

## Research Article

# A Decision-Making Model for Autonomous Vehicles at Urban Intersections Based on Conflict Resolution

Zi-jia Wang,<sup>1</sup> Xue-mei Chen ,<sup>1,2</sup> Pin Wang,<sup>3</sup> Meng-xi Li,<sup>1</sup> Yang-jia-xin Ou,<sup>1</sup> and Han Zhang<sup>4</sup>

<sup>1</sup>Beijing Institute of Technology, School of Mechanical Engineering, Intelligent Vehicle Research Institute, 5 South Zhong Guan Cun Street, Haidian District, Beijing 100081, China

<sup>2</sup>Advanced Technology Research Institute, Beijing Institute of Technology, Jinan 250001, Shandong, China

<sup>3</sup>University of California, Berkeley, 1357 South 46<sup>th</sup> Street, Richmond, CA 94804, USA

<sup>4</sup>Shandong Hi-Speed Construction Management Group Co., Ltd., Jinan 250001, Shandong, China

Correspondence should be addressed to Xue-mei Chen; chenxue781@126.com

Received 26 March 2020; Revised 6 November 2020; Accepted 19 November 2020; Published 13 February 2021

Academic Editor: Wenqing Wu

Copyright © 2021 Zi-jia Wang 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.

The decision-making models that are able to deal with complex and dynamic urban intersections are critical for the development of autonomous vehicles. A key challenge in operating autonomous vehicles robustly is to accurately detect the trajectories of other participants and to consider safety and efficiency concurrently into interactions between vehicles. In this work, we propose an approach for developing a tactical decision-making model for vehicles which is capable of predicting the trajectories of incoming vehicles and employs the conflict resolution theory to model vehicle interactions. The proposed algorithm can help autonomous vehicles cross intersections safely. Firstly, Gaussian process regression models were trained with the data collected at intersections using subgrade sensors and a retrofit autonomous vehicle to predict the trajectories of incoming vehicles. Then, we proposed a multiobjective optimization problem (MOP) decision-making model based on efficient conflict resolution theory at intersections. After that, a nondominated sorting genetic algorithm (NSGA-II) and deep deterministic policy gradient (DDPG) are employed to select the optimal motions in comparison with each other. Finally, a simulation and verification platform was built based on Matlab/Simulink and PreScan. The reliability and effectiveness of the tactical decision-making model was verified by simulations. The results indicate that DDPG is more reliable and effective than NSGA-II to solve the MOP model, which provides a theoretical basis for the in-depth study of decision-making in a complex and uncertain intersection environment.

## 1. Introduction

Today's driving-assistance systems have made traffic more efficient and safer and show considerable improvements towards the availability of autonomous driving. To develop the next generation of driver assistance systems or even self-driving systems, the algorithms that are capable of handling complex situations are required. Many researchers have proposed some approaches about perception [1], path planning [2], and control [3]. However, the decision-making of autonomous driving at intersections is still one of the major bottlenecks. The primary reason for the difficulty in analyzing crossing behavior is that most models may only work when given long-term, accurate predictions of the

trajectories of other participants. To address this problem, this paper will focus on developing a tactical decision-making model for autonomous vehicles in intersection crossing scenarios.

The problems of robust tactical decision-making for autonomous vehicles in a complex and dynamic urban environment have been investigated quite extensively by many organizations and researchers, such as Google [4], Carnegie Mellon University [5], Berkeley [6], and Baidu [7]. The UCB utilized a minimal future distance and a two-level dynamic threshold to perform collision prediction tasks at urban intersections [8]. BMW and the University of Munich came up with a decision-making model based on partially observable Markov decision processes [9]. NVIDIA used a

deep convolutional neural network (DCNN) to establish an end-to-end driving model [10].

In recent years, more and more researchers have begun studying decision-making behavior. Chen [11] established a vehicle decision model in an urban environment using a hierarchical finite state machine method for different drivers and road environment characteristics. Liu et al. [12] adopted the control prediction theory and the reinforcement learning theory to obtain a decision model. However, these models cannot be adapted to urban intersections. Ma et al. [13] proposed a decision-making framework titled ‘‘Plan-Decision-Action’’ for autonomous vehicles at complex urban intersections. Zhong et al. [14] proposed a model-learning-based actor-critic algorithm with the Gaussian process approximator to solve the problems with continuous state and action spaces. Xiong et al. [15] used a Hidden Markov model to predict other vehicles’ intentions and built a decision-making model for vehicles at intersections. Lv et al. [16] combined offline and online machine learning methods to establish a personalized decision model that could simulate the characteristics of driver behavior. Chen et al. [17] used the rough-set theory to extract different drivers’ decision rules. Chen et al. [18] used a novel RSAN (rough-set artificial neural network) method to learn decisions made by excellent human drivers. Chen et al. [19] proposed a merging strategy based on the least squares policy iteration (LSPI) algorithm and selected a basis function that included the reciprocal of TTC, relative distance, and relative velocity to represent the state space and discretize the action space. However, these studies did not take the overall interaction scenarios into consideration and can only be adopted for short-term trajectory prediction.

This paper focuses on the decision-making process of autonomous vehicles in an urban environment and develops a vehicle trajectory prediction model based on Gaussian process regression (GPR) [20], which can generate long-term predictions of incoming vehicles. The problem of conflict resolution among vehicles at intersections is modeled as a multiobjective optimization problem (MOP), in which the acceleration, as the only decision variable, is used to control the vehicles. The main contributions of this work are the presentations of two solutions of intersection multiobjective optimization problems. First, the noninferior genetic algorithm (NSGA-II) is applied to maximize the overall driving benefit of system; the other one considers the deep deterministic policy gradient (DDPG) algorithm of reinforcement learning with continuous actions. Its expected gradient of the action-value function means that DDPG can be estimated much more stable than the usual stochastic policy gradient. A simulation and verification platform was built to validate the results based on Matlab/Simulink and PreScan, and the proposed MOP decision-making method and calculation algorithms were verified in several typical scenarios.

The remainder of this paper is organized as follows: Section 2 elaborates upon the methodology used in this study, which includes an introduction of Gaussian process regression, nondominated sorting genetic algorithm (NSGA-II), and deep deterministic policy gradient

algorithm of reinforcement learning. Section 3 describes data acquisition and data processing. Section 4 proposes the GPR models for trajectory prediction and the MOP decision-making model based on efficient conflict resolution at intersections, which is solved by NSGA-II and DDPG. The simulation verification platform to evaluate the effectiveness and reliability of the proposed model and performance between two algorithms is introduced in Section 5. In Section 6, conclusions and future work are presented.

## 2. Methodology

*2.1. Gaussian Process Regression Model.* Gaussian process regression (GPR) is a statistical method that can make full use of raw data by considering its temporal trends and periodic changes to establish a suitable predictive model. This model has been used to predict the trajectories of vehicles and has been proven to be efficient. Compared with LSTM, its main advantage is that it is more robust when dealing with data with noise, making it more suitable for urban intersections.

The log likelihood function of the sample data is shown as follows:

$$\log p(\mathbf{y}|\mathbf{X}) = -\frac{1}{2}\mathbf{y}^T\mathbf{C}^{-1}\mathbf{y} - \frac{1}{2}\log|\mathbf{C}| - \frac{N}{2}\log(2\pi). \quad (1)$$

The joint distribution of the model’s observations and training data is shown as follows:

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{y}_* \end{bmatrix} \sim N\left(0, \begin{bmatrix} \mathbf{C} & \mathbf{K}_* \\ \mathbf{K}_*^T & \mathbf{C}(\mathbf{x}_*, \mathbf{x}_*) \end{bmatrix}\right), \quad (2)$$

where  $\mathbf{K}_* = [\mathbf{C}(\mathbf{x}_*, \mathbf{x}_1), \mathbf{C}(\mathbf{x}_*, \mathbf{x}_2), \dots, \mathbf{C}(\mathbf{x}_*, \mathbf{x}_n)]^T$  is the covariance matrix between the test data  $\mathbf{x}_*$  and the training data and  $\mathbf{C}(\mathbf{x}_*, \mathbf{x}_*)$  is the covariance matrix of the test data itself.

