

Article

A Secure Trajectory Planning Method for Connected Autonomous Vehicles at Mining Site

Han Li, Yiming Li, Peng Chen *, Guizhen Yu and Yaping Liao

School of Transportation Science and Engineering, Beihang University, Beijing 150028, China; leehan@buaa.edu.cn (H.L.)

* Correspondence: cpeng@buaa.edu.cn

Abstract: Recently, with the assistance of 5G networks and the Internet of Things, specialized applications of autonomous driving to mining sites have been explored, with the goal of realizing the unmanned operation of mining systems and enhancing the safety of the mining industry. After receiving the loading task, the autonomous driving system will generate a feasible trajectory for the mining truck. It requires that the trajectory be generated in advanced within a limited-time high-latency network. In addition, the secure trajectory planning for mining sites involves factors in the complex environment and an unstable network. Thus, a secure trajectory planning method for autonomous trucks at mining sites is proposed. It simplifies the planning by decoupling the planning into front-end path searching and back-end trajectory generation. First, the planner enhances the Hybrid A* search algorithm to find the hauling path within the boundary of the mining site, and then, it post-processes the path with a well-designed symmetric optimization-based method. Then, considering the interaction with other autonomous trucks, a topology-guided search method for secure decision making is proposed, considering the possibility of cybersecurity. The proposed method was validated in real scenarios of the mining environment. The results verify that the planner can generate the secure trajectory under network delay 2.0 s conditions.

Keywords: connected autonomous vehicle; secure trajectory planning; mining site; cyberattack



Citation: Li, H.; Li, Y.; Chen, P.; Yu, G.; Liao, Y. A Secure Trajectory Planning Method for Connected Autonomous Vehicles at Mining Site. *Symmetry* **2023**, *15*, 1973. <https://doi.org/10.3390/sym15111973>

Academic Editor: Christos Volos

Received: 4 October 2023

Revised: 20 October 2023

Accepted: 22 October 2023

Published: 25 October 2023



Copyright: © 2023 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

Recently, autonomous vehicles have been applied for transporting ore at mining sites [1–3]. The aim is to improve productivity by decreasing labor costs and accidents and increasing working time. Trajectory planning is the core technology of autonomous vehicles. Since the haul roads at mining sites have typical characteristics of a large curvature and large planning area, which bring network latency to the planning module, it is of great significance to investigate a trajectory planning method suitable for the off-road environment of mining sites, as shown in Figure 1.

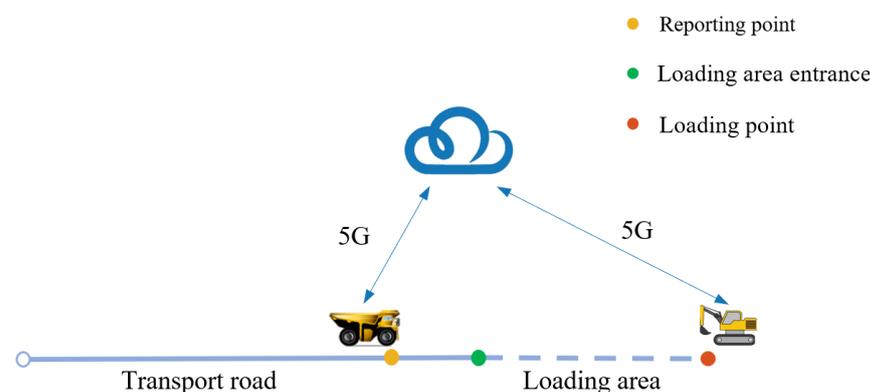


Figure 1. The network flow of autonomous mining trucks at a mining site.

Numerous studies have been conducted in the field of trajectory planning over the past few decades, including path planning [4] and motion planning [5,6]. Two comprehensive surveys of these methods and strategies can be found in [7,8]. However, most of them deal with autonomous driving for the on-road environment and cannot be adapted to a mining site with an unstable network.

As for the secure trajectory planning at mining sites, Maekawa and Noda employed a path generation method for autonomous driving utilized in mining operations. The method applied Dijkstra's algorithm and cubic B-spline curves [9] to generate collision-free paths and then adjusted the control points of the B-spline curves for continuous curvature and safety. Similar to the previous study, Suzuki and Usami [10] proposed an algorithm for generating collision-free two-lane paths for autonomous driving at mining sites. It used the A* algorithm to generate the initial path and then performed interpolation using quartic (degree 4) B-spline curves, which relaxed the curvature change rate. Usami et al. [11] introduced a two-lane path planning algorithm for autonomous vehicles in a two-and-a-half-dimensional environment. The method extended path planning to two-and-a-half dimensions and designed cross-slopes in high-curvature regions of the path. The cross-slopes guaranteed that vehicles would not need to slow down in high-curvature regions. Zhang et al. [12] proposed a hybrid trajectory planning method, including path planning and speed planning, in a highly constrained environment. For path planning, the method employed a heuristic A* graph search algorithm [13] to obtain the global path and then introduced a multiple-stage sampling algorithm for optimization. For speed planning, the method utilized an optimization-based method to construct the speed profile. In addition, numerous methods have been proposed for autonomous parking, which conforms to the truck operation at mining sites. Dolgov et al. [13] proposed a practical search method for autonomous driving. The first stage generated the global path using a heuristic search algorithm with non-holonomic constraints; the second stage used conjugate gradient (CG) descent to post-process the global path. Chen and Wu [14] introduced a trajectory planning method by solving an optimal control problem. The Gaussian pseudo-spectral method was applied to obtain the parking trajectory. By utilizing a homotopic method, the obstacle avoidance was transformed into constraints, and the solution was set as the initial guess for offline planning.

Oliveira et al. [15] proposed a sharpness-continuous Dubins-like planner for heavy-duty vehicles, which replaced the clothoid segments to achieve sharpness continuity based on a smoother kinematic model [16]. Due to the existence of obstacles at the mining site, it is difficult to generate the hauling path only using the curve-based method (e.g., the continuous curvature steer) without resorting to the search method. To the best of our knowledge, most of the previous studies dealing with the trajectory planning problem at mining sites incurred high computational costs, and they tended to ignore the interactions between autonomous vehicles. Thus, an online trajectory planning method is proposed in this study for autonomous mining trucks (AMTs) at mining sites. It considers the interaction between AMTs based on a hierarchical structure consisting of two modules: path searching (front-end) and trajectory generation (back-end). The front-end finds the haul path while adhering to nonholonomic constraints. The back-end conducts the decision making and generates a feasible speed profile based on the global transport task and surrounding environment information.

The contributions of this study can be summarized as follows: (1) Considering the network condition of the mining site and AMTs, this study provided an improved Hybrid A* (IHA*) with a comprehensive evaluation function and dynamic expanding step. In addition, this study established a dynamic time interval and a symmetric switching corridor in the optimization, thereby enhancing the quality of the haul path and the success rate of finding solutions. (2) A topology-guided maneuver decision-making algorithm was proposed for the secure planning under cyberattack, which enables maintaining a safe headway with other AMTs and find the maneuver sequence in a cheap way. Within a

limited distance, the nonlinear programming (NLP) method is able to minimize the arriving time and obtain the speed profile based on the maneuver sequence.

The rest of this study is organized as follows. Section 2 presents the problem description. Section 3 provides an IHA* algorithm and the optimization-based method for post-processing the haul path. Section 4 illustrates the decision-making process and the generation of speed profile. Last, Section 5 shows the experiment results, and Section 6 concludes this study.

2. Problem Description

2.1. Secure Trajectory Planning Problem

Figures 1 and 2 depict the workflow of AMTs, which arrive at the reporting point and receive their loading tasks from the cloud platform. The cloud platform specifies the loading task according to the operating state of the mining transportation system. If the network has latency or cybersecurity, the planning module may fail to generate a feasible trajectory before the AMT reaches the entrance of the loading area. In such situations, the AMT will stop and wait, and the engine will idle. It is only when a planned trajectory is available that the AMT will accelerate from zero, which consumes considerable fuel and decreases the operation efficiency. Figure 2a provides an illustration of a haul road at an actual mining site, which serves as the study scenario to explore the trajectory planning problem. The mining areas and haul roads are depicted in gray and yellow, respectively, with a width of 400 m and a length of 600 m.

