

Research Article

A Multiobjective Cooperative Driving Framework Based on Evolutionary Algorithm and Multitask Learning

Xia Jiang ¹, Jian Zhang ^{1,2}, Qing-yang Li,¹ and Tian-yi Chen³

¹Jiangsu Key Laboratory of Urban ITS,

Jiangsu Province Collaborative Innovation Center of Modern Urban Traffic Technologies and Jiangsu Province Collaborative Innovation Center for Technology and Application of Internet of Things, School of Transportation, Southeast University, Nanjing 210096, China

²School of Engineering, Tibet University, Lhasa, Tibet 850000, China

³Department of Civil & Environmental Engineering, University of Wisconsin-Madison, 1221 Engineering Hall, 1415 Engineering Drive, Madison, WI 53706, USA

Correspondence should be addressed to Jian Zhang; jiangzhang@seu.edu.cn

Received 23 July 2021; Revised 18 December 2021; Accepted 30 December 2021; Published 21 January 2022

Academic Editor: Peng Hang

Copyright © 2022 Xia Jiang 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 development of connected and automated vehicle (CAV) techniques brings an upcoming revolution to traffic management. The control of CAVs in potential conflict areas such as on-ramps and intersections will be complex to traffic management when considering their deployment. There is still a lack of a general framework for dispatching CAVs in these bottlenecks, which is expected to ensure safety, traffic efficiency, and energy consumption in real time. This study aimed to fill the technique gap, and a comprehensive cooperative intelligent driving framework is put forward to study the problem, which can be used in both on-ramp and intersection scenarios. Based on a multi-objective evolutionary algorithm, CAVs are denoted as a sequence to be searched in solution space, while a multitask learning neural network with adaptive loss function is implemented for optimization target feedback to surrogate the simulation test procedure. The simulation results show that the proposed framework can get satisfying performance with low time and energy consumption. It can reduce time consumption by up to 16.51% for the on-ramp scenario and 9.8% for the intersection scenario, while reducing energy consumption by up to 16.39% and 11.39% for the two scenarios. Meanwhile, an analysis of computation time is carried out, illuminating the flexibility and controllability of the new strategy.

1. Introduction

Connected and automated vehicles are considered to play an important role in improving traffic efficiency and saving energy [1]. The fickle driving behaviors can easily lead to a series of problems, including traffic congestion, energy consumption, and accident [2–4], but transport systems consisting of intelligent vehicles can make a difference using vehicle-to-everything (V2X) communication and advanced control techniques [5–7].

The development of connected and automated vehicles (CAVs) brings both opportunities and challenges to traffic management. As the bottlenecks in traffic organization, intersection and on-ramp become the research hot spots in the domain [8–10]. Conventionally, the vehicles must adhere to the traffic signals in urban

scenarios, and corresponding studies are proposed to optimize the trajectory of vehicles in this case [1, 11, 12]. Considering the traffic environment composed of CAVs, traffic signals can be eliminated because the information on the road can be fully obtained [13], while the vehicles on the road can be fully controlled. It is possible to implement cooperative control for CAVs through V2X communication [14]. Thus, the design of a cooperative driving strategy through the use of real-time traffic information becomes particularly important. Ann and Colombo [15] pointed out that an effective cooperative driving framework can work in different traffic scenarios such as intersections, merging roadways, and roundabouts. On account of the significance of cooperative driving, the researchers proposed many theoretical

methods to solve the problem for different scenarios. Grand cooperative driving challenges were also organized to promote its development in practice [16].

Some of the existing studies belong to the optimization-based method. Yan et al. [17] proposed a dynamic programming algorithm to evacuate vehicles at the intersection as soon as possible. Zhu and Ukkusuri [18] put forward a linear programming model to dispatch vehicles at autonomous intersections in order to minimize total travel time. Besides, mixed-integer linear programming (MILP) is widely used to obtain solutions [19–21]. However, Li and Wang [13] proposed a framework based on the optimization principle, which utilized a tree search algorithm to achieve the same purpose. All of the listed studies focus on searching for optimal solutions based on different prior hypotheses.

Relevant studies pointed out that the key to solving the problem is determining the right-of-way for CAVs approaching the merging area [22–24]. In other words, the vehicles can be formulated as a passing sequence in the form of arrays, and the performance of the schedule strategy hinges on the way to generate the best passing order among a large number of possible solutions.

In terms of generating passing orders, the existing studies can be classified into two categories. One is the rule-based strategy, which uses some heuristic rules to determine the passing order of vehicles. Dresner and Stone [25, 26] proposed a reservation-based system and assigned right-of-way to vehicles on a first-come-first-served (FCFS) basis. Although the effectiveness of the FCFS method can be proved [26, 27], its rule-based nature always leads to feasible but not optimal solutions. Moreover, the reservation-based strategy cannot outperform traditional signal control in some cases [28, 29]. While the rule-based strategies cannot always perform very well, the other approach to generate passing orders is introduced, called “planning-based strategy” [13]. Meng et al. demonstrated that the planning-based strategy could consistently outperform the FCFS method in intersection scenarios by comprehensive simulations [30]. Actually, the planning-based strategy is a framework that can search for optimal solutions in a huge solution space. The strategy is essentially a traversal problem with intolerable computational complexity. Therefore, consequent studies focused on the reduction in computing time. Xu et al. [31] proposed a grouping-based strategy, which groups CAVs to reduce the count of possible solutions. In their other study, a Monte Carlo tree is built to keep the trade-off between coordination performance and computation time [32]. Meanwhile, Zhang et al. [33] reported a framework that utilized a neural network to surrogate the simulation test process with the intent to reduce computation time. However, the only optimization target they considered is about traffic efficiency indexes such as passing time or total delay, while the value of other targets such as energy consumption or queue length is difficult to acquire. This is caused by the weakness of the trajectory interpretation algorithm in their studies.

Therefore, there is still a lack of a real-time, multi-objective cooperative driving strategy that can be maneuverable and reliable. To this end, we design a multi-objective

discrete evolutionary algorithm (MODEA) to search for (near) optimal passing orders, which combines the non-dominated sorting method [34] and state transition algorithm [35]. A multitask learning model is proposed to be a regressor, which can give feedback of objective values to MODEA. The scenario is simulated by Simulation of Urban MObility (SUMO) [36]. The simulation results indicate that the framework can be applied to different scenarios, performing well even under a high concurrency environment.

The rest of the study is arranged as follows: Section “Problem Statement” gives the general form of cooperative driving problems and traffic scenarios the paper studied. Section “Methodology” presents the framework we proposed, including the MODEA and multitask learning method in detail. Section “Simulation and Analysis” provides the simulation results of a series of experiments. Finally, conclusions are given in Section “Conclusion.”

2. Problem Statement

Highway on-ramps and urban unsignalized intersections are two typical scenarios for cooperative driving (see Figure 1). Rios-Torres and Malikopoulos [37] pointed out that the two mainstream frameworks in the cooperative driving field are centralized coordination and decentralized coordination, respectively, while the method proposed in this study belongs to the former. The centralized frameworks rely on a central controller responsible for computing and sending control commands. The controller has a communication range (CR) that defines the boundary of communication and control. This article denotes the CR as a circle, which is widely adopted in previous studies [30, 33, 38]. Only vehicles within the CR will communicate with the controller and be controlled.

Some followed assumptions are listed to make the analysis and implementation easier:

- (i) Lane-changing behaviors are prohibited in CR for safety consideration.
- (ii) The system has no interference from pedestrians and non-motor vehicles.
- (iii) All CAVs can transmit id, position, speed, and other precise information to the controller spontaneously.
- (iv) The vehicles are homogeneous pure electric CAVs for estimating energy consumption. The energy model can be found in [39].

The general form of the objective function in cooperative driving can be defined as follows:

$$\min_x F(x), \quad (1)$$

where F is the function that represents queue length, energy consumption, or traffic delay, and x is the independent variable that will give rise to the optimization target. In this study, two objects are considered: (a) the minimization of time consumption to evacuate all CAVs in CR and (b) electricity consumption for CAVs in the process of a scheduling scheme.

