Research Article

An MEC Architecture-Oriented Improved RRT Algorithm for Regional Trajectory Planning

Qingwen Han, Ziyun Huang, Lingqiu Zeng, Yunqu Wu, Lei Ye, Hui Zu, and Yu Lei

1 School of Microelectronics and Communication Engineering, Chongqing University, Chongqing 400044, China
2 School of Computer Science, Chongqing University, Chongqing 400044, China
3 Chongqing Vehicle Test and Research Institute Co. Ltd., Chongqing 401122, China
4 Chongqing Automobile Collaborative Innovation Center (CQACIC), Chongqing University, Chongqing 400044, China

CorrespondenceshouldbeaddressedtoLeiYe;yelei@cqu.edu.cn

Received 15 October 2020; Revised 17 December 2020; Accepted 15 April 2021; Published 27 April 2021

Academic Editor: Adrian Kliks

Copyright © 2021 Qingwen Han et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Multi-access Edge Computing (MEC), which could provide real-time computing ability, is considered as an effective approach to improve performance of Vehicular Ad Hoc Network (VANET). MEC could process regional vehicles information and generate real-time road hazard features, which could be used to realize trajectory planning progress of vehicles. In this paper, an MEC-oriented VANET infrastructure is presented, and a road hazard feature-based trajectory planning method is proposed. Back Propagation (BP) neural network is employed to predict road hazard feature changing, while a hazard-based cost function is defined. Then, an improved Rapidly Exploring Random Tree (RRT) algorithm is proposed for novel regional trajectory planning. A joint simulation is done based on SUMO and NS3 platforms. Simulation results verify the effectiveness and stability of the proposed algorithm.

1. Introduction

The rapid development of V2X technology makes it possible to share real-time information among all traffic participants, such as vehicles, pedestrians, and public transportation facilities. V2X communication, which helps road drivers avoid potential threats by exchanging safe warning information, such as environment feature, bad driving state, etc., is considered as an important approach to improve road safety and driving safety [1]. Until now, most of autonomous driving systems are constructed based on active safe technology, which employ on-board sensors and cameras to collect corresponding information, such as road state, surrounding vehicle’s state, and pedestrian, to make real-time decision-making control [2]. However, road test results show that active safe autonomous vehicles hardly adapt to complex road environment.

Trajectory planning technique, which aims to find a collision-free path from an initial state to a goal state with optimal or near-optimal path cost, becomes one of useful applications in autonomous vehicles [3]. However, most of the current researches on the movement of vehicles is mainly in the micro and macro perspective; the former is mainly concerned about the movement of a single vehicle on the road but ignore the overall situation, while the latter is mainly concerned about road traffic state but ignore the relationship between road vehicles, both of which are difficult to achieve the regional vehicles trajectory planning. Obviously, if road vehicles look down from heaven with God’s eyes, namely, mesoscopic, they could get the characteristics of specific road section and generate a preset mesoscopic trajectory, which could greatly decrease computing burden of motion level control and enhance the real-time feature of autonomous control. The only question is how to get God’s perspective.

Fortunately, along with the development of VANET, both researchers and industries believe that Multi-access Edge Computing (MEC), which connects with RSU [1], is
useful in getting road hazard distribution feature, which could be employed as basis of mesoscopic trajectory planning. Therefore, based on MEC architecture, most recent approaches for trajectory planning take on-road vehicle status into consideration. Under MEC architecture, mesoscopic trajectory generation progress includes two critical problems: dynamic road hazard feature evaluation and collision-free trajectory generation.

In our previous work, an Ising model is proposed to divide the road into a finite number of adjacent grids that are used as the basic unit, which realize the evaluation of road hazard [4]. To get the dynamic feature of road hazard feature, in this paper, an Ising model-based hazard map generation method and a two-level BP neural network are employed to get the dynamic hazard feature of specific road segment, which is used as basic criteria for collision-free trajectory generation. Hence, in this paper, a dynamic road hazard distribution map-oriented regional trajectory planning method is presented; meanwhile, an improved RRT algorithm is proposed to generate mesoscopic trajectory.

Specifically, the contributions of this paper are as follows:

(i) A road hazard feature-based trajectory planning method is proposed.
(ii) BP neural network is employed to predict road hazard feature changing, while a hazard-based cost function is defined.
(iii) An improved RRT algorithm is proposed for novel regional trajectory planning.

The rest of this article is organized as follows. Section 2 presents the related work, and Section 3 describes the MEC framework used in this paper. In Section 4, the road hazard distribution characteristic evaluation and prediction method is presented. The proposed trajectory planning algorithm is discussed in Section 5, while the corresponding experiment set and results analysis are given in Section 6. Finally, conclusions and future works are discussed in Section 7.

