State-driven Priority Scheduling Mechanisms for Driverless Vehicles Approaching Intersections

Kailong Zhang*,†, Member, IEEE, Dafang Zhang*, Arnaud de La Fortelle†, Member, IEEE, Xiao Wu*, and Jean Grégoire†

*School of Computer, Northwestern Polytechnical University, 710072, Xi’an, China
†Robotics Centre, Mines ParisTech, 75006, Paris, France

Abstract—Scheduling driverless vehicles with different priorities to pass through intersections efficiently and safely has been becoming an important Passing-Through Intersection (PTI) problem in the field of Intelligent Traffic System (ITS). Considering new emerging features with possible priorities, a novel centralized priority scheduling mechanism is mainly explored in this study. First, related pivotal aspects of environment and driverless vehicles are modeled by fusing their physical and kinematic characters. Based on these models, PTI-related motions are further abstracted as several reservation-oriented standard states and actions. Then, an event-triggered and state-driven autonomous control procedure is designed. By mapping vehicular relations in spatiotemporal domain into time-distance windows, a universal passing-through principle, rules and priority-based scheduling mechanisms are proposed and described in detail. Finally, a priority scheduling algorithm sPriorFIFO is proposed and designed. These models and mechanisms are then implemented within an algorithm simulator, through which scheduling performances are verified and evaluated.

Index Terms—Intelligent Traffic System (ITS), Driverless vehicle, Intersection, Critical section (CS), Reservation, State-driven, Priority scheduling.

I. INTRODUCTION

In the last two decades, considerable interest to the field of Intelligent Traffic System (ITS) has been noted, and several prominent studies have been conducted to make vehicles and traffic spaces more and more autonomous and intelligent [1] [2]. By employing intelligent embedded systems equipped with novel technologies, e.g. environment sensing, intelligent recognition and control, Global Positioning System (GPS) and onboard digital map, driverless vehicles have been coming to the real life, typically as the Google driverless car, Stanley of Stanford university and European CyberCars [3]. Compared with regular vehicles, driverless vehicles are more convenient and safer because they are more autonomous and driven by intelligent pilot systems instead of human drivers, whose distraction or misjudgment is considered as the leading cause of over 90% of accidents [4]. Such vehicles are envisioned to provide autonomous and convenient services that will advance the development of ITS and innovate future lifestyle.

In current studies on ITS, the autonomous cooperation problem among driverless vehicles has been becoming increasingly significant and receiving considerable research attentions; such research involves the scheduling of vehicles when Passing-Through Intersections (PTI) [5] [6] as well as vehicular coordinations on roads [7]. And, scheduling driverless vehicles with different priorities to pass through intersections is one another branch of the common PTI problems that has been extensively studied recently [8]. Focusing on such novel competitive problems, this paper explores the essence of such problems and studies corresponding reservation-oriented priority scheduling mechanisms. After analyzing related traffic phenomena, the unified essence of them is abstracted as "the competitive reserving and optimal utilizing Critical Sections (CSs)". Based on this abstract, we carried out our study and contribute in the following aspects: (1) Related Traffic models of traffic objects, mainly covering the pivotal aspects of the environment and of driverless vehicles, are firstly established by fusing the physical and kinematic characters of these objects; (2) An event-triggered and state-driven control mechanism is proposed, in which a set of reservation-based actions and vehicular states are defined to present possible PTI-related vehicular behaviors; (3) After mapping vehicular relations in the spatiotemporal domain into new relations of time-distance windows, a universal reservation-oriented priority scheduling mechanism is proposed, including the relevant passing-through principle and some vital rules; (4) Based on the aforementioned works, a priority scheduling algorithm sPriorFIFO is designed and implemented within a traffic simulator, through which all proposed methods are simulated and scheduling performances are verified and evaluated.

The rest of this paper is arranged as follows. Section 2 presents the problem and related works. In Section 3, related traffic objects and vehicular states are abstracted and modeled. In Section 4, a universal priority-based passing-through principle and reservation mechanisms are presented in detail. A priority scheduling algorithm is proposed in Section 5 and the simulated verification results are analyzed in Section 6. Finally, conclusions are drawn and plans for future work are provided in Section 7.