Therefore, the output of the model can be found with (3). By calculating the mean and variance of the output  $y_*$  of the model, the predicted mean  $\widehat{y}_*$  and predictive confidence  $\sigma_*^2$  of the model can be obtained separately:

$$\begin{cases} \widehat{y}_* = E(y_*) = \mathbf{K}_*^T\mathbf{C}^{-1}\mathbf{y}, \\ \sigma_*^2 = \text{Var}(y_*) = \mathbf{C}(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{K}_*^T\mathbf{C}^{-1}\mathbf{K}_*. \end{cases} \quad (3)$$

*2.2. Nondominated Sorting Genetic Algorithm.* In 2000, a new nondominated sorting genetic algorithm (NSGA-II) was proposed by Srinivas and Deb on the basis of the NSGA, which is a theory and method of handling the Pareto optima in multiobjective optimization problems. It is one of the most popular multiobjective genetic algorithms (GAs) in studying complex system analysis, and diversity results discovery. The structure of the algorithm is as shown in Figure 1.

The step-by-step procedure shows that NSGA-II algorithm is simple and straightforward. First, a combined population  $Rt = Pt \cup Qt$  is formed. The population  $Rt$  is of size  $2N$ . Then, the population  $Rt$  is sorted according to nondomination. Since all previous and current population

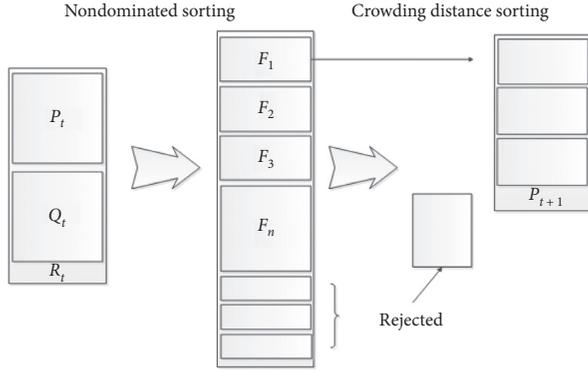


FIGURE 1: Structure of NSGA-II.

members are included in  $R_t$ , elitism is ensured. Now, solutions belonging to the best nondominated set  $F_1$  are of best solutions in the combined population and must be emphasized more than any other solution in the combined population. If the size of  $F_1$  is smaller than  $N$ , we definitely choose all members of the set  $F_1$  for the new population  $P_{t+1}$ . The remaining members of the population  $P_{t+1}$  are chosen from subsequent nondominated fronts in the order of their ranking. Thus, solutions from the set  $F_2$  are chosen next, followed by solutions from the set  $F_3$ , and so on. This procedure is continued until no more sets can be accommodated. Say that the set is the last nondominated set beyond which no other set can be accommodated [21].

**2.3. Deep Deterministic Policy Gradient.** The interactive learning process of reinforcement learning is similar to human learning, which can be represented as a Markov decision process consist of  $(s, a, P, r)$ . In 2013, the DQN [22] algorithm was proposed by DeepMind, opening a new era of deep reinforcement learning. The core improvement of the algorithm is to use experience replay and build a second target network [23], which eliminate the correlation between the training samples and improve stability of training. Some algorithms evolved by DQN have made great progress in discrete action control problem, but it is difficult to learn continuous strategy control problem. In 2015, DeepMind proposed the DDPG algorithm based on the DPG and DQN algorithms [24], importing the normalization mechanism in deep learning [25]. Experiments show that the proposed algorithm performs well on multiple kinds of continuous control problems.

The DDPG algorithm is an improved actor-critic method. In the actor-critic algorithm, the actor function  $\pi(s|\phi)$  generates an action given the current state. The critic evaluates an action-value function  $Q(s, a|\theta)$  based on the output from actor, as well as the current state. The TD (temporal-difference) errors produced from the critic drive the learning in the critic network, and then actor network is updated based on policy gradient.

The DDPG algorithm combines the advantages of the actor-critic and DQN algorithms so that the converge becomes easier. In other words, DDPG introduces some concepts from DQN, which are employing the target

network and estimate network for both of the actor and critic. Moreover, the policy of the DDPG algorithm is no longer stochastic but deterministic. It means the only real action is outputted from the actor network instead of telling probability of different actions. The critic network is updated based on

$$L = \frac{1}{N} \sum_i^N (Q(s_t, a_t|\theta^Q) - y_i)^2, \quad (4)$$

where  $y_i = r_i + \gamma Q'(s_{t+1}, a_t|\theta^Q)$  is the Q value estimated by target network and  $N$  indicates the total number of mini-batch size. The actor network is updated by means of the gradient term

$$\nabla_{\theta^\mu} J \approx \frac{1}{N} \sum_i^N \nabla_a Q(s, a|\theta^Q)|_{s=s_i, a=\mu(s_i)} \nabla_{\theta^\mu} \mu(s|\theta^\mu)|_{s=s_i}, \quad (5)$$

where  $Q(s, a|\theta^Q)$  is from critic estimate network. Furthermore, the DDPG algorithm solves continuous action space problem by means of experience replay and asynchronous updating. The updates of the target critic and target actor networks are as follows:

$$\theta^{Q'} \leftarrow \tau \theta^Q + (1 - \tau) \theta^{Q'}. \quad (6)$$

### 3. Data

The data were collected from the intersections of Wei Gong Cun Road using subgrade sensors and a retrofit autonomous vehicle as the training and testing samples of the trajectory prediction model. The details are discussed in the following section.

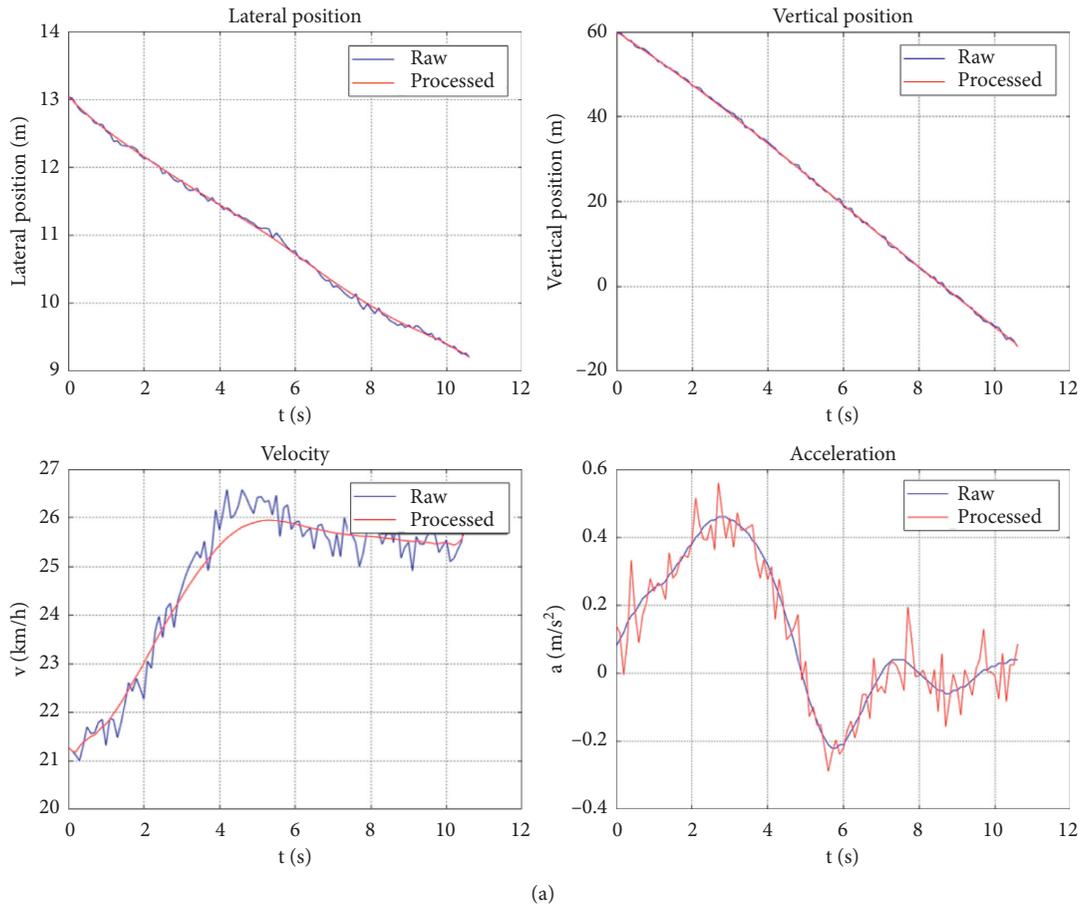
**3.1. Subgrade Data Acquisition.** The camera for subgrade data acquisition was installed on the BIT Science and Technology Building. The vehicles' locations  $(x, y, z)$ , velocities  $(v)$ , and accelerations  $(a)$  were extracted. The symmetric exponential moving average (SEMA) method [26] was adopted to smooth out the training data.