2. Related Work

In the past few years, many researchers have done a lot of researches in Advanced Driver Assistance Systems (ADAS) and Autonomous Driving (AD) functions design. They are devoted to assisting drivers in avoiding road risks by obtaining real-time status data of the driver, the vehicle, and the surrounding environment to improve driving safety and improve the vehicle track. Hence, vehicle trajectory planning-related technologies have attracted many researchers’ attention. In this section, we analyze related works from three aspects, which are network architecture, road hazard state evaluation and prediction, and collision-free trajectory generation.

2.1. Network Architecture. In the last decade, cloud computing, which aims at centralized computing, storage, and network management in the clouds [5], is considered as an effective approach to support intelligent applications. However, cloud computing could not satisfy the demand of time-critical applications, especially for intelligent connected vehicles. Hence, to meet the requirements of autonomous driving, literature [6] proposed an MEC-oriented architecture, which takes full advantage of roadside equipment and vehicle on-board equipment, to overcome the shortcomings of centralized cloud computing in terms of latency and throughput. MEC servers, which connect with RSUs, collect real-time vehicle information, execute corresponding processing task, and provide processing results to road vehicles [1]. In the literature [7], an MEC server is deployed at RSU to improve real-time feature and then could provide real-time service for autonomous vehicles, such as local driving trajectory. In literature [8], the authors discuss about the effectiveness and real-time feature of MEC framework and think that RSU-oriented MEC could provide real-time service for autonomous vehicles, such as local driving trajectory plan, regional vehicle distribution information sharing, etc. Hence, in this paper, we assume that vehicular network is feasible and road vehicle’s information can be efficiently accessed and processed by edge computing approach. Under the assumed condition, an MEC-oriented processing procedure is proposed to provide real-time service for autonomous vehicles, such as local driving trajectory plan, regional vehicle distribution information sharing, etc. [8].

2.2. Road Hazard State Evaluation and Prediction. For road hazard state evaluation and prediction, risk avoidance is one of the hot points that should be considered in the design of Advanced Driver Assistance Systems (ADAS) and Autonomous Driving (AD) functions. Due to the complex road environment, it is unfeasible to evaluate all possible states of the involved traffic participants [9]. Hence, a road hazard distribution map is proposed, which is useful to decision makers in ADAS/AD system [10].

It is well known that a mesoscopic trajectory is decided by road grid hazard state on arrival time instead of present moment. That is said, we should consider not only current hazard state, but also the dynamic feature of hazard distribution map. Then, the performance of prediction method greatly influences effectiveness of output trajectory.

Road hazard not only depends on the cruising features of ego-car, but also is influenced by the surrounding driving contexts, such as road condition, traffic density, regional average speed, surrounding traffic participants’ characteristics, etc. Eggert [11] proposed a microscopic risk model, which could estimate the risk that a dangerous event like an accident can happen within a future time interval, being useful to driving trajectory schedule. Moreover, he declares that predictive risk map is useful in driving strategy generation process. However, the risky probability is a statistic parameter, which is calculated based on historic data, and hardly to evaluate real-time feature of road traffic. In [12], a vehicle-passing scenario is employed to illustrate future collision event probability and survival function for the case of multiple risks. However, vehicle-passing scenario proposed in [12] only considers single passing vehicle row. In
practice, instead of single vehicle row, an ego-car should try to pass vehicle group in object region, which includes multirow and multicolumn vehicles. In our previous work, a revised Ising model is proposed to explain road hazard distribution factor [4] and to take both joint threat of regional vehicles and regional threat distribution function into account in risk estimation progress. However, literature [4] did not consider the dynamic feature of hazard distribution map.

As mentioned above, a mesoscopic trajectory is decided by road grid hazard state on arrival time instead of present moment so that road hazard prediction should be considered. Moreover, traffic prediction approach, such as short-time multistep traffic prediction model [13] and BP neural network [14], could be used to predict traffic trend of a series discrete time points in the future. Based on the above, in this paper, we employ Ising model-based hazard map generation method and a two-level BP neural network to get the dynamic hazard feature of specific road segment.

2.3. Collision-Free Trajectory Generation. In general, regional trajectory planning, which considers both real-time road traffic state and autonomous vehicle cruising information, is used to generate a reliable driving trajectory for the autonomous vehicle within a future short time. However, without God’s perspective, autonomous vehicles used active-safety approach, which realizes decision-making process according to the real-time information providing by on-broad equipment, such as radars, sensors, and cameras, instead of calculating a collision-free trajectory in advance [15].

