strict requirements for the running trajectory of vehicles. Therefore, the core of path planning for intelligent vehicles is to coordinate the environment of the drift and the kinematics of vehicles to obtain an optimal path trajectory that guides the vehicles to drive autonomously.

2.2. Path Planning Methods

Based on high-precision positioning and unmanned driving technology, many unmanned equipment path planning algorithms have been studied, which are mainly divided into artificial potential field methods, graph search algorithms, and sampling-based algorithms. The artificial potential field, proposed by Khatib, is a virtual force method, which makes the equipment subject to the repulsive forces from obstacles and gravity and from the target at the same time [15–18]. This method is simple to calculate, and the obtained path is safe and smooth, but it easily falls into a local optimal solution. The graph search algorithm converts the map for path planning into a graph and obtains the optimal path through graph theory, including the Dijkstra algorithm, A* algorithm, etc. [19–24]. This method takes into account both efficiency and completeness, but the map needs to be rasterized to complete the graph conversion, resulting in poor path smoothness. The sampling-based algorithm narrows the search space by discrete sampling in a continuous space. It is a Monte Carlo method with uniform space, including the Probabilistic Road Map Method (PRM), Rapid Random Extended Tree Method (RRT), etc. [25–29]. It has the advantages of fast search speed and simple environment modeling, but it cannot obtain a global optimal solution, and its efficiency is greatly affected by its step size and sampling mode.

The RRT algorithm [30] was proposed by Lavelle et al. in 1998. It is a random sampling algorithm that uses incremental growth to achieve rapid search in non-convex high-dimensional spaces. The RRT algorithm does not need to rasterize the search space and has the advantage of high search space coverage. It is suitable for dealing with scenes containing obstacles and motion constraints. Therefore, it is widely used in path planning for intelligent devices. The RRT algorithm is a Monte Carlo method. It usually takes the starting point as the root node and generates a random extended tree through random sampling. When the child node reaches the target area, the sampling is completed, and a feasible path is obtained.

The sampling of the RRT algorithm is random, and the generated path is a feasible path rather than an optimal path. Therefore, a variety of improved methods are derived. The RRT* algorithm [31] was improved based on the RRT algorithm, and the goal is to improve the performance of the RRT algorithm in order to ascertain the optimal path. The RRT* algorithm continuously optimizes the path during the search process by reselecting the parent node and rerouting. With the increase in iterations, the obtained path gradually approaches the optimal path.

There is relatively little research on path planning in underground mining, and currently it is mainly focused on underground disaster relief, surveying, and mapping. Ma et al. [32] proposed a path planning method considering gas concentration distributions. The global working path for a coal mine robot was planned based on the Dijkstra algorithm and the ant colony algorithm, then local path adjustments were carried out. The research object was coal mine robots, and the scene was disaster relief. Mauricio [33] proposed a strategy of exploration and mapping for multi-robot systems in underground mines where toxic gas concentrations are unknown. The principal algorithm was behavior control. Papachristos et al. [34] considered the challenge of autonomous navigation, exploration, and mapping in underground mines using aerial robots, and proposed an optimized multimodal sensor fusion approach combined with a local environment morphology-aware exploration path planning strategy. The research objects were four-rotor drones, and the scene was underground surveying and mapping. Gamache et al. [35] set up a shortest-path algorithm for solving the fleet management problem in underground mines with considera-

tion for dispatching, routing, and scheduling vehicles. The solution approach was based on a shortest-path algorithm. They considered all single-lane bi-directional road segments of the haulage network. The research focused more on mining scheduling than vehicle path planning. The solution provided the direction of the vehicle at an intersection, rather than the trajectory of a single device. It can be considered as a form of cooperative scheduling, which relates to the upper-level control of intelligent vehicles. Larsson [36] developed a new flexible infrastructure-less guidance system for autonomous tramming of center-articulated underground mining vehicles. The results showed that it was capable of autonomous navigation in tunnel-like environments. However, the process of path planning was not described. Tian [37] presented a novel strategy for autonomous graph-based exploration path planning in subterranean environments. Yuan [38] focused more on path planning and an obstacle avoidance mechanism under the complex geological conditions of a coal mine. Dang [39] presented a novel strategy for autonomous graph-based exploration path planning in subterranean environments. Song [40] considered both the distance of the path and some hybrid costs to obtain a global path. Bai [41] proposed a multisensor data fusion algorithm based on genetic algorithm optimization of the variably structured fuzzy neural network. Ma [42] improved both the distance function and the selection of child nodes. The feature of this paper is the full consideration of the environment with a vectorized map and the articulated kinematics of underground mines. A comparison between some typical underground mine path planning studies is shown in Table 1.

Table 1. Comparison of typical underground mine path planning research.

Research	Algorithms	Scenarios	Path Type	Map Type	Equipment
[32]	Dijkstra, Ant colony	Rescue	Global	Rasterized	Mine robots
[33]	Scanning algorithms	Dangerous environment in coal mines	Local	Real-time sensing	Multi-robot systems
[34]	Optimized multimodal sensor fusion approach	Navigation, mapping	Navigation	Real-time sensing	Aerial robots
[35]	Enumeration algorithm	Production	Global	Topological	Underground vehicles
[36]	Feature detection algorithm	Production	Navigation	Real-time sensing	Underground articulated vehicles
[37]	Artificial potential field	Rescue	Global	Rasterized	Mine robots and UAVs
[38]	A* algorithm	Production	Global and local	Rasterized	Underground four-wheeled vehicles
[39]	Graph-based exploration path planning	Exploration, mapping	Global and local	Real-time sensing	UAVs
[40]	Ant colony algorithm	Not mentioned	Global	Rasterized	Mine robots
[41]	Genetic algorithm	Rescue	Navigation	Real-time sensing	Rescue snake robot
[42]	D* algorithm	Not mentioned	Global	Rasterized	Mine robots
This paper	Improved RRT* algorithm	Production	Global	Vectorized	Underground articulated vehicles

3. Constraints Formulation

Autonomously driven underground intelligent vehicles initiate a process of interaction between the underground environment and the vehicle. Before path planning, it is necessary to establish the environment, vehicle features, and interaction constraints.

3.1. Drift Environment Formulation

Drifts are the main environments for underground intelligent vehicles. These intelligent vehicles start at the stope filled with ore, drive through the drifts, then reach the orepass, and offload the ore. The point cloud is a common method for intelligent mine environmental modeling, which is generated by laser scanners [43]. Figure 3 shows the point cloud data obtained through SLAM, which is a typical drift environment. A typical design profile of a drift is shown in Figure 4 [44]. Where vehicles are required to travel through drifts, the vehicle cross-section will fix the dimensions of the opening. Underground intelligent vehicles do not make vertical movements, so it is possible to process 3D point cloud data into a 2D map by extracting the waistline and then converting the map into a graph for the path planning algorithm.