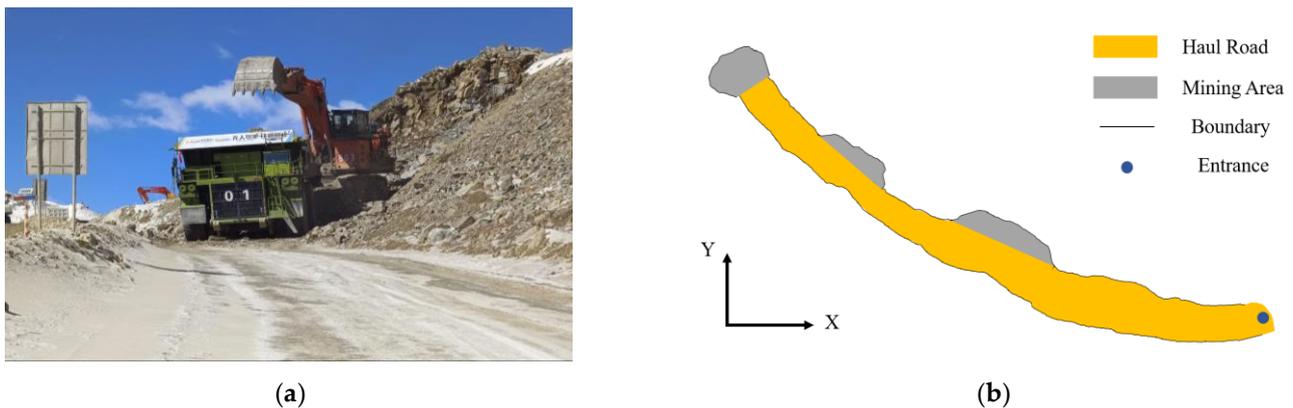


Figure 2. Illustration of trajectory planning. (a) An autonomous mining truck in operation. (b) Top view of mining site (600 × 400).

Compared to path planning in a public road scenario, path planning in a mining site scenario presents several distinct challenges: (1) the obstacles at mining site are irregular, with some areas being vast while others are narrow; (2) the minimum turning radius of the AMT is large, and the planner is required to generate a smoother path than that on public roads; (3) the large planning area results in much longer paths, which require significant computational effort; (4) the planner should be capable of interacting with other AMTs to avoid collisions, especially in the event of a cyberattack.

2.2. Vehicle Kinematic Modeling

Table 1 presents the main parameters of the dump truck. The dynamic model for the truck can be simplified based on the three-degree-of-freedom model. However, compared to the passenger car, it is difficult to calibrate and validate the model for the following reasons: (1) there is no dedicated testbed for dump trucks to accurately estimate their inertia; (2) due to significant load variation and tire wear, it is difficult to estimate the cornering stiffness. Moreover, due to the limitation of road curvature at mining site, the

speed of trucks is limited to low values. Thus, the kinematic motion model [17] is suitable to be used in this study, which is formulated as shown below.

$$\begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{\theta}(t) \\ \dot{v}(t) \end{bmatrix} = \begin{bmatrix} v(t) \cos(\theta(t)) \\ v(t) \sin(\theta(t)) \\ \frac{v(t) \tan(\delta(t))}{l_f} \\ a(t) \end{bmatrix} \quad (1)$$

where \dot{x} and \dot{y} denote the derivatives of central point (x, y) of the vehicle, respectively; v and a denote the velocity and acceleration, respectively; and θ and δ denote the orientation angle and steering angle, respectively.

Table 1. Main parameters of dump truck.

Parameters	Value	Description	Unit
w_v	9.4	Vehicle width	m
l_v	15.35	Vehicle length	m
l_f	6.0	Wheeling length	m
m_v	330.0	Vehicle mass	ton
h_v	7.82	Vehicle height	m
ρ_{\min}	16.2	Minimum turning radius	m
v_{\max}	59.0	Maximum velocity	km/h
l_{heavy}	240.0	Rated dead weight	ton

3. Generation of Path

The method for generating the haul path is introduced in this section. Given the boundary information, the position, and the orientation of the loading points, the planner initially generates an initial path. To ensure smoothness and completeness, the path is then converted into a trajectory and refined using an optimization-based method, which minimizes the penalties related to travel distance, smoothness, and collision cost.

3.1. Secure Path Generation

Due to the large planning area of the mining site, previous search algorithms are unable to meet real-time requirements with a short expansion step. If the step is too large, the algorithms may become trapped in a narrow planning area because they cannot generate a safe sub-node using the defined motion primitives.

To generate a safe path, the potential field of the planning space is computed by the Voronoi field [18] based on the boundary information. Then, a heuristic searching algorithm is employed to search the path between the start position and goal position [13].

3.1.1. Comprehensive Evaluation Function

The evaluation function is defined by a linear combination of the trajectory cost function $g(n)$, heuristic function $h(n)$, and obstacle function $o(n)$:

$$f(n) = g(n) + \alpha h(n) + o(n) \quad (2)$$

$$g(n) = g(n_p) + d_{\text{step}}(n) \quad (3)$$

$g(n)$ denotes the cost of the path from the start position to node n , where n_p refers to the parent node of n , and $d_{\text{step}}(n)$ is the expanding distance [12] for generating the next node candidates. The parameter α is the heuristic factor, which determines the weight of the heuristic function. A small α can result in a heavy computation burden during the search process, while a large α can accelerate the searching process by exploring the nodes close to the goal. However, an excessively large α may lead to longer path lengths. To strike a balance between computation efficiency and optimality results in the search process, the

heuristic factor α is assigned a constant value of 1.1 based on simulation experiments. The heuristic function $h(n)$ denotes the cost of the path from node n to the goal, which is defined as follows:

$$h(n) = h_{RS} \quad (4)$$

where h_{RS} denotes the length of the Reeds–Shepp (RS) curve from the current node to the goal. It considers the kinematic constraints by setting the maximum curvature in the RS curve. The essential reason for adopting the RS model rather than the continuous curvature steer method is that the RS model can achieve higher computational efficiency and the shortest path length, which can reflect the heuristic value more accurately [19].

Compared to the conventional Hybrid A*, IHA* removes the heuristic function related to the path length of A*, specifically, the holonomic-with-obstacles path [20]. This decision is based on the complex mining environment and truck–shovel operations, where loading positions and road boundaries may change in real time. The planner recalculates the path length using A* for each task. The obstacle function $o(n)$ is defined as follows:

$$o(n) = \begin{cases} 0 & d_{\text{obs}}(n) > d_{\text{thr}} \\ (d_{\text{obs}}(n) - d_{\text{thr}})^2 & d_{\text{obs}}(n) \leq d_{\text{thr}} \end{cases} \quad (5)$$

The obstacle function evaluates the cost from node n to the closest obstacle. d_{thr} denotes the threshold of obstacle clearance and $d_{\text{obs}}(n)$ the shortest collision distance between node n and the closest obstacle. In this study, the shortest collision distance was directly obtained by the Voronoi field.

3.1.2. Dynamic Expanding Step

According to the distribution of obstacle in the planning area, IHA* adjusts the expanding step to find the search results. The expanding step, denoted as $d_{\text{step}}(n)$, for each node is dynamically adjusted based on the surrounding information.

$$d_{\text{step}}(n) = \max(d_{\text{min}}, d_{\text{dy}}) \quad (6)$$

$$d_{\text{dy}} = \min(d_{\text{obs}}(n), d_{\text{max}}(n)) \quad (7)$$

where d_{dy} is the dynamic expanding step, and d_{min} and d_{max} represent the minimum and maximum expanding steps, respectively. It should be noted that many previous studies determined the expanding step based on the truck's center of mass. While such an approach might be suitable for passage cars, it is less appropriate for dump trucks with their larger size, which can be up to four times wider than a passage car. In this case, each corner of the dump truck, such as the left, middle, and right, has a different relative distance to the boundary of the driving area or obstacles, as shown in Figure 3.

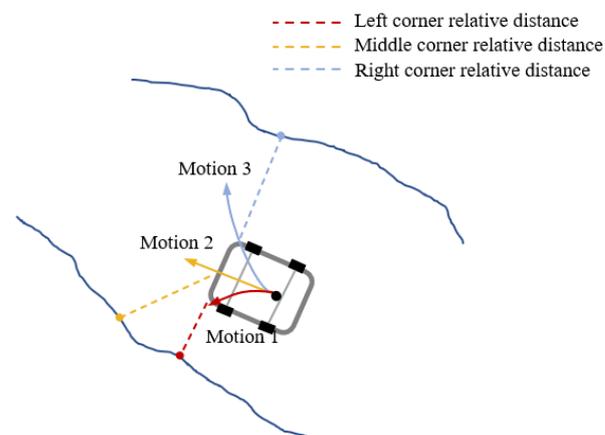


Figure 3. Dynamic expanding step considering the large size of AMT.