Finding a collision-free path from the initial state to the target state with optimal or near-optimal path costs is one of the key technologies to achieve automated vehicles. Traditional trajectory planning methods could be divided into four categories, which are specific function-based methods, optimal constraint-based control methods, machine learning-based methods, and path-based search methods, respectively [16]. Specific function-based methods typically use a specific trajectory function for path planning, such as cubic polynomial functions [16]. However, this kind of methods is not applicable for complex road environment. Methods based on optimal constraint control, such as Artificial Potential Field [17], are to supplement some necessary optional constraint indicators and then plan a better driving trajectory. Due to algorithm limitation, optimal constraint-based control methods are easy to stray into local optimal. It is true that machine learning-based methods are useful to construct a good planning model [18]. Unfortunately, the output of this method shows a poor environment adaptive feature. Path search-based methods, such as Dijkstra algorithm, A* algorithm, Rapid Exploration Random Tree (RRT) algorithm, etc. [19], usually discretize the road and then start the path search based on the discretization results. Due to their environment adaptability, path search-based planning algorithms become the most widely used trajectory planning method.

On-road vehicles who want to obtain a collision-free trajectory should find a valid searching method. RRT algorithm, which uses random samples from the search space to realize collision detection, can effectively solve the problem of trajectory planning in high-dimensional space and complex constraints. Then, a predictive hazard map should be used as searching basis for RRT algorithm [20]. Literature [21] presented an improved RRT-based motion planner for autonomous vehicles to effectively navigate in cluttered environments with narrow passages. In literature [22], the authors used connectivity feature as the basis of random tree expansion. Hence, in this paper, connectivity feature should be denoted by grid hazard value, which is the basic parameter of our proposed improved RRT algorithm. Moreover, a dynamic hazard feature-based cost function is defined to fulfill trajectory searching progress.

3. MEC Network Architecture

Multi-access Edge Computing (MEC) technology, which is introduced by the European Telecommunications Standards Institute (ETSI) Industry Specification Group (ISG) on MEC [23], offers cloud-computing capabilities within the radio access network (RAN) and an information technology service environment at the edge of the mobile network, close to mobile subscribers. Edge, which refers to both the base stations themselves [e.g., evolved node B (eNB), radio network controller (RNC), etc.] and data centers close to the radio network (e.g., located at RSUs), allows the third parties to deploy innovative applications and services toward mobile subscribers, enterprises, and vertical segments, flexibly and rapidly, while the common consumer may experience improved performance and new services offered by the MEC system [24].

The MEC framework, as presented in Figure 1, includes three layers, namely, cloud layer, MEC layer, and general vehicle layer. The first layer is cloud layer, which is used to store historic data and execute model training task. The second layer is the MEC layer, which includes edge computing nodes. All edge computing nodes are located at the back end of RSUs and execute regional centralize computing task. Each RSU maintains a local database and connects with core cloud through wired connection. The third layer is network node layer, which includes general vehicles. All road vehicles are assumed to equip with PC5 interface. Specifically, via PC5 interface, road vehicles can receive application-oriented information and update basic safety messages (BSMs) to RSUs.

The service flow is described as follows.

Step 1. Road vehicles periodically upload BSM, which includes driving information, such as cruising speed, location, heading, etc., to home RSUs.

Step 2. RSUs collect regional vehicles information. MECs at the back end of RSUs calculate out local hazard distribution feature and upload calculation results to cloud layer.

Step 3. Cloud layer collects calculation outputs of MEC layer and store them as historic data set $E_h$. Then, BP
model training process is done. The trained model is sent back to MEC layer via wired network.

**Step 4.** Based on the trained model, MEC layer executes an incremental learning process of BP neural network and predicts dynamic feature of regional hazard state and then hands out prediction results to road vehicles.

**Step 5.** According to the dynamic regional hazard feature, road vehicles generate a collision-free trajectory via improved RRT algorithm.

### 4. Road Hazard Distribution

#### 4.1. Road Hazard Distribution

In our previous work, Ising model, a model in statistical mechanics to study the magnetic dipole moments of atomic spins, is used to explain the intervehicle relationship. Two factors, which are vehicle health level and intervehicle relationship factor [4], are defined to denote regional hazard level.

The intervehicle relationship [4] between vehicles $i$ and $j$ at time $t$ is defined as

$$ R(i, j, t) = f[H(i, t), H(j, t), d_{ij}(t), \rho(t), \Delta v_{ij}(t)], $$

where $H(i, t)$ and $H(j, t)$ represent the health level of vehicles $i$ and $j$, respectively, $\rho$ is vehicle density of object region, $\Delta v_{ij}$ is the speed difference between vehicles $i$ and $j$, and $d_{ij}$ is the distance between vehicles $i$ and $j$. The health level [25] of the vehicle is determined by the driving behavior of the driver and the physical properties of the vehicle.

