Adaptive Intelligent Sensing Control Method for Traffic Lights under Real-Time Vehicle Conditions Based on Logic Rules

Yang Liu[1], Jiaojiao He[1], Jingwei Chen[2,3*], Bo Mi[1], Ling Zhao[1], Xinyu Liu[1], and Linhan Yang[1]
[1] School of Information Science and Engineering, Chongqing Jiaotong University, Chongqing 400074, China
[2] Chongqing Key Laboratory of Secure Computing for Biology, Chongqing Institute of Green and Intelligent Technology, Chinese Academy of Sciences, Chongqing 400714, China
[3] Chongqing College, University of Chinese Academy of Science, Chongqing 400714, China
[e-mail: liuyang13@cqjtu.edu.cn, hejiaojiao@mails.cqjtu.edu.cn, chenjingwei@cigit.ac.cn, mi_bo@163.com, zhao.ling@163.com, xinyuliu@mails.cqjtu.edu.cn, linhanyang@mails.cqjtu.edu.cn]
*Corresponding author: Jingwei Chen

Received April 1, 2024; revised May 29, 2024; revised July 26, 2024; revised September 6, 2024; accepted December 1, 2024; published December 31, 2024

Abstract

Compared with traditional adaptive algorithms, the complex and changing operating environment of intelligent networks demands high real-time accuracy in data transmission, necessitating an accurate and adaptive traffic light control strategy. Machine learning (ML) techniques can predict traffic conditions based on historical data and real-time information. However, some scholars have mentioned that ML techniques are still deficient in real-time response and in coping with random traffic accidents. Traditional Reinforcement Learning (RL) requires repeated trial and error operations. Applying traditional RL techniques to the optimal control of traffic lights may lead to more serious traffic congestion in some cases. Therefore, this paper combines perceptual control with real-time adaptive control methods to provide a rule-based reasoning method for adaptive intelligent perception and precise control of traffic signals under real-time smart grid-connected hybrid vehicle conditions. IIoT (Industrial Internet of Things) devices are utilized to monitor the parking queue length, pedestrian flow, vehicle flow, and traffic flow in each lane in real time to dynamically adjust the green light duration. By adjusting the light priority according to real-time vehicle and road conditions, this method solves the problem of wasting green lights when random accidents occur in a certain lane, optimizes traffic light settings, achieves real-time precise control of lane flow, and improves the adaptability and precision of traffic lights.

Keywords: Adaptive, intelligent sensing, rule-based reasoning, real-time control, IIoT (industrial internet of things).

http://doi.org/10.3837/tiis.2024.12.003 **ISSN : 1976-7277**

1. Introduction

With the increasing maturity of artificial intelligence, wireless communication technology, sensing technology, image analysis and processing technology[1], vehicle networking technology[2], and unmanned technology[3]. Intelligent technologies have been integrated into everyday life, scientific education, military deployment, and other civil and military fields[4]. In the context of accelerating urbanization and the continuous application of unmanned vehicles, the problem of traffic congestion has become increasingly serious. RL-based methods for traffic signal control usually require a large number of update iterations and repeated trial-and-error in order for the traditional RL model to produce a more desirable result in a simulated environment. X. Yang[5], H. Wei[6], Y. Liu[7], M. Noaeen[8], Y. He[9] have studied the application of RL in traffic signal control. ML-based technology is based on historical data and real-time information to predict traffic conditions, there are still challenges in the accuracy and reliability of ML models for real-time and random events. Signal control at individual intersections mainly includes timing control, perception control, and real-time adaptive control[9-11]. This paper combines perception control and real-time adaptive control methods, utilizing traffic parameters determined by vehicle detectors to optimize signal control schemes in real-time. The objective is to maximize the benefits of control at individual intersections to adapt to the stochastic variations in traffic demand.

Compared to traditional adaptive algorithms for traffic signal timing, the real-time intelligent connected mixed traffic environment is complex and constantly changing, with a massive number of terminal sensor devices. Additionally, the intelligent connected vehicle system demands high real-time data transmission and precision. Although current traffic signal control methods based on historical data have achieved good results in vehicle travel prediction, traffic diversion, and route planning[12,13], they are insufficient in real-time performance and handling random traffic incidents. On the other hand, while real-time optimization control methods for intelligent traffic signal lights [8,14] exhibit certain real-time capabilities in signal control and can intelligently adjust signal lights based on predicted traffic flow[15], they are not precise enough. It is not possible to control the corresponding lane timing at large intersections with high traffic volumes, nor is it possible to effectively respond to random traffic accidents. Therefore, in response to the frequent occurrence of random traffic incidents in real-time complex traffic conditions, there is an urgent need to address the issue of precise traffic signal control strategies. Based on this background, the innovations and contributions of this paper are as follows:

(1) This paper adopts automatic inference rules with a strict mathematical logic basis, which is more precise and rigorous than the rule-based inference method based on fuzzy control[16]. In the latter, precise numerical inputs are converted to membership values of fuzzy sets during the fuzzification process, and this conversion leads to loss of precision.

(2) The method proposed in this paper performs real-time priority setting for traffic signal control system. It analyzes the current road traffic condition through real-time monitoring system and IoT data. In addition, it adjusts the signal duration setting and the corresponding lane indicator conversion in a timely manner. This method is conducive to improving road capacity and effectively solving the problem of green light waste.

(3) In the last part of this paper, based on a simulation model of a large intersection, the method of this paper is compared with other signal control methods, and the experimental results based on the evaluation index system show that our proposed method is more effective in reducing the average delay of the intersection and improving the traffic efficiency in the case of large traffic flow and random accidents.

2. Fundamentals

This paper proposes an adaptive and intelligent perception-based control method for real-time intelligent connected mixed traffic. It discusses the applications and challenges of RL and ML in intelligent traffic control. The section covers logical rule inference, priority queues, updates, and the selection of optimal coordinated sets of green lights in the method.

2.1 Background

With the rapid development of AI technology, there is a strong push to enhance the intelligence of autonomous driving systems[17]. ML, RL, and other AI algorithms have transitioned from theory to industrial application. ML is applied in traffic management to predict and plan road traffic, deploy resources, and optimize signal control through data analysis and modeling. RL has a powerful learning framework that adapts strategies based on the environment to maximize expected benefits[6,18].

The use of ML and RL in traffic light control is a growing trend, with promising results in recent studies. However, existing research has not validated these methods using real-world traffic data and has primarily focused on rewards without analyzing the underlying policies. In addition, challenges remain in the accuracy and reliability of ML models due to factors like data quality and timeliness[19-20]. RL learns through trial and error[21], which can be risky in real-world applications like traffic signal malfunctions[22]. In addition, the more congested the traffic is and the more stochastic it is, the harder it is for the RL method to learn the optimal policy. To meet the requirements of a safe and efficient control system, this paper adopts automatic inference rules based on mathematical logic for real-time priority setting[23]. It analyzes current traffic conditions using real-time monitoring and sensor data, adjusting signal duration and lane indicators accordingly. This approach effectively enhances road capacity and improves the travel experience for pedestrians and drivers[24].

2.2 Logical Reasoning Rule Base

Based on traffic regulations, road topology, and constraints related to traffic flow control, specific rules are formulated for each signal group within the area. These rules generate an inference rule library:

$$S = \{s_i : G(l_i) \to \wedge_{\mathrm{j}} R(l_j) \mid l_i, l_j \in L; 0 \le i \le (n-1)\}$$

S denotes the rule inference library and s_i denotes the ith rule in the library. The rule s_i corresponds to the condition $G(l_i)$, which indicates that signal group l_i is green, and $R(l_i)$ represents the condition that signal group l_j is red. The expression "$\to$ " denotes that when signal group l_i is green, signal group l_j must be red, where $i, j \in [0, n\text{-}1]$ are integers, and $\wedge$ represents the logical AND operator. The specific rule schematic of s_i is schematized in **Fig. 1**.

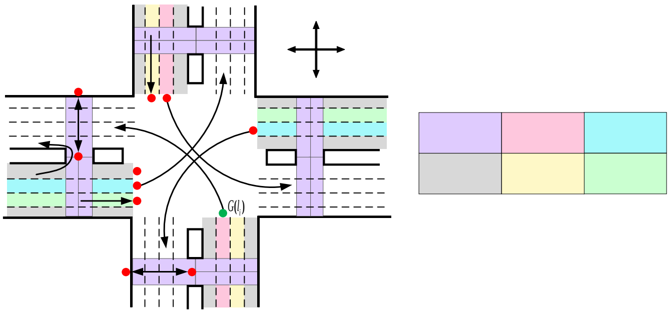


Fig. 1. Schematic diagram of rules

In **Fig. 1**, the solid line from the south exit roadway to the west entrance roadway represents the route of green light l_i , with the starting point marked by a green dot. The rest of the lines in the diagram have red dots at their endpoints, which indicate all the routes with red lights corresponding to that rule. This rule states that when l_i is green, while ensuring safe and efficient passage, l_0,l_{i-1},l_{i+1},l_j, and l_{j+1}should all display red lights. During this time, these segments are not allowed for passage.

2.3 Priority Queue

Based on the data from road monitoring, radar sensors, and feedback from intelligent connected vehicles within the area, road condition analysis is conducted. Following the principle that the longer the queue length of vehicles at each intersection lane, the greater the number of pedestrians waiting at pedestrian crossings, and the lighter the congestion of vehicles on the road ahead, the higher the priority, each signal light element is evaluated in terms of priority. The priority queue for the current signal light group is then initialized.

Road condition analysis is conducted in the current cycle using regional road monitoring, radar sensors, and intelligent networked vehicle feedback data. Deep learning (DL) methods are employed to process the images from road monitoring, vehicle speeds from radar sensor feedback, and vehicle speeds along with driving routes from intelligent networked vehicles. The DL methods used can include frame differencing algorithms or pyramid Lucas-Kanade optical flow algorithms, among others.