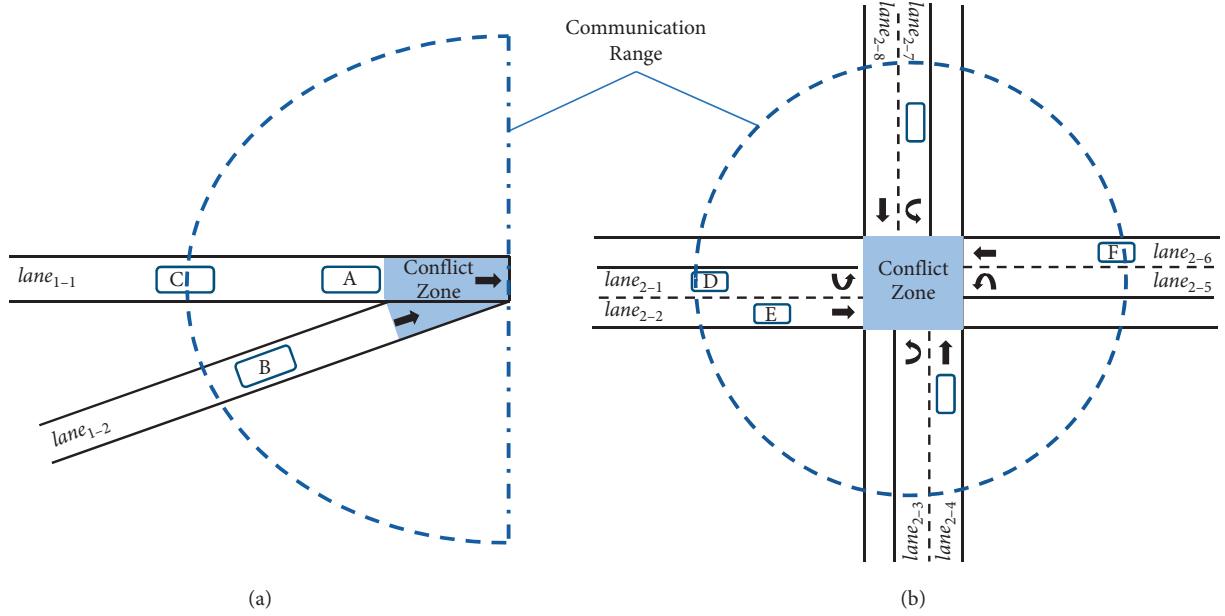


FIGURE 1: Two typical cooperative driving scenarios in general road networks. (a) The on-ramp in highway and (b) the unsignalized intersection in urban road.

The input of the function x denotes a passing sequence, which can be denoted as follows:

$$x = [CAV_1, CAV_2, \dots, CAV_n], \quad (2)$$

where n is the number of vehicles in CR. Let f_1 be the time consumption to evacuate all CAVs in CR, and f_2 be the corresponding electricity consumption, and (1) can be transformed to as follows:

$$\min_x [f_1(x), f_2(x)], \quad (3)$$

Here, $f_1(x)$ can be denoted as follows:

$$f_1(x) = t_e(CAV_n), \quad (4)$$

where t_e represents the time when the vehicle CAV_n exits from CR. $f_2(x)$ can be denoted as follows:

$$f_2(x) = \sum_{i=1}^n \int_0^{t_e(CAV_i)} e(CAV_i), \quad i \in (1, n), \quad (5)$$

where e represents the energy consumption of CAV_i in discrete time, and readers can refer to [39] for the stepwise energy consumption model.

3. Methodology

Figure 2 illustrates the procedure of the framework this study proposed. The framework uses MODEA with non-dominated sorting and multitask neural network to reduce computation time and implement multi-objective optimization. A population-based evolutionary algorithm is used to search solutions in solution space, while the fitness value of every individual can be obtained from a neural network, which plays the role of target regressor. Then, the framework will be introduced in detail.

3.1. Multitask Learning Model. It is found that carrying out learning for tasks jointly can improve the performance compared with conducting them individually [40]. Thus, in this study, a multitask deep learning model is trained to target the evolutionary algorithm's feedback. Therefore, the task of the model is learning for target yield in each traffic state. Here, we consider the time consumption and electricity consumption as the targets defined in equations (4) and (5).

For performing the regression task, the input should be appropriately expressed. As in equation (2), a passing sequence can be denoted as an array including CAV ids. We define the encoding of a single CAV as follows:

$$E_i = (p_i, v_i, a_i, \text{encode}(\text{lane}_i)), \quad (6)$$

where p_i is the position of CAV_i from the beginning of the lane, and v_i is the speed, while a_i represents the acceleration of CAV_i , and p_i, v_i, a_i will be normalized for input into the model.

In addition, $\text{encode}(\text{lane}_i)$ is the encoding of the lane that the driving vehicle belongs to. The encoding method is different according to the different traffic scenarios. For the on-ramp scenario shown in Figure 1(a), the one-hot encoding is applied. However, in the intersection scenario, considering the spatial relationship, we combine with approach direction and driving direction. Figure 3 shows the encoding process that takes the scenarios in Figure 1 as an example. For instance, vehicle D is coming from the west approach, and it will turn left at the intersection, so the encoding of its lane is $(1, 0, 0, 0, 0, 1)$. Finally, a passing sequence can be formulated as the concatenation of encodings of CAVs.

When the vectorized representations of passing sequences are constructed, a neural network model can be built to take the vectors as input. Similar to TextCNN [41], we also use the convolutional neural network (CNN)

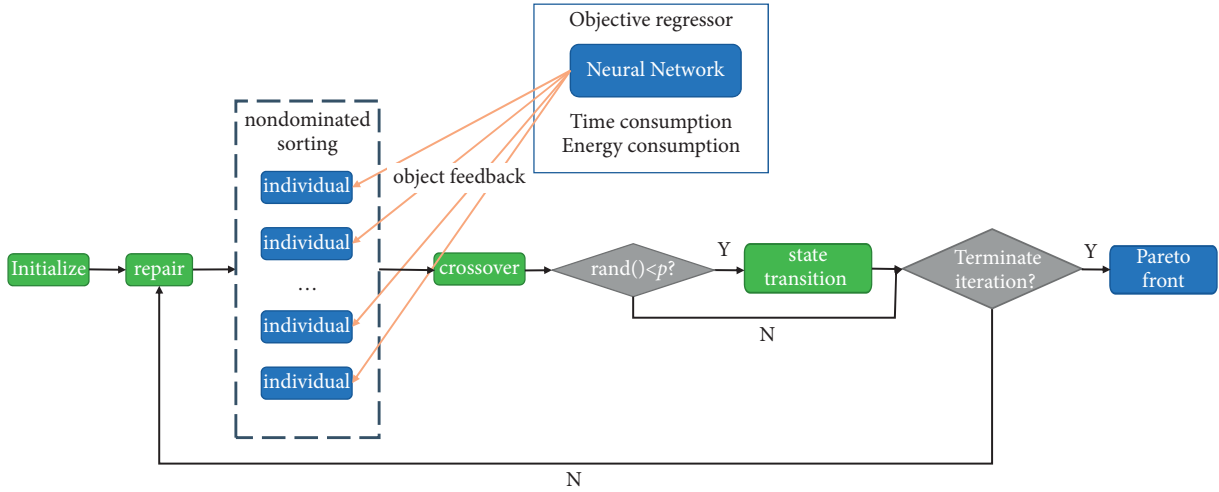


FIGURE 2: Flowchart of the proposed framework.

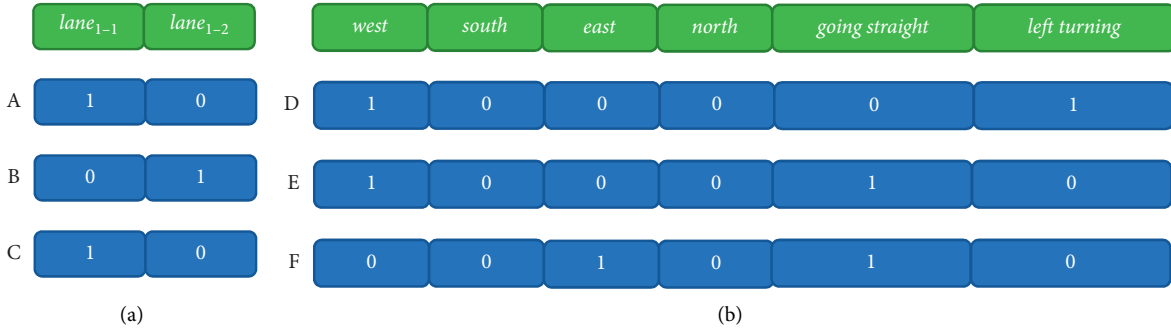


FIGURE 3: Encoding of lane of vehicles, which takes the state in Figure 1 as an example. (a) Example of Figure 1(a) and (b) example of Figure 1(b).

to carry out the learning process, whereas CNN can extract the features from original data automatically [42]. The structure of the CNN-based multitask learning model is shown in Figure 4. The backbone part takes sequence vectors consisting of several CAV encodings as inputs and extracts latent feature representations for them; then, the specific task part takes the feature representations as input and output time consumption and energy consumption of the sequences in a specific traffic scenario. In the backbone part, one-dimensional convolution layers with different scales of kernel size are applied to extract features.