Then intervehicle relation energy of vehicle $i$ at time $t$ is

$$ E[i, t] = \sum_{j=1}^{n} R(i, j, t). $$

The radiant energy of $E[i, t]$ on each grid is

$$ E(i, grid, t) = \sum_{i=1}^{n} \frac{E[i, t]}{L_i^2}, $$

where $L_i$ is the distance between vehicle $i$ and object grid center.

The energy of each grid is

$$ E(grid, t) = \sum_{i=1}^{n} E(i, grid, t). $$

Hence, the radiant energy on each grid of the road can be obtained, and the corresponding grid energy radiation model is the road hazard distribution model constructed in this paper.

#### 4.2. Dynamic Trend of Grid Feature

Here, we assume that front road is composed of several grids with hazard state vectors. Then,

$$ S_t = \{E_1^t, E_2^t, \ldots, E_n^t\}, $$

where $S_t$ represents hazard state vector of road at time $t$, which is composed of the hazard of each grid of the road at time $t$, and $E_i^t$ is the hazard of grid $i$ at time $t$. 

![Figure 1: MEC network architecture.](image-url)
Using discrete time step indices \( t, t + 1, \ldots, t + s \) (time step size \( \Delta t \)), hazard state vector sequences are additionally introduced as follows:

\[
S_{t+s} = \{S_t, S_{t+1}, \ldots, S_{t+s}\},
\]

which describes the hazard state of the road from \( t \) (now) until the time \( t + s \) (future).

As mentioned earlier, considering the computing power of RSU, we employ two-level BP neural network to fulfill dynamic hazard feature prediction. The proposed network is shown in Figure 2.

As shown in Figure 2, two levels of training process, which are cloud-level training and MEC-level training, are used. Due to its strong computing power, cloud layer is used to store historic data and fulfill model training process. Training results are sent to MEC, which is deployed with RSU. According to training results, MEC executes predicting task.

The two-level training process is described as follows.

4.2.1. Cloud-Level Training. The input of BP neural network is the road hazard value set \( \{S_1, S_2, \ldots, S_n\} \). The trained model is sent to RSU.

4.2.2. MEC-Level Training. According to the trained model, MEC conducts incremental training using the hazard data from the past time \( t - m \) to \( t \) to obtain \( S_{t+1} \). Then, incremental training process output \( S_{t+2}, S_{t+3}, \ldots \), and so on, \( S_{t+s} \).

Here, BP neural network includes one input layer, one output layer, and one hidden layer, while the error rate is limited to 0.001 during training. Moreover, in order to save computation resources, the training time is limited to 10000.

4.3. Feasible Grid Discovery. In some way, a collision-free path is generated according to the feasible grid distribution feature of front road. Hence, this problem is transformed into a problem of feasible grid discovery. In general, a low-level hazard grid should be considered as a feasible grid. However, the dynamic feature of grid hazard should be taken into account either.

Assuming the threshold of hazard detection is \( E_{th} \), grid feasibility, denoted by \( f_g \), is defined as threshold-crossing frequency of grid hazard value. Then, we assume that the threshold of grid feasibility is \( f_{gh} \). Therefore, a feasible grid evaluation mechanism is proposed as follows:

If \( E_{grid}(i,t) < E_{th} \), grid \( i \) is considered as an optional feasible grid at time \( t \) and denoted by grid \( (i,t) \in G_{fg}(t) \).

Calculate \( f_g \) of all grids belonging to \( G_{fg}(t) \). If \( f_g(i,t) < f_{gh} \), grid \( i \) is set as a feasible grid at time \( t \). Then, the set of feasible grids at time \( t \) should be denoted by \( G_f(t) \).

Using discrete time step indices \( t, t + 1, \ldots, t + s \) (time step size \( \Delta t \)), feasible grids set sequences are additionally introduced as follows:

\[
G_f(t+s) = \{G_f(t), G_f(t+1), \ldots, G_f(t+s)\}.
\]

The connectivity area of feasible grids should be considered as the basis of collision-free path generation.

5. Trajectory Planning

Trajectory planning based on real-time assessment of road area hazard allows drivers to obtain the road conditions of the current driving environment and can improve road driving safety and driving efficiency to a certain extent. However, the road environment is complex and vehicle’s velocity is fast, so the effect of path planning based on real-time data is disappointing. Therefore, this paper proposes a path planning method based on dynamic assessment of road hazard. Section 4 has already introduced methods for predicting hazard, so this section will introduce the trajectory planning algorithm and a cost function defined in this paper in turn.

5.1. RRT Algorithm. The RRT algorithm is a sampling-based incremental iterative trajectory planning algorithm. The algorithm mainly uses the random and incremental generation process of sampling points in the state space to extend the direction of the path search to the unknown blank area, thus achieving fast and efficient search. The core of the algorithm is to use a random search method and can be applied to path search in complex high-dimensional space. The whole RRT algorithm node expansion search process is shown in Figure 3 below.