The formula for calculating the priority weight of any signal l_i is:

$$W_i = P_i \cdot W_p + v_i \cdot W_v + Len_i \cdot W_{Len} + C_i \cdot W_c + J_i \cdot W_J \,. \tag{1}$$

W_i——Priority of signal element l_i.
P_i——Number of pedestrians at sidewalk intersections corresponding to l_i.
v_i——Average speed of vehicles in the lane corresponding to l_i.
Len_i——Length of lane parking holding corresponding to l_i.
J_i——Lane congestion for the lane corresponding to l_i (0/1).
C_i——Whether a green-to-red light event occurred for l_i (0/1).

In the equations, when there are no waiting pedestrians, or the road ahead is blocked and not passable, or there are no vehicles passing through the section, the values of P_i, v_i and Len_i are all 0. $C_i=1$ indicates that a green-to-red light event occurred previously, while $C_i=0$ indicates no such event occurred. When the average speed $v_i=0$, $J_i=1$ represents congestion, and $J_i=0$ represents no congestion. The initial values are $C_i=0$ and $J_i=0$. W_p, W_v, W_{Len}, W_c, and W_J are predetermined weight parameters corresponding to pedestrian count, average lane speed, lane parking holding length, event, and congestion, respectively. W_p, W_v, and W_{Len} are set as positive values, while W_c is set as a negative value, and W_J is set as negative infinity.

In real-world scenarios, road cameras are often used to capture real-time footage of the road. It is necessary to recognize pedestrians and vehicles in the footage in order to calculate P_i, v_i and Len_i values:

(1) Number of waiting pedestrians (P_i) at sidewalk[25] intersections corresponding to signal group l_i: The number of waiting pedestrians on a sidewalk refers to the total count of pedestrians waiting to cross the street during a specific time period[26], which is shown in Equation (2):

$$P_i = SP_i \times t_p \tag{2}$$

SP_i——Number of pedestrians passing through the sidewalk in a unit of time.

t_p——Time period of observation, measured in seconds (s).

(2) Average speed of vehicles (v_i) in the lane ahead of the signal: The simplest and most intuitive indicator of traffic congestion is the speed of vehicles[27]. Any type of road is considered congested when the speed is below 10 km/h. Based on this, this paper considers the average vehicle speed as a quantitative indicator of traffic congestion, $v_i=\bar{v}$.The calculation method for the average travel speed is shown in the following equation:

$$v_i = \bar{v} = \frac{nL}{\sum_{i=1}^{n} t_i} \tag{3}$$

L——Length of the road segment, excluding intersections (km).

t_i——Time it takes for vehicle i to pass through the road segment,(h).

n——Number of vehicles measured.

(3) Length of the queue of vehicles (Len_i) in the lane at the intersection: In traffic management, the queue length refers to the length of the vehicle queue formed when vehicles stop and wait at a traffic signal or lane during a red light. It represents the number of vehicles waiting to pass in a lane[28]. In this paper, the following calculation method is used:

$$Len_i = d_s \times N_v \tag{4}$$

d_s——Safe distance between adjacent vehicles in the queue, (km).

N_v——Number of vehicles in the queue waiting to pass.

Next, the signal elements are arranged in descending order based on their priority values to obtain the initial priority queue $Priority = \{l_0, l_1, \ldots\ldots, l_{n-1}\}$, where l_i, $i=0,\ldots, n-1$ represents different signal groups.

In order to improve the accuracy of the evaluation results as well as to fully reflect the evaluation needs of the actual operation of intersections, an intersection access efficiency evaluation index system consisting of passing time, average delay, and stopping rate is established[29-33]. Typically, lower result values for these indicators indicate more efficient access. The following definitions were consulted from the VISSIM professional manual[35]:

(1) Intersection Passing time. The formula given in the VISSIM manual is to use the total traveling time t_e, plus the total delay time t_s, to get the passing time T through the intersection (based on the same section length measured on each road segment for data collection). This is shown in Equation (5) below.

$$T = t_e + t_s \tag{5}$$

(2) The average intersection delay is calculated as the time from when a vehicle enters the intersection to when it exits the intersection. The average intersection delay d is equal to the total delay time t_s, divided by the sum of the number of vehicles in the network, V_e, and the number of vehicles that have arrived, V_s. The specific Equation is (6):

$$d = t_s / (V_e + V_s) \tag{6}$$

(3) Intersection Stopping Rate. The stopping rate is the total number of vehicles stopping after entering the queue as a percentage of the total flow. In this paper, the stopping rate is defined as the percentage of the total number of stops q and the effective sample vehicles traveling within the intersection after the vehicles enter the queuing state. The effective sample vehicles here are equal to the number of vehicles in the road network V_e plus the number of vehicles that have arrived V_s, as shown in Equation (7):

$$Q = q / \left(V_e + V_s\right) \tag{7}$$

2.4 Acquisition of coordinated set elements

Based on real-time traffic monitoring and sensor data analysis of the current conditions in the lane ahead, the capacity and saturation of the lane are determined. The initial green light duration for each signal group is initialized as GreenT = $\{t_0, t_1, ..., t_{n-1}\}$. For safety reasons, the green light duration is generally within the range of [30s,120s]. If the lane ahead is congested and not passable,t_i=0. A two-dimensional array, PriorityGreen[n][2], is used to store the priority element information and the corresponding green light duration for each signal in the priority queue, PriorityGreen[i][0]=l_i, PriorityGreen[i][1]=t_i. Considering the current road condition monitoring, elements representing lanes or sidewalks are selected one by one from the initial *Priority* queue for green lights. Each element is evaluated against the mutually exclusive relationships in the inference rule library S. If the requirements are not met, the next element in the queue is selected. The resulting groups of green, yellow, and red lights that can be activated simultaneously in this round are denoted as $G_i=\{l_k, l_j, ..., l_m\}$,and the number of elements in G_i is recorded as num_i. A two-dimensional array CGreen[n][2] is used to store the information of the G_i groups in the current round, where CGreen[n][0]=$\{G_i\}$ and CGreen[n][1] = $\{num_i\}$.The coordination set CGreen is continuously updated, prioritizing the principle of allowing as many vehicles as possible from different directions to pass. The optimal coordinated green light set for the current round, CGreen1[i][j], is selected. When updating the G_i set it is important to avoid that the element to the right of " $\rightarrow$ " in the added rule contradicts an already existing green light, and let $SR_i = s_i \rightarrow \wedge_j R\left(l_j\right)$, s_i denotes the rule whose " $\rightarrow$ "left element is l_i, and SR_i denotes all the red-lighted elements to the right of that rule. The expressions for updating the red and green lights are as follows:

$$G_n = \bigcup_{i=0}^{n-1} \begin{cases} G_0, \left(i = 0 \wedge t_i = 0\right) \\ G_0, \left(i = 0 \wedge t_i \neq 0 \wedge l_i \in R_0\right) \\ G_0 \cup \left\{l_i\right\}, \left(i = 0 \wedge t_i \neq 0 \wedge l_i \notin R_0\right) \\ G_{i-1}, \left(i > 0 \wedge l_i \in R_{i-1}\right) \\ G_{i-1}, \left(i > 0 \wedge t_i = 0\right) \\ G_{i-1}, \left(i > 0 \wedge \exists r \in SR_i, r \in G_{i-1}\right) \\ G_{i-1} \cup \left\{l_i\right\}, \begin{pmatrix} i > 0 \wedge t_i \neq 0 \wedge l_i \notin R_{i-1} \\ \wedge \forall r, r \in SR_i, r \notin G_{i-1} \end{pmatrix} \end{cases} \tag{8}$$

$$Red_n = \bigcup_{i=0}^{n-1} \begin{cases} R_0, \left(i = 0 \wedge t_i = 0\right) \\ R_0, \left(i = 0 \wedge t_i \neq 0 \wedge l_i \in R_0\right) \\ R_0 \cup \left\{SR_i\right\}, \left(i = 0 \wedge t_i \neq 0 \wedge l_i \notin R_0\right) \\ R_{i-1}, \left(i > 0 \wedge t_i = 0\right) \\ R_{i-1}, \left(i > 0 \wedge \exists r \in SR_i, r \in G_{i-1}\right) \\ R_{i-1} \cup \left\{SR_i\right\}, \begin{pmatrix} i > 0 \wedge t_i \neq 0 \wedge l_i \notin R_{i-1} \\ \wedge \forall r, r \in SR_i, r \notin G_{i-1} \end{pmatrix} \end{cases} \tag{9}$$

$$\text{CGreen}[\,][0]=\{G_0,G_1,\ldots,G_{n-1}\} \tag{10}$$

In the above expressions, l_i = PriorityGreen[i][0], where n is the total number of traffic lights. G_n and R_n are initialized as empty sets. Move the first element of the initial $Priority$ queue to the end, resulting in $Priority_j = \{l_0, l_1, \ldots, l_{n-1}\}$. Repeat this operation until l_0 becomes the first element of the priority queue again:

$$Priority_{j+1} = \{l_i, \ldots, l_{n-1}, l_0, \ldots, l_{i-1} | \forall i \in \{0, \ldots, n-1\}, Priority_j = \{l_{i-1}, \ldots, l_{n-1}, l_0, \ldots, l_{i-2}\}\} \tag{11}$$

2.5 Selection of the optimal coordination set

Based on the obtained coordinated green light set CGreen, assuming the number of distinct light groups in CGreen is denoted as n. This paper provides an algorithm suitable for selecting the optimal coordinated green light set CGreen1. The algorithm takes into account the number and characteristics of elements in the green light set G, as well as data information such as pedestrian and vehicle flow.

The algorithm formula is as follows:

$$WG_i = PN_i \cdot W_{PN} + vN_i \cdot W_{vN} + PR_i \cdot W_{PR} \tag{12}$$

WG_i ——Priority of the coordinated green light set.

PN_i ——Number of signal light groups corresponding to pedestrian lanes in G.

vN_i ——Number of traffic lanes for vehicle passage in G.

PR_i ——Whether G contains a queue head element in the current priority (0/1).

In the formulas, if the first signal light group in the current cycle's priority queue is included, $PR_i = 1$; otherwise, $PR_i = 0$. The weights parameters for PN_i, vN_i, and PR_i are denoted as W_{PN}, W_{vN}, and W_{PR}, respectively, which are predetermined values.