The larger the relative distance, the larger the expansion step size. Thus, this study employs a point mass model to explore three motion primitives based on the expanding steps at three different corners. This approach of using adaptive expansion step sizes enhances computational efficiency while ensuring the quality of the planned path. The state transition from point (x_n, y_n, θ_n) to $(x_{n+1}, y_{n+1}, \theta_{n+1})$ is defined as follows:

$$\begin{aligned} x_{n+1} &= x_n + d_{\text{step}}^{\text{mo}}(n)(\Delta x_{\text{mo}} \cos(\theta_n) - \Delta y_{\text{mo}} \sin(\theta_n)) \\ y_{n+1} &= y_n + d_{\text{step}}^{\text{mo}}(n)(\Delta x_{\text{mo}} \sin(\theta_n) + \Delta y_{\text{mo}} \cos(\theta_n)) \end{aligned} \quad (8)$$

where Δx_{mo} , Δy_{mo} , and $d_{\text{step}}^{\text{mo}}(n)$ represent different motion primitives (left, straight and right) and their corresponding expanding step sizes. In this study, the motion primitive with $d_{\text{obs}}(n)$ less than d_{lim} will be abandoned according to engineering experience.

3.1.3. Bi-Directional Search

This study introduces a bi-directional search strategy that expands simultaneously from both the endpoint to the starting point. The extension node serves as a sub-target point, and when the forward search is extended to this sub-target node, it connects the front and back paths. It should be noted that employing a greedy search strategy in reverse search can rapidly expand nodes toward the starting point due to the narrow nature of stopping endpoints. Consequently, the cost evaluation for sub-target nodes solely encompasses the trajectory cost function, as shown in Equation (3).

3.2. Secure Path Post-Processing

The raw path generated by IHA* can be suboptimal though it meets all constraints. There are several reasons to improve the path quality:

- The discontinuities of the curvature profile could lead to jitters if directly provided to the low-level control module.
- The relatively longer length of the raw path could result in increased fuel consumption.
- Although IHA* avoids collision during expansion, the resulting path may be close to the boundary of the haul road.

Thus, post-processing is implemented using an optimization-based (OP) method to refine the path. The details of the OP method are described as follows: for the raw path defined by N points (p_0, p_1, \dots, p_N) , it is interpolated with a B-spline [9] and converted into a trajectory (s_0, s_1, \dots, s_N) with the desired speed. Each trajectory point includes state variables $(x, y, \theta, v, \Delta t)$ and control variables (a, δ) , which can be obtained by the pure-pursuit [21] and constant velocity method [22]. Therefore, at the beginning, the acceleration a is set as zero, and the time interval Δt is set as a uniform value of 0.1 s. The steering angle δ for the i_{th} point is calculated by:

$$\delta_i = \text{atan} \left(\frac{(\theta_{i+1} - \theta_i)l_f}{\sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}} \right) \quad (9)$$

where l_f is the front wheel length of the truck. Instead of using a fixed time interval during the refinement [23,24], the method adds the schedule of dynamic time allocation. In other words, the time intervals between two adjacent points differ from each other. This approach enhances the flexibility of the optimization method. The path generated and its corresponding control variables will be used as input for the OP method.

3.2.1. Optimization Function

The optimization variables $z \in R^{7N}$ consist of vehicle states $(x, y, \theta, v, \Delta t)$ and control variables (a, δ) at each step n . The overall objective function f_{total} is defined as the sum of consistency, safety, dynamical feasibility, and optimality.

$$f_{\text{total}}(z) = w_{\text{ref}}f_{\text{ref}} + w_c f_c + w_d f_d + w_s \int ds \quad (10)$$

where f_{ref} denotes the reference cost, and f_c and f_d denote the collision and control derivatives costs, respectively. $\int ds$ denotes the cost of the path length. Additionally, w_{ref} , w_c , w_d , and w_s are the weights of the reference, collision, control derivatives and path length costs, respectively. The reference cost evaluates the difference between the raw path P^* and the final haul path. Its function aims to maintain path consistency. The relative distance between P_i and P_i^* is defined as follows:

$$f_{\text{ref}} = \sum_{i=0}^N \frac{\|P_i - P_i^*\|}{2} \quad (11)$$

Different from an on-road environment, the mining site owns numerous irregular boundaries rather than polygonal obstacles. The boundary of the haul road cannot be represented by a specific shape. Thus, this study establishes a collision cost function as a soft constraint to guarantee the path safety:

$$f_c = \sum_{i=0}^N f_c^i \quad (12)$$

where

$$f_c^i = \begin{cases} d_{\text{obs}}^i & d_{\text{obs}}^i \leq d_{\text{thd}} \\ 0 & d_{\text{obs}}^i > d_{\text{thd}} \end{cases} \quad (13)$$

Similarly, the control derivatives cost f_d is formulated as:

$$f_d = \sum_{i=1}^N \left(\frac{u_i - u_{i-1}}{\Delta t_{i-1}} \right)^2 \quad (14)$$

where u denotes the control variables $\{a, \delta\}$, and Δt_{i-1} is the time interval between the $i-1$ th and i th path points. The trajectory is represented as a set of N discrete points set. The travel time t_{total} of the optimized path is calculated as follows:

$$t_{\text{total}} = \sum_{i=0}^N \Delta t_i \quad (15)$$

By minimizing the derivatives of control variables, the smoothness and feasibility of the path can be significantly improved [25]. The relative distance Δs is defined as follows:

$$f_s = \sum_{i=0}^N \Delta s_i = \|P_i - P_{i-1}\|_2 \quad (16)$$

3.2.2. Nonholonomic Constraints

The trajectory must be generated while adhering to the nonholonomic constraints during optimization. Based on the kinematic model in Equation (1), the nonholonomic constraints are formulated as follows:

$$\dot{z}_{i+1} = f(z_i, u_i, \Delta t_i) \quad (17)$$

where Δt_i denotes the time interval between i and $i + 1$, and u_i represents the control variables (a, δ) at time i . It is important to ensure that the velocity along the trajectory does not exceed the maximum value v_{\max} defined by the global task. Additionally, the control variables and their change rates should be constrained within the permitted range to ensure dynamical feasibility:

$$\begin{aligned} v_{\min} \leq v_i \leq v_{\max} \text{ and } a_{\min} \leq a_i \leq a_{\max} \\ \delta_{\min} \leq \delta_i \leq \delta_{\max} \\ \left| \frac{\delta_i - \delta_{i-1}}{\Delta t_{i-1}} \right| \leq \dot{\delta}_{\max}, \Delta t_i > 0 \end{aligned} \quad (18)$$

During the loading and unloading process, AMTs need to conduct backward maneuvers to reach specific positions and orientations. In the IHA* algorithm, the RS curve [26] is utilized to accelerate the search efficiency. The parking path is obtained by connecting the start and goal points with the RS curve, as shown in Figure 4.

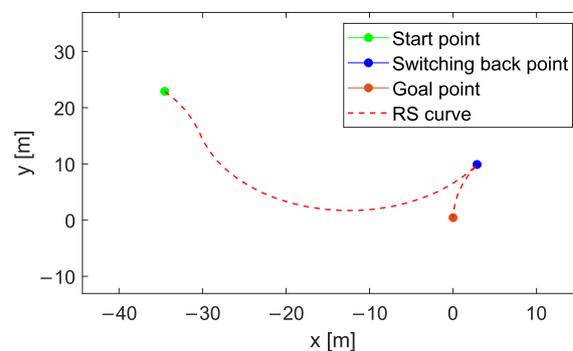


Figure 4. RS curves with switching-back points.

Although the generation of curves is constrained by the minimum turning radius, the curvature discontinuity is usually ignored in the expanding of the RS curve, which poses challenges to the control module. The actuators (e.g., steer and brake) of the dump truck are all hydraulically controlled, which leads to a large delay (about 2 s) from command sending to state response. Zhang [20] smoothed the forward driving and reverse driving path, respectively, which fixed the switching point at the original position. In such a case, the optimality of the path might be compromised [27]. In this study, the planner sets up a corridor where the switching-back point can be adjusted to guarantee the continuous curvature and short path length. By referring to the minimum turning radius, the upper and lower bounds of the switching symmetric corridor can be defined:

$$\begin{aligned} x_{\min} &= \min(x_s, x_{sw}, x_s^{lf}, x_s^{rf}, x_g, x_g^{lf}, x_g^{rf}), \\ x_{\max} &= \max(x_s, x_{sw}, x_s^{lf}, x_s^{rf}, x_g, x_g^{lf}, x_g^{rf}), \\ y_{\min} &= \min(y_s, y_{sw}, y_s^{lf}, y_s^{rf}, y_g, y_g^{lf}, y_g^{rf}), \\ y_{\max} &= \max(y_s, y_{sw}, y_s^{lf}, y_s^{rf}, y_g, y_g^{lf}, y_g^{rf}) \end{aligned} \quad (19)$$

where (x_s, y_s) , (x_g, y_g) , and (x_{sw}, y_{sw}) denote the coordinates of P_s , P_g and P_{sw} , respectively. (x_{slf}, y_{slf}) and (x_{srf}, y_{srf}) represent the coordinates of P_{slf} and P_{srf} where the vehicle conducts the quarter circular motion of left turn or right turn with the minimum turning radius in the forward direction from P_s , respectively. Similarly, (x_{glf}, y_{glf}) and (x_{grf}, y_{grf}) represent the coordinates of the positions from P_g (see Figure 5).