However, the RRT algorithm also has some drawbacks: (1) random sampling leads to slow convergence [11]; (2) the randomness of RRT algorithm lacks certain guidance information, which makes the path results more tortuous and not smooth enough.

5.2. Cost Function Definition. To overcome the shortcomings of the RRT algorithm, based on the thought of referenced RRT * algorithm [19], this paper proposes a cost function that combines the current road hazard and future hazard distributions. Incorporating this function into the RRT algorithm can make the RRT algorithm preferentially select the mesh with a low risk, thus indicating the direction for the RRT algorithm.

In our experiments, we define the cost function mainly composed of the road hazard distribution result \( E(grid,t) \) based on the Ising model mentioned in Section 4, trend characteristic \( S_{fg} \), and grid feasibility analysis result \( G_{fg}(t+s) \). Here, we define a function called FindMinCost to represent the cost function, the pseudocode is shown in Algorithm 1, and the corresponding explanation is presented in Table 1.

The specific description of the function is as follows:

**Step 1.** Start with \( x_{nearest} \), use the depth traversal method to extend \( k \) units backwards in units of grid, with only one grid per column. Use \( G_{fg}(t+s) \) to find all possible extension paths that are feasible and have no obstacles and form a path set \( R_{ij} \).

**Step 2.** Calculate the hazard of grid \( E(grid,t) \) and predict the grid hazard trend \( S_{fg} \), of each grid in \( R_{ij} \),
Figure 2: Two-level BP neural network.

Figure 3: Extended search process of RRT.

Algorithm 1: Description of FindMinCost.

Table 1: Explanation of variables and functions in FindMinCost.
which are used to evaluate the sum of hazards for each path at time $t$. Then, calculate cumulative hazard value of each trajectory at time $t + s$ and select the trajectory with the lowest cumulative hazard value as selected trajectory, which is denoted by $X_{\text{min}}$.

Step 3. Select the node closest to $x_{\text{nearest}}$ in path $X_{\text{min}}$ as the new node $x_{\text{new}}$.

5.3. Improved RRT Algorithm. The improved RRT algorithm mainly uses spatial analysis to divide the road into grids of the same size. Each step of the search tree $T$ extension will select the grid with the lowest possible risk, so that the vehicle travels to the road grid area where the road risk is low, thereby ensuring the safe passage of the vehicle through the road area.

After adding the cost function mentioned in the previous section, the pseudocode for improving the RRT algorithm is shown in Algorithm 2, and the corresponding variables and functions are explained in Table 2.

The specific algorithm flow is described as follows:

- **Step 1.** Initialize the random tree $T$ and add the starting point to the structure of tree $T$.
- **Step 2.** Select the random point $x_{\text{rand}}$ in the searching space and find the point $x_{\text{nearest}}$ from the tree that is closest to $x_{\text{rand}}$.
- **Step 3.** Starting from $x_{\text{nearest}}$, use the cost function to calculate the path with the least risk and determine the new node $x_{\text{new}}$ that should be expanded next.
- **Step 4.** Determine if there is an obstacle on the $x_{\text{nearest}} - x_{\text{new}}$ connection path; if it exists, return to step 2 to start the next iteration; if not, add the edge formed by $x_{\text{nearest}} - x_{\text{new}}$ and the new node $x_{\text{new}}$ to tree $T$. After repeated iterations, the final path planning result can be obtained.

In the improved RRT algorithm in this paper, the cost function is an important basis for random tree expansion. We focus on the hazard value of generated trajectories and corresponding dynamic feature of hazard value. To improve the safety of trajectory, the lowest hazard value grid should be selected as expansion node of the random tree.

6. Experimental Results and Analysis

6.1. Simulation Platform. All experiments are done based on SUMO and NS3 platforms. Road traffic scenarios are built by SUMO platform, while V2X communication progress is simulated by NS3 platform. SUMO + NS3 joint simulation process is shown in Figure 4.

The joint simulation process includes four procedures as follows:

6.1.1. Synchronization. NS3 platform sends clock synchronization message to SUMO platform to ensure that two platforms remain in sync.

6.1.2. Vehicle Cruising Information Acquisition. NS3 platform sends vehicle information request to SUMO, while SUMO feeds back corresponding information to NS3. Significantly, this procedure is realized based on a special communication protocol.

6.1.3. Vehicle Control Information Generating. NS3 platform performs communication process simulation and regional vehicles cruising feature analysis and then generates corresponding vehicles control information.

6.1.4. Vehicle Control Message Feedback and Control. NS3 sends generated vehicle control message to SUMO platform. The format of feedback control message is TraCI. According to received control message, SUMO executes vehicle control progress.