The weight parameters W_p, W_v, W_{Len}, W_c, W_J, W_{PN}, W_{vN}, and W_{PR} mentioned in Equation (1) and (6) can be obtained through training using manually labeled historical data. The allocation of weights depends on the specific situation and the optimization objective, in this paper, the objective is to reduce delay and improve traffic efficiency. For example, in (1) because congestion has an important impact on traffic smoothness and efficiency, so if there is congestion ahead, the signal should not be lit as much as possible, that is, the priority value should be the lowest, so J_i corresponding to the weights $W_J = -\infty$. In (6) according to the order of the elements of the priority queue, the higher the priority the more elements should be in the green light set and play a decisive role, so the weight value W_{PR} should be the highest.

Using (6) and based on the data information for each intersection, the priority WG_i, J_i for each set is calculated. The index m is determined as the set with the highest priority value. After calculating the priority for each set, the set with the highest priority is selected as the optimal coordinated green light set CGreen1.

$$WG_m = \max(WG_0, WG_l, \ldots\ldots, WG_{n-l}) \tag{13}$$

$$m = index(\max(WG_i)), (i = 0, l, \ldots\ldots, n-l) \tag{14}$$

$$|\text{CGreen1}[i][j]| = |\text{Green}_\text{m}| \tag{15}$$

The light groups in the CGreen1 set are illuminated with green lights for a duration of GreenT, as predetermined. The remaining light groups are displayed as red lights. The light groups currently displaying green, yellow, and red lights are moved to the end of the priority queue $Priority$ in order, and the priority queue $Priority$ is updated.

2.6 Adjustment of the optimal coordination set

Check if there are green, yellow, and red light groups in the detection timer that are approaching the upper limit. If any green light element in CGreen1 reaches the upper limit within $15s$, it is added to the EL light group, and preparations for the next round begin. If there are no instances of reaching the upper limit within $15s$, the monitoring detector is used to observe if there is congestion in the lanes ahead while the green, yellow, and red light groups are displaying green lights. If there is no congestion, monitoring continues. If congestion is detected, the green light l_i of that lane, where $l_i \in CGreen1$, is added to the EL set, transitioning to flashing yellow lights. Additionally, based on real-time monitoring data from road condition sensors, the current capacity and saturation level of the lane section ahead are analyzed. The new durations for each group of green, yellow, and red lights under the current conditions are updated as GreenT = $\{t_1', t_2', ..., t_n'\}$, typically within the range of $[30s, 120s]$ for safety considerations. If lane k is congested and cannot be cleared, $t_k' = 0$. The remaining green lights in $CGreen1 / EL$ with $t_k' \neq 0$ are retained. Therefore, $CGreen1' = CGreen1 / EL$, and $R' = R \cup EL$. $CGreen1'$ and R' contain the updated elements.

To improve road traffic efficiency, starting from the first element of *Priority*, continue selecting lanes (or pedestrian lanes) that can have green lights one by one. The expression for updating the coordinated green light set is as follows:

$$CGreen' = \bigcup_{i=0}^{n-1} \begin{cases} CGreen1', (i = 0 \wedge t_i = 0) \\ CGreen1', (i = 0 \wedge t_i \neq 0 \wedge l_i \in CGreen') \\ CGreen1', (i = 0 \wedge t_i \neq 0 \wedge l_i \in R') \\ CGreen1' \cup \{l_i\}, (i = 0 \wedge t_i \neq 0 \wedge l_i \notin CGreen1' \wedge SR_i \in R') \\ CGreen1'_{i-1}, (i > 0 \wedge l_i \in R'_{i-1}) \\ CGreen1', (i-1, (i > 0 \wedge t_i = 0) \\ CGreen1'_{i-1} \cup \{l_i\}, \begin{pmatrix} i > 0 \wedge t_i \neq 0 \wedge l_i \notin CGreen1'_{i-1} \wedge SR_i \in R'_{i-1} \\ \wedge \forall l_i, l_i \notin SR_i, l_i \notin CGreen1'_{i-1} \end{pmatrix} \end{cases} \tag{16}$$

If an element from the queue already exists in the CGreen1′ set, skip it. For each selection, evaluate the mutually exclusive relationships defined in the inference rule library S. If the requirements are not met, continue evaluating the next element in the queue. Similarly, select all the green, yellow, and red light groups that can simultaneously display green lights in this round, denoted as CGreen′, and use the weight Equation to identify the set in CGreen′ with the most elements, denoted as $CGreen1 = \{l_i', l_j', ..., l_m'\}$.When the countdown timer of the green light groups in the EL set reaches $3s$, the yellow lights of those groups that need to transition from green to red start flashing. Simultaneously, the light groups in the CGreen1 set display green lights, while the remaining light groups display red lights. Additionally, the newly added green light groups in the timer are updated according to the times in $GreenT'$. The timer durations of the previous round's green lights remain unchanged. Furthermore, update the priority queue *Priority* after the light transition in this round. Move the light groups currently displaying green, yellow, and red lights to the end of the queue in that order.

3. Adaptive intelligent perception precision control method

The proposed method uses logical reasoning rules to control traffic lights. It employs a CPU-based system with real-time regulation. It handles mixed traffic scenarios, including connected vehicles, manual vehicles, pedestrians, signal lights, and timers. IIoT devices like monitors, sensors, lane controllers, ECUs, and control systems are utilized.

The method establishes inference rules based on traffic regulations, road topology, and flow control constraints. It analyzes road conditions using monitoring, sensors, and connected vehicle feedback. DL methods initialize and prioritize signal light groups, determining their green light duration. Rules are sorted based on priority. Each rule infers green and red light sets, forming a coordinated set. Elements are rearranged until all rules are included. A weight model selects the best element. Signal light groups for the next cycle are controlled based on the optimal element's green and red light sets. Congestion and green light durations are checked. If conditions are not met, the process repeats. Otherwise, green lights on congested roads and those completing their duration transition to red lights through a yellow light process. Steps are repeated for subsequent cycles to control the intelligent traffic light system. See **Fig. 2** for the flowchart.

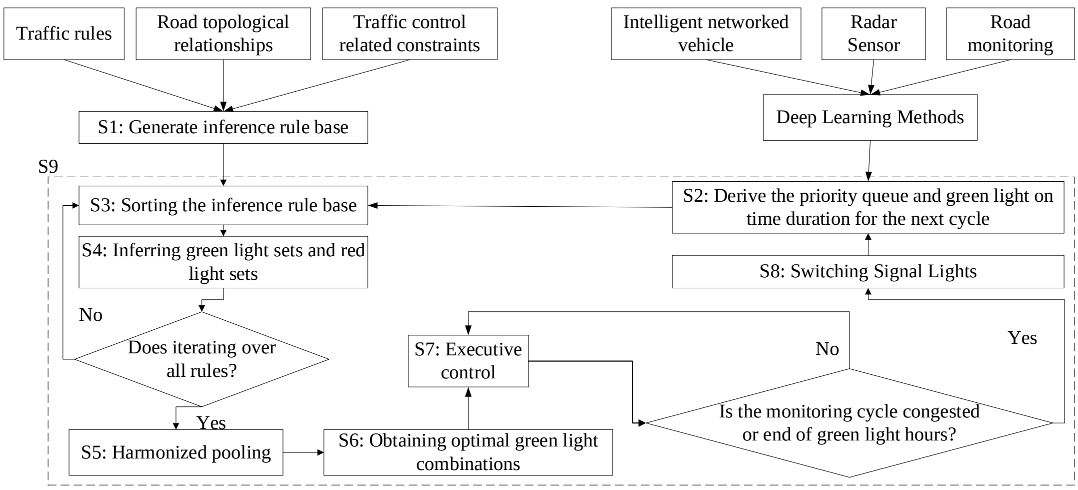


Fig. 2. Flowchart for real-time intelligent networked hybrid vehicle traffic light control

3.1 Algorithm for getting priority queues

Based on the (1) for signal light elements, the *Priority* value for each element can be calculated, and by sorting them in descending order, the initial *Priority* queue can be obtained.

Definition 1. Let $L=\{l_i\},i=0,1,\ldots\ldots,n-1$ represent the initial n signal light elements. P_i, v_i and Len_i denote the number of pedestrians waiting, average vehicle speed, and queue length of vehicles in the corresponding lane for signal light l_i. Using a two-dimensional array W[2][n] to store the *Priority* values and corresponding coordinates of the n signal light elements, $W[0][k]= W_i, W[1][k]=i$. Using L, and P_i, v_i and Len_i obtained from Equations (2)-(4) as inputs to the algorithm, the output priority queue *Priority*.

Algorithm1: Algorithm for getting priority queues

Input: $P_i, v_i, Len_i, L \leftarrow \{l_i\}, i \leftarrow 0,1,\ldots\ldots n\text{-}1$

Output: *Priority*[*n*]

When $W[0][k]$ is the current maximum value, *m* is the corresponding $W[1][k]$

Again: Iterate through all the signal elements sequentially

Put the weight value and subscript of the signal element into $W[2][n]$

For $i \leftarrow 0$ ***to*** n ***do***

  $\{W[0][k] \leftarrow P_i \cdot W_p + v_i \cdot W_v + Len_i \cdot W_{Len} + C_i \cdot W_c + J_i \cdot W_J$

  $W[1][k] \leftarrow i;\ k{+}{+};i{+}{+}\}$

Select the subscripts corresponding to the largest weight values in $W[2][n]$ in order

Store l_i into *Priority*[n] in order

For $i \leftarrow 0$ ***to*** k ***do***

  $\{m \leftarrow W[1][k]$

  Priority[n]=*Priority*[n]∪*L*[*m*]; *k*++}

Return *Priority*[n], go back to Again

Based on the description of Definition 1, the priority queue can be obtained as shown in **Algorithm 1**. By **Algorithm 1**, the initial priority queue *Priority*=$\{l_0, l_1, \ldots\ldots, l_{n-1}\}$ can be obtained. l_i, $i=0,\ldots, n\text{-}1$ represent different groups of signal lights, arranged in descending order based on their priority values.