**3.2. Vehicle Data Acquisition.** The vehicle data were collected with a BYD line-controlled autonomous vehicle which was retrofitted by the BIT Intelligent Vehicle Research Institute. The retrofit autonomous vehicle "Surui" [27] was equipped with several kinds of sensors, as shown in Figure 2(b).

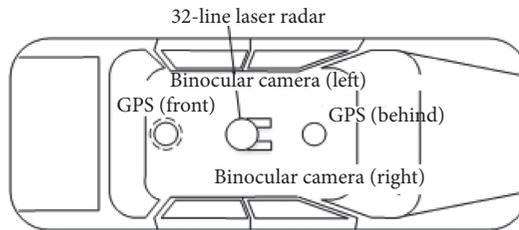
The camera and LIDAR sensor were able to detect, track, and localize dynamic objects. The outputs of the fusion algorithm are the positions of vehicles.

### 4. Model

**4.1. Analysis of Driving Behavior at Intersections.** Due to the different driving directions and routes of vehicles at intersections, a collision may occur. As shown in Figure 3(a), a blue unmanned vehicle (unmanned vehicle, UV) may collide with a yellow manned vehicle (manned vehicle, MV) that

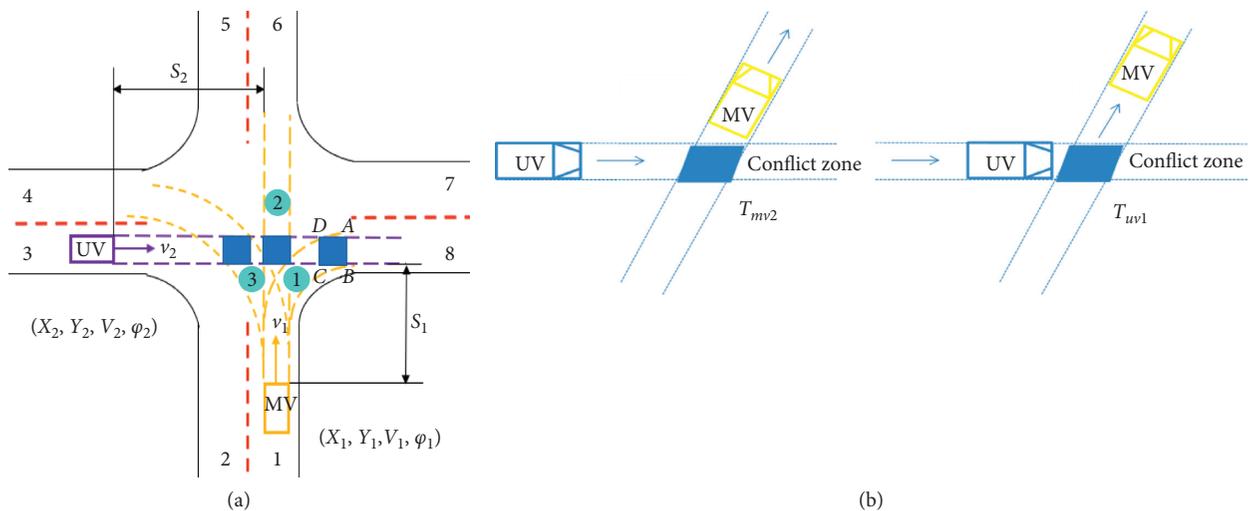


(a)



(b)

FIGURE 2: (a) The raw and processed data. (b) The layout of sensors on the BYD retrofit vehicle.



(a)

(b)

FIGURE 3: Scenes on crossing intersection with no signal.

may go straight and turn right or turn left at blue areas which called conflict zones. Therefore, a decision-making model was established to avoid collisions in these spaces based on vehicles crossing the intersection at different times. This paper only focused on the impacts of motor.

#### 4.2. Research on Vehicle Trajectory Prediction

**4.2.1. Feature Motion Parameter Extraction.** Vehicle course angle and azimuth were extracted to distinguish if a vehicle turned or not, because these two parameters change linearly with time when vehicles turn. By utilizing vehicles' motion parameters to recognize driving patterns, incoming vehicles' trajectories were predicted effectively. Real-time acceleration was used to distinguish if a vehicle kept driving or gave way to incoming vehicles because vehicles' real-time accelerations for the two patterns are distributed across different ranges.

**4.2.2. Trajectory Prediction Model.** A trajectory prediction model based on the GPR model was used to predict the trajectories of MVs. The training process of GPR models [28] is shown in Figure 4(a).

In this paper, the data collected from the subgrade sensors were used for training the GPR models and optimizing its hyper parameters.  $(x(t), y(t), v_x(t), v_y(t))$  were the inputs, while  $a_x(t)$  was the output. A square exponential covariance function (SE) was adopted as the kernel function, because it can accurately describe the nonlinear relationships between the inputs and outputs. A conjugate gradient optimization algorithm was then adopted to search for optimal parameters. When the error fell below 0.001, the results were regarded as convergent.

After training the prediction model, as this paper paid more attention to straight driving MVs, the CA (constant acceleration) [29] kinematic formula is utilized to calculate the follow-up trajectories more accurately, as shown in Figure 4(b).

**4.3. A Decision-Making Model Based on Efficient Conflict Resolution.** An appropriate parameter should be selected to analyze the traffic conflict. TTC (time to collision) is a widely used parameter in traffic conflict research, but it is generally used for scenes such as highway and is improper to evaluate the danger degree of vehicles collision at intersections. We use EPET (estimating postencroachment time) as the safety indicator which describes the time difference between vehicles passing through the center of conflict zone and can effectively evaluate collision danger between vehicles at any angles, as shown in Figure 3(b):

$$\text{EPET} = |T_{uv} - T_{mv_i}|, \quad (7)$$

where  $T_{uv}$  and  $T_{mv_i}$  are, respectively, the time when UV and MV(i) arrive the conflict zone. EPET is expected a larger value which means smaller risk of collision.