Figure 3. Typical drift environment point cloud data by SLAM.

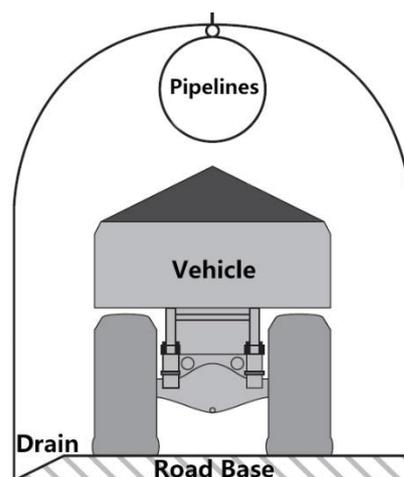


Figure 4. A typical design profile of a drift.

This paper uses the vectorized method instead of the common rasterization method as the map preprocessing method. The drifts are narrow and long with complicated surfaces. In the process of rasterization, the grid size has a great influence. Large-size grids cannot express the small edges and corners of the drifts well, resulting in a lack of detailed map information, and collisions during driving of the vehicles. Small grids lead to large total grids, which result in calculations being carried out in increments and a reduction in

efficiency. Therefore, the rasterization method has certain limitations in processing the drift environment. A vectorized map can effectively improve these shortcomings. It expresses map information such as points, lines, and areas by recording coordinates. The points are represented by the north coordinate and east coordinate. The lines are represented by a series of ordered coordinates. The surfaces are represented by a series of ordered and closed coordinates. We recorded the coordinates of the scattered points on the map boundary through dense interpolation and connected them to form lines. The dataset included coordinate points, lines, and polygons, named as $Polygon_{map}$ in the following. The effect comparison between the rasterized map and vectorized map is shown in Figure 5. The rasterized map used a 22×41 matrix, and the dataset was 26.4 kb, as shown in Figure 5a. The vectorized map included 17 points, 17 lines, and 1 polygon. The dataset was 0.9 kb, as shown in Figure 5b. The vectorized map has great advantages in map refinement and data size.

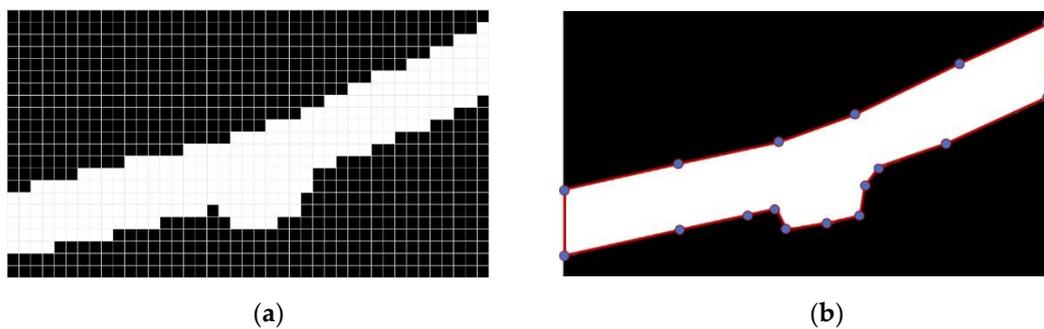


Figure 5. Comparison of the rasterized map and vectorized map. (a) The rasterized map; (b) the vectorized map.

3.2. Kinematics of Vehicles

Intelligent vehicle path planning needs to consider the kinematics to realize steering and obstacle avoidance. Articulated vehicles are considered in this paper, which are composed of a front body and rear body, and the vehicle bodies are connected through the articulated points. Articulated vehicles in underground mines are usually rear-wheel drives, and the steering is completed by controlling the relative position between the front and rear bodies through the expansion and contraction of the steering cylinders. Non-articulated vehicles can be abstracted as articulated vehicles with a rear body length of 0 to achieve the universality of this article. Assuming that the tire and the ground have pure rolling friction, the movement process of the vehicles can be simplified to rigid body plane movement, as shown in Figure 6.

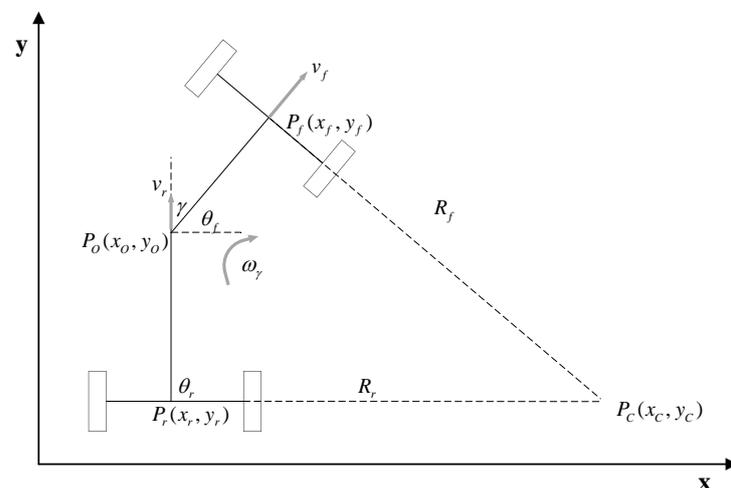


Figure 6. The simplified movement process of the vehicles.

In order to describe the position of the vehicle, the kinematics formula needs to be established. We established a Cartesian coordinate system for articulated vehicles. We set the instantaneous steering center as $P_C(x_C, y_C)$, the axle center of the front body as $P_f(x_f, y_f)$, the axle center of the rear body as $P_r(x_r, y_r)$, the articulated point as $P_O(x_O, y_O)$, the front and rear linear velocities as v_f and v_r , respectively, the headings as θ_f and θ_r , the radius of the front and rear body as R_f and R_r , respectively, the steering angle as γ , and the angular velocity as ω_γ . Then the kinematics model of the articulated vehicles can be described as follows [45].

$$\begin{bmatrix} \dot{x}_r \\ \dot{y}_r \\ \dot{\theta}_f \\ \dot{\gamma}_f \end{bmatrix} = \begin{bmatrix} \cos\theta_f \\ \sin\theta_f \\ \frac{\sin\gamma}{R_f\cos\gamma+R_r} \\ 0 \end{bmatrix} v_f + \begin{bmatrix} 0 \\ 0 \\ \frac{R_r}{R_f\cos\gamma+R_r} \\ 1 \end{bmatrix} \omega_r \tag{1}$$

Then, the position equation of the vehicles can be derived to avoid collision with drifts or obstacles; that is, the collision detection should be performed on the geometric shape of the vehicles, drifts, and obstacles. The Oriented Bounding Box (OBB) method was used to transform each entity into multiple bounding boxes in different directions for intersection testing. On the premise of authenticity, it is assumed that the front and rear bodies of the articulated vehicles are two rectangles, the width is w , and the length of the front and rear bodies are l_f and l_r , respectively, as shown in Figure 7.