3.2 Algorithm for obtaining the optimal coordination set

Based on the inference rules and the priority queue *Priority* obtained from Algorithm 1, a traversal is performed starting from the first element of *Priority*. By using Equations (8)-(11) and Equation (12), the green light set *G*, red light set *R*, and the optimal coordinated set of green lights based on rule inference can be obtained for each round.

Definition 2. Let G_n and R_n represent the green light coordinated set and the corresponding red light set for each round, respectively. CGreen1[*n*][2] represents the optimal coordinated set. With *n* signal light elements, a two-dimensional array PriorityGreen[*n*][2] is used to store the priority queue obtained from Algorithm 1, and the duration for each signal light to be green. PriorityGreen[*i*][0]=*Priority*[*i*]=l_i, PriorityGreen[*i*][1]=GreenT[*i*]=$t_i, \forall\, t_i \geq 0$, i=0, 1, ……,n-1. Included among these, PriorityGreen[*n*][2] and the green light coordinated set CGreen[*n*][2] are used as input to the algorithm. Where CGreen[*n*][0]=$\{G_i\}$ and CGreen[*n*][1]=$\{num_i\}$, num_i represents the number of elements in the corresponding set. G_n and R_n are initially empty sets. The algorithm also obtains the optimal coordinated set of green lights CGreen1[*n*][2] based on (12).

Algorithm2: Algorithm for obtaining the optimal coordination set.

Input: PriorityGreen[n][0], $S[n]$.
Output: G_n, R_n.

l←PriorityGreen[0][0]
while PriorityGreen[n][0]$\neq l$ **do{**
$WG[n]\leftarrow PN[n]\leftarrow vN[n]\leftarrow P[n]\leftarrow\{0\}$
For $i\leftarrow 0$ ***to*** n ***do***{
if $i\leftarrow 0$ ***then***{
  if $t_i\leftarrow 0$ ***then*** $\{G_n\leftarrow G_n\cup G_0; R_n\leftarrow R_n\cup R_0\}$
  else{
    if $l_i\in R_0$ ***then***$\{G_n\leftarrow G_n\cup G_0; R_n\leftarrow R_n\cup R_0\}$
    else$\{G_n\leftarrow G_n\cup\{l_i\}; R_n\leftarrow R_n\cup\{s_i\rightarrow\cap_j R(l_j)\,|\,l_i,l_i\in L; 0\leq i\leq(n\text{-}1)\}\}\}\}$
else{
  if $l_i\in R_{i-1}$ *or* $t_i\leftarrow 0$ ***then***$\{G_n\leftarrow G_n\cup G_{i-1}; R_n\leftarrow R_n\cup R_{i-1}\}$
  else if$(\exists r\in G_{i-1},\ r\in\{s_i\rightarrow\cap_j R(l_j)\,|\,l_i,l_i\in L;0\leq i\leq(n\text{-}1)\})$
    $\{G_n\leftarrow G_n\cup G_{i-1};\ R_n\leftarrow R_n\cup R_{i-1}\}$
  else if$(t_i>0$ *and* $l_i\notin R_{i-1}$ *and* $\forall r\in\{s_i\rightarrow\cap_j R(l_j)\,|\,l_i,l_i\in L;0\leq i\leq(n\text{-}1)\},r\notin G_{i-1})$
    $\{G_n\leftarrow G_{i-1}\cup\{l_i\};\ R_n\leftarrow R_n\cup\{s_i\rightarrow\cap_j R(l_j)\,|\,l_i,l_i\in L;0\leq i\leq(n\text{-}1)\}\}\}$
Move the first element of the current PriorityGreen[n][0] to the end of the queue }
CGreen[n][0]←$\{G_n|n\leftarrow 0,1,\ldots\ldots n\text{-}1\}$
CGreen[n][1]←$\{num_0,num_1,\ldots\ldots,num_{n-1}\}\}$
For $i\leftarrow 0$ ***to*** n ***do*** **{*For*** $j\leftarrow 0$ *to* num_i ***do***
  if *element in* $G_i\leftarrow l$ ***then*** $\{PR[i]\leftarrow 1\}$
  if *element in* G_i*contains 'P'* ***then*** $\{PN[i]\leftarrow PN[i]+1\}$
  else$\{vN[i]\leftarrow vN[i]+1;\}\}$
$WG_i\leftarrow PN_i\cdot W_{PN}+vN_i\cdot W_{vN}+PR_i\cdot W_{PR}\}$
$m\leftarrow index(max(WG_i)),\ (i=0,1,\ldots\ldots,n\text{-}1)$
CGreen1[n][0]←G_m;CGreen1[n][1]←num_m

Based on the principle of establishing the inference rule library S. Where each rule is represented as $s_i\rightarrow G(l_i)$ for the i-th group of green lights and $s_i\rightarrow\wedge_j R(l_j)$ for all red elements corresponding to that green light. Based on the **Definition 2**, the algorithm for updating the green light and red light sets can be obtained, as shown in **Algorithm 2**.

3.3 Adaptive perceptual control methods

Based on the n signal light elements in L and the principle of ensuring traffic safety while maximizing efficiency, n rule $S[n]$ are set as input for the algorithm, where $S=\{s_i:G(l_i)\rightarrow\wedge_j R(l_j)\,|\,l_i,l_j\in L;0\leq i\leq(n-1)\}$.

Definition 3. Let $CGreen1[n][2]$ be the optimal coordinated set, obtained from the priority queue generated by **Algorithm 1** and the n green light coordinated sets based on rule inference obtained from **Algorithm 2**. The optimal coordinated set is determined using (12). Based on this, the $CGreen1[n][2]$ is further optimized based on real-time traffic congestion conditions. It serves as the initial green light coordinated set G_0 and is subjected to **Algorithm 2** and subsequent operations. the algorithm for obtaining the real-time optimal green light coordinated set for each round can be obtained, as shown in **Algorithm 3**.

Algorithm3: All steps of the algorithm proposed in this paper.

Input: $S[n]$, $L[n]$.
Output: CGreen1[n][2].

Priority[j]←{ $l_i,\ldots,l_{n-1},l_0,\ldots,l_{i-1}$ |∀i∈{0,…,n-1},Priority[j-1]←{ $l_{i-1},\ldots,l_{n-1},l_0,\ldots,l_{i-2}$ }}
PriorityGreen[n][0]← *Priority*[n]
PriorityGreen[n][1]← GreenT←t_i
l←PriorityGreen[0][0]; EL←{0}
Obtain the optimal coordination set CGreen1[n][2] of the last round from **Algorithm2**
Call **Algorithm1** and **Algorithm2** again, update CGreen1[n][2], and store it into CGreen1'[n][2]
CGreen1'[n][0]←CGreen1[n][0]
CGreen1'[n][1]←CGreen1[n][1]
For i←0 ***to*** num_m ***do{***
    if CGreen1[n][1]<15 ***then***{ EL←EL∪CGreen1[i][0]; R'←R∪CGreen1[i][1]
    CGreen1'[i][0]←CGreen1'[i][0]/CGreen1[i][0] }
    if J_i←0 ***then*** {CGreen1'[i][1]←0 }}
R_0←R'; CGreen1← G_0 ←CGreen1'[n][0]
The light in the EL lights up yellow, the light in CGreen1 is green, and the light in R is red

Algorithm 3 utilizes real-time traffic data to accurately control and reduce ineffective waiting time. It also dynamically adjusts the control lanes based on the real-time traffic conditions. This approach significantly reduces wastage of signal light timing and maximizes road utilization.

4. Case study

4.1 Background of the case

At the traffic intersection, assume there are four sets of red and green lights for each direction (16 sets total), four vehicle information collection points, four monitoring devices, and 16 pedestrian traffic lights. Each signal light group has its own timer (32 in total).

Directions are identified as SNi, NSi, WEi, EWi: SN (south to north), NS (north to south), WE (west to east), and EW (east to west). The index i indicates the turn type: i=1 (U-turn), i=2 (left turn), i=3 (straight), i=4 (right turn). L1 = {SNi, NSi, WEi, EWi | i =1,2,3,4}. Pedestrian walkways are divided into two segments for each direction, with four sets each having four parts: NPi, SPi, EPi, WPi, where i=1 (front half), i=2 (rear half). L2={NPi, SPi, EPi, WPi | i =1,2}. The intersection has four radar sensors (SNC1, WEC1, NSC1, EWC1) and four road monitoring devices(SNRV, WERV, NSRV, EWRV). The schematic diagram is shown in **Fig. 3**.

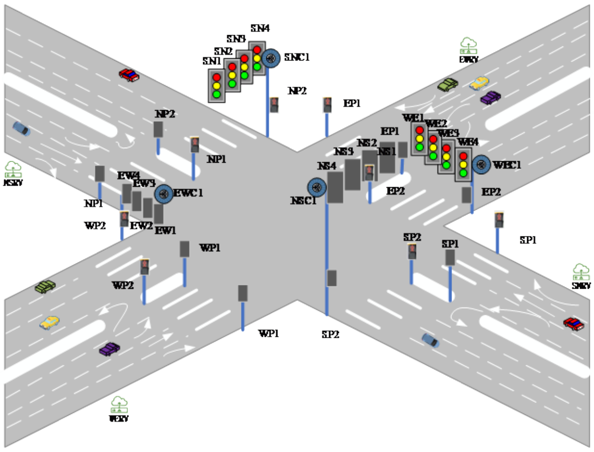


Fig. 3. illustrates a schematic diagram of the intersection environment in the case study.

Assuming historical data during peak hours, in the time interval of [15s,30s], congestion occurs on the NP2 segment due to students leaving school. To address this congestion, this paper proposes an intelligent adaptive traffic light control method based on real-time traffic conditions.

4.2 The application of adaptive intelligent perception control methods.

The set of unique identifier names for all signal light groups is denoted as $L=L1\cup L2$. The control cycle duration is T=15s. The initial moment of the current cycle is $t_0=0$.