While ensuring safety, an appropriate speed is expected, which stands for efficiency during crossing the intersection. Using these criteria, we define the following measure combining safety and efficiency:

$$\begin{aligned} U &= U_{\text{safe}}(\text{EPET}) + U_{\text{efficiency}}(V) \\ &= -\exp(-\text{EPET}) - \frac{(V_{\text{MV}} - V_{\text{cri}})^2}{V_{\text{cri}}^2}, \end{aligned} \quad (8)$$

where  $U$  is the profit function, and we expect a larger  $U$  that represents more ideal motions during the crossing for vehicles.  $V_{\text{cri}}$  is the expected speed for MV, which is set to 40 km/h according to the driving rules at the intersections. The reason for defining the  $U$  negative is ensuring efficiency in the following model, e.g., deep reinforcement learning.

As the states and actions of vehicles are continuous, we use acceleration  $a$  as a parameter to control target. A constrained model of multiobjective optimization problem (MOP) is proposed based on conflict resolution at intersection, and the goal of which is to maximize profit of the system. The interaction between vehicles is quantized by importing a variable parameter  $P$ , and vehicles will cross the intersection in competition when  $P$  is zero. When  $P$  is 1, vehicles will be in cooperation completely.

The mathematical model of MOP is usually expressed as follows:

$$\begin{cases} \min f(X), \\ h_i(X) = 0, \quad i = 1, 2, \dots, m, \\ g_j(X) \geq 0, \quad j = 1, 2, \dots, l, \end{cases} \quad (9)$$

where  $f(X)$  is object function and  $h_i(X) = 0$  and  $g_j(X) \geq 0$  are constraint conditions. For solving the maximum of  $U$ , it can be transformed into finding the minimum of negative function. Therefore, we can then establish

$$\begin{aligned} \max \quad & U = U_1 + p \cdot U_2, \\ \text{s.t.} \quad & a_{\min} \leq a_i \leq a_{\max}, i = 1, 2, \\ & 0 \leq v_i \leq v_{\max}, i = 1, 2. \end{aligned} \quad (10)$$

$v_{\max}$  depends on the speed limit at the intersections, and  $a_{\max}$  and  $a_{\min}$  represent comfort requirement during driving, which are defined as  $\pm 2m/s^2$  in this paper.

#### 4.4. The Calculation Method Based on NSGA-II

**4.4.1. Constraint Condition.** To ensure safety, a simplified circle model for vehicles is established, as shown in Figure 5.

We set a safety constraint for no overlap between the excircles of vehicles:

$$\sqrt{((x_1(t) - x_2(t))^2 + (y_1(t) - y_2(t))^2)} \neq R_1 + R_2, \quad (11)$$

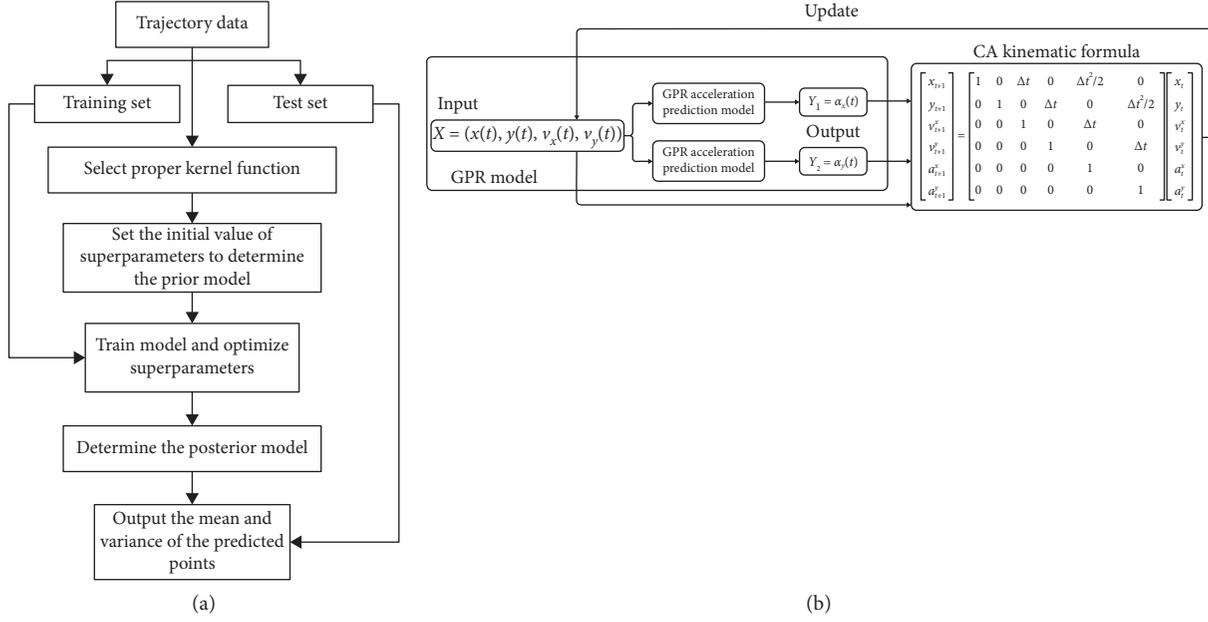


FIGURE 4: (a) The training process of the GPR model. (b) The trajectory prediction model.

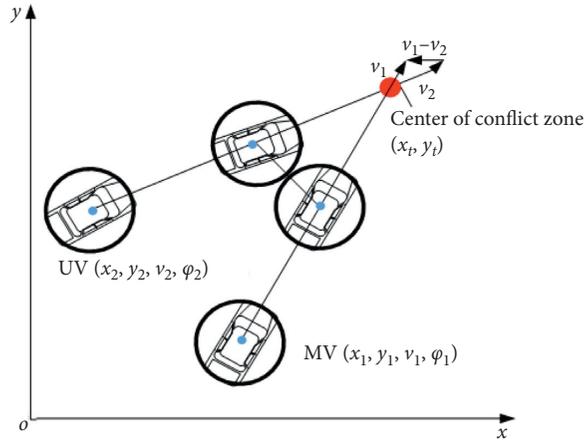


FIGURE 5: Simplified circle model.

where  $R_i = \sqrt{L_i^2 + W_i^2}/2$ , where  $L$  and  $W$  are, respectively, the length and width of vehicles.

The formula for the motion state of vehicles is as follows:

$$\begin{cases} x_i(t) = x_i(0) - \left[ v_i(0)t + \frac{1}{2}a_i(0)t^2 \right] \cos \varphi_i, \\ y_i(t) = y_i(0) - \left[ v_i(0)t + \frac{1}{2}a_i(0)t^2 \right] \sin \varphi_i, \end{cases} \quad 0 \leq t \leq T, \quad (12)$$

where  $(x_i(0), y_i(0))$  is initial position and  $\varphi_1$  is orientation.

**4.4.2. Process of Decision Making.** For the model of MOP, we perform an optimal solution based on NSGA-II, and the process is shown in Figure 6.

There are two stages in the solution process: the first phase is decision making at the initial moment and performing the action with the known information, and the second phase is to update the position and velocity of vehicles with dynamic information and then regenerate optimal motions.