II. BACKGROUND

A. Problem Statement

Intersections located at regular crossroads or T-junctions are regarded as typical scarce resources that contain several permanent CSs, indicated as static CSs(S-CSs), where traffic
flows must be well organized by rules. As shown in Fig.1, at the intersection of four bidirectional roads (E, S, W, and N), each road has three legal travel directions: left, straight and right. Thus, disregarding size limitation, all allowed paths and CSs can be abstracted as shown in Fig.2, in which CS$_4$ to CS$_{19}$ are S-CSs, while CS$_0$ to CS$_3$ are four conditional sections wherein conflict depends on whether each path has an independent exit. Under such circumstance, traditional PTI problems occur when vehicles pass through these CSs along different paths, such as $<$e$_1$, n$_1$,>, $<$e$_2$, w$_4$,>, and $<$w$_1$, s$_1$,> at CS$_5$, CS$_{17}$, and CS$_{19}$ separately in Fig.1. Obviously, organizing these vehicles to pass through competitive sections in an orderly and safe manner is the key aspect to solving the PTI problem. Recently, several scheduling methods, such as the traffic light and managements based on Vehicle to Agent(V2A) or Vehicle to Infrastructure(V2I) models, have been studied [9].

At present, situations are becoming increasingly complex in the envisioned intelligent traffic scenes, where driverless vehicles are used in diverse situations and exhibit different urgency degrees [8] [10] called priorities, such as ambulances with high(H) priority, noted as vehicles(H), shuttle vehicles with medium(M) priority (vehicles(M)), and personnel vehicles with low(L) priority (vehicles(L)) etc. Under such situations, the PTI problem will differ from the traditional ones. Thus, when two vehicles with different priorities compete for a CS, the vehicle with higher priority must be guaranteed to safely pass through in prior, such as s$_2$ and w$_4$ in Fig.1. However, via current time-based scheduling methods, such as First-In-First-Out(FIFO) and Queue scheduling [11], vehicles are always delayed by those vehicles in front of them because such methods mainly depend on vehicle arriving time or the length of queues rather than on urgency degree. Typically, a vehicle with high priority is blocked by vehicles with low priorities in front of it. We call this phenomenon Vehicular Priority Inversion(VPI) in our research. Hence, the connotations to solve this new PTI problem should cover the following aspects: (1) How to enable a traffic Agent to discern the statuses of arriving vehicles; (2) How to authorize vehicles with higher priorities to pass through in prior; (3) How to eliminate the VPI phenomena at CSs as effectively as possible. Considering these key issues, our research is conducted and explained in the following sections.

B. Related Work and Start-of-art

Numerous works have recently contributed to make vehicular embedded systems be real-time and smart, which are key foundations for intelligent behaviors. In particular, Milanés et al. [12] proposed a Cooperative Adaptive Cruise Control(CACC) system, in which they employed Vehicle to Vehicle(V2V) communication to acquire augment sensor data, and two controllers to manage vehicular approaching maneuver and regulate vehicle-following respectively, which improve vehicle intelligence in formation travel. Gonzalez et al. [13] presented a new control architecture for CyberCars in Cybernetic Transportation Systems(CTS). In this architecture, the decision-making logic is presented based on multiple driving modes and mainly involves global and local planning stages. Rahul et al. [14] proposed an enumerative behavior-based planning and decision-making algorithm without inter-vehicle communication. This algorithm can be considered as a step towards achieving autonomous traffic with both autonomous and non-autonomous vehicles. Furda et al. [15] regarded driverless city vehicles as safety-critical objects, and mainly studied a two-stage real-time decision-making method to improve road safety in city traffic, wherein data from a world model and a path planner are fused to enhance accuracy of decision. In their recent work, Furda et al. [16] optimized the pivot real-time decision-making issue for autonomous city vehicles, which can select a most appropriate maneuver with multiple criteria decision-making.