6.2. Simulation Parameters Setting. Here, SUMO and NS3 platform parameters setting process should be considered.

6.2.1. Simulation Parameters Setting for SUMO. Here, a four-lane straight road is considered to construct simulation. Corresponding parameters setting are given in Table 3.

6.2.2. Simulation Parameters Setting for NS3. The communication scenario of NS3 is set with 20 vehicle nodes and 2 roadside units. Corresponding parameters are shown in Table 4.

6.3. Simulation Results Analysis. In this part, we shall analyze simulation results from three aspects. Firstly, the simulation process is discussed. Then, the validity of the proposed method is considered. At last, the effectiveness of generation trajectory is analyzed.

6.3.1. Simulation Process. Here, we assume initial prediction time is 29.445 s. Time step $\Delta t$ is set as 200 ms, and the length of the historical hazard array $E_h$ is set to 11, while the future time $s$ is set as 5.

Simulation results of proposed algorithm are shown in Figure 5. Here, snapshots of 4 time points are given.

Figure 5(a) shows the snapshot of 29.445 s. Ego-car cruises along the leftmost lane. RSUs collect road vehicles information and use improved RRT algorithm to calculate grid hazard state vector sequences. And then, add the grid with minimum hazard value into the tree which the ego-car should go next. This process is performed every 200 ms. Therefore, in a snapshot of 31.445 s, the ego-car reached the position shown in (b) from position shown in (a).

Then, at the next 2 time points, which are 31.445 s and 33.445 s, ego-car executes twice lane change procedures and change cruising lane from the leftmost lane to the rightmost one.
6.3.2. Validity Analysis. To verify the validity of proposed algorithm, the prediction error is studied and the multivariate linear model method and the Long Short-Term Memory (LSTM) neural network are used to compare with it.

Here, we consider two scenarios, stable hazard value scenario and variable hazard value scenario.

### Table 2: Explanation of variables and functions in the algorithm of improved RRT.

| Variable/function | Explanation |
|-------------------|-------------|
| $T$               | The extended search tree |
| $x_{\text{init}}$ | Initial node, or the starting point of path planning |
| RandomSample      | Function of random sampling |
| $x_{\text{rand}}$ | Random points obtained by random sampling |
| Nearest $(T,x_{\text{rand}})$ | Function of finding the nearest point to $x_{\text{rand}}$ in tree $T$ |
| $x_{\text{nearest}}$ | The nearest point to $x_{\text{rand}}$ |
| FindMinCost $(x_{\text{nearest}}, k, \text{laneNum})$ | Function of finding the minimum cost path |
| $x_{\text{new}}$ | The new node to be expanded in next step |
| InsertNode $(x_{\text{min}}, x_{\text{new}}, T)$ | Function of adding new nodes to the search tree |

#### Algorithm 2: Algorithm of improved RRT.

1. $T \leftarrow \text{InitializeTree}$
2. $T \leftarrow \text{InsertNode} (\emptyset, x_{\text{init}}, T)$
3. for $i = 0$ to $N$
4.     $x_{\text{rand}} \leftarrow \text{RandomSample}$
5.     $x_{\text{nearest}} \leftarrow \text{Nearest} (T, x_{\text{rand}})$
6.     $x_{\text{new}} \leftarrow \text{FindMinCost} (x_{\text{nearest}}, k, \text{laneNum})$
7.     if Obstaclefree $(x_{\text{new}})$ then
8.         $T \leftarrow \text{InsertNode} (x_{\text{min}}, x_{\text{new}}, T)$
9. end if
10. end for
11. return $T$

### Table 3: Parameter setting of SUMO platform.

| Parameter               | Value  |
|-------------------------|--------|
| Road length             | 1000 m |
| Number of lanes         | 4      |
| Lane width               | 3.2 m  |
| Number of vehicles      | 20, 30 |
| Vehicle length          | 5.0 m  |
| Vehicle width           | 1.8 m  |
| Maximum speed           | 50 m/s |
| Maximum acceleration    | 2.6 m/s^2 |
| Maximum deceleration    | 4.5 m/s^2 |
| Minimum spacing         | 2.5 m  |

### Table 4: Parameter setting of NS3 platform.

| Parameter               | Value   |
|-------------------------|---------|
| Number of vehicle nodes | 20, 30  |
| Number of RSUs          | 2       |
| Vehicle node start and end coordinates | (0,0), (1000,0) |
| RSU node coordinates     | (300,50), (700,50) |
| Cloud node coordinates   | (0,100), (100,100), ... |
| Simulation duration      | 110 s   |

**Stable Hazard Value Scenario.** The prediction performance of stable hazard value scenario is shown in Figure 6, while the corresponding parameters are listed in Table 5.
As shown in Figure 6, the prediction performance of the proposed method is better than that of multivariate linear model method and LSTM neural network method. The relative prediction error of the proposed method is lower than 11.98%, and the average relative prediction error is 6.482%.