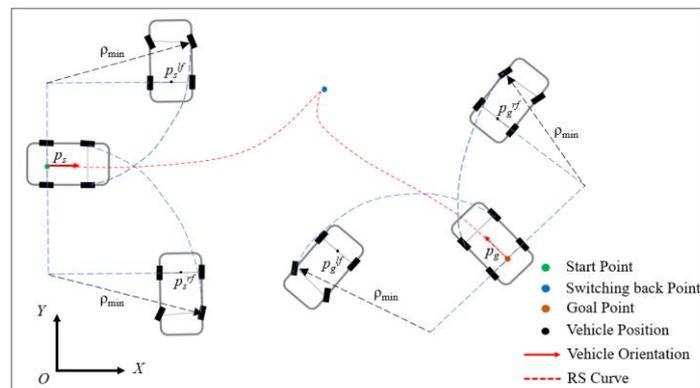


Figure 5. Construction of switching symmetric corridor.

The AMT must decrease the speed to zero when switching gears at the switching-back point. According to the transportation rules of the mining site, reversing will be the last motion if the trajectory includes reverse motion. Thus, the kinematic constraints including corresponding velocity v_{sw} and acceleration a_{sw} at the sw -th point (switching-back point) are defined as follows:

$$\begin{aligned} v_{sw} &= 0, & a_{sw} &= 0, \\ v_i &\leq 0, & \forall i &\geq sw \end{aligned} \quad (20)$$

The proposed optimization method resolved the switching point problem by constraining the control input and its change rate (i.e., steer and acceleration). In addition, the AMT can stop in the switching symmetric corridor and reverse flexibly in the narrow loading area with continuous curvature. The derived speed profile is used as the reference by the low-level control module. By solving Equation (10) numerically, collocation points representing the haul path can be derived.

4. Interactive Speed Planning Considering Cyberattack

Generally, the speed planning module provides a feasible speed profile for the AMT to travel along the haul path. As aforementioned, interactions exist among AMTs during transportation. According to the state of the other AMTs, the ego AMT should decide whether to yield or accelerate when approaching a merging area. However, in the face of network attacks, planning algorithms need to improve their reliability to ensure safe driving. Thus, a dynamic speed planning strategy is developed, considering the interactions with other AMTs. It adopts a hierarchical structure [28] including maneuver decision making and speed profile generation.

4.1. Topology-Guided Secure Maneuver Decision Making

The haul path generated in Section 2 only considers static objects. This study constructs a temporal–spatial (T-S) graph [29] that captures the topological relation among AMTs. The priorities strategies exist for road allocation among AMTs:

- AMTs must follow the sequence of loading tasks based on the task orders issued by the cloud platform. First come, first serve.
- When interacting with other AMTs, the AMT with a higher priority will maintain its original trajectory, whereas the lower one is required to adjust its trajectory.

By limiting the distance (represented by time to collision (TTC) [30]) between the vertices of AMT trajectories in the T-S graph, collision can be avoided when the network has latency. According to the reaction time and braking time of the system [29,31], the TTC is set as 3.5 s.

As shown in Figure 6, truck V1 must decide whether to overtake in the merging area. The topological relationships between AMTs are depicted in the T-S graph shown in Figure 7. Taking truck V2 as an example, points A1 and B1 represent the maneuvers of accelerating and yielding, respectively.

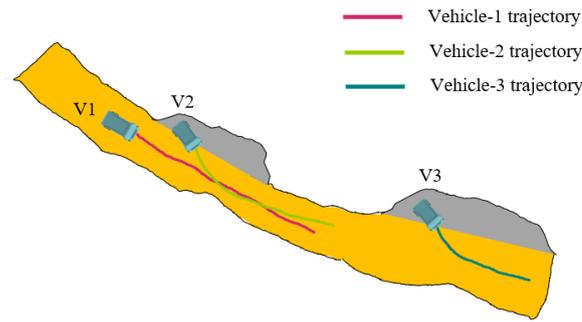


Figure 6. Illustration of merging area at mining site.

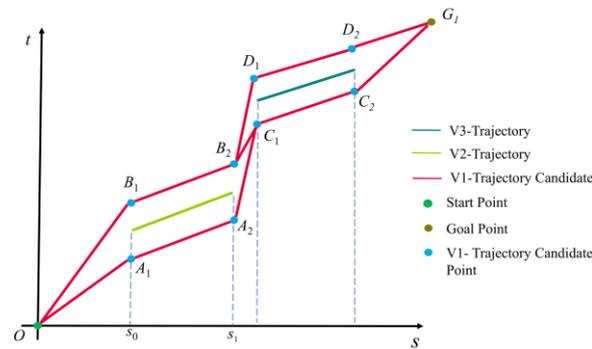


Figure 7. Schematic diagram of AMT merging in T-S graph.

By setting the desired arrival time at the goal position, the planner connects the corresponding point of each maneuver to generate the edge sequence of the trajectory profile. For example, the edge sequence (B1–B2–C1–C2–G1) represents the situation where V1 first yields to V2 and then overtakes V3 before reaching the goal. It should be noted that the connecting vectors must satisfy the monotonicity.

$$\frac{T_{ego}(s_{i+1}) - T_{ego}(s_i)}{s_{i+1} - s_i} > 0, \forall s_{i+1} - s_i > 0 \quad (21)$$

where $T_{ego}(s_{i+1})$ and $T_{ego}(s_i)$ denote the timestamp of the ego vehicle's trajectory at station s_{i+1} and s_i . In this study, Dijkstra's algorithm [32] is applied to search the optimal graph vertices along the t -axis.

To ensure the consistency of the reference velocity profile, a cost function C_t is designed to facilitate the search for the maneuver sequence.

$$C_t = C_{ref}(m) + C_{vd}(m) \quad (22)$$

$$C_{ref}(m) = C_{ref}(m_p) + w_{ref} \sum_{k=0}^{N_e} (v_k - v_{ref,k})^2 \quad (23)$$

$$C_{vd}(m) = w_{vd}(v_m - v_{mp})^2 \quad (24)$$

where $C_{ref}(\cdot)$ denotes the cost of the edge, m_p denotes the parent node of m , v_k denotes the velocity of the k -th node, $v_{ref,k}$ denotes the reference velocity derived from the path post-processing, N_e denotes the number of path points on one edge, w_{ref} is the reference bias weight, and $C_{vd}(\cdot)$ denotes the average velocity deviation between the edge and its parent edge. The proposed method supports three or more than vehicles with lower priority considered as obstacles.

When connecting the graph vertices and extending to the goal, the edge with a lower cost is preferred for maneuver decision making. After several iterations, Dijkstra's

algorithm finds the optimal edge sequence, which will be used as the decision result. Instead of searching for a complete trajectory denoting the maneuver sequence [24,33], this study decreases the dimensions of the searching algorithm, allowing for a more cost-effective way to find the maneuver.

4.2. Speed Profile Generation

The speed profile generation is formulated as a time-optimal control problem. This study utilized the T-S graph and nonlinear programming (NLP) to solve this problem. The generated maneuver sequence can be represented as the bound constraints in the NLP method. By minimizing the arrival time t_f and control inputs a , the NLP method can be established. The objective function is formulated as a linear combination of acceleration, jerk, and arrival time t_f with their respective weights:

$$\operatorname{argmin}_{t,v,a} J(t_0, v_0, a_0, \dots, t_n, v_n, a_n) \quad (25)$$

$$J = w_a \sum_{i=0}^N a_i^2 + w_j \sum_{i=2}^{N-1} \operatorname{jerk}_i^2 + w_f t_f^2 \quad (26)$$

Similar to the trajectory optimization described in Section 3, the variable constraints for velocity v , acceleration a , and *jerk* are set as shown below:

$$v_i = \frac{s_{res}}{T(s_{i+1}) - T(s_i)} \quad (27)$$

$$a_i = \frac{v_{i+1} - v_i}{T(s_{i+1}) - T(s_i)} \quad (28)$$

$$\operatorname{jerk}_i = \frac{a_{i+1} - a_i}{T(s_{i+1}) - T(s_i)} \quad (29)$$

where s_{res} denotes the resolution of unit distance (i.e., 1 m) in the T-S graph. Compared to the optimization method in Section 3, the maneuver sequence is established as the constraint, which can be classified as following, yielding, and overtaking. For the following, the trajectory of the preceding vehicle is used as the lower boundary. The maneuver constraint is set as:

$$T_{ego}(s) + t_{lat} > T_j(s) \quad (30)$$

where T_{ego} and T_j denote the trajectories of the ego and j_{th} vehicle, respectively; and t_{lat} is the latency compensation. Regarding yielding, the maneuver constraints are defined as follows:

$$T_{ego}(s_{ns}) + t_{lat} > T_j(s_{ns}) \text{ and } T_{ego}(s_{ne}) + t_{lat} > T_j(s_{ne}) \quad (31)$$

where s_{ns} denotes the start position, while s_{ne} denotes the end position of the overlap between two AMTs' trajectories. In this scenario, the AMT slows down until the neighboring AMT has passed the merging area. Regarding overtaking, the maneuver constraints are defined as follows:

$$T_{ego}(s_{ns}) < T_j(s_{ns}) + t_{lat} \text{ and } T_{ego}(s_{ne}) < T_j(s_{ne}) + t_{lat} \quad (32)$$

In such a case, the AMT accelerates to pass the merging area before the neighboring AMT arrives at s_{ne} . Based on the constraints above, the optimizer follows the execution order of the maneuver sequence. Finally, the optimization problem is solved using the interior point optimization method [34]. With the trajectory generated in Section 3 serving as the initial guess, the planner can find the solution in real time. In the next section, the performance of the planner will be investigated via experiments.