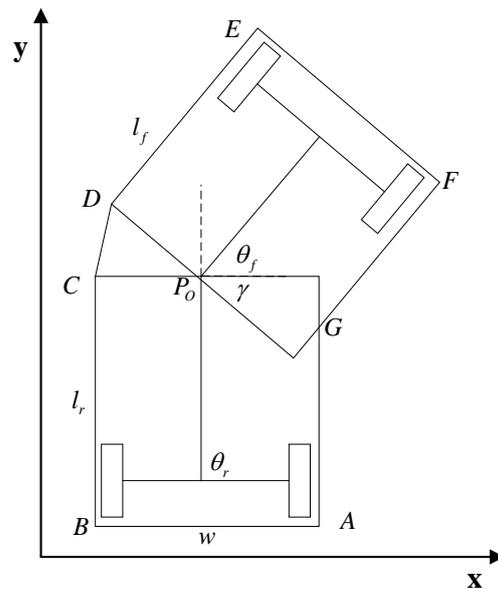


Figure 7. The geometric movement state of vehicles.

The OBB of the articulated vehicles ($Polygon_{car}$) can be represented by polygon ABCDEFG. According to the geometric relationship, the coordinates of each point of $Polygon_{car}$ can be expressed as the formulas

$$x_A = x_o - l_r\cos\theta_r + \frac{wsin\theta_r}{2} \tag{2}$$

$$y_A = y_o - l_r\sin\theta_r - \frac{wcos\theta_r}{2} \tag{3}$$

$$x_B = x_o - l_r\cos\theta_r - \frac{wsin\theta_r}{2} \tag{4}$$

$$y_B = y_o - l_r\sin\theta_r + \frac{wcos\theta_r}{2} \tag{5}$$

$$x_C = x_o - \frac{w \sin \theta_r}{2} \tag{6}$$

$$y_C = y_o + \frac{w \cos \theta_r}{2} \tag{7}$$

$$x_D = x_o - \frac{w \sin \theta_f}{2} \tag{8}$$

$$y_D = y_o + \frac{w \cos \theta_f}{2} \tag{9}$$

$$x_E = x_o + l_f \cos \theta_f - \frac{w \sin \theta_f}{2} \tag{10}$$

$$y_E = y_o + l_f \sin \theta_f + \frac{w \cos \theta_f}{2} \tag{11}$$

$$x_F = x_o + l_f \cos \theta_f + \frac{w \sin \theta_f}{2} \tag{12}$$

$$y_F = y_o + l_f \sin \theta_f - \frac{w \cos \theta_f}{2} \tag{13}$$

$$x_G = \frac{w \cos \theta_f^2 \cos \theta_r - w \cos \theta_f \cos \theta_r^2 - w \cos \theta_f \sin \theta_r^2 + w \cos \theta_r \sin \theta_f^2}{2(\cos \theta_f \sin \theta_r - \cos \theta_r \sin \theta_f)} + \frac{x_o \cos \theta_f \sin \theta_r - x_o \cos \theta_r \sin \theta_f}{(\cos \theta_f \sin \theta_r - \cos \theta_r \sin \theta_f)} \tag{14}$$

The collision can be detected by the intersection area between the map and the OBB of vehicles. If the vehicle is just within the feasible area of the map, it can be defined as no collision. We constructed the collision detection formula of the vehicle according to the $Polygon_{map}$ and $Polygon_{car}$, as shown in Formula (15).

$$Collision = \begin{cases} 1, & \emptyset < Polygon_{map} \cap Polygon_{car} < Polygon_{car} \\ 1, & Polygon_{map} \cap Polygon_{car} = \emptyset \\ 0, & Polygon_{map} \cap Polygon_{car} = Polygon_{car} \end{cases} \tag{15}$$

4. Improved RRT* Algorithm for Intelligent Vehicles

The RRT* algorithm has great advantages in search efficiency and search quality and has been successfully applied in unmanned vehicle driving, UAV navigation, etc.

For underground mines, the application of the RRT* algorithm must consider the following aspects:

- (1) The underground drift is long and narrow, and the available area of the entire map is small. The RRT* algorithm uses fixed-step full-map sampling, which results in low sampling efficiency in the scene of the drift map;
- (2) Drifts are usually constructed by a drilling and blasting method, and their surface will inevitably be irregular. As a result, the map of drifts cannot be as smooth as a regular road map, which will affect the smoothness of the solution path;
- (3) Underground vehicles are usually large in size, and the steering radius should be strictly controlled during their driving. Due to the randomness of the expansion, the RRT* algorithm cannot guarantee a path that meets the steering radius of the vehicles.

Aiming to adopt the use of intelligent vehicles in underground mines, this paper makes the following improvements to the RRT* algorithm:

- (1) Dynamic step size

The classic RRT* algorithm adopts a fixed step size expansion strategy. When the step size is small, the convergence speed is slow. When the step size is large, the vehicle easily collides with the drift wall, causing sampling failure and indirectly affecting the solution speed. The strategy of a fixed step size is: first, we randomly sampled x_{rand} in the map; secondly, we obtained its neighbor x_{near} ; then, we connected x_{rand} and x_{near} , and took the

length of the StepSize from x_{near} to obtain the point x_{new} ; if the collision detection was valid, an expansion was completed. Collision detection with a fixed step has a higher probability of failure. To solve this problem, a dynamic step size strategy was designed, and the step size was taken as a dynamic random function of CollisionSize (the distance from x_{near} to the collision point). When far from the obstacle, a larger step size was taken to ensure the speed of convergence; when the obstacle was closer, a smaller step size was taken to ensure the effectiveness of collision detection, as shown in Formula (16) and Figure 8.

$$DynamicSize = \begin{cases} StepSize & Collision = false \\ CollisionSize \times U[0,1] & Collision = true \end{cases} \quad (16)$$

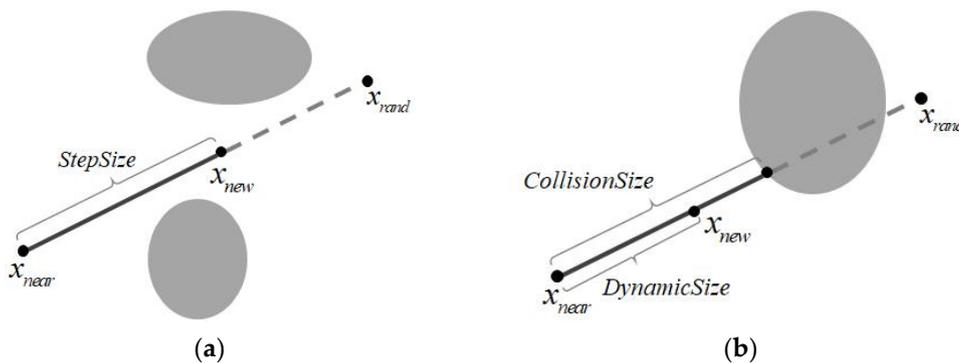


Figure 8. Comparison of fixed step size and dynamic step size. (a) Fixed step size; (b) dynamic step size.