After determining the basic structure of the neural network, the loss function should be specified to train the learning model towards the optimization goals. Here, considering the training process of two targets in two single-task models, the loss functions are considered as mean squared error (MSE), which is as follows:

$$MSE_1 = \frac{1}{n} \sum_{i=1}^n (\hat{f}_1(x_i) - f_1(x_i))^2, \tag{7}$$

$$MSE_2 = \frac{1}{n} \sum_{i=1}^n (\hat{f}_2(x_i) - f_2(x_i))^2,$$

where n is the count of test samples. \hat{f}_1 and \hat{f}_2 are predicting values, while f_1 and f_2 are ground truth. Generally, the loss function in the multitask learning model can be defined as the naive weighted sum of losses, which is as follows:

$$L = \omega_1 MSE_1 + \omega_2 MSE_2, \tag{8}$$

where the loss weights ω_1 and ω_2 are uniform or manually tuned. The performance of the model highly depends on the settings of the weight parameters. Cipolla et al. pointed out that the loss function can be calculated based on maximizing the Gaussian likelihood with homoscedastic uncertainty

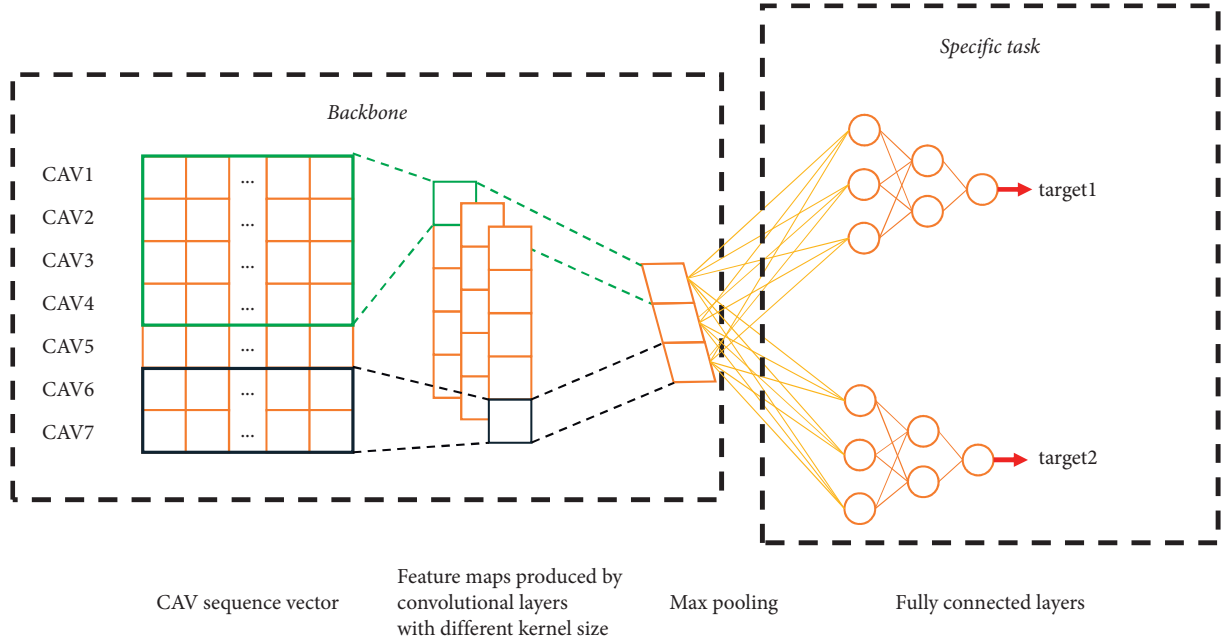


FIGURE 4: Structure of the proposed neural network. The backbone layer is for feature extracting, and the specific task layer is for different regression tasks.

[43]. As a result, let $f^W(x)$ be the outputs of neural network with weights W , and the likelihood as a Gaussian can be defined as follows:

$$p(f_j(x)|f^W(x)) = N(f^W(x), \sigma^2) (j = 1, 2), \quad (9)$$

where σ is a scalar that represents observation noise. Let $\tilde{f}_j(x)$ be the sufficient statistics; then, the multitask likelihood can be derived from the following:

$$\begin{aligned} p(f_1(x), f_2(x)|f^W(x)) &= p(f_1(x)|f^W(x)) \cdot p(f_2(x)|f^W(x)) \\ &= N(f_1(x); f^W(x), \sigma_1^2) N \\ &\quad \cdot (f_2(x); f^W(x), \sigma_2^2). \end{aligned} \quad (10)$$

Taking logarithmic form, the new loss function can be defined as follows:

$$-\log p(f_1(x), f_2(x)|f^W(x)) \propto \frac{1}{2\sigma_1^2} f_1(x) - f^W(x)^2 + \frac{1}{2\sigma_2^2} f_2(x) - f^W(x)^2 + \log \sigma_1 \sigma_2 = \frac{1}{2\sigma_1^2} \text{MSE}_1 + \frac{1}{2\sigma_2^2} \text{MSE}_2 + \log \sigma_1 \sigma_2. \quad (11)$$

Notice that σ_1 and σ_2 are the denominators in equation (11). To avoid division by zero errors, the logarithmic form is used for the actual training process:

$$\varsigma_j = \log \sigma_j^2 (j = 1, 2). \quad (12)$$

Finally, the loss function is given in equation (13), which can be adaptive during the training process.

$$L = \frac{1}{2} (e^{-\varsigma_1} \text{MSE}_1 + e^{-\varsigma_2} \text{MSE}_2 + \varsigma_1 + \varsigma_2). \quad (13)$$

4. Multi-Objective Discrete Evolutionary Algorithm

Generally, the average count of possible passing sequences in cooperative driving grows almost exponentially with the increase in numbers of CAVs in CR [30]. Thus, searching for

the best solution is hard when the number of CAVs is large, so this study proposes a population-based evolutionary algorithm to obtain (near) optimal passing order from this perspective.

In multi-objective optimization problems, the Pareto optimal solution is used to select according to the practical problem [44]. The conception of the Pareto optimal solution set is introduced as below. First, in this minimization problem, solution x^0 Pareto dominates x^1 only if:

$$\begin{aligned} f_j(x^0) &\leq f_j(x^1), \quad \forall j = 1, 2 \\ f_j(x^0) &< f_j(x^1), \quad \exists j = 1, 2. \end{aligned} \quad (14)$$

We use the corresponding symbol to denote the domination relationship:

$$x^0 \succ x^1, \quad (15)$$

which represents that x^0 dominates x^1 . If there is not any solution that dominates x^0 , then x^0 will be called the non-

dominated solution. Accordingly, the Pareto optimal solution set P_s can be defined as the set consisting of all the non-dominated solutions. Therefore, the primary purpose of the algorithm is to search corresponding Pareto optimal solutions. If there is more than one element in the Pareto optimal solution set, two kinds of heuristic strategies can be used:

- (i) Delay-first strategy (DFS): always choose the solution with minimal time consumption from P_s .
- (ii) Energy-first strategy (EFS): always choose the solution with minimal energy consumption from P_s .

The form of the candidate solutions in the algorithm is denoted as equation (2), while the initialization operation is generating n different integers with ranges from 1 to n . The feasible solutions make up a population in the evolutionary algorithm. Considering that lane-changing behavior is prohibited in CR, some solutions will be illegal. For example, in Figure 1(a), the passing order [C, A, B] cannot be accepted as candidate solution because A is supposed to be in front of C. Hence, a repair operation is applied to repair illegal sequence, which is defined as follows:

$$\dot{x} = M_r x, \quad (16)$$

where \dot{x} represents a passing sequence that can be a candidate, and M_r is a matrix that carries out the repair operation. The matrix is constructed according to the order of vehicles on the lanes. For unfeasible sequence [3, 1, 2], which represents "C-A-B" in Figure 1(a), M_r is as follows:

$$M_r = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}. \quad (17)$$

Then, [3, 1, 2] will be transformed to [1, 3, 2], which represents "A-C-B," and it will be legal.