5. Experiment Analysis

To validate the proposed method, the experiments were conducted at the Bayan Obo rare earth mineral deposit, as shown in Figure 8. The proposed method was implemented in C++ language code running on a PC with an Inter i7-6820EQ processor at 2.8 GHz and 16 GB of RAM, running the Ubuntu system. The AMT used is a TL 90 equipped with RTK-GPS, as shown in Figure 8b.



Figure 8. Experiment environment. (a) The actual environment of mining site at Bayan Obo. (b) The autonomous truck equipped with 5G antenna, RTK-GPS.

The detailed parameters in the experiments are shown in Table 2. Altogether, four types of experiments were conducted. The first was to verify the effectiveness of path generation. The second dealt with the parking scenario and investigated the scheme of switching-back point. The third experiment examined the performance of the speed-planning module, considering the interaction with other AMTs. Lastly, the experiments were conducted in a real mining site. Pure pursuit and the PID algorithm were adopted for longitudinal and lateral controllers at a frequency of 20 Hz, respectively.

Table 2. Parameters in IHA* and RRT* algorithms.

Parameters	Value	Description	Unit
α	1.1	Heuristic factor	\
d_{\min}	1.7	Minimum expanding step	m
d_{\max}	6.0	Maximum expanding step	m
d_{thr}	3.0	Obstacle threshold distance	m
d_{lim}	1.08	Exploration threshold distance	m
r_g	1.0	Grid unit resolution	m
r_d	5.0	Orientation angle resolution	deg
r_{sh}	5.0	RRT* searching step	m
$iter_{\max}$	100,000	RRT* maximum iteration	-

5.1. Evaluation of Path Generation at Mining Site

The scenarios were set for path planning with the same start point but different goal points in three mining areas with about 1 s network latency. According to the road boundary information, the planner generated the haul paths for the truck. For comparison, a series of benchmark methods were selected, including two basic planners (RRT* [35] and HA*) and two optimization-based planners (conjugate gradient (CG) and the proposed optimal control problem (OCP) solver in [34]). IHA*, RRT*, HA* and their combinations with the OP method (represented by IHA*-OP, HA*-OP, and RRT*-OP, respectively) were first compared. To further investigate the performance of the proposed optimization method, HA*-CG, HA*-OP, IHA*-OP, and IHA*-OCP were compared as well. The results are presented in Figures 9 and 10 and Table 3.

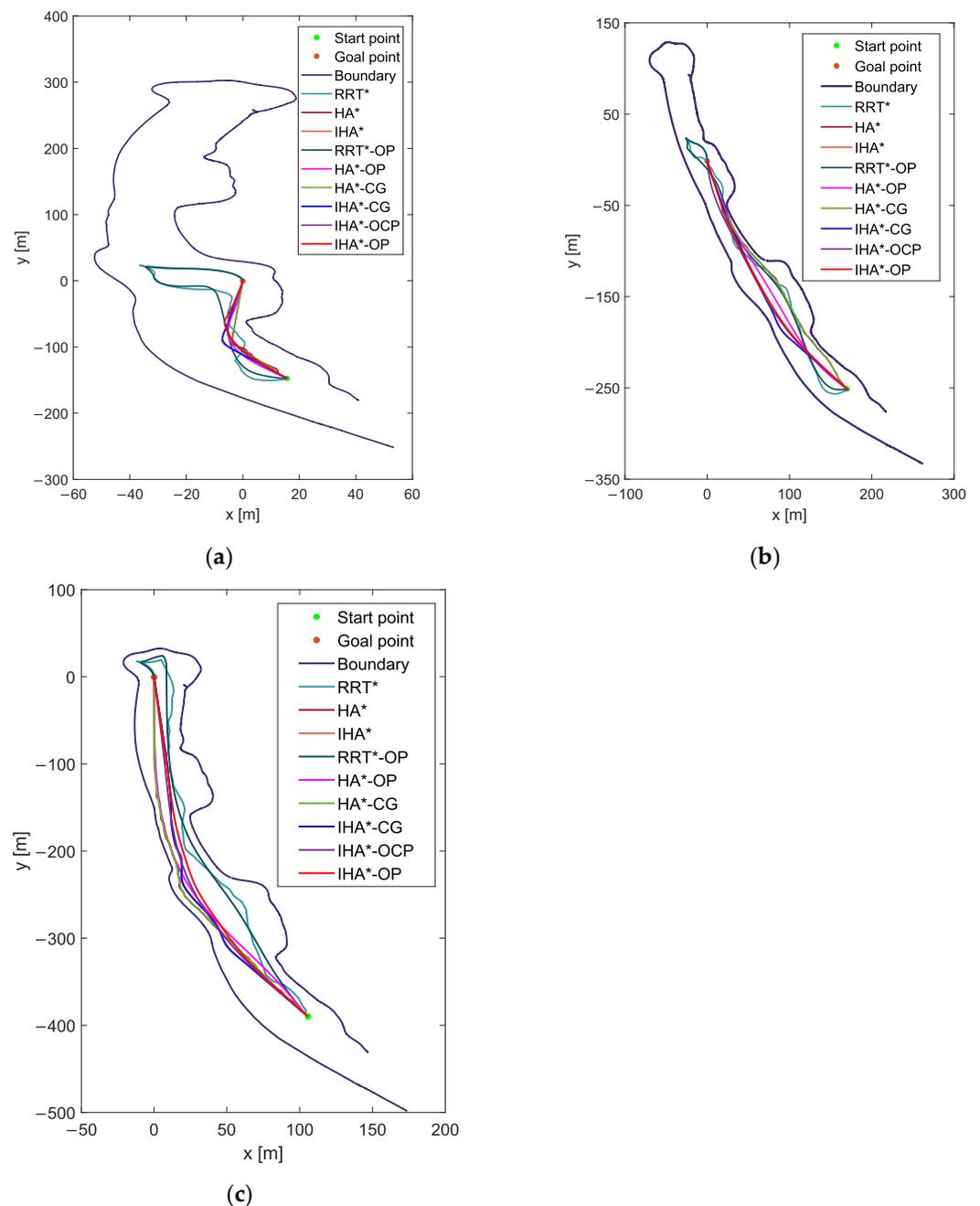


Figure 9. Planned paths with different goal points: (a) mining area 1, (b) mining area 2, (c) mining area 3.

It can be observed in Figure 9 that all the generated paths are collision-free with kinematic constraints. The paths generated by RRT* have a larger length and certain randomness, which poses challenges to the control module for tracing [17]. The paths generated by HA* and IHA* have a similar length, whereas HA* costs much more time than IHA*. The reason can be attributed to its setting of the fixed expansion step. By contrast, IHA* achieves better performance in terms of both maximum curvature and computation time.

To investigate the effectiveness of OP, the ablation experiments were conducted to examine the performance of HA*-OP, RRT*-OP, and IHA*-OP. It can be identified that OP supports shorten the initial path length and decrease the maximum curvature of the planned path, as shown in Figure 10a,b.