(2) Steering angle constraints

The steering process of vehicles is strictly constrained by the max steering angle β . Therefore, during the sampling process of the RRT* algorithm, the angle θ between the new path and the parent path should be less than β , as shown in Formula (17) and Figure 9.

$$\theta = |\dot{\gamma}_f| = \arccos \left(\frac{\overrightarrow{x_{parent}x_{near}} \cdot \overrightarrow{x_{near}x_{new}}}{\left| \overrightarrow{x_{parent}x_{near}} \right| \left| \overrightarrow{x_{near}x_{new}} \right|} \right) \leq \beta \quad (17)$$

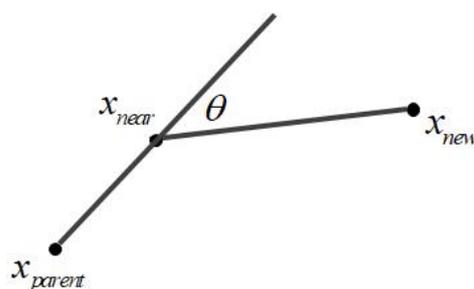


Figure 9. Geometric relation of steering angle constraints.

(3) Optimal tree reconnection

The classic RRT* algorithm uses random sampling, so the obtained path usually has certain turns, which lead to the deceleration of vehicles. Therefore, unnecessary turns should be avoided to lead the vehicles to drive straight. This will reduce the control difficulty of unmanned driving while reducing the path distance. The optimal tree reconnection process is as follows: we straightened and optimized the feasible path when the RRT* algorithm found a solution; we continuously traversed from the root node to the child

node; we searched for the child nodes that were directly connected without obstacles; we connected the two nodes and deleted the intermediate nodes. This process turned the path into a curve by reducing the number of nodes, as shown in Figures 10 and 11.

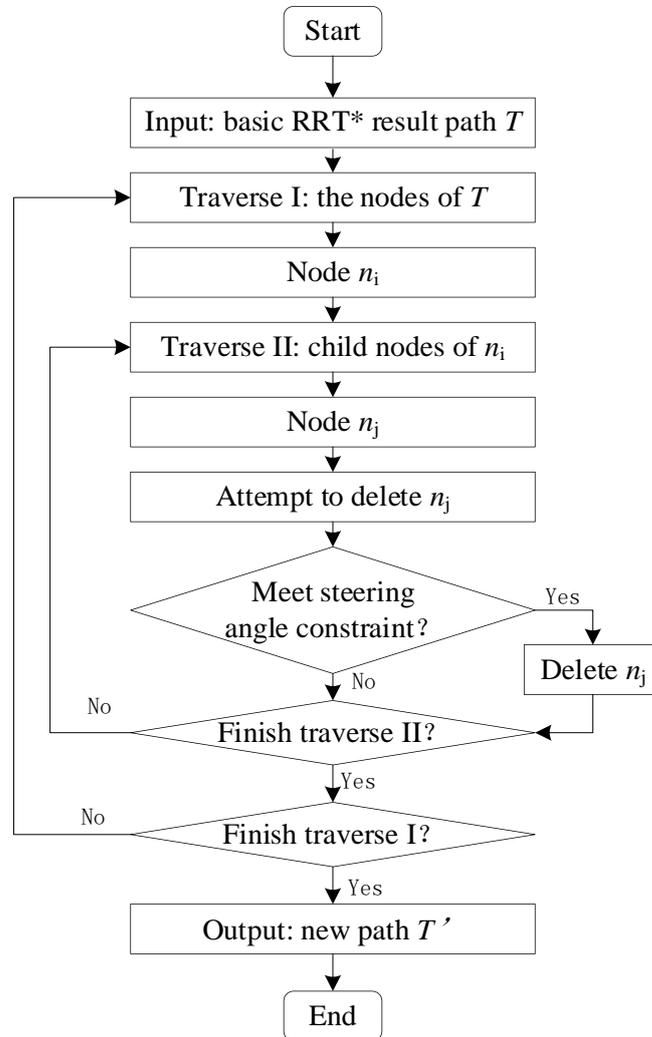


Figure 10. Process of optimal tree reconnection.

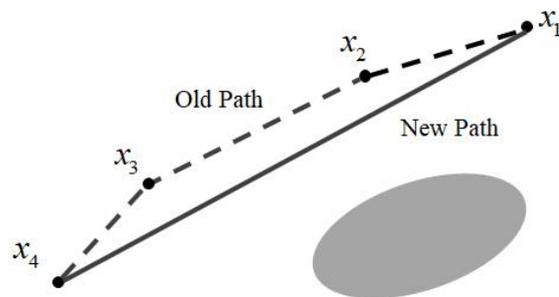


Figure 11. Geometric relation of optimal tree reconnection.

The pseudo code of the improved Algorithm 1 is as follows:

Algorithm 1 Improved RRT* Algorithm

Input: x_{start}, x_{goal}, Map
Output: A path T from x_{start} to x_{goal}

```

1  T.initialize();
2  for  $i = 1$  to  $n$  do
3    while true do
4       $x_{rand} \leftarrow \text{Sample}(Map)$ ;
5       $x_{near} \leftarrow \text{Near}(x_{rand}, T)$ ;
6       $DynamicSize \leftarrow \text{CollisionCheck}(x_{near}, Map)$ ;
7       $x_{new} \leftarrow \text{Steer}(x_{rand}, x_{near}, DynamicSize)$ ;
8      if  $\text{CollisionFree}(x_{new}, Map)$  and  $\text{Turnable}(x_{new}, x_{near}, x_{parent})$  then
9        break;
10     end
11   end
12    $X_{near\_neighbours} \leftarrow \text{NearNeighbour}(x_{new}, T)$ 
13   foreach  $x_{near\_neighbour} \in X_{near\_neighbours}$  do
14      $Test\_dis \leftarrow \text{Cost}(x_{new}) + \text{Distance}(x_{new}, x_{near\_neighbour})$ 
15     if  $\text{CollisionFree}(x_{new}, x_{near\_neighbour}, Map)$  and  $Test\_dis < \text{Cost}(x_{near\_neighbour})$  then
16        $x_{parent} \leftarrow \text{Parent}(x_{near\_neighbour})$ ;
17       Update( $T$ );
18     end
19   end
20   if  $x_{new} = x_{goal}$  then
21      $T \leftarrow \text{OptimalTreeReconnection}(T)$ ;
22     success();
23   end
24 end

```

5. Simulation Analysis**5.1. Simulation Environment**

In order to verify the adaptability of the improved RRT* algorithm, the classic RRT, the classic RRT*, and the improved RRT* algorithms are simulated and verified in the underground ore transportation scenario. The parameters of the vehicles come from the Scooptram ST3.5 diesel LHD, as shown in Figure 12 and Table 2. The verification map comes from a large underground mine in China, as shown in Figure 13a. The design size of the drifts was $4.4 \text{ m} \times 3.9 \text{ m}$. The ore is transported by an LHD from Stope #1 to Orepass #1. The map was preprocessed, and only the route of the LHD was retained. The simplified map is shown in Figure 13b.