The proposed algorithm uses selection operation, crossover operation, state transition with swap operation, shift operation, and symmetry operation for population evolution. Corresponding operations can be described as follows.

4.1. Selection Operation. Non-dominated sorting technique is used for layering individuals. Algorithm 1 shows the process of non-dominated sorting. In the algorithm, c is the non-dominated level, and X^N is the set of all the non-dominated solutions in P ; fitness represents the virtual value of individuals, which is used for selection operation. Eventually, the roulette wheel method is applied to choose individuals in the population; then, the crossover operation can be carried out. In the roulette wheel method, the selection probability of individual i is defined as follows:

$$\Pr(i) = \frac{c_f - \text{fitness}(i)}{\sum_{j=1}^{c_f-1} j}, \quad (18)$$

where c_f is the value of c after iterations in Algorithm 1.

4.2. Crossover Operation. Tie-breaking crossover is introduced in this study [45]. This operation can prevent two identical orders from appearing in a sequence, and the procedure is indicated in Figure 5. The start positions and length of subsequences are generated randomly, so the results after crossover could be with duplicated items. A crossover map will also be generated, and the crossover map is actually a random order of integers $0, 1, \dots, n-1$. Accordingly, the new sequences after exchange can be transformed by multiplying the length of the sequence and adding the crossover map. Finally, as shown in Figure 5, offspring can be produced by sorting operation according to phase 3.

4.3. State Transition. The state transition procedure is probabilistic in the light of predefined probability value p . In this study, the value of p is set to 0.2 to keep the trade-off between exploration and exploitation. The state transition operations include swap, shift, and symmetry [35]. Swap transformation is used for randomly exchanging subsequences in passing sequences; shift transformation is used for subsequence translation, and symmetry transformation means two subsequences symmetrical about a selected central point exchange their values. These operations can be implemented by several matrixes, which can be denoted as follows:

$$x_{k+1} = M_{\text{symmetry}}(M_{\text{shift}}(M_{\text{swap}}x_k)), \quad (19)$$

where x_{k+1} is a passing sequence after k iterations. M_{symmetry} , M_{shift} , and M_{swap} represent the matrix, which implements symmetry operation, shift operation, and swap operation, respectively. Figure 6 illustrates the three transformations. The length of subsequences is a hyperparameter for swap transformation and shift transformation. The values of these two operations are generated randomly according to the number of CAVs. While for symmetry operation, the length of subsequences and the position of the symmetry center can be generated randomly. Note that the boundary condition will be processed here when the indexes of elements may be out of bounds.

5. Vehicle Control

When a passing order is determined, CAVs can move in the light of the sequence. First of all, the motion of vehicles needs to be constrained by the speed limit and acceleration ability:

$$\begin{aligned} 0 &\leq v_i \leq v_{\max}, \\ d_{\max} &\leq a_i \leq a_{\max}, \end{aligned} \quad (20)$$

where v_{\max} denotes the maximum speed limit on the road, and d_{\max} is the maximum deceleration, while a_{\max} is the maximum acceleration constraint by vehicle dynamics.

The virtual vehicle mapping method is used in the framework to ensure safety [46, 47]. Taking the case in Figure 1(a) as an example, if the passing order is "A-C-B," then C will be mapped into lane₁₋₁. CAV B will then follow

```

Input: Init population  $P$ 
(1) :  $c \leftarrow 1$ 
(2) : while length ( $P$ ) > 0
(3) :    $B \leftarrow X^N$ 
(4) :   for each  $b \in B$ 
(5) :     fitness ( $b$ )  $\leftarrow c$ 
(6) :   end
(7) :    $c \leftarrow c + 1$ 
(8) :   Delete  $B$  from  $P$ 
(9) : end

```

ALGORITHM 1: Non-dominated sorting.

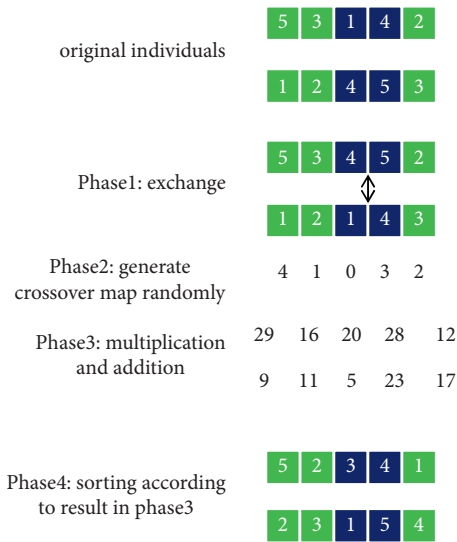


FIGURE 5: Procedure of tie-breaking crossover.

a virtual vehicle mapped by CAV C, which means the mode of motion of CAVs will be divided into two cases: free driving and car following, respectively. The control process of the CAVs in sequences can be given by Algorithm 2. Accordingly, Conflict is a function to judge whether there are potential conflicts between $x[i]$ and $x[j]$. Carfollowing is a function to guide vehicle $x[i]$ to follow vehicle $x[k]$. The equation of Carfollowing can be denoted as follows:

$$\min \int_{t=t_0}^{t_1} |p_k - p_i - \Delta s| dt, \quad (21)$$

where t_0 is the start time and t_1 is the time when $x[i]$ arrives at the conflict zone or stop line. In addition, Δs is the value of the safe gap between two consecutive CAVs. The gap here represents the distance from the front of the following vehicle to the rear of the leading vehicle. If $x[k]$ is a real CAV, the value is set to $\ddot{\Delta}s$, and if $x[k]$ is a virtual vehicle, a correction factor should be added to it, which can be denoted as follows:

$$\Delta s = \ddot{\Delta}s + b \times L_i^M, \quad (22)$$

where b is a bool variable, if $x[k]$ is virtual, the value of b will be 1, and L_M will be the distance for $x[i]$ to cross the conflict zone.

Using Algorithm 2, the first CAV in sequence drives freely, and a CAV with a minimal relative distance with the first CAV in the rest of the sequence is chosen as car following target.

Finally, if a passing sequence is determined, it will not be altered unless the set of CAVs in CR changes.

6. Simulation and Analysis

6.1. Simulation Platform and CNN Training. This study uses the microscopic traffic simulation software SUMO to study the cooperative driving strategy in two traffic scenarios in Figure 1. Under the premise of comprehensive consideration of reality, the simulation settings are given in Table 1. The simulation step is set to 0.2 s for smoother time-continuous control. The radius of CR in the on-ramp scenario is set to 1000 m by considering the communication capability [38]. Meanwhile, we set the radius parameter to 200 m in the urban intersection scenario because the speed of vehicles in this case is slow, while 200 m is enough for vehicle braking.

First of all, more than 50000 records were collected in SUMO for each traffic scenario to serve as the training data. The records include encoding of passing sequence and the combination of two regression targets. We use message-digest algorithm 5 (MD5) to delete duplicated data to ensure the uniqueness of the records. Because the length of CAV encoding in the two scenarios is 5 and 9, respectively, the convolution kernel sizes are set to [2, 3, 4] and [2, 5, 7] to extract different scales of features. The Adam optimizer is used to optimize the weights and biases for the network, and a step decay schedule for learning rate is implemented in the training process for better performance. Accordingly, the rest of the hyperparameters (e.g., batch size, the initial learning rate, and the scales of dense layers) were tuned automatically by applying tree-structured Parzen estimator (TPE), which can search significantly better results compared with random search methods [48].