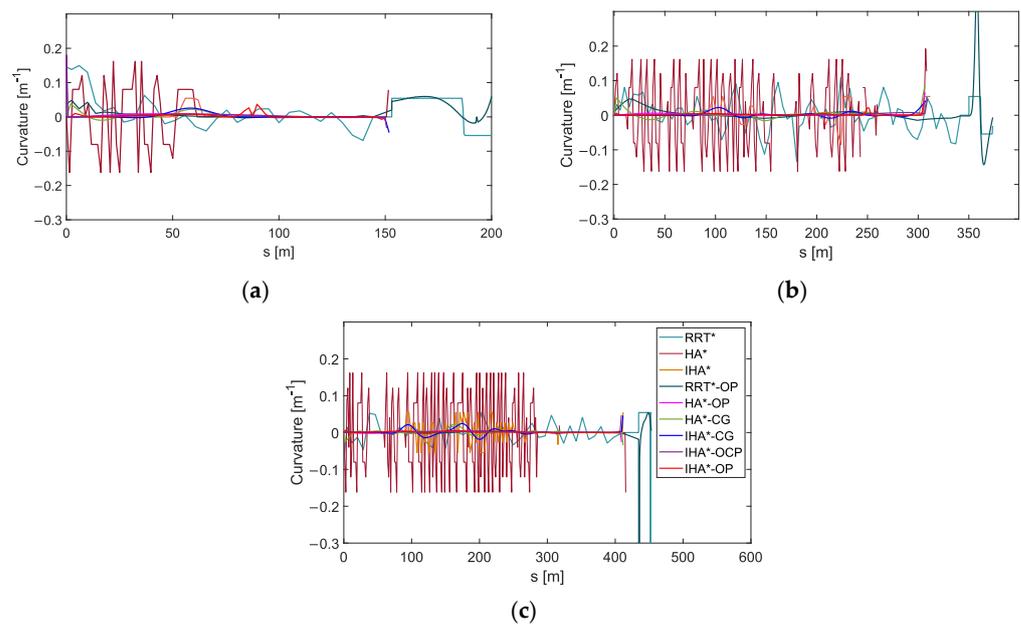


Figure 10. Curvature profiles of planned paths: (a) mining area 1, (b) mining area 2, (c) mining area 3.

Table 3. Comparison of generated paths for three mining areas.

Area	Planner	Path Length (m)	Maximum Curvature (m^{-1})	Computation Time (s)
Mining area 1	RRT*	253.34	0.145	0.283
	HA*	152.95	0.160	0.203
	IHA*	152.00	0.057	0.020
	RRT*-OP	242.07	0.072	0.939
	HA*-OP	150.40	0.033	0.734
	HA*-CG	151.20	0.039	0.312
	IHA*-CG	151.83	0.035	0.267
	IHA*-OCP	150.69	0.030	0.991
	IHA*-OP	148.95	0.032	0.486
Mining area 2	RRT*	387.67	−0.133	3.435
	HA*	309.80	0.160	0.962
	IHA*	313.50	−0.061	0.130
	RRT*-OP	382.58	0.045	4.462
	HA*-OP	302.60	0.065	1.685
	HA*-CG	306.88	0.087	1.122
	IHA*-CG	312.42	0.053	0.513
	IHA*-OCP	312.95	0.036	1.816
	IHA*-OP	311.54	0.038	0.859
Mining area 3	RRT*	475.74	-	11.389
	HA*	415.12	−0.140	1.223
	IHA*	416.98	−0.070	0.240
	RRT*-OP	464.85	1.862	12.403
	HA*-OP	410.30	0.023	2.132
	HA*-CG	412.71	0.032	1.425
	IHA*-CG	414.03	0.037	0.649
	IHA*-OCP	414.85	0.031	2.130
	IHA*-OP	410.28	0.027	0.987

For example, the maximum curvature of the path generated by HA* in mining area 1 is $0.160 m^{-1}$, and it decreases to $0.033 m^{-1}$ after implementing the OP method. For computation time, owing to the undesirable quality of generated paths by HA* and RRT*,

the OP method takes more time to post-process the raw planning results. Furthermore, a comparison between OP and the other two optimization-based planners (CG and OCP) was conducted. From Figure 10c, it can be observed that IHA*-OP outperforms IHA*-CG and IHA*-OCP in terms of path length. Although the maximum curvatures by IHA*-OP and IHA*-OCP are almost equivalent, IHA*-OP achieves more desirable computational efficiency. As for IHA*-CG, it costs less computation time than IHA*-OP and IHA*-OCP by implementing optimization using a gradient descent strategy. However, its path length and maximum curvature results are not optimal.

5.2. Switching-Back Point Adjustment

As aforementioned, parking often occurs in the loading operation. In such a scenario, AMTs must reverse before arriving at the loading position. The gear changes are inevitable, which requires the planner to generate a feasible path and provide it to the control module.

The paths generated by HA*, RRT*, IHA*, and their combinations with OP, CG, and OCP are shown in Figure 11, and the resulting orientations θ are shown in Figure 11b. Compared to RRT*, HA* and IHA*, all the methods with OP decreases the path length and relax the change rate of θ . Although HA* outperforms IHA* in terms of path length, its resultant path is closer to the boundary of the driving area, leading to increased driving risk. Compared to CG and OCP, OP could dynamically adjust the switching-back point and enhance the quality of the generated path. Table 4 shows that IHA*-OP achieves the shortest path length and computation time.

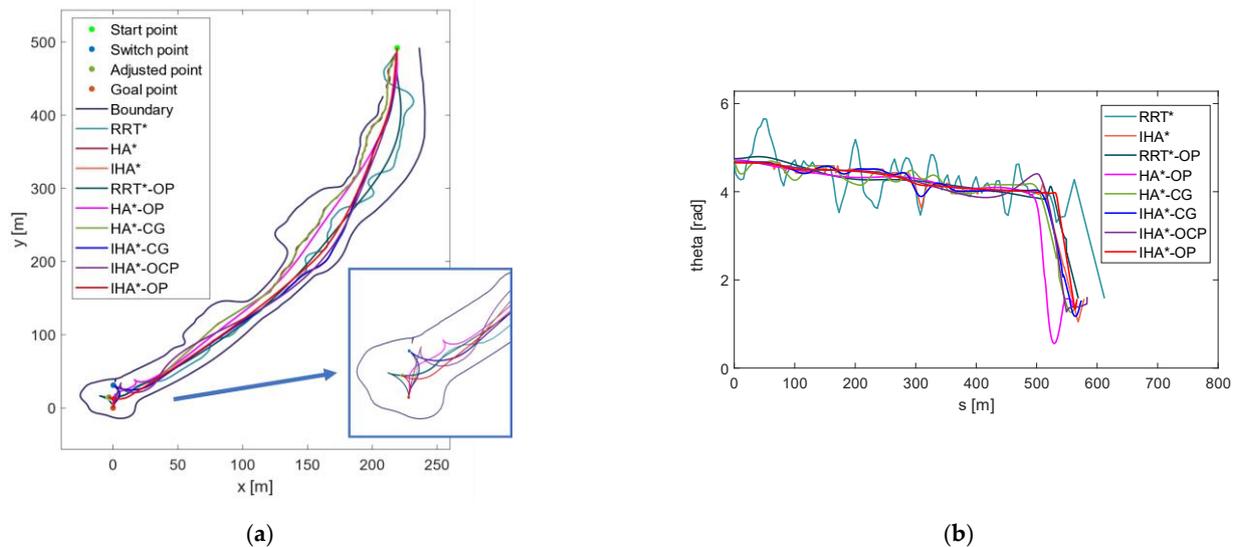


Figure 11. Result of switching-back point adjustment. (a) Adjustment of switching-back point; (b) Orientation θ of optimized paths.

Table 4. The comparison of path length and computation time.

Planner	Path Length (m)	Computation Time (s)
RRT*	609.67	7.85 s
HA*	577.46	6.39 s
IHA*	578.66	0.37 s
RRT*-OP	569.93	8.90 s
HA*-OP	563.28	8.07 s
HA*-CG	568.52	6.85 s
IHA*-CG	575.37	0.74 s
IHA*-OCP	584.25	9.06 s
IHA*-OP	565.23	1.07 s

In addition, it should be noted that the reversing path by OP, as illustrated in Figure 11, is also the shortest. It conforms to the parking behavior at the loading position since long-distance reversing driving would be potentially risky for AMT.

5.3. Validation of the Proposed Method in Real Environment

Furthermore, the proposed IHA*-OP was implemented on the cloud platform of the mining site in practice. In real operation, when one AMT (as shown in Figure 3) reaches the reporting point, it will request the loading task from the cloud platform. After receiving the task (i.e., the loading parking position and heading), the planning module will generate the trajectory and provide it to the AMT via 5G communication.

For validation purposes, two typical scenarios were established. The first one aims to examine the tracking of planned trajectories in large loading area, whereas the second is in a relatively narrow loading area. The results are presented in Figure 12. The boundary of the loading area is illustrated in black, whereas the planned trajectories by IHA*-OP and real tracking results are illustrated in blue and red, respectively. The analyses reveal that although there exists a certain overshoot in the process of trajectory tracking, the maximum lateral tracking error is less than 0.3 m in these two scenarios. The error can be further optimized by setting a shorter preview time in the lateral control module. The results above verify the effectiveness of the proposed method in a real environment.