Table 1. Rule Knowledge Base $\sum$

S	G	R
s_0	{SN1G}	{SP1R,SP2R,EW2R}
s_1	{SN2G}	{SP1R,EW2R,NS2R,NS3R,WP2R,WE1R,WE2R,WE3R}
s_2	{SN3G}	{SP1R,EW2R,EW3R,NP2R,NS2R,WE3R}
s_3	{SN4G}	{EP2R,SP1R}
s_4	{EW1G}	{EP1R,EP2R,NS2R}
s_5	{EW2G}	{EP1R,SP2R,NS2R,WE2R,WE3R,SN1R,SN2R,SN3R}
s_6	{EW3G}	{EP1R,WP2R,NS2R,NS3R,WE2R,SN3R}
s_7	{EW4G}	{EP1R,NP2R}
s_8	{NS1G}	{NP1R,NP2R,WE2R}
s_9	{NS2G}	{NP1R,EP2R,WE2R,SN2R,SN3R,EW1R,EW2R,EW3R}
s_{10}	{NS3G}	{NP1R,SP2R,WE2R,WE3R,SN2R,EW3R}
s_{11}	{NS4G}	{NP1R,WP2R}
s_{12}	{WE1G}	{WP1R,WP2R,SN2R}
s_{13}	{WE2G}	{WP1R,NP2R,SN2R,EW2R,EW3R,NS1R,NS2R,NS3R}
s_{14}	{WE3G}	{SN2R,SN3R,WP1R,EP2R,EW2R,NS3R}
s_{15}	{WE4G}	{WP1R,SP2R}
s_{16}	{NP1G}	{NS1R,NS2R,NS3R,NS4R}
s_{17}	{NP2G}	{EW4R,WE2R,NS1R,SN3R}
s_{18}	{EP1G}	{EW1R,EW2R,EW3R,EW4R}
s_{19}	{EP2G}	{WE3R,SN4R,EW1R,NS2R}
s_{20}	{SP1G}	{SN1R,SN2R,SN3R,SN4R}
s_{21}	{SP2G}	{WE4R,NS3R,EW2R,SN1R}
s_{22}	{WP1G}	{WE1R,WE2R,WE3R,WE4R}
s_{23}	{WP2G}	{WE1R,SN2R,EW3R,NS4R}

Based on traffic regulations and prioritizing safety, the operational rules for the 24 signal light groups are described as follows, and the inference rule knowledge base $\sum$ is established as shown in **Table 1**. In the table, *G* represents the green light element, and R represents all the red light elements corresponding to the rule.

The priority **Algorithm 1** assigns the following weights: $W_p = 0.2, W_v = 0.5$, $W_{Len} = 0.3$, $W_c = -100, W_J = -\infty$. By substituting these weight values into the algorithm equation mentioned above, obtain the priority equation for the coordinated set of green lights:

$$W_i = P_i \cdot 0.2 + v_i \cdot 0.5 + Len_i \cdot 0.3 + C_i \cdot (-100) + J_i \cdot (-\infty) \quad (17)$$

By substituting Equation (17) into **Algorithm 1**, obtain the initialized priority queue *Priority* [24]={SN3, SN1, WE2, EW3, EW4, NS1, NS4, NP2, WP2, EP2, SP1, SN2, SN4, WE1, WE3, WE4, NS2, NS3, NP1, SP2, WP1, EP1, EW1, EW2}.Through saturation detectors WERV,EWRV,NSRV, and SNRV monitoring the traffic saturation levels of the respective lanes at the current moment. If all saturation levels are less than ”1”, then set the predetermined green light time for each lane as Time[24] = {90, 60, 60, 60, 60, 60, 60, 60, 30, 30, 30, 60, 60, 60, 60, 60, 60, 60, 60, 30, 30, 30, 30, 60}.

4.3 Obtaining the coordinated set for the case study.

According to **Algorithm 2,** iterate from the first element to the last element of the updated *Priority* queue. Then, update the *Priority*[ ] queue until the first element of the current round, SN3, returns to the front of the *Priority*[ ] queue, ending the current round of iteration. By doing this, all possible green light sets for this round can be obtained. For easier observation and analysis, the green light sets are arranged in the order of S-E-N-W and duplicate sets are removed. Finally, 21 green light sets are obtained as shown in **Table 2**. Using (12), the optimal green light coordination set is selected from 21 green light sets.

Table 2. Coordinated Green Light Sets

CGreen[i][0]	G_i	**CGreen[i][1]**	**num_i**
CGreen[0][0]	{SN1,SN3,EP2,EW4,NS4,WE1,WE2,WE4}	CGreen[0][1]	8
CGreen[1][0]	{SP1,EP2,EW4,NS4,WE1,WE2,WE4}	CGreen[1][1]	7
CGreen[2][0]	{SP1,EP2,EW2,EW3,EW4,NS1,NS4,WE1,WE4 }	CGreen[2][1]	9
CGreen[3][0]	{SP1,EP2,EW2,EW4,NS1,NS3,NS4,WE1,WE4}	CGreen[3][1]	9
CGreen[4][0]	{ SP1,EP1, EP2,NS1,NS3, NS4, WE1,WE4 }	CGreen[4][1]	8
CGreen[5][0]	{ SP1,EP1, EP2,NP2,NS3, NS4, WE1,WE4 }	CGreen[5][1]	8
CGreen[6][0]	{ SP1, EP1,EP2,NP2,NS3,WP2,WE4 }	CGreen[6][1]	7
CGreen[7][0]	{ SP1, EP1,EP2,NS1,NS3,WP2,WE4 }	CGreen[7][1]	7
CGreen[8][0]	{ SP1, EP1,EP2,NS1,NS3,NS4,WE1,WE4 }	CGreen[8][1]	8
CGreen[9][0]	{SP1,EP1,NS1,NS2,NS4,WE1,WE3,WE4 }	CGreen[9][1]	8
CGreen[10][0]	{SN1, SN2,SN3, SN4,EP1,NP1, WE4 }	CGreen[10][1]	7
CGreen[11][0]	{SN1,SN4,EP1,NS1,NS2,NS4,WE1,WE3,WE4}	CGreen[11][1]	9
CGreen[12][0]	{ SN1,SN4,EP1,NS1,NS2,NS4,WE3,WE4 }	CGreen[12][1]	8
CGreen[13][0]	{ SN1,SN4,EP1,NS1,NS2,NS3,NS4,WE1,WE4}	CGreen[13][1]	9
CGreen[14][0]	{ SN1,SN4, EP1, NS1,NS2,NS3, NS4,WP1 }	CGreen[14][1]	8
CGreen[15][0]	{ SN1, SN3,EP1,EP2,NS1,NS3,NS4,WP1}	CGreen[15][1]	8
CGreen[16][0]	{ SP2,SN3,EP1,EP2,NP1,WP1,WP2 }	CGreen[16][1]	7
CGreen[17][0]	{SP2,SN2,SN3,EP1,EP2,NS1,NS4,WP1}	CGreen[17][1]	8
CGreen[18][0]	{ SN1,SN2, SN3,EP1,EP2, NS1,NS4, WP1 }	CGreen[18][1]	8
CGreen[19][0]	{ SN1,SN3,EP1,EP2,NS4,WE1,WE2,WE4}	CGreen[19][1]	8
CGreen[20][0]	{ SP1,EW1,EW2,EW3,EW4,NS1,NS4,WE1,WE4}	CGreen[20][1]	9

The weights assigned to (12) are as follows: W_{PN}=0.1, W_{vN}=1, W_{PR} =100. By substituting the weight values into the algorithm equation mentioned above, the priority equation for the coordinated green light set is as follows Equation (18):

$$WG_i = PN_i \cdot 0.1 + vN_i \cdot 1 + PR_i \cdot 100 \tag{18}$$

In the optimal coordinated green light set CGreen[0][0] = Green[0] = {SN1, SN3, EP2, EW4, NS4, WE1, WE2, WE4}, CGreen1[0][1] = 8, the vehicle passing lanes are SN1, SN3, EW4, NS4, WE1, WE2, WE4, totaling 7, vN_0 = 7; the pedestrian crossing lane is EP2, only 1, PN_0 = 1; and the set includes the first element of the current priority queue, SN3, PR_0 = 1. Therefore, the priority of CGreen[0] is: WG_0=1·0.1+7·1+1·100=107.1. Similarly, the priorities of other coordinated green light sets can be calculated, and then the set with the highest value can be selected as the optimal coordinated green light set. After calculation, it is found that CGreen[0][0] has the highest priority value. According to Equations (13) to (15), it can be concluded that CGreen[0][0] is the final coordinated green light set for the first round: CGreen1[0][0]= { SN3, SN1, WE2, EW4, NS4, EP2, WE1, WE4},CGreen1[0][1] = 8.

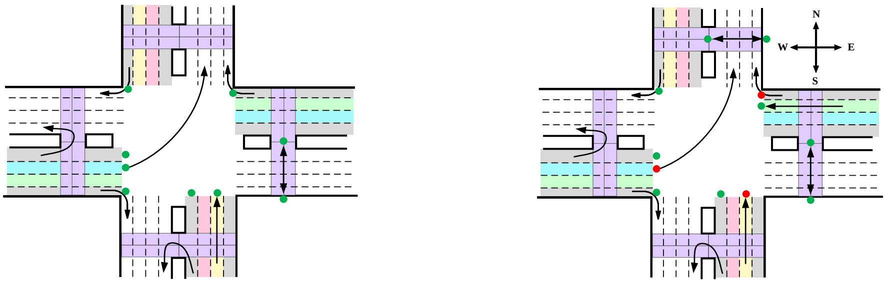
From the initial conditions, for CGreen1[0][0], the corresponding green light durations are *T*[8] = {90, 60, 60, 60, 60, 30, 60, 60}, and the remaining 16 elements in *Priority*[24] are set to display red lights. By moving all the elements in the optimal coordinated green light set CGreen1[0][0] to the end of *Priority*[24] at the beginning of this round, the updated *Priority*[24] = {EW3, NS1, NP2, WP2, SP1, SN2, SN4, WE3, NS2, NS3, NP1, SP2, WP1, EP1, EW1, EW2, SN3, SN1, WE2, EW4, NS4, EP2, WE1, WE4} can be obtained. The corresponding durations in *Time*[24] = {60, 60, 60, 30, 30, 60, 60, 60, 60, 60, 60, 30, 30, 30, 30, 60, 90, 60, 60, 60, 60, 30, 60, 60}.