Variable Hazard Value Scenario. The prediction performance of variable hazard value scenario is shown in Figure 7, while the corresponding parameters are listed in Table 6.

As shown in Figure 7, the proposed method prediction curve could track actual change of hazard value. However, the dynamic lag error is very obvious. On the other hand, the
LSTM has a higher error rate and linear prediction could not follow the dramatic change of hazard values and perform a high relatively error rate.

As shown in Table 6, the performance will deteriorate obviously when the hazard value changes fast and may cause some problem for trajectory planning. However, besides the fast change time point $t = 16$, the relative prediction error of proposed method is lower than 26.5%. Moreover, the corresponding average relative prediction error is 17.798%.

### 6.3.3. Effectiveness of Generation Trajectory

To verify the effectiveness of generated trajectory, a comparison experiment is repeated 10 times, while the vehicle number is selected as 20 and 30.

To verify the effectiveness of the proposed algorithm, classic RRT, improved RRT [22], improved RRT with linear prediction, and improved RRT with LSTM prediction are selected as contrast.

Here, we consider two parameters. One is travel time, while the other is accumulated hazard value of generation trajectory.

Experiment results are given in Figure 8.

As shown in Figure 8(a), overall, the travel times of all five algorithms are basically the same. For medium-density scenario (20 vehicles), the travel time of the proposed method is longer than that of linear prediction RRT. For high-density scenario (30 vehicles), the travel time of the proposed method is longer than that of improved RRT. The main reason for this situation is that the safest trajectory is not always the shortest one.

As shown in Figure 8(b), trajectory accumulated hazard value of proposed algorithm is lower than that of all other four algorithms and proves the effectiveness of generation trajectory.
(a)

Figure 8: Continued.
7. Conclusion

In this work, we presented a novel trajectory planning approach-based MEC framework for assistance driving and autonomous driving based on road hazard state. We used Ising model to evaluate the behavior of regional traffic participants and find feasible road grid.

In this paper, a classical robotic path searching method, RRT algorithm, is employed as a basic approach to generate regional trajectory. To overcome the weakness of RRT, such as bad environmental adaptability, output unstable, etc., a cost function based on current and future hazard distributions is defined to improve the performance of RRT algorithm. This paper uses a two-level BP neural network to predict future risk distribution based on past data. Simulation results show that the proposed method properly solves the flaws of typical RRT algorithm and could generate a reliable trajectory, which could help vehicle avoid potential road risk.

However, this paper only considers 4-lane straight road scenario. In our future work, other road scenarios, such as intersection, 2-lane straight road, pedestrian, and nonmotor vehicles, should be discussed to verify the universality of the proposed algorithm as well.

Data Availability

The simulation data supporting the system performance analysis can be obtained from https://github.com/hughyun/A-MEC-architecture-oriented-improved-RRT-Algorithm-for-Regional-Trajectory-Planning.git.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported in part by Chongqing Vehicle Test and Research Institute and Open Fund for Chongqing Engineering Research Center of Research and Testing for Automated Driving System and Intelligent Connected Vehicle under Project 20AKC17 and in part by the National Natural Science Foundation of China under Project 61601066.
References

[1] L. Zeng, J. Zhang, Q. Han et al., "A bus-oriented mobile FCNs infrastructure and intra-cluster BSM transmission mechanism," IEEE Access, vol. 7, pp. 24308–24320, 2019.

[2] K. Zhang, Y. Mao, S. Leng, Y. He, and Y. Zhang, "Predictive offloading in cloud-driven vehicles: using mobile-edge computing for a promising network paradigm," IEEE Vehicular Technology Magazine, vol. 12, no. 2, 2017.

[3] F. Damerow, Y. Li, T. Puphal, B. Flade, and J. Eggert, "Intersection warning system for occlusion risks using relational local dynamic maps," IEEE Intelligent Transportation Systems Magazine, vol. 10, no. 4, pp. 47–59, 2018.

[4] Q. Han, Z. Yin, L. Zeng et al., "The road regional hazard level evaluation method based on ising model," in Proceedings of the 2018 IEEE Intelligent Vehicles Symposium (IV), pp. 1937–1942, IEEE, Suzhou, China, June 2018.

[5] M. Armbrust, A. Fox, R. Griffith et al., "A view of cloud computing," Communications of the ACM, vol. 53, no. 4, pp. 50–58, 2010.

[6] H. Li, G. Shou, Y. Hu, and Z. Guo, "Mobile edge computing: progress and challenges," in Proceedings of the 2016 4th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud), pp. 83–84, IEEE, Oxford, UK, March 2016.