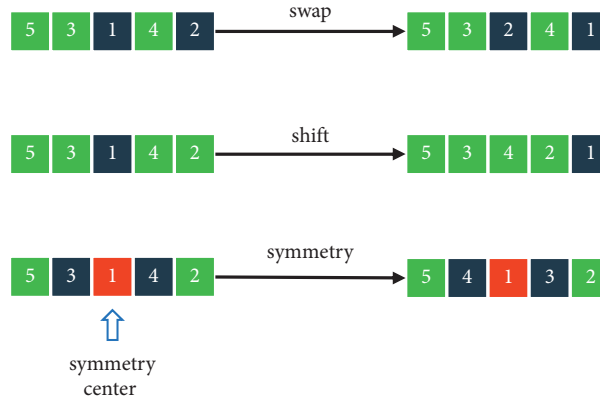


FIGURE 6: Sketch of swap operation, shift operation, and symmetry operation. The length of subsequences in this figure is 1.

```

Input: The passing sequence  $x$ 
(1) : for  $i \in 1: \text{length}(x)$ 
(2) :   if  $i = 1$ 
(3) :      $a_i = (v_i - v_{\max}) \&\& a_{\max}$ 
(4) :   else
(5) :      $k \leftarrow 1, rp_1 = p_1 - p_j - \Delta s$ 
(6) :     for  $j \in 2: i$ 
(7) :        $rp_j = p_i - p_j - \Delta s$ 
(8) :       if  $\text{Conflict}(x[i], x[j]) \&\& rp_j < rp_1$ 
(9) :          $k \leftarrow j$ 
(10) :      end
(11) :     Carfollowing( $x[i], x[k]$ )
(12) :   end
(13) : end
(14) : end

```

ALGORITHM 2: Simple sequence control.

6.2. Simulation Results. To evaluate the proposed strategy comprehensively, we carried out two kinds of simulations based on the pre-trained CNN model. One is a **discrete simulation**, which is used for observing the performance of the framework under different static numbers of vehicles to be scheduled. The other is a **continuous simulation**, which is served to evaluate the framework in different traffic demand levels using the trace data exported from SUMO.

We choose the FCFS strategy as a baseline, whereas it is generally used in the domain. The iteration step and population size in MODEA are set to 30 and 40, respectively. We generate different numbers of CAVs distributed in lanes randomly for the two scenarios, and the results of the discrete simulation are shown in Figure 7. Obviously, the proposed method always has a better performance than the FCFS method. While in the on-ramp scenario, the gap between the two methods becomes more significant with the increase in CAVs. Thus, the capability of global optimization of MODEA can be verified, while the rule-based FCFS method is regarded as weak to get satisfying solutions.

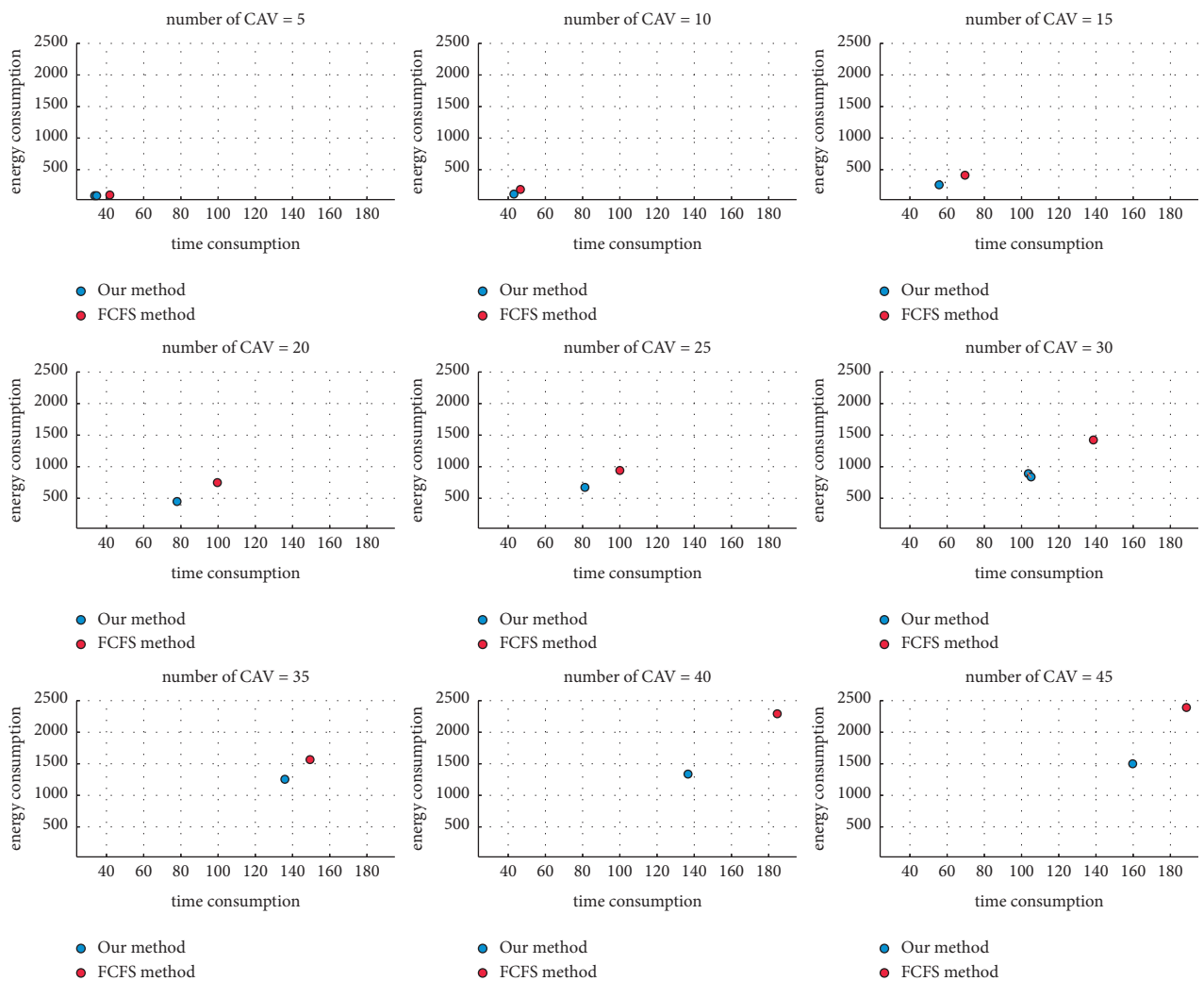
Meanwhile, when there is more than one solution in the Pareto front, the final sequence can be chosen manually according to specific requirements.

As for continuous simulation, different arrival rates of CAVs are deployed for 2000 simulation steps, and the trace data are exported per 4 times steps. The trace datasets include the information of CAVs such as position, speed, and acceleration, and then, we reload these data in SUMO and carry out simulations. In other words, the same trace data are used for result comparison so that the randomness can be eliminated.

All results presented are averaged over 10 independent runs, when the best results are shown in bold in Table 2. According to Table 2, there is no significant difference between DFS and EFS, which may be caused by the regression error of the neural network. However, with the increase in CAV arrival rate, the difference in results between FCFS and the proposed framework gets more remarkable. It demonstrates that the MODEA can optimize the two objectives jointly.

TABLE 1: Parameter configuration in simulation.

Item	Value	Unit
<i>Generic settings</i>		
Maximum acceleration of CAVs	2.6	m/s ²
Maximum deceleration of CAVs	4.5	m/s ²
Length of single lane	1000.0	m
Size of single simulation step	0.2	s
Safe gap Δs	2.5	m
<i>For on-ramp scenario</i>		
Road speed limit	120.0	km/h
Radius of CR	1000.0	m
<i>For intersection scenario</i>		
Road speed limit	60.0	km/h
Radius of CR	200.0	m



(a)

FIGURE 7: Continued.