4.4 Updating the Optimal Coordinated Set in the Case Study

According to Algorithm 3, assuming that after 15s of normal countdown for the green light, CGreen1[0][0] corresponds to the remaining durations becoming GreenT[8] = {75, 45, 45, 45, 45, 15, 45, 45}. It is observed that the green light element EP2 in CGreen1[0][0] has a remaining duration of only 15s, reaching the maximum limit. It should be included in the EL group for normal countdown and preparation for transitioning to yellow light.

Due to congestion on the northbound road, which is an outbound route with high traffic and pedestrian flow, the traffic lights NS1,WE2,SN3 and EW4 in that section will be affected. From the updated corresponding durations of the coordinated green light sets, it is found that SN3,WE2 and EW4 have remaining green light durations of 75s,45s and 45s, respectively, which have not reached the maximum limit. At this point, SN3,WE2 and EW4 can be included in the *EL* group for a non-natural transition, first transitioning to yellow light and then preparing to transition to red light. Set the next corresponding green light duration for SN3,WE2 and EW4 as t_i=0 and include SN3,WE2 and EW4 in the *EL* group. *EL*=*EL* ∪{SN3, WE2, EW4} = {EP2, SN3, WE2, EW4}. This process should be continuously monitored and the *EL* group should be updated accordingly. According to the road conditions, the green light set and red light set are updated by moving SN3, WE2, and EW4 from the green light set to the red light set. CGreen1'[0][0] = CGreen1[0][0] -{SN3, WE2, EW4} = {SN1, NS4, EP2, WE1, WE4}, and *R'*[0] = *R*[0] ∪ *EL*. The corresponding times for the congested signal lights in Time are set to 0. Therefore, the updated times $Time'[24]$ = {45, 45, 45, 15, 15, 45, 45, 45, 45, 45, 45, 15, 15, 15, 15, 45, 0, 45, 0, 0, 45, 15, 45, 45}.In order to maximize the utilization of green lights and road capacity, after determining the traffic lights that turn yellow due to congestion rather than natural causes, real-time road condition monitoring is combined. The current elements in *Priority*[24] and their corresponding inference rules are traversed again to find if there are new elements that can be added to the green light set as new green light elements. If an element in *Priority*[24] is already in the green light set CGreen1'[0][0], it is skipped. If an element in *Priority*[24] is not in CGreen1'[0][0], but the corresponding red light elements on the right side of the inference rule can only be found in the set *R'*[0], then it is considered that this element can replace the traffic light group that turned red due to congestion and be added to CGreen1'[0][0] as a green element.

For example, during the traversal process, from rule s_6, it is determined that EW3 is not in the current green light set CGreen1'[0][0], and the corresponding red light elements can only be found in the current red light set *R'*[0]. Therefore, EW3 can be added to the current green light set, CGreen1'[0][0] = CGreen1'[0][0] ∪ {EW3}. Continuing the traversal, from s_{17}, NP2 is not in the current CGreen1'[0][0], and the corresponding red light elements belong to the R'[0] set. Similarly, CGreen1'[0][0] can be updated as CGreen1'[0][0] = CGreen1'[0][0] ∪ {NP2} . Following this process, after completing the traversal, the new green light set is obtained as CGreen1[0][0] = CGreen1'[0][0] = {SN1, NS4, EP2, WE1, WE4, EW3, NP2}, where EW3 and NP2 are the newly added green light elements, while the red light set *R'*[0] remains unchanged.When new elements EW3 and NP2 are added to the green light coordination set, it is necessary to analyze the traffic capacity and saturation of the road segments ahead of the lanes based on real-time road condition monitors and sensor data. The updated duration of the current green light set CGreen1[0][0] is GreenT'= {60, 90, 60, 30, 30, 60, 60}. The optimal green light coordination set before the update is shown in **Fig. 4 (a)**, taking into account the congestion on the north road segment. The affected SN3, WE2, and EW4 in the green light set are converted to red lights, and the newly added green lights are incorporated into the CGreen1' set. The updated optimal coordination set is shown in **Fig. 4 (b)**.



(a) (b)

Fig. 4. Schematic Diagram of Implementation Case Rules.

After a countdown of $3s$, the traffic light groups *EL* = {EP2,SN3,WE2,EW4} need to prepare for blinking yellow lights as the green light countdown timer reaches 3 seconds. At the same time, the traffic light groups in the new CGreen1[0][0] = {SN1, NS4, EP2, WE1, WE4, EW3, NP2} set will display green lights. The newly added green light elements, EW3 and NP2, will be moved to the end of the priority queue *Priority*[24], resulting in an updated *Priority*[24] = {NS1, WP2, SP1, SN2, SN4, WE3, NS2, NS3, NP1, SP2, WP1, EP1, EW1, EW2, SN3, SN1, WE2, EW4, NS4, EP2, WE1, WE4, EW3, NP2}, Then proceed to the next round of selection sorting.

4.5 Algorithm Simulation Comparison

For ease of representation, the algorithm of this paper is denoted by RAISC (Rule-based Adaptive Intelligent Sensing Control) in the subsequent experimental descriptions and diagrams. Based on the RAISC theory introduced earlier, the simulation of this control method at planar intersections is carried out with the help of the microscopic software VISSIM[34]. The experiments in this paper are based on the simulation of a large cross-shaped intersection shown in **Fig. 3**. The measured traffic flow data of the target intersection in the morning peak (9:00-11:00), as shown in **Table 3**, to fully compare the timed signal control with the RAISC, four traffic flow schemes are designed, and simulated traffic accidents at the north entrance.

To evaluate the effectiveness of our model, we compare our model with these following methods based on the metrics system (passing time, lane delay, stopping rate) established in the previous Equations (5)-(7) and adjust the parameters for all methods.

(1) Fixed Timing (FT). Fixed time[36] control methods use predetermined cycle and phase times, and this method is widely used in situations where the traffic flow is stable.

(2) Self-Organizing Traffic Light Control (SOTL). This method accumulates the number of approaching vehicles through each red light counter and switches to green when a threshold is reached and avoids frequent switching[37]. The controller makes self-organizing decisions based on local traffic information to form a dynamic green wave, which improves traffic flow and reduces waiting time.

(3) Fuzzy Rule Inference System (MFIS). This system is based on fuzzy control[38] theory and dynamically adjusts the traffic light time through information fuzzification, fuzzy control rule formulation and defuzzification process, thus reducing traffic congestion and improving traffic efficiency.

Table 3. Intersection Traffic Flow

Lanes	Names	Symbols	Plan 1	Plan 2	Plan 3	Plan 4
South	Southbound U-turn	SN1	50	50	50	100
	Southbound left turn	SN2	900	1000	1200	1500
	Southbound through	SN3	700	800	900	1050
	Southbound right turn	SN4	50	50	50	100
	South pedestrian crossing	SP	200	400	600	660
North	Northbound U-turn	NS1	50	50	50	100
	Northbound left turn	NS2	800	900	1000	1200
	Northbound through	NS3	800	900	1000	1200
	Northbound right turn	NS4	50	50	100	130
	North pedestrian crossing	NP	200	400	600	640
East	Eastbound U-turn	EW1	50	50	50	100
	Eastbound left turn	EW2	50	50	50	100
	Eastbound through	EW3	700	800	1000	1200
	Eastbound right turn	EW4	700	800	900	1000
	East pedestrian crossing	EP	100	100	200	300
West	West bound U-turn	WE1	500	600	700	800
	West bound left turn	WE2	700	800	900	1000
	West bound through	WE3	100	120	150	200
	West bound right turn	WE4	50	50	50	100
	West pedestrian crossing	WP	100	100	200	240
Total traffic flow/(pcu·h^{-1})			6850	8070	9750	11720

The traffic simulation model established for the target intersection is shown in **Fig. 5**. Using the intersection access efficiency evaluation index system established by (5)-(7), we simulate the traffic conditions for periods of 100-350 seconds and 350-600 seconds at the target intersection. Taking the traffic flow shown in Plan 1 in **Table 3** as an example, the data results of the experimental simulation are visualized in the index system as shown in **Fig. 6**, with similar results for other schemes.

Because the values of each metric for the three lanes SN3, WE2, and EW3, which are greatly affected by congestion, are very different from those of the other lanes (for example, the delay of lane SN3 is as high as 243.35 under the FT control method), only lanes other than these three are included in the analysis of the experimental data visualization.

To visualize the effect of different control methods on the overall intersection traffic efficiency rather than individual lanes, the average queuing delay of the intersection is used as the evaluation index. The average queuing delay is calculated based on (6), with the traffic flow of each lane as the weight. A comparative analysis of the delay results simulated by the four methods is shown in **Table 4**, **Table 5** and **Fig. 7**. Where in both **Fig. 6** and **Fig. 7**, the solid red line is used to represent our proposed methodology.