For the PTI problem, considerable works have been launched from the perspective of V2A/V2V cooperation [4] [17]. Laurent et al. [3] proposed a Partial Motion Planner(PMP) algorithm with safety constraints and an optimized environment perception mechanism to drive CyberCars through intersections safely and autonomously, after coupling perception, planning, and V2V communication capabilities. Biswas et al. [4] generalized the Cooperative Collision Avoidance(CCA) problem and its implementation requirements in the context of a V2V wireless network, and proposed a communication mechanism for CCA to assist drivers to react emergency situations in a timely manner. After decomposing the scheduling autonomous vehicles to pass several adjacent
intersections into several isolated control problem with V2I. Fei et al. [11] proposed an Efficient Branch and Bound scheduling approach, and in the corresponding algorithm the average queue length and average waiting time are employed to improve the whole passing efficiency. Ismail et al. [18] proposed an iCACC tool to manage intersection and vehicle trajectory adaptively by using CACC, thus optimizing vehicle speed profiles to minimize delay and prevent crashes. Olivier et al. [19] presented centralized supervised reservation systems for crossroads with a proposed algorithm to determine the trajectories and speeds of all driverless vehicles approaching intersections. Li et al. [20] studied a cooperative driving mechanism at blind crossings with a proposed concept of safety driving patterns. With a group vehicular group communication, the traffic efficiency was promoted. Huang et al. [6] designed a reservation-based approach to make intersection control more intelligent and efficient based on [21], by taking advantages of vehicular networks and introducing new features of the real-world driving environment. Chanwoo et al. [22] presented a new algorithm to control traffic flow, balance flow efficiency and fairness among driverless vehicles, particularly by employing the concepts of IEEE 802.11 DCF/PCF mechanisms.

Moreover, some thoughts of Agent and MAS have been adopted to the autonomous PTI problem [23]. Dresner et al. [24] proposed a novel reservation-based multi-agent approach to alleviate traffic, particularly at intersections. By employing a strong agent driver and powerful interaction capabilities, this approach efficiently promotes traffic productivity. Ismail et al. [25] modeled driverless vehicles as autonomous agents and controllers as manager agents, respectively, and proposed a heuristic optimization algorithm at intersections to improve scheduling efficiency and safety. Mladenovic et al. [10] proposed a V2V cooperative and self-organizing control framework that enables driverless vehicles to adjust their trajectories intelligently based on a dynamic priority principle while approaching intersections. Kailong et al. [26] modeled a state-driven passing-through mechanism by combining the physical and kinematic properties of both the environment and driverless vehicles, and proposed centralized reservation-oriented scheduling algorithm. Fenghui et al. [27] simulated dynamical fleet planning of driverless vehicles of CTS with multi-agent system theories, and then proposed a planning algorithm to promote cooperation among such vehicles. In practice, most of these studies are typically verified via simulation methods, typically [25] and [28].

As analyzed above, V2V/V2A cooperation and Agent/MAS mechanisms have been increasingly employed to current studies in ITS domain, and these studies on the traditional PTI problem have proposed several possible solutions, to improve traffic safety and efficiency. However, few studies have focused on scheduling vehicles with different priorities and employed a state-driven reservation method. In particular, Mladenovic et al. [10] carried out a similar study, proposing a priority-related passing-through mechanism. It mainly considered the intelligent adjustment of vehicular trajectory and cooperation between two conflicting vehicles, instead of improving the passing efficiency of vehicles with different urgent degrees we concerned in this paper. Therefore, based on these studies and our previous work in [26], we further proposed a novel state-driven and reservation-oriented priority scheduling method, taking both advantages of the flexibility of FIFO and passing efficiency of Queue scheduling, and ultimately implement these mechanisms within a graphical traffic simulator.

### III. MODELS AND DEFINITIONS OF TRAFFIC OBJECTS

#### A. Lane, Path, and Trajectory

Lane, path, and trajectory are three fundamental objects that are environment and motion related. In our research, we first introduce a concept of piecewise lane, indicated as $\ell$, that is inseparable with constant width and doesn’t have any inflection point. This concept employs $\ell$ to present the current lane for each vehicle. Then, we define a set $\zeta_i^\ell$ to parcel all lanes of Area $A_i$. Each lane should be reachable and constrained by the logic expression: $\forall \ell \in \zeta_i^\ell (\exists \ell' \in \zeta_i^\ell \rightarrow \ell \rightarrow \ell') \lor (\exists \ell' \in \zeta_i^\ell \rightarrow \ell \rightarrow \ell')$, where $\rightarrow$ indicates a directional reachable relation. Formally, we define the $k^{th}$ lane $\ell_k$ as: $< id = k, \theta, s, e, w, r, p >$, as shown in Table I. Based on this definition, the approximate geographic and traffic features of any lane can be presented. Furthermore, one path $\rho$ can be described as a series of connected lanes which satisfy the following attributes: $\ell, \ell, \ell \in \rho, \rho_j \subseteq \zeta_i^\ell \land \exists \ell' \left( (\ell \rightarrow \ell' \land \ell \rightarrow \ell' = \ell') \lor (\ell \rightarrow \ell' \land \ell' \rightarrow \ell' = \ell') \right)$. Meanwhile, a vehicular trajectory can be defined as a planned path with starting and ending time.