**4.5. The Calculation Method Based on Deep Reinforcement Learning.** If we assume that the process of crossing intersections is a Markov decision process (MDP), it is practical to apply deep reinforcement learning for continuous action spaces. The input state is the speed of vehicles and distance from the center of vehicles to the center of conflict zone, i.e.,  $S = (V_{MV}, D_{MV}, V_{UV}, D_{UV})$ ; if there are more than one UV on the scene, all the speed and distance of UVs are appended into observation state. The output action is the acceleration  $a$

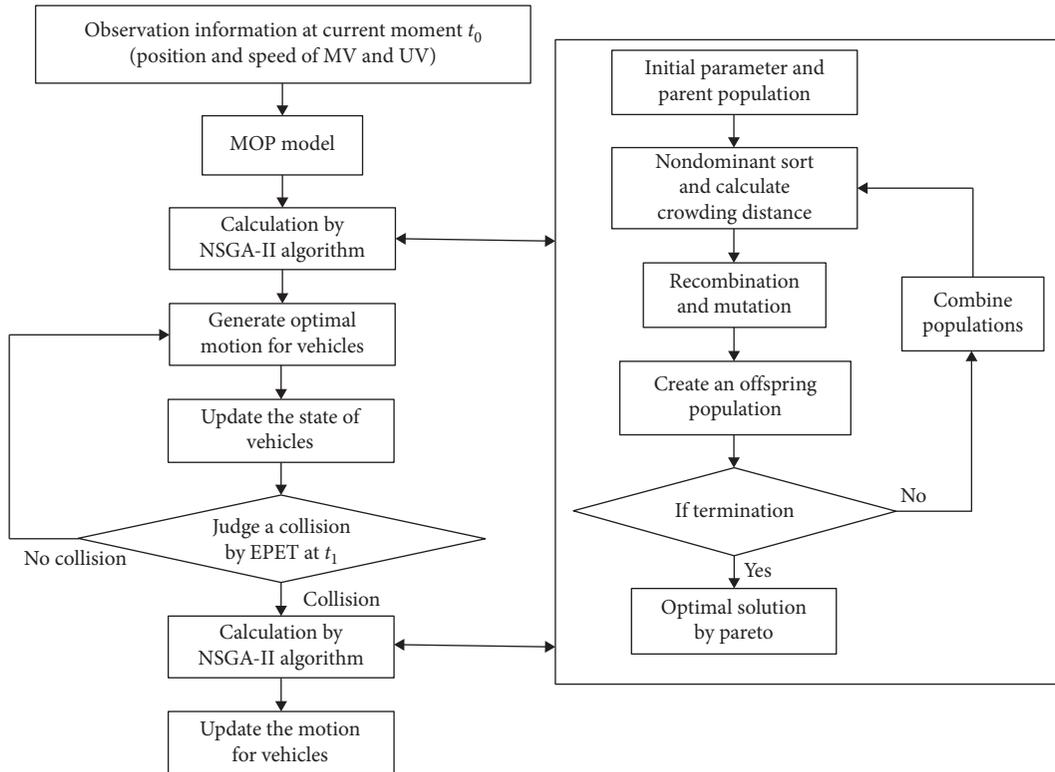


FIGURE 6: Process of the MOP model at intersections by the NSGA-II algorithm.

of MV. In this study, the reward function is built in the same way as (8),  $R = U$ , which is with consideration of safety and efficiency. We expect a larger total reward that means the sum of the rewards for each step and converge it through training based on policy gradient, which is the reason why we set the reward is negative. As for a positive reward function, a larger total reward may result from more step, which means more time to cross intersections by an inefficient policy. However, for a negative reward function, a larger total reward means a safe and efficient policy.

## 5. Discussion and Evaluation

In this section, we trained DDPG on OpenAI Gym and then tested the algorithms on PreScan to compare. This allowed us to verify the effectiveness and reliability of the proposed algorithms.

Simulation parameters are set as follows: we test the algorithms in single or multiple-vehicle scenes where there is one or more MVs driving straight from north to south, and a UV is expected to cross the intersection controlled by algorithms with no collision. The length and width of vehicle MV and UV are 4800 mm and 2178 mm, respectively, communication distance range is less than 200 m apart from each other, and speed limit at intersection is 60 km/h.

**5.1. Simulation and Verification Platform.** PreScan is a simulation environment for developing advanced driving assistant systems (ADASs) and intelligent vehicle (IV) systems. It is a platform that can be used to build 3D virtual

traffic scenes, generate vehicles, pedestrians, traffic lights, and other control modules, as shown in Figure 7(a). PreScan comes with a powerful graphics preprocessor, a high-end 3D visualization viewer, and a connection to standard MATLAB/Simulink. It is composed of various main modules. Some of these main modules represent a specific world. Multiple sensor readings were simulated and captured in the Sensor World.

We build a new task about intersection with multiple vehicles on OpenAI Gym, as shown in Figure 7(b). The deterministic actor policy network and critic policy network have the same architectures, which are multilayer perceptions with two hidden layers (64-64). For the metaexploration policy, we implemented a stochastic Gaussian policy with a mean network or variance network represented with a MLP with two hidden layers (64-64).

## 5.2. Analysis of Experimental Results

**5.2.1. Results of Prediction Model.** In this paper, the predictions of steering-vehicle trajectories and the straight vehicle trajectories are verified separately. These trajectories are divided into several different pieces to evaluate the prediction performance. The prediction lengths of the straight vehicle are 3 s, 4 s, 5 s, and 6 s. The prediction lengths of steering-vehicle are 3 s, 4 s, and 5 s. There are 80 trajectories in each group.

Figure 8(a) shows the prediction error of the straight vehicle trajectories. It can be found that the GPR model has better performance than the commonly used model in

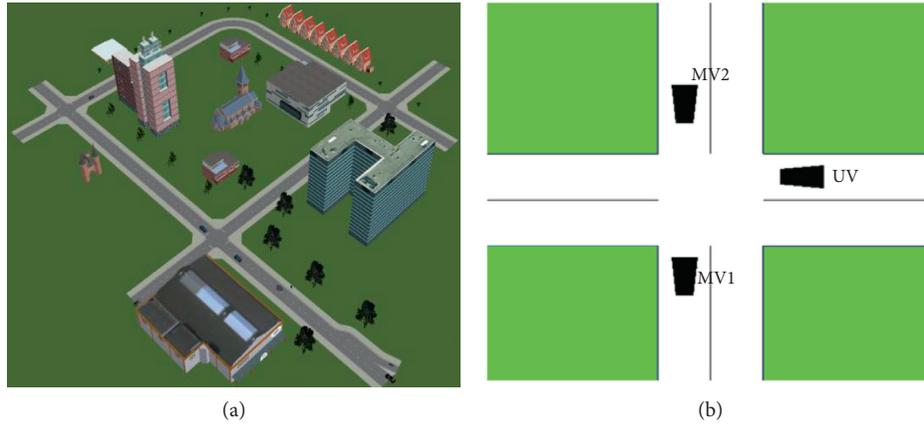


FIGURE 7: Simulation platform. (a) PreScan. (b) OpenAI Gym.

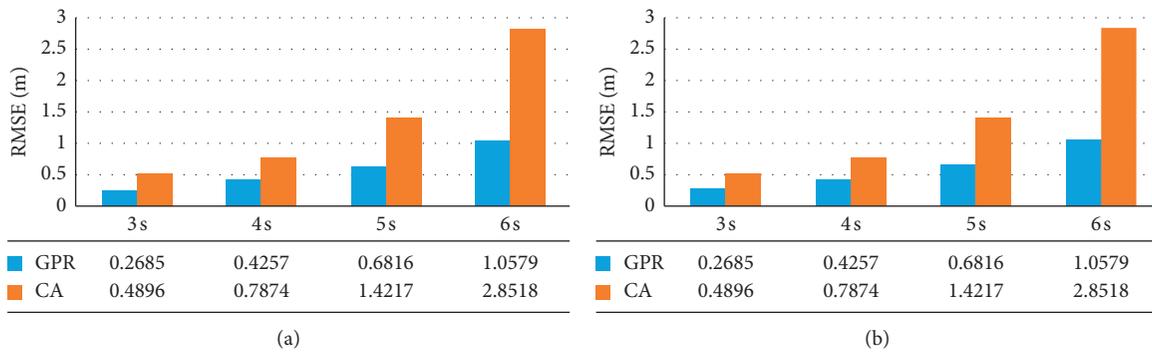


FIGURE 8: The trajectory prediction error of straight vehicle and left-turn vehicle. (a) Straight vehicle. (b) Left-turn vehicle.

prediction of straight vehicle trajectories. Figure 8(b) shows the prediction error of the steering-vehicle trajectories. It can be found that the GPR model is more accurate than the constant-rate steering motion model (CTRV).