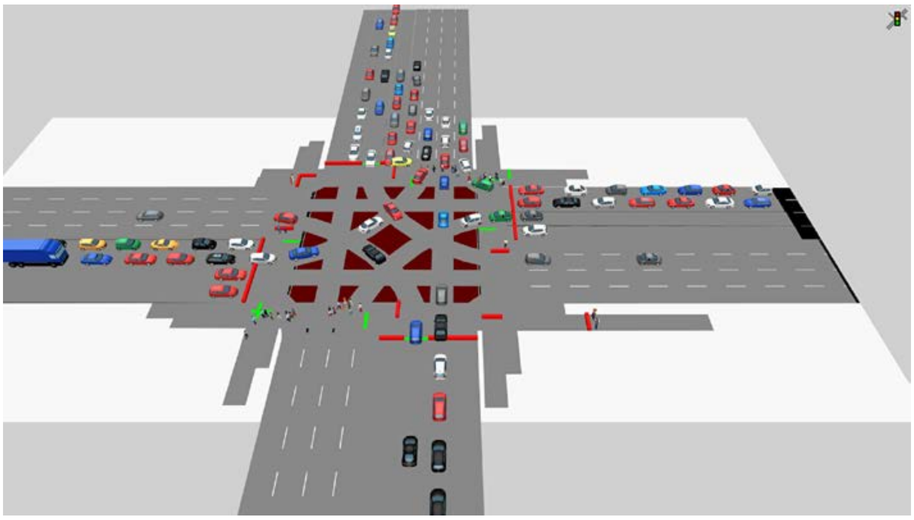
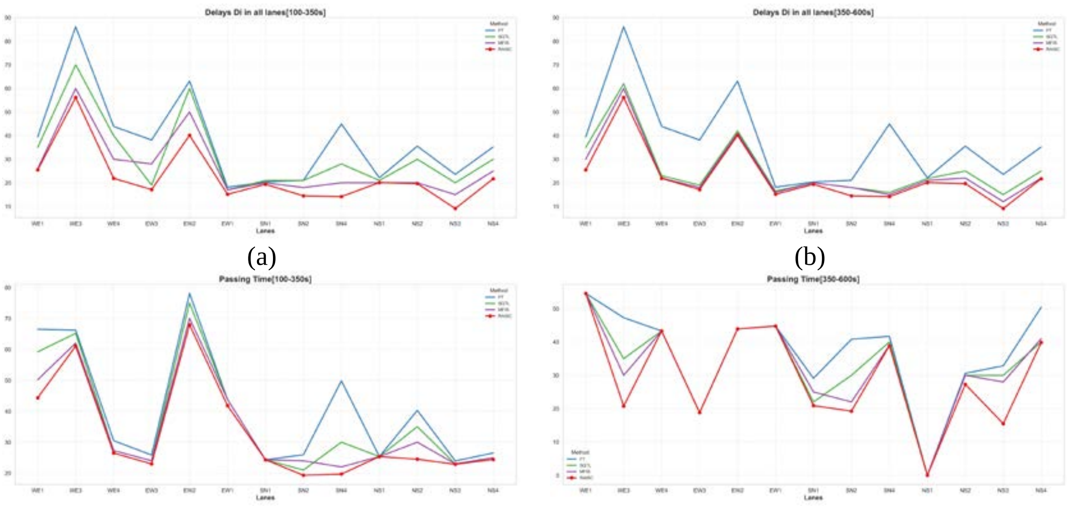
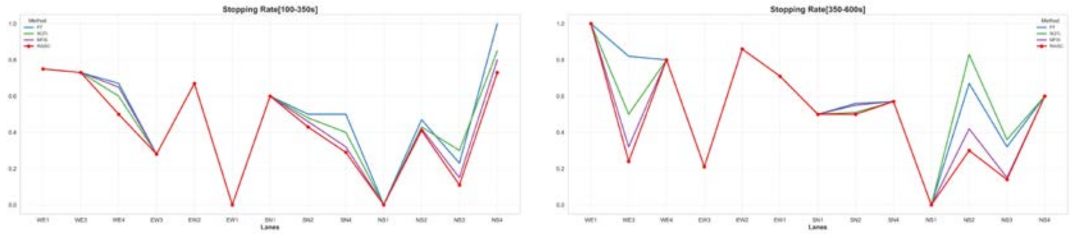


Fig. 5. Traffic Accident Scene Simulation



(a) (b)

(c) (d)



(e) (f)

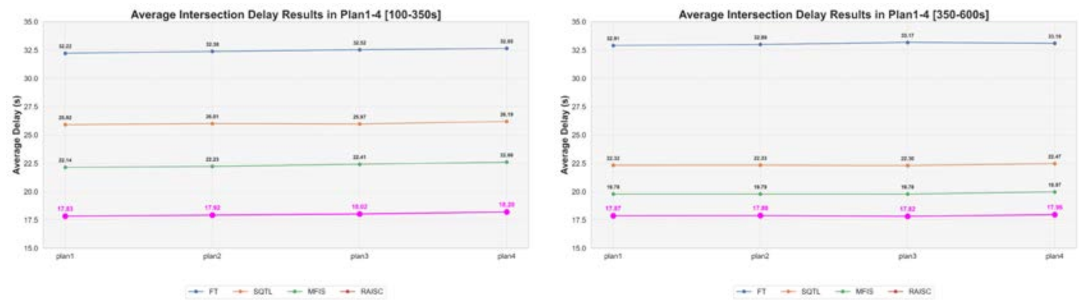
Fig. 6. Simulation results of each evaluation index

Table 4. Average Intersection Delay Results in Plan1-4 [100-350s]

Plan	FT	SQTL	MFIS	RAISC
1	32.2172	25.9157	22.1446	**17.8312**
2	32.3780	26.0064	22.2270	**17.9242**
3	32.5237	25.9725	22.4128	**18.0221**
4	32.6455	26.1933	22.5988	**18.1969**

Table 5. Average Intersection Delay Results in Plan1-4 [350-600s]

Plan	FT	SQTL	MFIS	RAISC
1	32.9072	22.3229	19.7831	**17.8676**
2	32.9919	22.3298	19.7859	**17.8836**
3	33.1740	22.3009	19.7798	**17.8174**
4	33.0958	22.4656	19.9722	**17.9639**



(a) (b)

Fig. 7. Visualization of average intersection delay in Plan1-4

The delays, passing times and stopping rates in each lane of the intersection during peak periods with high traffic flow are shown in **Fig. 6 (a)-(f)**. From the figures, it can be seen that the metric values of our proposed RAISC method are the lowest in 100s-350s, and in 350s-600s when accidents occur. For the whole intersection, the average delay of RAISC is significantly lower than that of FT, SQTL, and MFIS, as shown in **Fig. 7 (a)-(b)**. Among them, as shown in **Fig. 7 (b)**, the average delay of FT control method due to the lack of real-time adjustments stays high during the peak period when random accidents occur and is even higher than that of the previous period. Whereas SQTL, MFIS and our RAISC method all decrease. In comparison our RAISC method is more effective. And in Plan 3 and Plan 4, which have high traffic flow and random accidents, the effect of improving traffic efficiency and reducing the average delay at intersections is even more excellent. Due to the phenomenon of extracting the sample size, collection point positioning error and other phenomena, the accuracy of a few evaluation indexes is reduced, but in general it is within the acceptable range.

Therefore, when setting signal control at actual intersections, the adaptive perception signal control system can be preferred, especially in the peak and accident period, the advantages of adaptive perception signal control system are very obvious.

5. Conclusions

In this paper, a rule-based reasoning method for adaptive intelligent sensing and precise control of traffic signals under real-time intelligent grid-connected hybrid vehicle conditions is proposed. The core of the method is to dynamically adjust the green light duration by real-time monitoring of the parking queue length, pedestrian flow, vehicle flow, and traffic volume of each lane, and to adjust the green light priority according to the vehicle and road conditions. When a random accident occurs in a lane during peak hours and causes congestion, the traditional fixed timing method cannot flexibly adjust the control system, resulting in wasted green light time and high intersection delays. The method proposed in this paper enables precise real-time control and avoids the waste of green light time. In this paper, examples are given to further explain this method, and simulation comparisons are made with Fixed Timing, Self-Organizing Traffic Light Control, Fuzzy Rule Inference System, and many other methods. Through the experimental results, it can be seen that the method proposed in this paper effectively reduces the travel time, stopping rate, and lane delay values for each lane of the intersection, as well as the overall average delay value.

In summary, the method proposed in this paper is especially useful during peak hours that are prone to random traffic accidents and high traffic flow, and the method can improve traffic efficiency to a large extent compared to traditional methods. Of course, in addition to the optimal control of individual intersection signals, research on the control of arterial and regional signals can also be included in the scope of future work to further improve the algorithm for a wider range of optimal control. Moreover, the implementation of the method relies on a large number of sensors and data acquisition equipment, which increases the cost of system construction and maintenance. In the future, more efficient data acquisition and processing methods can be explored to reduce system costs.

Acknowledgement

This work was supported by the Chongqing Municipal Natural Science Foundation General Project CSTB2023NSCQ-MSX0441, and Chongqing Research Program (2022YSZX-JCX0011CSTB, CSTB2023YSZX-JCX0008) and CAS "Light of West China" Program, and the National Scholarship Council for Study Abroad[2020]50, and the National Natural Science Foundation of China Project NSFC 61903053 and 62073051, and the Science and Technology Research Project of the Chongqing Municipal Education Commission of P.R.China under Grants KJZD-K 201900704.

References

[1] Y. Fu, C. Li, F. R. Yu, T. H. Luan and Y. Zhang, "A Survey of Driving Safety With Sensing, Vehicular Communications, and Artificial Intelligence-Based Collision Avoidance," *IEEE Transactions on Intelligent Transportation Systems*, vol.23, no.7, pp.6142-6163, 2022. Article (CrossRef Link)