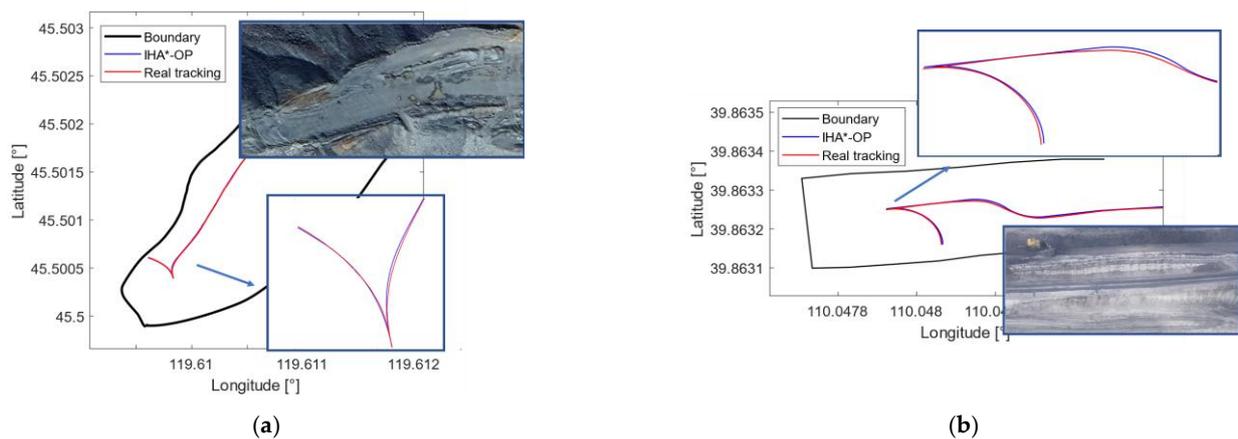


Figure 12. Validation results in large loading area and narrow loading area. (a) Large loading area; (b) narrow loading area.

5.4. Interaction with Other AMTs under Cyberattack

5.4.1. Scenario 1: Car Following

A car-following scenario was set as follows. After receiving the load task, AMT-1 runs along the haul path at the desired speed of 4 m/s. AMT-2 is at the position 30 m ahead of AMT-1 and runs at 3 m/s. The detailed setting of the parameters is provided in Table 5. AMT-1 will lose the connection to AMT-2 under cyberattack. According to the rules at the mining site, AMT-1 is required to follow AMT-2 ahead by adapting its speed profile, as shown in Figure 13. Such a scenario was designed to examine the car-following performance of the proposed method.

Table 5. Simulation parameters and task parameters.

	v_{ego} (m/s)	v_{des} (m/s)
Scenario. 1	4.0	4.0
Scenario. 2	4.0	4.0

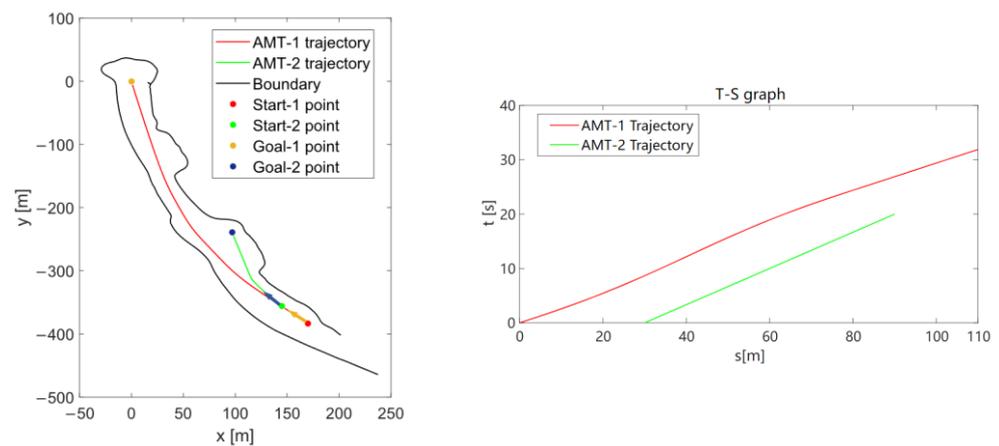


Figure 13. Desired paths and speed profile of two AMTs in the car-following scenario.

The trajectories and speed profiles of AMTs are shown in Figure 14. It can be identified that AMT-1 decreases its speed and maintains a safe time headway. At 90 m, AMT-2 deviates from the haul path and steers to the loading point. Thus, AMT-1 accelerates to the desired speed and maintains the speed along the haul path. At about 100 m, it starts to decrease the speed and finally stops at the goal.

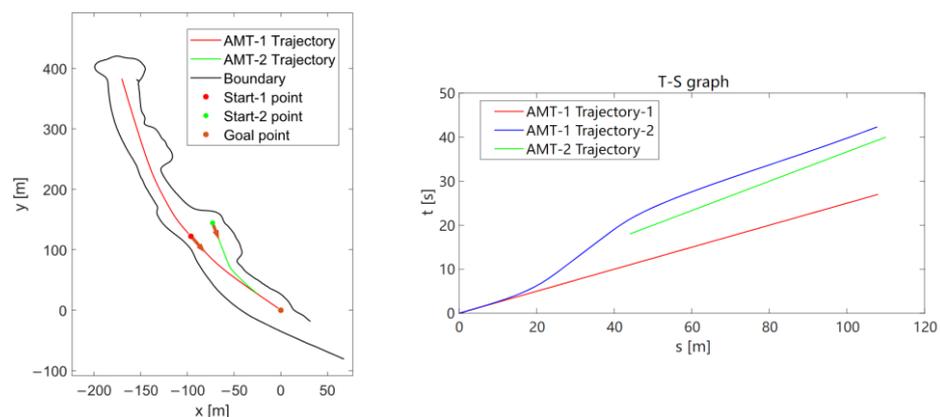


Figure 14. Desired paths and speed profile of two AMTs in the merging scenario.

5.4.2. Scenario 1: Car Merging

A merging scenario was set as follows. AMT-1 is loaded with ore runs at the exit trajectory. AMT-2 is at the position 45 m ahead of AMT-1 and runs at 3.2 m/s along the desired trajectory. The trajectories of the two AMTs are shown in Figure 14. With the cyberattack, AMT-2 will lose the connection to AMT-1. Such a scenario was designed to examine the car-meeting performance of the proposed method.

AMT-1 needs to decide whether to yield or accelerate for passing. According to the T-S graph, two options exist: the first is to yield and decrease the speed, corresponding to Trajectory-1; the second is to accelerate for passing, corresponding to Trajectory-2. Compared to the first option, the second one achieves a shorter travel time. Thus, the planner decides to pass the merging area without deceleration. Trajectory-2 would be selected as the optimal speed profile.

6. Conclusions

In this study, a secure trajectory planning method for connected autonomous vehicles at the mining site has been proposed, which contributes to the systematic secure planning framework of the mining site. By considering the operational characteristics of the mining site and AMTs comprehensively, this study improved the Hybrid A* to find the raw path

in real time. Next, by resorting to a topology-guided method, the planner formulated the secure decision making as the graph vertexes searching problem, which generated the maneuver sequence in a cheap way. Based on the maneuver sequence, the planner found the time-optimal speed profile by using the NLP while considering the cyberattack simultaneously. Last, the validation results via a series of experiments demonstrate that the proposed method can generate feasible trajectories in real time.

7. Future Work

The study also has limitations which can be addressed in future work. The proposed method conducts the secure decision making based on the priority rules to resolve the cyberattack problem. The whole system based on the cloud platform adopts the idea of centralized control. In order to enhance the system robustness, the distributed model prediction control (MPC) for multi-AMT cooperation via V2V is supposed to be incorporated in the trajectory planning.

Author Contributions: Conceptualization, H.L.; methodology, H.L.; resources, Y.L. (Yiming Li); data curation, P.C.; writing—review and editing, G.Y. writing—review and editing, Y.L. (Yaping Liao). All authors have read and agreed to the published version of the manuscript.

Funding: H.L. acknowledges the support by the National Key Research and Development Program of China, No. 2022YFB4703700.

Data Availability Statement: The data presented in this study are openly available in https://github.com/studentlemon/miningsite_map (accessed on 3 October 2023).