Figure 12. Scooptram ST3.5 diesel LHD.

Table 2. Parameters of the Scooptram ST3.5 diesel LHD.

Parameter	Value
Max steering angle	42.5°
Width	2120 mm
Front body length	4130 mm
Rear body length	4330 mm

Data Source: Epiroc official website.

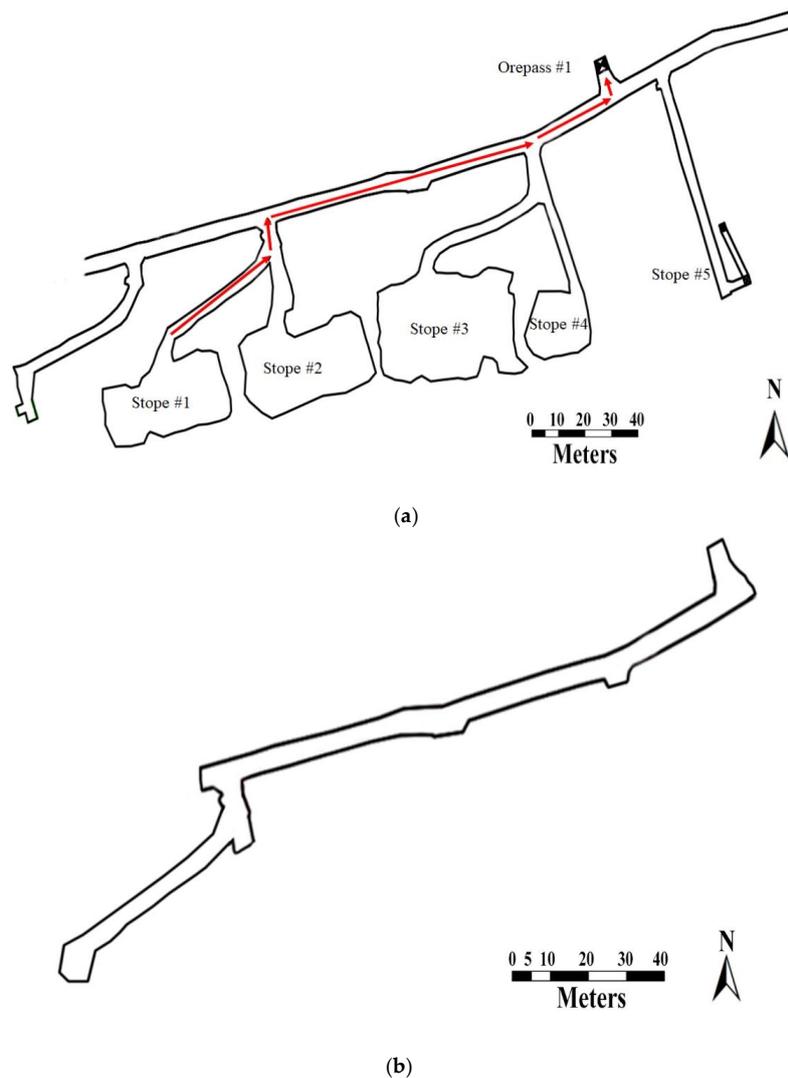


Figure 13. The map of the case study. (a) The original map; (b) the simplified map.

The case study simulated the operation process of the LHD from the stope to the orepass and verified the algorithm's ability to plan a feasible path in a long and narrow space. The LHD is required to complete ore transportation with the minimum distance under safe conditions and kinematic constraints. The simulation process was developed with Python 3.7, the operating system was Windows 10 × 64 bit, the CPU was Intel Core i7-8550U, and the memory was 16 GB. The simulation environment included Scipy 1.6.2, Shapely 1.8.0, and Matplotlib 3.3.4. Scipy was used to create the formulas. Shapely was used to calculate the OBB of vehicles and map polygons. Matplotlib was used to show the path.

5.2. Simulation Results

Comparative simulation experiments of the classic RRT algorithm, classic RRT* algorithm, and improved RRT* algorithm were carried out, and the results are shown in Figure 14. The red “X” represents the starting point and end point of the path planning, the blue line represents the wall of the drifts, and the horizontal and vertical axes represent the east and north coordinates. The yellow line represents the result of the classic RRT algorithm, the green line represents the result of the classic RRT* algorithm, and the red line represents the result of the improved RRT* algorithm.

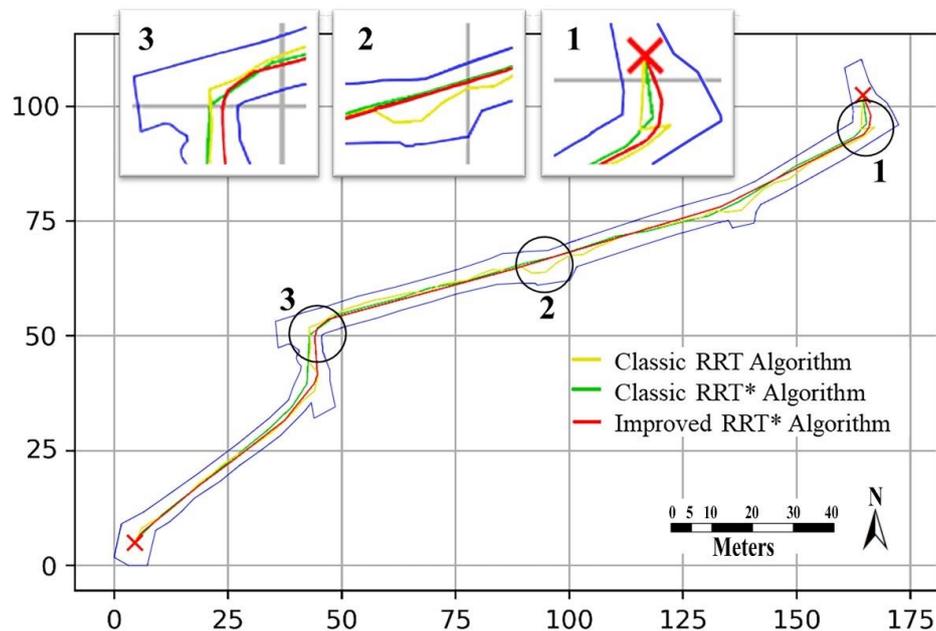


Figure 14. The simulation results.