[2] Z. Ullah, F. Al-Turjman, L. Mostarda and R. Gagliardi, "Applications of Artificial Intelligence and Machine learning in smart cities," *Computer Communications*, vol.154, pp.313-323, 2020. Article (CrossRef Link)
[3] Y. Ma, Z. Wang, H. Yang and L. Yang, "Artificial intelligence applications in the development of autonomous vehicles: A survey," *IEEE/CAA Journal of Automatica Sinica*, vol.7, no.2, pp.315-329, 2020. Article (CrossRef Link)
[4] F. A. Butt, J. N. Chattha, J. Ahmad, M. U. Zia, M. Rizwan and I. H. Naqvi, "On the Integration of Enabling Wireless Technologies and Sensor Fusion for Next-Generation Connected and Autonomous Vehicles," *IEEE Access*, vol.10, pp.14643-14668, 2022. Article (CrossRef Link)
[5] X. Yang, D. Qin, J. Liu, Y. Li, Y. Zhu and L. Ma, "Deep reinforcement learning in NOMA-assisted UAV networks for path selection and resource offloading," *Ad Hoc Networks*, vol.151, 2023. Article(CrossRef Link)
[6] H. Wei, G. Zheng, V. Gayah and Z. Li, "Recent Advances in Reinforcement Learning for Traffic Signal Control: A Survey of Models and Evaluation," *ACM SIGKDD Explorations Newsletter*, vol.22, no.2, pp.12-18, 2021. Article (CrossRef Link)
[7] Y. Liu, G. Liu, Y. Wu, W. He, Y. Zhang and Z. Chen, "Reinforcement-Learning-Based Decision and Control for Autonomous Vehicle at Two-Way Single-Lane Unsignalized Intersection," *Electronics*, vol.11, no.8, 2022. Article (CrossRef Link)
[8] M. Noaeen, A. Naik, L. Goodman, et al., "Reinforcement learning in urban network traffic signal control: A systematic literature review," *Expert Systems with Applications*, vol.199, 2022. Article (CrossRef Link)
[9] Y. He, Y. Liu, L. Yang and X. Qu, "Deep Adaptive Control: Deep Reinforcement Learning-Based Adaptive Vehicle Trajectory Control Algorithms for Different Risk Levels," *IEEE Transactions on Intelligent Vehicles*, vol.9, no.1, pp.1654-1666, 2023. Article (CrossRef Link)
[10] H. Lin, Y. Han, W. Cai and B. Jin, "Traffic Signal Optimization Based on Fuzzy Control and Differential Evolution Algorithm," *IEEE Transactions on Intelligent Transportation Systems*, vol.24, no.8, pp.8555-8566, 2023. Article (CrossRef Link)
[11] Z. Zhong, M. Nejad and E. E. Lee, "Autonomous and Semiautonomous Intersection Management: A Survey," *IEEE Intelligent Transportation Systems Magazine*, vol.13, no.2, pp.53-70, 2021. Article (CrossRef Link)
[12] T. Chu, J. Wang, L. Codecà and Z. Li, "Multi-Agent Deep Reinforcement Learning for Large-Scale Traffic Signal Control," *IEEE Transactions on Intelligent Transportation Systems*, vol.21, no.3, pp.1086-1095, 2020. Article (CrossRef Link)
[13] H. Wang, M. Zhu, W. Hong, C. Wang, G. Tao and Y. Wang, "Optimizing Signal Timing Control for Large Urban Traffic Networks Using an Adaptive Linear Quadratic Regulator Control Strategy," *IEEE Transactions on Intelligent Transportation Systems*, vol.23, no.1, pp.333-343, 2022. Article (CrossRef Link)
[14] S. A. Celtek, A. Durdu and M. E. M. Alı, "Real-time traffic signal control with swarm optimization methods," *Measurement*, vol.166, 2020. Article (CrossRef Link)
[15] J. Jin and X. Ma, "A Multi-Objective Agent-Based Control Approach With Application in Intelligent Traffic Signal System," *IEEE Transactions on Intelligent Transportation Systems*, vol.20, no.10, pp.3900-3912, 2019. Article (CrossRef Link)
[16] A. Atta, S. Abbas, M. A. Khan, et al., "An adaptive approach: Smart traffic congestion control system," *Journal of King Saud University - Computer and Information Sciences*, vol.32, no.9, pp.1012-1019, 2020. Article (CrossRef Link)
[17] B. R. Kiran, I. Sobh, V. Talpaert, et al., "Deep reinforcement learning for autonomous driving: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol.23, no.6, pp.4909-4926, 2022. Article (CrossRef Link)
[18] J. Garcıa, F. Fernández, "A Comprehensive Survey on Safe Reinforcement Learning," *Journal of Machine Learning Research*, vol.16, no.1, pp.1437-1480, 2015. Article (CrossRef Link)
[19] A. Boukerche and J. Wang, "Machine Learning-based traffic prediction models for Intelligent Transportation Systems," *Computer Networks*, vol.181, 2020. Article (CrossRef Link)

[20] Z. Liu, Y. Liu, Q. Meng and Q. Cheng, "A tailored machine learning approach for urban transport network flow estimation," *Transportation Research Part C: Emerging Technologies*, vol.108, pp.130-150, 2019. Article (CrossRef Link)
[21] H. Wei, G. Zheng, V. Gayah, et al., "Recent Advances in Reinforcement Learning for Traffic Signal Control: A Survey of Models and Evaluation," *ACM SIGKDD Explorations Newsletter*, vol.22, no.2, pp.12-18, 2021. Article (CrossRef Link)
[22] H. Wei, G. Zheng, H. Yao, et al., "IntelliLight: A Reinforcement Learning Approach for Intelligent Traffic Light Control," in *Proc. of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp.2496-2505, 2018. Article (CrossRef Link)
[23] F.-M. Luo, T. Xu, H. Lai, X.-H. Chen, W. Zhang and Y. Yu, "A survey on model-based reinforcement learning," *Science China Information Sciences*, vol.67, no.2, 2024. Article (CrossRef Link)
[24] N. Kumar, S. S. Rahman and N. Dhakad, "Fuzzy Inference Enabled Deep Reinforcement Learning-Based Traffic Light Control for Intelligent Transportation System," *IEEE Transactions on Intelligent Transportation Systems*, vol.22, no.8, pp.4919-4928, 2021. Article (CrossRef Link)
[25] G. Filomena, L. Kirsch, A. Schwering and J. A. Verstegen, "Empirical characterisation of agents' spatial behaviour in pedestrian movement simulation," *Journal of Environmental Psychology*, vol.82, 2022. Article (CrossRef Link)
[26] N. Molyneaux, R. Scarinci and M. Bierlaire, "Design and analysis of control strategies for pedestrian flows," *Transportation*, vol.48, no.4, pp.1767-1807, 2021. Article (CrossRef Link)
[27] I. Kravcovas, A. Vaitkus and R. Kleizienė, "Comparison of Pavement Performance Models for Urban Road Management System," *The Baltic Journal of Road and Bridge Engineering*, vol.15, no.3, pp.111-129, 2020. Article (CrossRef Link)
[28] M. Eom and B.-I. Kim, "The traffic signal control problem for intersections: a review," *European Transport Research Review*, vol.12, pp.1-20, 2020. Article (CrossRef Link)
[29] M. Tursun, T. Qugula, J. Ma, et al., "Data-driven Evaluation of Transit Efficiency at Intersection," *Science Technology and Engineering*, vol.23, no.34, pp.14857-14864, 2023. Article (CrossRef Link)
[30] J. Ko, H. W. Cho, J. I. Kim, et al., "Intelligence in traffic simulation model: Modeling congested network," *Journal of Intelligent & Fuzzy Systems*, vol.40, no.4, pp.7917-7923, 2021. Article (CrossRef Link)
[31] L. Chen, J. Li, Y. Li and J. Yao, "Comprehensive Evaluation of Operational Efficiency of Intersections in Arterial Considering Pedestrians Yield Rule," *Journal of Advanced Transportation*, vol.2022, 2022. Article (CrossRef Link)
[32] L. Tang, Y. Liu, J. Li, R. Qi, S. Zheng, B. Chen and H. Yang, "Pedestrian crossing design and analysis for symmetric intersections: Efficiency and safety," *Transportation research part A: policy and practice*, vol.142, pp.187-206, 2020. Article (CrossRef Link)
[33] M. Barthauer and B. Friedrich, "Evaluation of presorted and presignaled intersections with respect to traffic efficiency and traffic safety," *Transportation Research Procedia*, vol.47, pp.307-314, 2020. Article (CrossRef Link)
[34] J. Fabianova, P. Michalik, J. Janekova, et al., "Design and evaluation of a new intersection model to minimize congestions using VISSIM software," *Open Engineering*, vol.10, no.1, pp.48-56, 2020. Article (CrossRef Link)
[35] PTV Group, PTV Vissim 7 User Manual, Karlsruhe, Germany: PTV AG, 2014. [Online]. Available: https://pdfcoffee.com/qdownload/vissim-7-manual-transfer-ro-25nov-ccb8e5-pdf-free.html
[36] A. J. Miller, "Settings for Fixed-Cycle Traffic Signals," *Journal of the Operational Research Society*, vol.14, no.4, pp.373-386,1963. Article (CrossRef Link)
[37] S.-B. Cools, C. Gershenson, and B. D'Hooghe, "Self-Organizing Traffic Lights: A Realistic Simulation," *Advances in Applied Self-Organizing Systems*, pp.45-55, 2013. Article (CrossRef Link)
[38] L.Y. Deng and W. T. He, "Research on fuzzy control method of single intersection traffic signal," *Foreign Electronic Measurement Technology*, vol.37, no.4, pp.83-86, 2018. Article (CrossRef Link)

Yang Liu was born in 1984. She received the Ph.D. degree in computer software and theory from the University Electronic Science and Technology of China, Chengdu, China, in 2013. Her research interests include formal verification, information safety, data processing, and fault-tolerant control.

Jiaojiao He was born in 2000 and graduated from Chongqing Three Gorges College in 2023 with a bachelor's degree in Information and Computing Science. She is currently pursuing her master's degree in the School of Information Science and Engineering at Chongqing Jiaotong University. Her research interests are formal verification and intelligent transportation control.

Chen Jingwei was born in 1984. He received the Ph. D. degree from University of Chinese Academy of Sciences in 2013. His research interests include Theory, Algorithms, and Application of Lattices, especially the lattice-based cryptography and homomorphic encryption.

Bo Mi was born in 1982. He received the Ph.D. degree in computer system architecture from Chongqing University, China, in June 2009. Since 2011, he has been a Professor with the College of Information Science and Engineering, Chongqing Jiaotong University, China. His current research interests include intelligent transportation, vehicular ad-hoc networks, and cryptography.

Ling Zhao received the B.S., M.S., and Ph.D. degrees in mechanical and electronic engineering from Chongqing University, in 2002, 2005, and 2010, respectively. Since 2020, she has been a Professor with the College of Information Science and Engineering, Chongqing Jiaotong University, Chongqing, China. Her research interests include fault diagnosis and fault-tolerant control of dynamic systems, fault diagnosis of complex systems, and reliability engineering.

Xinyu Liu was born in 1999 and graduated from Chengdu Neusoft University in 2022 with a bachelor's degree in Information Engineering. He is currently pursuing his master's degree in the School of Information Science and Engineering, Chongqing Jiaotong University. His research interests include intelligent musical arrangement and information security.

Linhan Yang was born in 2000, a native of Wanzhou, Chongqing, and a master's student at Chongqing Jiaotong University, with main research interests in information security and ciphertext computing.