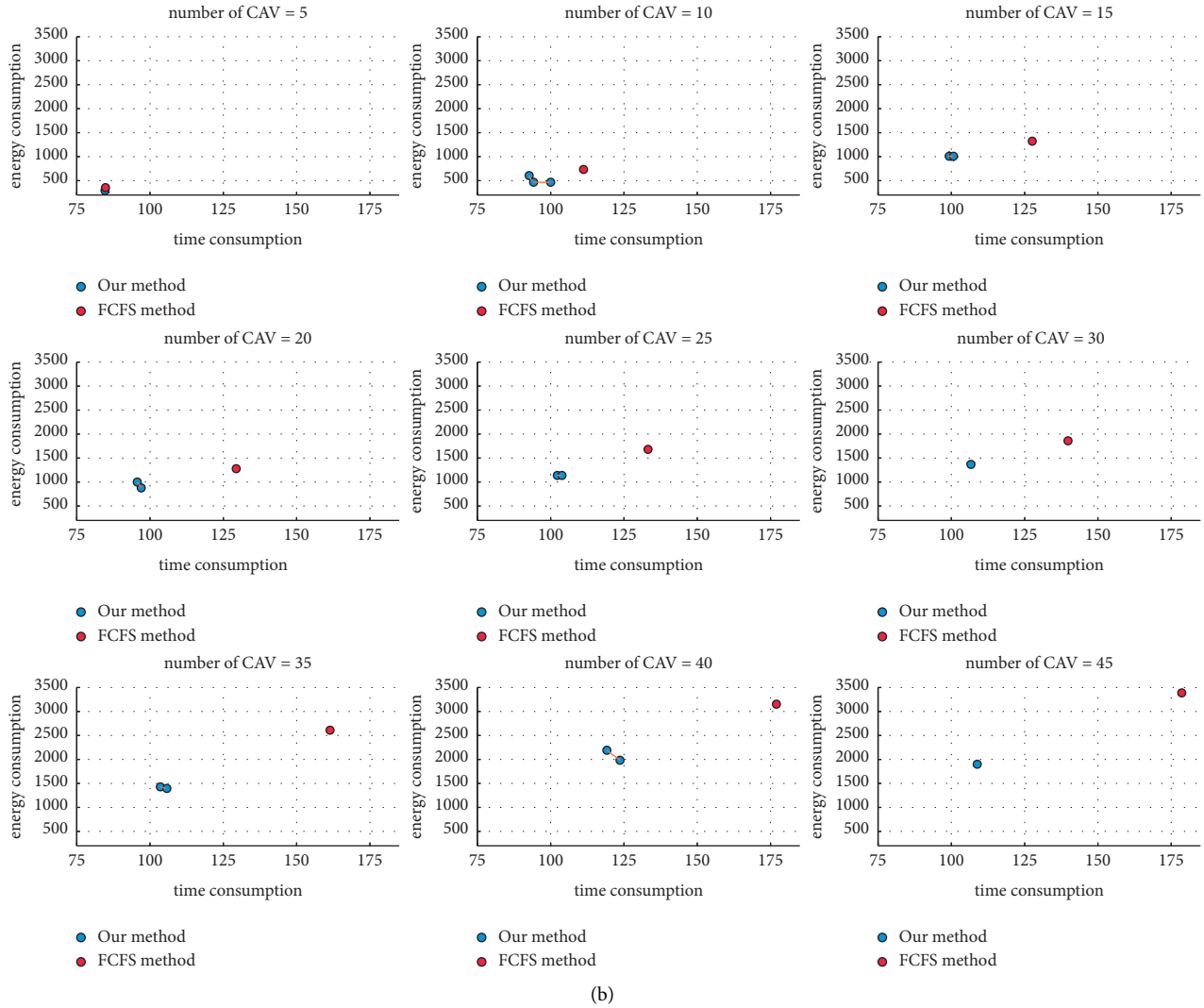


FIGURE 7: Performance of the proposed framework compared with FCFS strategy under different CAV number circumstances. (a) For on-ramp scenario and (b) for intersection scenario.

7. Discussion about Computation Time

In cooperative driving tasks, the computation time of algorithms is vital to ensure safety and efficiency. We focus on the time performance of the proposed framework in this part, and we only consider the on-ramp scenario for evaluating computation time because the time complexity of the algorithm in the two scenarios is equal. All experiments were conducted using Julia programming language on Windows 10 operating system with Intel CORE i7-10750H CPU. Meanwhile, BenchmarkTools.jl package is used to precisely evaluate the computation time performance [49].

As Figure 8 shows, the computation time of the proposed method mainly depends on the population size of MODEA, while the number of CAVs in CR has little effect on the computation complexity, which means that we can control the computation time flexibly by setting the population size of the algorithm manually.

Meanwhile, the influence of computation time on the traffic system should be discussed. First, safety is always the most primary goal to be achieved. The impact of computing time on safety considerations will be reflected in the safe gap Δs . The Δs can be roughly revised with the time consumption t_i :

TABLE 2: Simulation result comparison.

	FCFS		MODEA-DFS		MODEA-EFS		Optimization Rate (%)	
	Time cost (s)	Energy cost (Wh)	Time cost (s)	Energy cost (Wh)	Time cost (s)	Energy cost (Wh)	Time	Energy
<i>On-ramp scenario</i>								
400 veh/(lane · h)	71.5 ± 9.2	302.9 ± 88.0	71.1 ± 9.1	302.2 ± 88.8	71.1 ± 9.2	298.8 ± 86.8	0.56%	1.35%
600 veh/(lane · h)	75.4 ± 7.7	488.8 ± 132.4	74.6 ± 7.9	479.0 ± 129.5	75.0 ± 7.6	475.3 ± 128.0	1.06%	2.76%
800 veh/(lane · h)	80.5 ± 6.8	679.8 ± 172.8	79.0 ± 7.0	658.9 ± 168.7	79.9 ± 7.2	653.0 ± 167.2	1.86%	3.94%
1000 veh/(lane · h)	90.1 ± 7.9	933.5 ± 239.2	82.8 ± 6.6	833.3 ± 220.0	83.2 ± 6.9	831.7 ± 210.5	8.10%	10.91%
1200 veh/(lane · h)	106.6 ± 10.5	1276.3 ± 324.1	89.0 ± 8.4	1070.9 ± 277.6	90.0 ± 8.1	1067.1 ± 274.9	16.51%	16.39%
<i>Intersection scenario</i>								
400 veh/(lane · h)	41.4 ± 12.7	140.2 ± 49.5	41.0 ± 12.9	137.8 ± 48.3	41.7 ± 12.8	140.0 ± 49.4	0.10%	1.71%
600 veh/(lane · h)	58.6 ± 15.6	293.5 ± 150.5	56.7 ± 14.4	287.3 ± 141.3	57.1 ± 14.5	288.9 ± 142.9	3.24%	2.11%
800 veh/(lane · h)	106.5 ± 44.2	951.5 ± 661.3	98.2 ± 37.4	861.9 ± 574.0	98.6 ± 38.5	861.4 ± 572.1	7.79%	9.47%
1000 veh/(lane · h)	141.8 ± 60.9	1468.5 ± 829.0	127.9 ± 35.9	1323.5 ± 639.4	130.4 ± 38.5	1301.3 ± 610.2	9.80%	11.39%
1200 veh/(lane · h)	197.4 ± 44.9	2572.5 ± 812.1	182.3 ± 42.9	2384.1 ± 780.9	186.7 ± 57.8	2352.6 ± 779.8	7.65%	8.55%

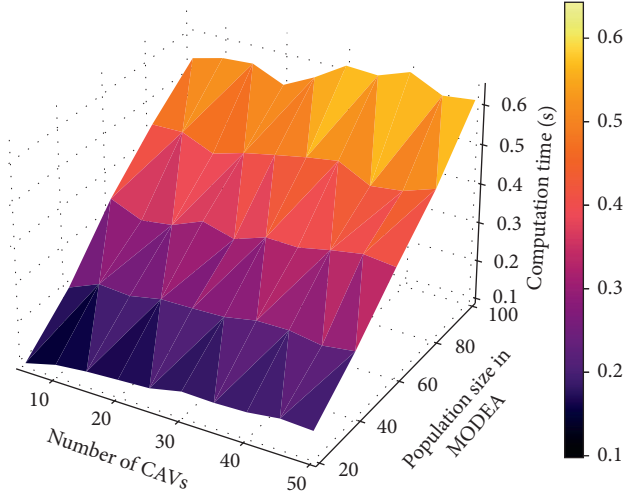


FIGURE 8: Average computation time of the proposed framework based on the different number of CAVs and population size in MODEA.

$$\Delta s' = \ddot{\Delta} s + t_d \times v_{\max}, \quad (23)$$

where v_{\max} is used for ensuring safety under any circumstance, so that Δs will be changed in simulations in terms of equation (22).

Then, we carry out a series of simulations using the same trace data exported from SUMO to compare the performance of the control framework under different computation delays. In the test, the delay caused by

computation varies from 0.1 s to 0.4 s, while DFS is chosen to get solutions. Figures 9(a) and 9(b) show the time consumption and energy consumption under different circumstances. On average, the FCFS rule will outperform the proposed framework in the time consumption aspect when the computation delay reaches 0.3 s, and it will have almost identical performance in the electricity consumption aspect when the computation delay reaches 0.4 s.