It can be seen from Figure 13 that the path generated by the classic RRT algorithm had robust randomness, and there were a lot of irregular corners, such as Circle 1 and Circle 2. In contrast, the smoothness of the path generated by the classic RRT* algorithm was greatly improved, but the steering angle at the bend of the drift was too sharp, which was not suitable for the steering angle of the vehicles, such as Circle 2 and Circle 3.

Ten independent random simulations were performed on each algorithm to offset the random deviation of a single experiment. The results are shown in Table 3. The average path length obtained by the improved RRT* algorithm was much lower than that of the classic RRT algorithm but had only a small reduction compared with the classic RRT* algorithm. The main reason is that the reconnection in the classic RRT* algorithm can quickly approach the theoretically shortest time. The improved RRT* algorithm inherited this feature, and there was no more room for improvement. For the average search time, the performance of the improved RRT* algorithm was between the classic RRT algorithm and the classic RRT* algorithm. The same reason also led to the increment in average search nodes. Due to the optimal tree reconnection, the improved RRT* algorithm had a significant advantage over the classic algorithm in terms of average path nodes. This parameter reduced the control points during vehicle driving and reduced the difficulty of automatic driving. The steering angle constraints made the improved RRT* algorithm result fully meet the steering requirements, and the optimal tree reconnection increased the smoothness of the path, so the device can directly follow the path without further adjustment, avoiding multiple calculations. In general, the improved RRT* algorithm greatly improved the quality of the path while appropriately sacrificing the solution speed.

Table 3. Statistics of 10 independent random simulations.

Parameters	Classic RRT	Classic RRT*	Improved RRT*
Average path length (m)	211.11	189.86	189.54
Average search time (s)	168.94	44.16	86.12
Average of search node count	561.60	267.30	360.00
Average of path node count	32.00	28.80	16.20
Effective ratio of steering angle	81.87%	92.71%	100.00%

Obstacles in underground drifts are common, such as faulty vehicles and stacked materials. Further verification was conducted with known obstacles, as shown in Figure 15. Two scenarios were considered with both avoidable obstacles and unavoidable obstacles in the drift. The red line represents the final result, the yellow line represents the invalid leaf of a random tree, and the blue point represents the obstacle. For avoidable obstacles, the algorithm could pass them using a smooth curve without more additional sampling being necessary. For unavoidable obstacles, the algorithm stopped sampling after a certain number of samples.

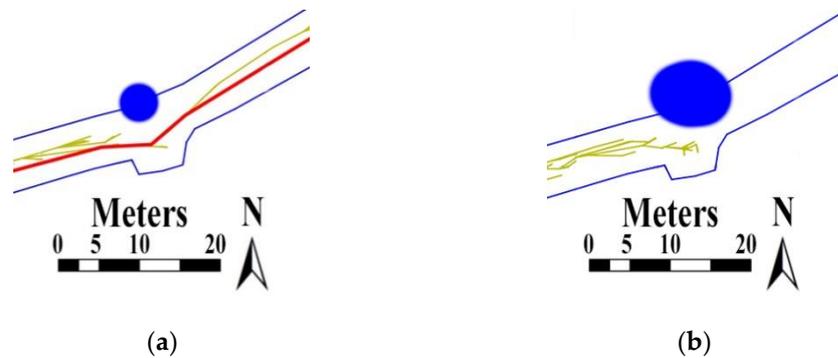


Figure 15. The simulation result with known obstacles. (a) With avoidable obstacles; (b) with unavoidable obstacles.

The kidnapping problem of intelligent vehicles might occur due to navigation failure or other reasons. For the verification of the kidnapping problem, we assumed that the vehicle planned to reach point B from point A but reached point B' for kidnapping reasons. Two scenarios were considered with both turnable kidnapping and unturnable kidnapping for the vehicle, as shown in Figure 16. For turnable kidnapping, it will reach the front point of the original path by the maximum steering angle. For unturnable kidnapping, it will drive astern to the back point of the original path by the maximum steering angle.

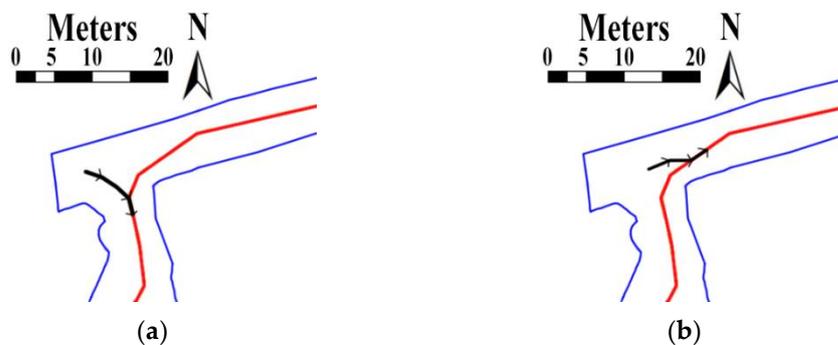


Figure 16. The simulation result for the kidnapping problem. (a) With turnable kidnapping; (b) with unturnable kidnapping.

5.3. Discussion

With the aim of the unmanned driving of intelligent vehicles in underground mines, we improved the path planning algorithm to adapt to the complex drift environment based on the RRT* algorithm. Many existing algorithms have to rasterize the map, but rasterized maps are not suitable for the drift environment. We constructed a vectorized drift environment map and then selected the RRT* algorithm to improve it. The vectorized map can effectively restore the details of the roadway environment and can also reduce the dataset. Combined with the articulated structure of underground intelligent vehicles, the dynamic characteristics were analyzed, and then the constraints were constructed. It strengthened the consideration of complex vehicle structures in this field. The process of the classical RRT* algorithm was analyzed, and then its shortcomings in adaptability to underground mining were extracted. On this basis, three improvements were proposed: a dynamic step size solved the algorithm efficiency problem; steering angle constraints solved the vehicle dynamics problem; optimal tree reconnection solved the control difficulty problem. By way of a simulation case study, the improved RRT* algorithm obtained a path suitable for underground intelligent vehicles within a reasonable time. Its results increased the effective ratio of the steering angle to 100%, fully met the vehicle's requirements, eliminated the secondary optimization of the path, greatly reduced the average number of path nodes, and simplified the vehicle's automatic driving control. Many existing algorithms have to rasterize the map.