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

Abbreviations

CG	Conjugate gradient
AMT	Autonomous mining truck
IHA*	Improved Hybrid A*
NLP	Nonlinear programming
RS	Reeds–Shepp
RRT*	Random rapid tree
OP	Optimization process
T-S	Temporal–spatial
TTC	Time to collision
OCP	Optimal control programming
PID	Proportional integral differential
RTK-GPS	Real-time kinematic–global position system

References

- Zhu, K.L.; Chen, W.T.; Jiao, L.; Wang, J.X.; Peng, Y.Y.; Zhang, L. Online training data acquisition for federated learning in cloud–edge networks. *Comput. Netw.* **2023**, *223*, 109556. [CrossRef]
- Dou, F.Q.; Huang, Y.J.; Liu, L.; Wang, H.; Meng, Y.; Zhao, L.Q. Path planning and tracking for autonomous mining articulated vehicles. *Int. J. Heavy Veh. Syst.* **2019**, *26*, 315–330. [CrossRef]
- Wang, P.C.; He, X.Z.; Wei, Y.; Wu, X.K.; Wang, Y.P. Damping behavior analysis for connected automated vehicles with linear car following control. *Transp. Res. Part C Emerg. Technol.* **2022**, *138*, 103617. [CrossRef]
- Maekawa, T.; Noda, T.; Tamura, S.; Ozaki, T.; Machida, K.I. Curvature continuous path generation for autonomous vehicle using B-spline curves. *Comput. Aided Des.* **2010**, *42*, 350–359. [CrossRef]
- Wang, H.; Huang, Y.J.; Khajepour, A.; Zhang, Y.; Rasekhipour, Y.; Cao, D.P. Crash mitigation in motion planning for autonomous vehicles. *IEEE Trans. Intell. Transp. Syst.* **2019**, *20*, 3313–3323. [CrossRef]
- Zhang, H.; Hu, M.J.; Nguyen, A.T.; Wang, Y.P.; Shi, Y. Preface for Cyber-Attack Detection and Resilient Control of Intelligent and Connected Vehicles. *Automot. Innov.* **2023**, *6*, 143–145. [CrossRef]
- Claussmann, L.; Revilloud, M.; Dominique, G.; Glaser, S. A review of motion planning for highway autonomous driving. *IEEE Trans. Intell. Transp. Syst.* **2020**, *21*, 1826–1848. [CrossRef]
- Gonzalez, D.; Perez, J.; Milanés, V.; Nashashibi, F. A review of motion planning techniques for automated vehicles. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 1135–1145. [CrossRef]

9. Amini, A.; Chen, Y.; Elayyadi, M.; Radeva, P. Tag surface reconstruction and tracking of myocardial beads from SPAMM-MRI with parametric B-spline surfaces. *IEEE Trans. Med. Imaging* **2001**, *20*, 94–103. [[CrossRef](#)]
10. Taketoshi, S.; Riku, U.; Takashi, M. Automatic two-lane path generation for autonomous vehicles using quartic B-spline curves. *IEEE Trans. Intell. Veh.* **2018**, *4*, 547–558.
11. Usami, R.; Kobashi, Y.; Onuma, T.; Maekawa, T. Two-lane path planning of autonomous vehicles in 2.5D environments. *IEEE Trans. Intell. Veh.* **2020**, *5*, 281–293. [[CrossRef](#)]
12. Zhang, Y.; Chen, H.Y.; Waslander, S.L.; Gong, J.W.; Xiong, G.M.; Yang, T.; Liu, K. Hybrid trajectory planning for autonomous driving in highly constrained environments. *IEEE Access* **2018**, *6*, 32800–32819. [[CrossRef](#)]
13. Kim, H.; Choi, Y. Location estimation of autonomous driving robot and 3D tunnel mapping in underground mines using pattern matched LiDAR sequential images. *Int. J. Min. Sci. Technol.* **2021**, *31*, 779–788. [[CrossRef](#)]
14. Chen, C.; Wu, B.; Xuan, L.; Chen, J.; Wang, T.T.; Qian, L.J. A trajectory planning method for autonomous valet parking via solving an optimal control problem. *Sensors* **2020**, *20*, 6435–6445. [[CrossRef](#)] [[PubMed](#)]
15. Oliveira, R.; Lima, F.P.; Cirillo, M.; Mårtensson, J.; Wahlberg, B. Trajectory generation using sharpness continuous dubins-like paths with applications in control of heavy-duty vehicles. In Proceedings of the IEEE Conference on Decision and Control, Limassol, Cyprus, 12–15 June 2018.
16. Botros, A.; Smith, S.L. Tunable trajectory planner using G3 curves. *IEEE Trans. Intell. Veh.* **2022**, *7*, 273–285. [[CrossRef](#)]
17. Chai, R.Q.; Tsourdos, A.; Savvaris, A.; Chai, S.S.; Xia, Y.Q.; Chen, L.P. Multiobjective optimal parking maneuver planning of autonomous wheeled vehicles. *IEEE Trans. Ind. Electron.* **2020**, *67*, 10809–10821. [[CrossRef](#)]
18. Xiang, J.; Zhong, C.W.; Wei, W. A Varied Weights Method for the Kinematic Control of Redundant Manipulators with Multiple Constraints. *IEEE Trans. Robot.* **2012**, *28*, 330–340. [[CrossRef](#)]
19. Fraichard, Y.; Scheuer, A. From reeds and shepp’s to continuous-curvature paths. *IEEE Trans. Robot.* **2004**, *20*, 1025–1035. [[CrossRef](#)]
20. Zhang, S.Y.; Jian, Z.Q.; Deng, X.D.; Chen, S.T.; Nan, Z.X.; Zheng, N. Hierarchical motion planning for autonomous driving in large-scale complex scenarios. *IEEE Trans. Intell. Transp. Syst.* **2022**, *2*, 13291–13305. [[CrossRef](#)]
21. Scheuer, A.; Fraichard, T. Continuous-curvature path planning for car-like vehicles. In Proceedings of the IEEE International Conference on Intelligent Robots and Systems, Grenoble, France, 11 September 1997; Volume 1, pp. 997–1003.
22. Xie, G.T.; Gao, H.B.; Qian, L.J.; Huang, B.; Li, K.Q.; Wang, J.Q. Vehicle trajectory prediction by integrating physics- and maneuver-Based approaches using interactive multiple models. *IEEE Trans. Ind. Electron.* **2017**, *65*, 5999–6008. [[CrossRef](#)]
23. Lim, W.; Lee, S.; Sunwoo, M.; Jo, K. Hybrid trajectory planning for autonomous driving in on-road dynamic scenarios. *IEEE Trans. Intell. Transp. Syst.* **2021**, *22*, 341–355. [[CrossRef](#)]
24. Lim, W.; Lee, S.; Sunwoo, M.; Jo, K. Hierarchical trajectory planning of an autonomous car based on the integration of a sampling and an optimization method. *IEEE Trans. Intell. Transp. Syst.* **2018**, *19*, 613–626. [[CrossRef](#)]
25. Zhou, B.; Gao, F.; Wang, L.; Liu, C.; Shen, S. Robust and efficient quadrotor trajectory generation for fast autonomous flight. *IEEE Robot. Autom. Lett.* **2019**, *4*, 3529–3536. [[CrossRef](#)]
26. Reeds, A.J.; Shepp, A.L. Optimal paths for a car that goes both forwards and backwards. *Pac. J. Math.* **1990**, *145*, 367–393. [[CrossRef](#)]
27. Boyali, A.; Thompson, S. Autonomous parking by successive convexification and compound state triggers. In Proceedings of the IEEE Intelligent Transportation System Conference, Rhodes, Greece, 20–23 September 2020.
28. Wang, P.C.; Wu, X.K.; He, X.Z. Vibration-Theoretic Approach to Vulnerability Analysis of Nonlinear Vehicle Platoons. *IEEE Trans. Intell. Transp. Syst.* **2023**; early access. [[CrossRef](#)]
29. Qian, L.L.; Xu, X.; Zeng, Y.J.; Li, X.H.; Sun, Z.P.; Song, H. Synchronous maneuver searching and trajectory planning for autonomous vehicles in dynamic traffic environments. *IEEE Intell. Transp. Syst. Mag.* **2022**, *14*, 57–73. [[CrossRef](#)]
30. Hillenbrand, J.; Spieker, M.A.; Kroschel, K. A multilevel collision mitigation approach—Its situation assessment, decision making, and performance tradeoffs. *IEEE Trans. Intell. Transp. Syst.* **2006**, *7*, 528–540. [[CrossRef](#)]
31. Xu, C.; Zhao, W.Z.; Wang, C.Y. An integrated threat assessment algorithm for decision-making of autonomous driving vehicles. *IEEE Trans. Intell. Transp. Syst.* **2020**, *21*, 2510–2521. [[CrossRef](#)]
32. Dijkstra, E. A note on two problems in connexion with graphs. *Num. Math.* **1959**, *1*, 269–274. [[CrossRef](#)]
33. Meng, Y.; Wu, Y.M.; Gu, Q.; Liu, L. A decoupled trajectory planning framework based on the integration of lattice searching and convex Optimization. *IEEE Access* **2019**, *7*, 130530–130551. [[CrossRef](#)]
34. Wächter, A.; Biegler, T.L. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Math. Program.* **2006**, *106*, 25–57. [[CrossRef](#)]
35. Karaman, S.; Frazzoli, E. Optimal kinodynamic motion planning using incremental sampling-based methods. In Proceedings of the IEEE Conference on Decision and Control, Atlanta, GA, USA, 15–17 December 2010.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.