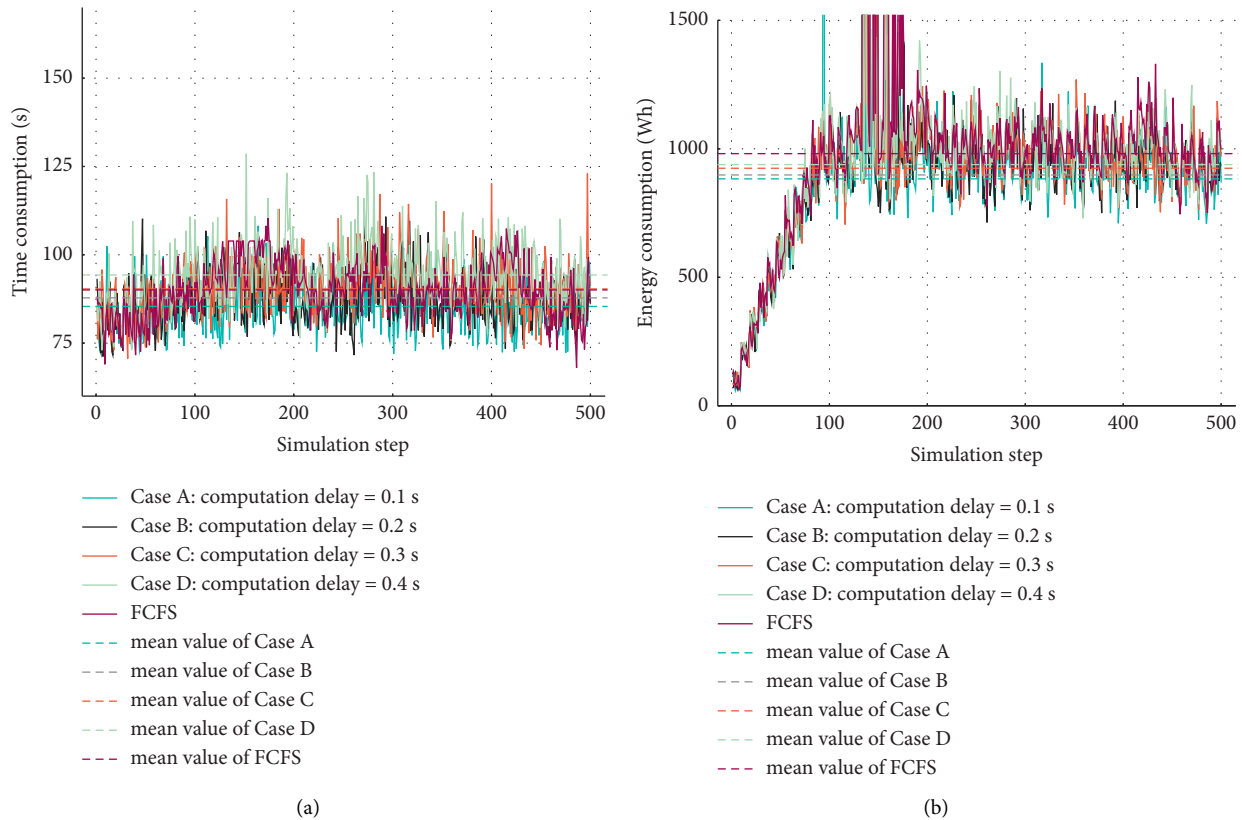


FIGURE 9: Comparison for values of objective function under different computation delays. (a) Time consumption and (b) electricity consumption.

8. Conclusions

Over the last few years, many methods have been put forward in the cooperative driving field, but the controllability of optimization objectives and the efficiency of algorithms are still difficult to deal with. Based on the combination of evolutionary algorithm and machine learning technique, this study proposes an intelligent framework that considers both the delay and the energy consumption of vehicles. An encoding approach of CAVs is implemented, and a passing sequence of CAVs is approximately regarded as a sentence in natural language so that the TextCNN can be applied to extract features. Compared with other frameworks, it has some significant advantages:

- (i) Controllability and flexibility: the optimization objectives and computation time can be adjusted manually, and it can be instrumental under different design requirements.
- (ii) General applicability: similar to FCFS protocol, the framework can be applied in different cooperative driving scenarios such as intersection and on-ramps.

In future research, a more concrete vehicle control method is supposed to be studied for practicability. Moreover, the neural network this study implements can only deal with a finite number of cases because the input length for the network is

fixed. Therefore, the maximum number of CAVs must be assigned, and the zero paddings will be used if the number of CAVs is less than the predefined maximum length. Hence, the form of the neural network and CAV encodings can be further studied for better performance; for example, the encoder-decoder structure can be applied to study the cases of different numbers of CAVs. Finally, the lane-changing behavior of vehicles and pedestrian crossing rules can be considered in the system. However, a more complex but more realistic system will be put before us to study.

Data Availability

The data used to support the findings of this study are produced by simulations.

Conflicts of Interest

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

Acknowledgments

This study was supported in part by the Key R&D Program of Jiangsu Province in China (No. BE2020013), the Opening Project of Key Laboratory of Intelligent Transportation Systems Technologies, Ministry of Communications, P.R.

China (Grant No. 2020-8501), and the R&D Project of China Communications Construction Company (Grant 2019-ZJKJ-ZDZX02-2). The work of the first author was supported in part by the Postgraduate Research & Practice Innovation Program of Jiangsu Province (Grant No. SJCX21_0062).