However, we must admit that in order to achieve the path planning effect, a large number of invalid samples were discarded, which led to an increase in calculation time. This algorithm can improve the sampling efficiency and shorten the calculation time through parallel calculation. This will be improved in future research to further reduce the calculation time. In addition, the simulation case study was completed in this paper, but no on-site industrial experiment was carried out. The unmanned driving design of underground intelligent vehicles coordinates with multiple modules, including communication, sensors, SLAM, mechanical control, etc. It is also necessary to shut down some mining operations to ensure the safety of the experiment area. Due to these difficulties, this research only completed the path planning algorithm module, and in the future, an on-site industrial experiment will be completed after the preparation of each module.

6. Conclusions

This paper proposed a path planning method based on an improved RRT* algorithm for solving the problem of path planning for underground intelligent vehicles on an articulated structure and in drift environment conditions. Through a vectorized drift map and using the kinematics of vehicles, the constraints of articulated underground intelligent vehicles can be ascertained. The RRT* algorithm is an efficient sampling-based path planning algorithm, but it cannot meet the constraints of articulated underground intelligent vehicles. To solve this problem, this paper proposed an improved RRT* algorithm, including dynamic step size, steering angle constraints, and optimal tree reconnection. A simulation case study proved that the algorithm was effective and could solve the problem of underground intelligent vehicle path planning.

However, the method in this paper still has limitations, and future research will focus on the following aspects. (1) The solution time is still unsatisfactory because 86.12s cannot meet the application requirement for underground unmanned driving. Vehicles need to obtain a path within several seconds. A parallel calculation will be used to increase the solution speed and further reduce the calculation time. (2) There is still no joint debugging with intelligent vehicles. After the preparation of the industrial site, it will be combined with other modules to complete on-site industrial experiments and test the gap between the simulated and actual performance.

Author Contributions: Conceptualization, N.H. and H.W.; methodology, H.W.; software, J.H. and L.C.; validation, H.W., G.L. and L.C.; formal analysis, H.W.; data curation, L.C.; writing—original draft

preparation, H.W.; writing—review and editing, G.L. and N.H.; visualization, J.H.; supervision, N.H.; funding acquisition, G.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Natural Science Foundation of China, grant number 52074022 and National Key R&D Program of China, grant number 2018YFC0604400.

Data Availability Statement: The parameters of the Scooptram ST3.5 diesel LHD were obtained from <https://www.epiroc.com/en-us/products/loaders-and-trucks/diesel-loaders/scooptram-st3-5> (accessed on 12 December 2021). The source code associated with the algorithms introduced in the paper is available from the corresponding author upon request. The environment data are not available according to the privacy policy.

Acknowledgments: The authors would like to thank SD Gold for drift environment data support and Epiroc for LHD parameters support.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Guo, H.; Zhu, K.; Ding, C.; Li, L. Intelligent optimization for project scheduling of the first mining face in coal mining. *Expert Syst. Appl.* **2010**, *37*, 1294–1301. [\[CrossRef\]](#)
2. Li, J.G.; Zhan, K. Intelligent Mining Technology for an Underground Metal Mine Based on Unmanned Equipment. *Engineering* **2018**, *4*, 11. [\[CrossRef\]](#)
3. Lw, J.; Abrahamsson, L.; Johansson, J. Mining 4.0—The Impact of New Technology from a Work Place Perspective. *Min. Metall. Explor.* **2019**, *36*, 701–707.
4. Sánchez, F.; Hartlieb, P. Innovation in the Mining Industry: Technological Trends and a Case Study of the Challenges of Disruptive Innovation. *Min. Metall. Explor.* **2020**, *37*, 1385–1399. [\[CrossRef\]](#)
5. Chen, W.; Wang, X. Coal Mine Safety Intelligent Monitoring Based on Wireless Sensor Network. *IEEE Sens. J.* **2020**, *21*, 25465–25471. [\[CrossRef\]](#)
6. Roberge, V.; Tarbouchi, M.; LaBonte, G. Comparison of Parallel Genetic Algorithm and Particle Swarm Optimization for Real-Time UAV Path Planning. *Ind. Inform. IEEE Trans.* **2012**, *9*, 132–141. [\[CrossRef\]](#)
7. Tokekar, P.; Hook, J.V.; Mulla, D.; Isler, V. Sensor Planning for a Symbiotic UAV and UGV System for Precision Agriculture. *IEEE Trans. Robot.* **2016**, *32*, 1498–1511. [\[CrossRef\]](#)
8. Chugh, Y.; Parkinson, H. Automation trends in room-and-pillar continuous-mining systems in the USA. *J. S. Afr. Inst. Min. Metall.* **1981**, *81*, 57–65.
9. Scoble, M. Canadian mining automation evolution: The digital mine en route to minewide automation. *Int. J. Rock Mech. Min. Sci. Geomech. Abstr.* **1995**, *32*, 351A.
10. Kot-Niewiadomska, A. ROBOMINERS Project—The future of mining. In Proceedings of the 30th International Conference on the Present and Future of the Mining and Geology, Demanovska Dolina, Slovakia, 3–4 October 2019; Robominers: Demanovska Dolina, Slovakia, 2019; pp. 1–12.
11. Bearne, G. Innovation in mining: Rio Tinto’s Mine of the Future (TM) programme. *Alum. Int. Today* **2014**, *26*, 15.
12. Mullan, L. *Spearheading a Digital Transformation in the Canadian Mining Industry*; Swedish Mining Automation Group: Toronto, ON, Canada, 2017; pp. 1–3.
13. Altafini, C. Why to use an articulated vehicle in underground mining operations? In Proceedings of the 1999 IEEE International Conference on Robotics and Automation, Detroit, MI, USA, 10–15 May 1999; pp. 3020–3025.
14. Rodrigo, P. *Mining Haul Roads: Theory and Practice*; CRC Press: Boca Raton, FL, USA, 2019.
15. Khatib, O. Real-Time Obstacle Avoidance System for Manipulators and Mobile Robots. *Int. J. Robot. Res.* **1986**, *5*, 90–98. [\[CrossRef\]](#)
16. Luo, G.; Yu, J.I.; Mei, Y.O.; Zhang, S.Y. UAV Path Planning in Mixed-Obstacle Environment via Artificial Potential Field Method Improved by Additional Control Force. *Asian J. Control* **2015**, *17*, 1600–1610. [\[CrossRef\]](#)
17. Malone, N.; Chiang, H.T.; Lesser, K.; Oishi, M.; Tapia, L. Hybrid Dynamic Moving Obstacle Avoidance Using a Stochastic Reachable Set-Based Potential Field. *IEEE Trans. Robot.* **2017**, *33*, 1124–1138. [\[CrossRef\]](#)
18. Yang, W.; Wu, P.; Zhou, X.; Lv, H.; Liu, X.; Zhang, G.; Hou, Z.; Wang, W. Improved Artificial Potential Field and Dynamic Window Method for Amphibious Robot Fish Path Planning. *Appl. Sci.* **2021**, *11*, 2114. [\[CrossRef\]](#)
19. Seder, M.; Baotic, M.; Petrovic, I. Receding Horizon Control for Convergent Navigation of a Differential Drive Mobile Robot. *IEEE Trans. Control Syst. Technol.* **2017**, *25*, 653–660. [\[CrossRef\]](#)
20. Saad, M.; Salameh, A.I.; Abdallah, S.; El-Moursy, A.; Cheng, C.T. A Composite Metric Routing Approach for Energy-Efficient Shortest Path Planning on Natural Terrains. *Appl. Sci.* **2021**, *11*, 6939. [\[CrossRef\]](#)
21. Biyela, P.; Rawatlal, R. Development of an optimal state transition graph for trajectory optimisation of dynamic systems by application of Dijkstra’s algorithm. *Comput. Chem. Eng.* **2019**, *125*, 569–586. [\[CrossRef\]](#)
22. Goerzen, C.; Kong, Z.; Mettler, B. A Survey of Motion Planning Algorithms from the Perspective of Autonomous UAV Guidance. *J. Intell. Robot. Syst.* **2010**, *57*, 65. [\[CrossRef\]](#)