### 5.2.2. Effect of MOP Model. Scenario 1: single-vehicle scenario

Figure 9(a) depicts the interaction between a UV and an incoming MV. Two experiments were carried out in the simulation platform. The difference between the two experiments was whether the UV was controlled by the tactical decision-making algorithm or not. In the first experiment, without the proposed algorithm, a collision between the MV and UV happened at  $t = 5.8$  s. In the other experiment, the main vehicle was controlled by the proposed algorithm. When the two vehicles met at the intersection, the main vehicle predicted the trajectory of the other vehicle, which is shown in Figure 9(b). In this experiment, deceleration was the optimal choice. The desired velocities given by the decision-making algorithm and the actual velocity changes are shown in Figure 9(c). There was no collision because the algorithm chose to yield to the incoming vehicle.

Figure 9(c) shows that with the decision-making algorithm, the main vehicle decelerates before entering

conflict zone, thus slowing down to give way to the incoming vehicle. Figures 9(d) and 9(e) show the distances and TTCs of the two vehicles. Before the algorithm is executed, both the distance and the TTC curves of the two vehicles pass through  $x = 0$ , indicating that a collision occurs at this time. After the algorithm is executed, the distance and the TTC remain within the safe range, indicating that no collision occurred.

### 5.2.3. Comparison of NSGA-II and DDPG Algorithm. Scenario 2: multiple-vehicle scenario

To compare the performances of the DDPG and NSGA-II algorithms, we conducted two groups of experiments on the same scene, in which  $D_{MV1}$  and  $D_{MV2}$  were, respectively, 10 m and 32 m, and the initial position of the UV, i.e.,  $D_{UV}$ , was 30 m. We set MV1 and MV2 to drive with a constant speed of 40 km/h. Subsequently, we trained the DDPG algorithm based on the MOP model, tested the performance in group B, and compared it with that of NSGA-II in group A, as shown in Figure 10.

For group A, the UV adopts a yield strategy wherein it slows down before  $t = 3$  s to wait for MV1 and MV2 to cross the intersection and then accelerates after the MVs move away. As shown in Figure 10(a), as the speed becomes

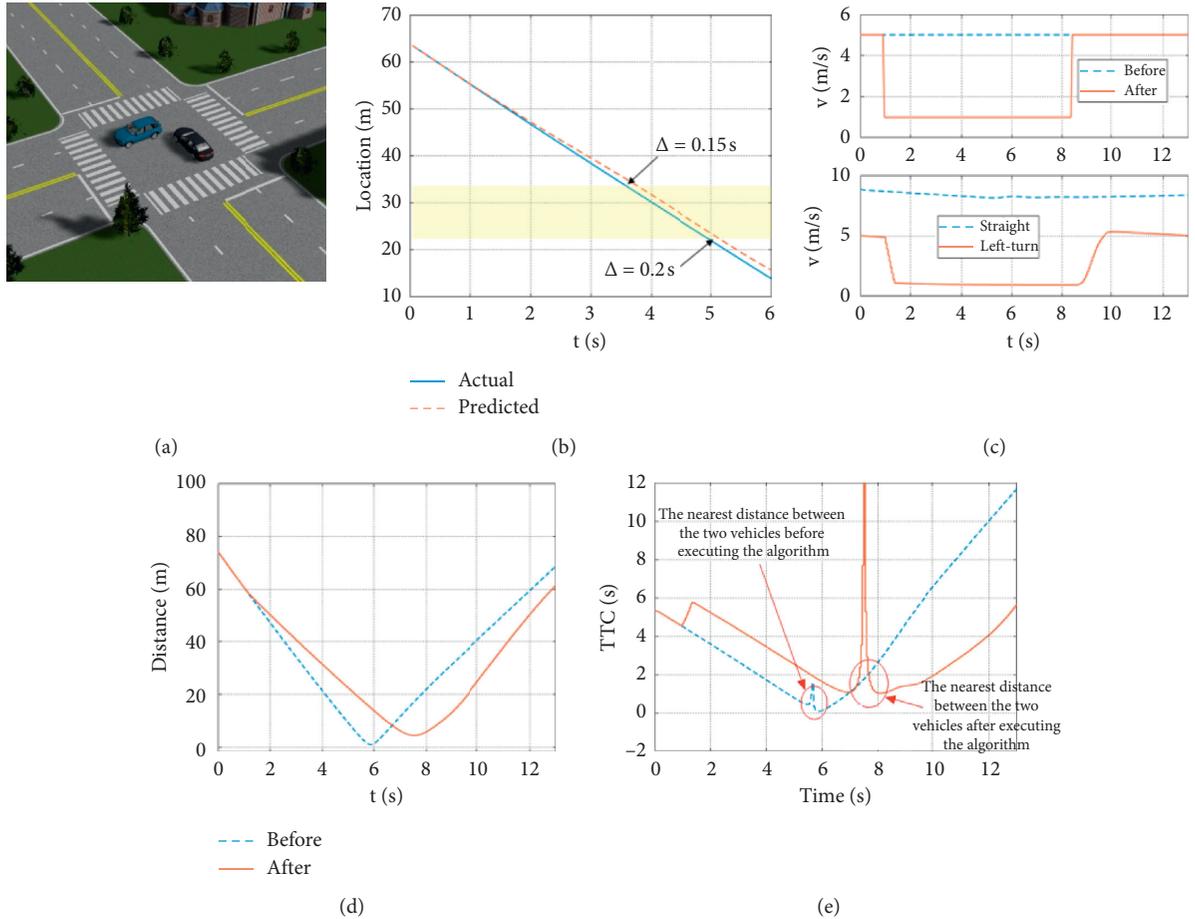


FIGURE 9: (a) Scenario 1. (b) Predicted trajectory. (c) Velocity. (d) Distance. (e) TTC.

increasingly lower than the expected speed, the reward appears to decline until  $t=3\text{ s}$  and increases thereafter. A higher crossing time means a higher accumulation of the negative reward, which leads to a lower total reward of  $-44.184$ .

Figure 10(b) shows that the UV passes through the intersection between the two MVs with an efficient strategy in group B; as shown in the bottom image in Figure 10(b), the UV reaches the conflict zone at  $t=2\text{ s}$ , approximately  $0.5\text{ s}$  earlier than MV2. In the image, the shaded area represents the conflict zone in consideration of the size of the vehicles. With the efficient strategy of the DDPG, the UV maintains an acceleration of  $2\text{ m/s}^2$  during the entire process of passing through the intersection, thus achieving a much higher total reward than that in group A.

The comparison data in Table 1 show that the passing through time for the UV of group B is approximately  $1.5\text{ s}$  lower than that of group A, which means that the DDPG algorithm reduces the traffic delay and improves the

efficiency with which the UV passes through the intersection. Moreover, the rate of change in the acceleration of the UV is lower in group B, which implies a lower energy consumption. In general, the DDPG algorithm is more efficient than NSGA-II.

The stability of the DDPG and NSGA-II algorithms was studied by performing a new task wherein the initial speed of the UV was varied from  $30\text{ km/h}$  to  $55\text{ km/h}$ .