References

- [1] J. Zhang, S. Dong, Z. Li, B. Ran, R. Li, and H. Wang, "An eco-driving signal control model for divisible electric platoons in cooperative vehicle-infrastructure systems," *IEEE Access*, vol. 7, pp. 83277–83285, 2019.
- [2] A. A. Malikopoulos and J. P. Aguilar, "An optimization framework for driver feedback systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 2, pp. 955–964, 2013.
- [3] Y. Ma, K. Tang, S. Chen, A. J. Khattak, and Y. Pan, "On-line aggressive driving identification based on in-vehicle kinematic parameters under naturalistic driving conditions," *Transportation Research Part C: Emerging Technologies*, vol. 114, pp. 554–571, 2020.
- [4] B. Peng, M. F. Keskin, B. Kulcsár, and H. Wymeersch, "Connected autonomous vehicles for improving mixed traffic efficiency in unsignalized intersections with deep reinforcement learning," *Communications in Transportation Research*, vol. 1, Article ID 100017, 2021.
- [5] C. Dong, H. Wang, Y. Li, W. Wang, and Z. Zhang, "Route control strategies for autonomous vehicles exiting to off-ramps," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 7, pp. 3104–3116, 2020.
- [6] Y. Zheng, B. Ran, X. Qu, J. Zhang, and Y. Lin, "Cooperative lane changing strategies to improve traffic operation and safety nearby freeway off-ramps in a connected and automated vehicles environment," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 11, pp. 4605–4614, 2020.
- [7] J. Larsson, M. F. Keskin, B. Peng, B. Kulcsár, and H. Wymeersch, "Pro-social control of connected automated vehicles in mixed-autonomy multi-lane highway traffic," *Communications in Transportation Research*, vol. 1, Article ID 100019, 2021.
- [8] Y. Han, M. Ramezani, A. Hegyi, Y. Yuan, and S. Hoogendoorn, "Hierarchical ramp metering in freeways: an aggregated modeling and control approach," *Transportation Research Part C: Emerging Technologies*, vol. 110, pp. 1–19, 2020.
- [9] H. Yu, P. Liu, Y. Fan, and G. Zhang, "Developing a decentralized signal control strategy considering link storage capacity," *Transportation Research Part C: Emerging Technologies*, vol. 124, Article ID 102971, 2021.
- [10] Y. Guo, T. Sayed, and M. Essa, "Real-time conflict-based Bayesian Tobit models for safety evaluation of signalized intersections," *Accident Analysis & Prevention*, vol. 144, Article ID 105660, 2020.
- [11] B. Varga, T. Tettamanti, B. Kulcsár, and X. Qu, "Public transport trajectory planning with probabilistic guarantees," *Transportation Research Part B: Methodological*, vol. 139, pp. 81–101, 2020.
- [12] M. Li, X. Wu, X. He, G. Yu, and Y. Wang, "An eco-driving system for electric vehicles with signal control under V2X environment," *Transportation Research Part C: Emerging Technologies*, vol. 93, pp. 335–350, 2018.
- [13] L. Li and F.-Y. Wang, "Cooperative driving at blind crossings using intervehicle communication," *IEEE Transactions on Vehicular Technology*, vol. 55, no. 6, pp. 1712–1724, 2006.
- [14] Z. Wang, Y. Bian, S. E. Shladover, G. Wu, S. E. Li, and M. J. Barth, "A survey on cooperative longitudinal motion control of multiple connected and automated vehicles," *IEEE Intelligent Transportation Systems Magazine*, vol. 12, no. 1, pp. 4–24, 2020.
- [15] H. Ahn and A. Colombo, "Abstraction-based safety verification and control of cooperative vehicles at road intersections," *IEEE Transactions on Automatic Control*, vol. 65, no. 10, pp. 4061–4074, 2020.
- [16] V. Dolk, J. d. Ouden, S. Steeghs et al., "Cooperative automated driving for various traffic scenarios: experimental validation in the GCDC 2016," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 4, pp. 1308–1321, 2018.
- [17] F. Yan, M. Dridi, and A. El Moudni, "Autonomous vehicle sequencing algorithm at isolated intersections," in *Proceedings of the 2009 12th International IEEE Conference on Intelligent Transportation Systems*, pp. 1–6, St. Louis, MO, USA, October 2009.
- [18] F. Zhu and S. V. Ukkusuri, "A linear programming formulation for autonomous intersection control within a dynamic traffic assignment and connected vehicle environment," *Transportation Research Part C: Emerging Technologies*, vol. 55, pp. 363–378, 2015.
- [19] M. W. Levin and D. Rey, "Conflict-point formulation of intersection control for autonomous vehicles," *Transportation Research Part C: Emerging Technologies*, vol. 85, pp. 528–547, 2017.
- [20] E. R. Müller, R. C. Carlson, and W. K. Junior, "Intersection control for automated vehicles with MILP," *IFAC-PapersOnLine*, vol. 49, no. 3, pp. 37–42, 2016.
- [21] C. Yu, W. Sun, H. X. Liu, and X. Yang, "Managing connected and automated vehicles at isolated intersections: from reservation- to optimization-based methods," *Transportation Research Part B: Methodological*, vol. 122, pp. 416–435, 2019.
- [22] S. Ilgin Guler, M. Menendez, and L. Meier, "Using connected vehicle technology to improve the efficiency of intersections," *Transportation Research Part C: Emerging Technologies*, vol. 46, pp. 121–131, 2014.
- [23] J. Wu, A. Abbas-Turki, and A. El Moudni, "Cooperative driving: an ant colony system for autonomous intersection management," *Applied Intelligence*, vol. 37, no. 2, pp. 207–222, 2012.
- [24] H. Xu, Y. Zhang, C. G. Cassandras, L. Li, and S. Feng, "A bi-level cooperative driving strategy allowing lane changes," *Transportation Research Part C: Emerging Technologies*, vol. 120, Article ID 102773, 2020.
- [25] K. Dresner and P. Stone, "Multiagent traffic management: a reservation-based intersection control mechanism," in *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems*, vol. 2, pp. 530–537, New York, NY, USA, July 2004.
- [26] K. Dresner and P. Stone, "A multiagent approach to autonomous intersection management," *Journal of Artificial Intelligence Research*, vol. 31, pp. 591–656, 2008.
- [27] D. Fajardo, T.-C. Au, S. T. Waller, P. Stone, and D. Yang, "Automated intersection control," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2259, no. 1, pp. 223–232, 2011.
- [28] M. W. Levin, S. D. Boyles, and R. Patel, "Paradoxes of reservation-based intersection controls in traffic networks,"

- Transportation Research Part A: Policy and Practice*, vol. 90, pp. 14–25, 2016.
- [29] E. Lukose, M. W. Levin, and S. D. Boyles, “Incorporating insights from signal optimization into reservation-based intersection controls,” *Journal of Intelligent Transportation Systems*, vol. 23, no. 3, pp. 250–264, 2019.
- [30] Y. Meng, L. Li, F.-Y. Wang, K. Li, and Z. Li, “Analysis of cooperative driving strategies for nonsignalized intersections,” *IEEE Transactions on Vehicular Technology*, vol. 67, no. 4, pp. 2900–2911, 2017.
- [31] H. Xu, S. Feng, Y. Zhang, and L. Li, “A grouping-based cooperative driving strategy for CAVs merging problems,” *IEEE Transactions on Vehicular Technology*, vol. 68, no. 6, pp. 6125–6136, 2019.
- [32] H. Xu, Y. Zhang, L. Li, and W. Li, “Cooperative driving at unsignalized intersections using tree search,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 11, 2019.
- [33] J. Zhang, X. Jiang, Z. Liu, L. Zheng, and B. Ran, “A study on autonomous intersection management: planning-based strategy improved by convolutional neural network,” *KSCE Journal of Civil Engineering*, vol. 25, no. 10, pp. 3995–4004, 2021.
- [34] N. Srinivas and K. Deb, “Multiobjective optimization using nondominated sorting in genetic algorithms,” *Evolutionary Computation*, vol. 2, no. 3, pp. 221–248, 1994.
- [35] X. Zhou, D. Y. Gao, C. Yang, and W. Gui, “Discrete state transition algorithm for unconstrained integer optimization problems,” *Neurocomputing*, vol. 173, pp. 864–874, 2016.
- [36] P. A. Lopez, E. Wiessner, M. Behrisch et al., “Microscopic traffic simulation using SUMO,” in *Proceedings of the 2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pp. 2575–2582, Maui, HI, USA, November 2018.
- [37] J. Rios-Torres and A. A. Malikopoulos, “A survey on the coordination of connected and automated vehicles at intersections and merging at highway on-ramps,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 5, pp. 1066–1077, 2017.
- [38] Z. Li, M. Pourmehr, L. Elefteriadou, and S. Ranka, “Intersection control optimization for automated vehicles using genetic algorithm,” *Journal of Transportation Engineering, Part A: Systems*, vol. 144, no. 12, Article ID 04018074, 2018.
- [39] T. Kurczveil, P. Á. López, and E. Schnieder, “Implementation of an energy model and a charging infrastructure in SUMO,” in *Simulation of Urban MObility User Conference*, pp. 33–43, Springer, Berlin, Heidelberg, 2013.
- [40] K. Zhang, L. Zheng, Z. Liu, and N. Jia, “A deep learning based multitask model for network-wide traffic speed prediction,” *Neurocomputing*, vol. 396, pp. 438–450, 2020.
- [41] Y. Kim, “Convolutional Neural Networks for Sentence Classification,” arXiv preprint arXiv:1408.5882, 2014.
- [42] Y. Liu, C. Lyu, X. Liu, and Z. Liu, “Automatic feature engineering for bus passenger flow prediction based on modular convolutional neural network,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 4, pp. 2349–2358, 2021.
- [43] R. Cipolla, Y. Gal, and A. Kendall, “Multi-task learning using uncertainty to weigh losses for scene geometry and semantics,” *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7482–7491, Salt Lake City, UT, USA, June 2018.
- [44] J. Mou, “Intersection traffic control based on multi-objective optimization,” *IEEE Access*, vol. 8, pp. 61615–61620, 2020.
- [45] P. W. Poon and J. N. Carter, “Genetic algorithm crossover operators for ordering applications,” *Computers & Operations Research*, vol. 22, no. 1, pp. 135–147, 1995.
- [46] E. G. Debada and D. Gillet, “Virtual vehicle-based cooperative maneuver planning for connected automated vehicles at single-lane roundabouts,” *IEEE Intelligent Transportation Systems Magazine*, vol. 10, no. 4, pp. 35–46, 2018.
- [47] A. Uno, T. Sakaguchi, and S. Tsugawa, “A merging control algorithm based on inter-vehicle communication,” in *Proceedings of the 1999 IEEE/IEEE/JSAI International Conference on Intelligent Transportation Systems (Cat. No.99TH8383)*, vol. 5, pp. 783–787, Tokyo, Japan, October 1999.
- [48] J. S. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, “Algorithms for hyper-parameter optimization,” in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 2546–2554, Granada, Spain, December 2011.
- [49] J. Chen and J. Revels, “Robust benchmarking in noisy environments,” arXiv preprint arXiv:1608.04295, 2016.