[7] H. Bao, Q. Liu, C. Huang, and X. Jia, "Minimal road-side unit placement for delay-bounded applications in bus Ad-hoc networks," in Proceedings of the 2017 IEEE 36th International Performance Computing and Communications Conference (IPCCC), pp. 1–7, IEEE, San Diego, CA, USA, December 2017.

[8] Q. Han, X. Zhang, J. Zhang et al., "Research on resource scheduling and allocation mechanism of computation and transmission under MEC framework," in Proceedings of the 2019 IEEE Intelligent Transportation Systems Conference (ITSC), pp. 437–442, Auckland, NZ, USA, October 2019.

[9] F. Damerow and J. Eggert, "Balancing risk against utility: behavior planning using predictive risk maps," in Proceedings of the 2015 IEEE Intelligent Vehicles Symposium (IV), pp. 857–864, IEEE, Seoul, South Korea, June 2015.

[10] F. Damerow and J. Eggert, "Predictive risk maps," in Proceedings of the 17th International IEEE Conference on Intelligent Transportation Systems (ITSC), pp. 703–710, IEEE, The Hague, The Netherlands, September 2014.

[11] J. Eggert, "Solomon curve 2020: relating microscopic risk models with accident statistics," in Proceedings of the 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), pp. 2293–2300, IEEE, Rio De Janeiro, Brazil, November 2016.

[12] J. Eggert, "Predictive risk estimation for intelligent adas functions," in Proceedings of the 17th International IEEE Conference on Intelligent Transportation Systems (ITSC), pp. 711–718, IEEE, The Hague, The Netherlands, September 2014.

[13] X. Zhan, S. Zhang, W. Y. Szeto, and X. Chen, "Multi-step-ahead traffic speed forecasting using multi-output gradient boosting regression tree," Journal of Intelligent Transportation Systems, vol. 24, no. 2, pp. 125–141, 2020.

[14] L. Ziweng, G. Junwei, Z. Bin, L. Xin, and M. Zengtao, "Short-term traffic flow forecasting model of optimized BP neural network based on genetic algorithm," in Proceedings of the 32nd Chinese Control Conference, pp. 8125–8129, IEEE, Xi’an, China, July 2013.

[15] A. Nakamura, Y. C. Liu, and B. G. Kim, "ShortTerm multi-vehicle trajectory planning for collision avoidance," IEEE Transactions on Vehicular Technology, vol. 99, p. 1, 2020.

[16] D. González, J. Pérez, V. Milánés, and F. Nashashibi, "A review of motion planning techniques for automated vehicles," IEEE Transactions on Intelligent Transportation Systems, vol. 17, no. 4, pp. 1135–1145, 2015.

[17] L. An, T. Chen, A. Cheng, and W. Fang, "Simulation on the path planning of intelligent vehicles based on artificial potential field algorithm," Automotive Engineering, vol. 39, pp. 1451–1456, 2017.

[18] F. Zhang, J. Leitner, M. Milford, B. Upcroft, and P. Corke, "Towards vision-based deep reinforcement learning for robotic motion control," 2015, https://arxiv.org/abs/1511.03791.

[19] I. Noreen, A. Khan, and Z. Habib, "A comparison of RRT, RRT* and RRT*-smart path planning algorithms," International Journal of Computer Science and Network Security (IJCNS), vol. 16, no. 10, p. 20, 2016.

[20] L. Ma, J. Xue, K. Kawabata, J. Zhu, C. Ma, and N. Zheng, "Efficient sampling-based motion planning for on-road autonomous driving," IEEE Transactions on Intelligent Transportation Systems, vol. 16, no. 4, pp. 1961–1976, 2015.

[21] M. Du, J. Chen, P. Zhao et al., "An improved RRT-based motion planner for autonomous vehicle in cluttered environments," in Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA), pp. 4674–4679, IEEE, Hong Kong, China, June 2014.

[22] D. Zhang, Y. Xu, and X. Yao, "An improved path planning algorithm for unmanned aerial vehicle based on rrt-connect," in Proceedings of the 2018 37th Chinese Control Conference (CCC), pp. 4854–4858, IEEE, Wuhan, China, July 2018.

[23] ETSI, "Mobile edge computing: a key technology towards 5G,” 2015, http://www.etsi.org/images/files/ET-SIWhitePapers/etsi_wp11_mec_a_key_technology_towards_5g.pdf.

[24] D. Sabella, A. Vaillant, P. Kuure, U. Rauschenbach, and F. Giust, "Mobile-edge computing architecture: the role of MEC in the internet of things," IEEE Consumer Electronics Magazine, vol. 5, no. 4, pp. 84–91, 2016.

[25] L. Ye, D. Chen, L. Zeng et al., "Adaptive SWM-oriented clustering algorithm based on vehicle health evaluation for V2V communication," Computer Engineering and Applications, vol. 54, pp. 74–80, 2018.