We built a single-vehicle scene, where there is only one UV, and imported the trained actor policy of the DDPG to output the motions of the UV. We then imported the NSGA-II algorithm as a compared group to observe the performance on the same task 10 times. As shown in Figure 11, because the NSGA-II algorithm was recalculated each time, the total reward is quite different at the same initial speed of the UV. On the other hand, the DDPG gives a more stable and efficient result, and the average of the total rewards of the DDPG is higher than that of NSGA-II. Furthermore, the averages of the total

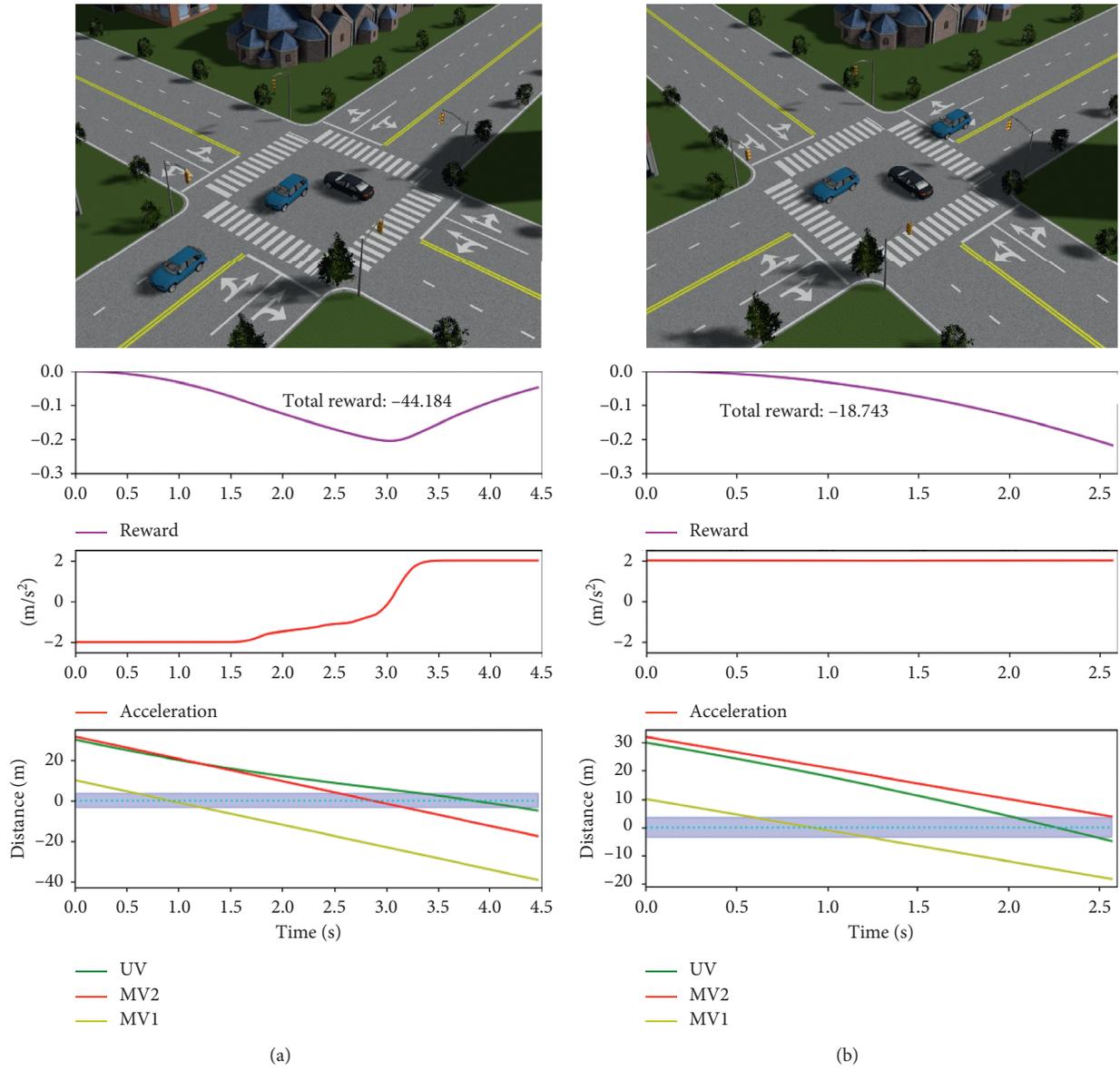


FIGURE 10: Comparison of the performance of the two algorithms. (a) NSGA-II. (b) DDPG.

TABLE 1: Comparison data of two algorithms.

Algorithm	Time to cross the conflict zone for UV (s)	Total reward	$\dot{a}_{max}(m/s^3)$
NSGA-II	3.75	-44.184	5
DDPG	2.25	-18.743	0

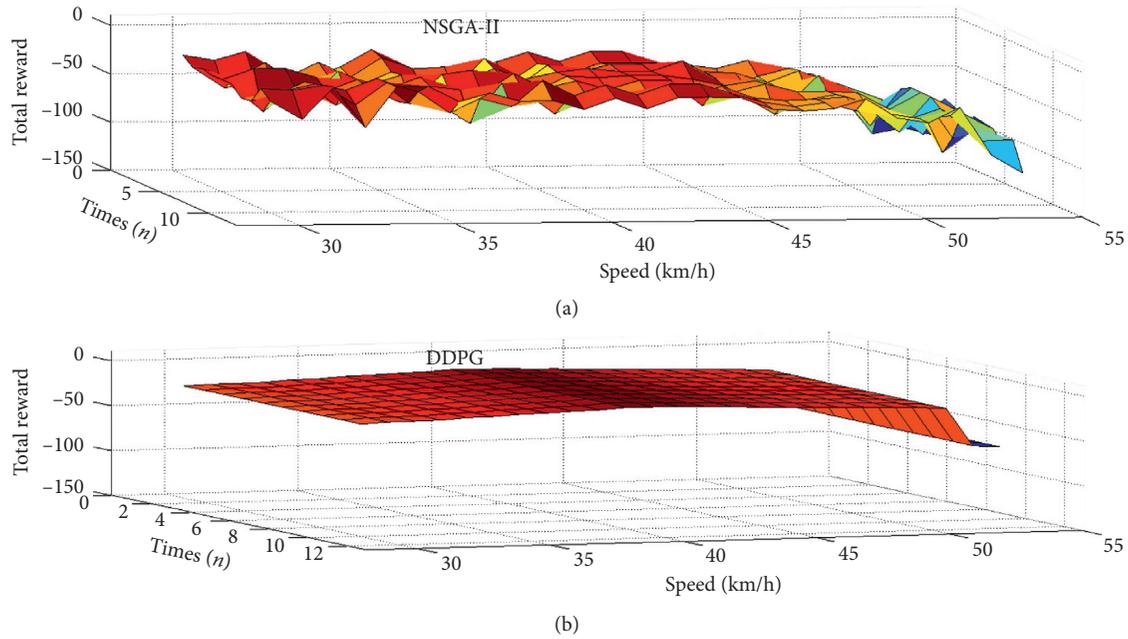


FIGURE 11: Stability and robustness of two algorithms.

rewards of the two algorithms decrease when the initial speed is greater than 50 km/h, which indicates the possibility of a collision.