23. Christie, G.A.; Shoemaker, A.; Kochersberger, K.B.; Tokekar, P.; Mclean, L.; Leonessa, A. Radiation search operations using scene understanding with autonomous UAV and UGV. *J. Field Robot.* **2017**, *34*, 1450–1468. [[CrossRef](#)]
24. Zhuang, H.; Dong, K.; Qi, Y.; Wang, N.; Dong, L. Multi-Destination Path Planning Method Research of Mobile Robots Based on Goal of Passing through the Fewest Obstacles. *Appl. Sci.* **2021**, *11*, 7378. [[CrossRef](#)]
25. Kavraki, L.E.; Svestka, P.; Latombe, J.C.; Overmars, M.H. Probabilistic Roadmaps for Path Planning in High-Dimensional Configuration Spaces. *IEEE Trans. Robot. Autom.* **1996**, *12*, 566–580. [[CrossRef](#)]
26. Park, B.; Chung, W.K. Efficient environment representation for mobile robot path planning using CVT-PRM with Halton sampling. *Electron. Lett.* **2012**, *48*, 1397–1399. [[CrossRef](#)]
27. LaValle, S.M.; Kuffner, J.J., Jr. Randomized kinodynamic planning. *Int. J. Robot. Res.* **2001**, *20*, 378–400. [[CrossRef](#)]
28. Cui, R.; Yang, L.; Yan, W. Mutual Information-Based Multi-AUV Path Planning for Scalar Field Sampling Using Multidimensional RRT*. *IEEE Trans. Syst. Man Cybern. Syst.* **2017**, *46*, 993–1004. [[CrossRef](#)]
29. Shome, R.; Solovey, K.; Dobson, A.; Halperin, D.; Bekris, K.E. dRRT*: Scalable and informed asymptotically-optimal multi-robot motion planning. *Auton. Robot.* **2020**, *44*, 443–467. [[CrossRef](#)]
30. Lavalle, S.M. Rapidly-Exploring Random Trees: A New Tool for Path Planning. *Res. Rep.* **1998**, RT98-11, 1–4.
31. Karaman, S.; Frazzoli, E. Optimal Kinodynamic Motion Planning using Incremental Sampling-based Methods. In Proceedings of the IEEE Conference on Decision & Control, Atlanta, GA, USA, 15–17 December 2010; pp. 7681–7687.
32. Ma, X.; Mao, R. Path planning for coal mine robot to avoid obstacle in gas distribution area. *Int. J. Adv. Robot. Syst.* **2018**, *15*, 172988141775150. [[CrossRef](#)]
33. Mauricio, A.; Nieves, A.; Castillo, Y.; Hilaraca, K.; Fonseca, C.; Gallardo, J.; Rodríguez, R.; Rodríguez, G. Multi-robot exploration and mapping strategy in underground mines by behavior control. In *Multibody Mechatronic Systems*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 101–110.
34. Papachristos, C.; Khattak, S.; Mascarich, F.; Dang, T.; Alexis, K. Autonomous Aerial Robotic Exploration of Subterranean Environments relying on Morphology-aware Path Planning. In Proceedings of the 2019 International Conference on Unmanned Aircraft Systems (ICUAS), Atlanta, GA, USA, 11–14 June 2019; pp. 299–305.
35. Gamache, M.; Grimard, R.; Cohen, P. A shortest-path algorithm for solving the fleet management problem in underground mines. *Eur. J. Oper. Res.* **2005**, *166*, 497–506. [[CrossRef](#)]
36. Larsson, J.; Broxvall, M.; Saffiotti, A. Flexible infrastructure free navigation for vehicles in underground mines. In Proceedings of the 2008 IEEE 4th International IEEE Conference Intelligent Systems, Varna, Bulgaria, 6–8 September 2008; pp. 2-45-2-50.
37. Tian, Z.; Gao, X.; Zhang, M. Path planning based on the improved artificial potential field of coal mine dynamic target navigation. *J. China Coal Soc.* **2016**, *41* (Suppl. S2), 589–597.
38. Yuan, X.; Hao, M. Research on key technology of coal mine auxiliary transportation robot. *Ind. Mine Autom.* **2020**, *46*, 8–14.
39. Dang, T.; Mascarich, F.; Khattak, S.; Papachristos, C.; Alexis, K. Graph-based path planning for autonomous robotic exploration in subterranean environments. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 3–8 November 2019; pp. 3105–3112.
40. Song, B.; Miao, H.; Xu, L. Path planning for coal mine robot via improved ant colony optimization algorithm. *Syst. Sci. Control Eng.* **2021**, *9*, 283–289. [[CrossRef](#)]
41. Bai, Y.; Hou, Y.B. Research of environmental modeling method of coal mine rescue snake robot based on information fusion. In Proceedings of the 2017 IEEE 20th International Conference on Information Fusion (Fusion), Xi'an, China, 10–13 July 2017; pp. 1–8.
42. Ma, T.; Lv, J.; Guo, M. Downhole robot path planning based on improved D* algorithm. In Proceedings of the 2020 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC), Macau, China, 21–24 August 2020; pp. 1–5.
43. Liu, Z. Application of handheld 3D laser scanner based on slam technology in tongkuangshan mine. *China Mine Eng.* **2021**, *50*, 13–16. [[CrossRef](#)]
44. Peter, D. *SME Mining Engineering Handbook*; Society for Mining, Metallurgy, and Exploration: Englewood, CO, USA, 2011; pp. 1148–1149.
45. Bai, G.; Liu, L.; Meng, Y.; Luo, W.; Gu, Q.; Ma, B. Path Tracking of Mining Vehicles Based on Nonlinear Model Predictive Control. *Appl. Sci.* **2019**, *9*, 1372. [[CrossRef](#)]