## 6. Conclusion and Future Work

To improve the safety and efficiency of autonomous vehicles, this paper proposed a MOP decision-making model based on efficient conflict resolution for autonomous vehicles at urban intersections, which considers the complexity of urban intersections and the uncertainties of vehicle behavior. The prediction algorithm for incoming vehicles was studied, and we compare the performance for UV at intersections based on the decision-making model by NSGA-II and DDPG. The main conclusions are listed as follows:

- (1) The trajectory prediction model fits the predicted trajectory by learning the probability distribution of a large amount of trajectory data, and the accuracy of the model depends on the quantity and quality of the training data. The incoming vehicle trajectory data collected in this paper was limited and was unable to cover all the incoming vehicle motion patterns.
- (2) The MOP decision-making model performs well, which can avoid a collision for vehicles happened at intersections. Compared to a traditional machine learning algorithm, NSGA-II, the performance of DDPG algorithm is more stable and effective to solve the MOP model at intersections, and UVs perform the more appropriate and efficient motions by DDPG.

The decision making of autonomous vehicles is influenced by human-vehicle-road (environmental) factors. Due to limits on the length of this article, the impacts of pedestrians, non-motor vehicles, road structure types, and traffic flow density on decision-making were not considered in this study. In the future, the impacts of these factors will be studied and discussed. The interactions between people and vehicles will be considered to further improve the decision-making model of driving behavior under real road conditions.

## Data Availability

The data used to support the findings of this study are provided in the Supplementary Materials section.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This work was supported in part by the Youth Science Fund (no. 51705021), Automobile Industry Joint Fund (no. U1764261) of the National Natural Science Foundation of China, Beijing Municipal Science and Technology Project (no.Z191100007419010), and Key Laboratory for New Technology Application of Road Conveyance of Jiangsu Province (no. BM20082061706).

## Supplementary Materials

The data collected from the intersections of Wei Gong Cun Road in Beijing using subgrade sensors and a retrofit

autonomous vehicle were as the training and testing samples of trajectory prediction model. And the data were divided into three categories: left-turn vehicles, right-turn vehicles, and straight vehicles. Every category included the vehicles, information like location, speed, acceleration, and so on. (*Supplementary Materials*)

## References

- [1] M. Paul Mureşan, I. Giosan, and S. Nedeveschi, "Stabilization and validation of 3D object position using multimodal sensor fusion and semantic segmentation," *Sensors*, vol. 20, no. 4, 2020.
- [2] H. Kim, J. Cho, D. Kim et al., "Intervention minimized semi-autonomous control using decoupled model predictive control," in *Proceedings of the Intelligent Vehicles Symposium. IEEE*, Las Vegas, NV, USA, July 2017.
- [3] M. R. Boukhari, A. Chaibet, M. Boukhniifer et al., "Exteroceptive fault-tolerant control for autonomous and safe driving," in *Automation Challenges of Socio-technical Systems* Wiley Online Library, New Jersey, NY, USA, 2019.
- [4] S. Gibbs, *Google Sibling Waymo Launches Fully Autonomous Ride-Hailing Service*, The Guardian, London, UK, 2017.
- [5] S. Zelinski, T. Koo, and S. Sastry, "Optimization-based formation reconfiguration planning for autonomous vehicles," in *Proceedings of the 2003 IEEE International Conference on Robotics and Automation*, September 2003.
- [6] C. Urmson, J. C. Baker, B. P. Salesky Rybski, W. Whittaker, D. Ferguson, and M. Darms, "Autonomous driving in traffic: boss and the urban challenge," *AI Magazine*, vol. 30, no. 2, pp. 17–28, 2009.
- [7] H. Whittaker, Z. Fan, C. Liu et al., "Baidu apollo em motion planner," 2018, <http://arxiv.org/abs/1807.08048>.
- [8] P. Wang and C.-Y. Chan, "Vehicle collision prediction at intersections based on comparison of minimal distance between vehicles and dynamic thresholds," *Iet Intelligent Transport Systems*, vol. 11, no. 10, pp. 676–684, 2017.
- [9] C. Hubmann, M. Becker, D. Althoff et al., "Decision making for autonomous driving considering interaction and uncertain prediction of surrounding vehicles," in *Proceedings of the 2017 IEEE Intelligent Vehicles Symposium (IV)*, June 2017.
- [10] M. Bojarski, D. Del Testa, D. Dworakowski et al., "End to end learning for self-driving cars," 2016, <http://arxiv.org/abs/1604.07316>.
- [11] J. Chen, *Research on Decision Making System of Autonomous Vehicle in Urban Environments*, University of Science and Technology of China, Hefei, China, 2014.
- [12] C. Liu, R. Zheng, and Q. Guo, "A decision-making method for autonomous vehicles based on simulation and reinforcement learning," in *Proceedings of the International Conference on Machine Learning & Cybernetics*, Tianjin, China, July 2013.
- [13] Z. Ma, J. Sun, and Y. Wang, "A two-dimensional simulation model for modelling turning vehicles at mixed-flow intersections," *Transportation Research Part C: Emerging Technologies*, vol. 75, pp. 103–119, 2017.
- [14] S. Zhong, J. Tan, H. Dong et al., "Modeling-learning-based actor-critic algorithm with Gaussian process approximator," *Journal of Grid Computing*, vol. 18, pp. 181–195, 2020.
- [15] G. Xiong, Y. Li, S. Wang, X. Li, and P. Liu, "HMM and HSS based social behavior of intelligent vehicles for freeway entrance ramp," *International Journal of Control and Automation*, vol. 7, no. 10, pp. 79–90, 2014.
- [16] C. Lv, C. Li, Y. Xing, C. Lu et al., "Hybrid-learning-based classification and quantitative inference of driver braking intensity of an electrified vehicle," *IEEE Transactions on Vehicular Technology*, vol. 99, no. 1, 2018.
- [17] X. Chen, G. Tian, C.-Y. Chan, Y. Miao, J. Gong, and Y. Jiang, "Bionic lane driving of autonomous vehicles in complex urban environments: decision-making analysis," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2559, no. 1, pp. 120–130, 2016.
- [18] X. Chen, M. Jin, M. Yi-song, and Q. Zhang, "Driving decision-making analysis of car-following for autonomous vehicle under complex urban environment," *Journal of Central South University*, vol. 24, pp. 1476–1482, 2017.
- [19] X.-m. Chen, Q. Zhang, Z.-h. Zhang, G.-m. Liu et al., "Research on intelligent merging decision-making of unmanned vehicles based on reinforcement learning," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, pp. 91–96, Changshu, Suzhou, China, July 2018.
- [20] M. Chen, C. Pan, B. Yin et al., "Ship navigation trajectory prediction based on Gaussian process regression," *Technology Innovation and Application*, vol. 31, pp. 28–29, 2017.
- [21] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, 2002.
- [22] V. Mnih, K. Kavukcuoglu, D. Silver et al., "Playing Atari with deep reinforcement learning," 2013, <http://arxiv.org/abs/1312.5602>.
- [23] L. J. Lin, *Reinforcement Learning for Robots Using Neural Networks*, Carnegie-Mellon University, Pittsburgh, PA, USA, 1993.
- [24] T. P. Lillicrap, J. J. Hunt, A. Pritzel et al., "Continuous control with deep reinforcement learning," 2015, <http://arxiv.org/abs/1509.02971>.
- [25] S. Ioffe and C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," 2015, <http://arxiv.org/abs/1502.03167>.
- [26] M. Stuart, "FSM design and verification," *Electronic Engineering*, vol. 71, pp. 17–18, 1999.
- [27] Y. Gu, Y. Hashimoto, L. T. Hsu et al., "Motion planning based on learning models of pedestrian and driver behaviors," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, Rio de Janeiro, Brazil, November 2016.
- [28] M. Du, *Trajectory Prediction Method of Surrounding Vehicles at Urban Intersections Based on Motion Modes Recognition*, Beijing Institute of Technology, Beijing, China, 2019.
- [29] N. Zhao, W. Chen, Y. Xuan et al., "Focus and shift of visual attention in driving scenes," *Ergonomics*, vol. 17, no. 4, pp. 85–88, 2011.