#### B. Intersection and S-CSs

A special critical traffic zone $\gamma_j (\varphi_j \subseteq A_i)$ generally has a set of entry and exit paths, indicated as $\zeta_j^p (\zeta_j^p \subseteq \zeta_j^\ell)$. This zone will form an intersection when the condition: $\zeta_j^p \neq \varnothing, \text{and} |\zeta_j^p| > 1$, $\exists \rho_m \in \zeta_j^p (\exists \rho_n \in \zeta_j^p \rightarrow \rho_m \times \rho_n \neq \varnothing)$ is satisfied, where $\rho_m$ and $\rho_n$ are two different paths in $\zeta_j^p$, and $\times$ is a logical cross operator. At an intersection, each place located on crossed lanes is regarded as a S-CS $\gamma$, that is, the smallest segment unit that can be allocated to only one vehicle at once time. Considering all such sections in a lane coordinate system, the $k^{th}$ section $\gamma_k$ can be parameterized as $: \{id = k, c, l, w, \rho, \rho \}$, as shown in Table I. For example, Fig.3(a) shows that the cross place of paths $\rho_1$ and $\rho_2$ will form a diamond critical section $\gamma$, with four vertices $< c_1, c_2, c_3, c_4 >$, edges $< c_1 c_2, c_2 c_3, c_3 c_4, c_4 c_1 >$, and $\zeta_j^p = \{ \rho_1, \rho_2 \}$ with a horizontal angle $\theta_{12}$. In reality, such definition is insufficient and $\gamma$ is only a theoretical CS, because when a vehicle on $\rho_2$ stays within this area, vehicles on $\rho_1$ are also forbidden in zones $z_1$ and $z_2$ aside from in $\gamma$. Consequently, we can further refine such critical regions. As an transformation of Fig.3(a), Fig.3(b) shows that $\gamma$ is refined into $\gamma_1$ and $\gamma_2$, with vertices $< c_2, c_6, c_4, c_8 >$ and $< c_2, c_7, c_4, c_5 >$, respectively. When two paths are vertical, $\theta_{12} = \frac{\pi}{2}$, $\gamma_1$ and $\gamma_2$ are all equal to $\gamma$. Fig.3(c) and Fig.3(d) show the solution to three crossing paths. This concept can be adapted to almost all CSs in intersections.
C. Vehicle Model and Its Reservation-oriented Actions

After abstracting physical and kinematic characters, we define a cooperative and adaptive model of a driverless vehicle \( v_i \) as:

\[
< s, \alpha, v, a, \omega >, \ 
< x_c, y_c, \hat{\theta}_c, \ell_c >, \ 
< x, y, \theta >,
\]

as described in Table I. This model covers basic mechanical features and necessary kinematic characteristics, wherein the elements, such priority and vector of CSs, are necessary to support our reservation-based passing-through mechanism. With the vehicular posture tuple \(< x, y, \theta >\), typical integral relations exist for \( \dot{x_c}, \dot{y_c} \)
and \( \hat{\theta}_c \), as shown in Formula(1), where \(< v.x_0, v.y_0, v.\theta_0 >\) are the initial parameters.

During maneuvering, vehicular motions transform along with changing situations; however, these motions are also certain in a space of finite states and actions. That is, a vehicle must be in a known state or action at anytime. After inducing characters of different typical motions, such as cruising, passing-through and so on, we introduce a unified action model \( \alpha \) that is presented as a tuple with five elements. The \( i^{th} \) action \( \alpha_i \) is defined as:

\[
< i, n, p, \zeta_c, M_c >,
\]

as shown in Table I, where \( M_c \) is a transition matrix of statuses. When \( \varepsilon_{ij} \) is set to 1, a valid transition exist from \( s_i \) to \( s_j \), and \( M_c \) is a condition matrix where each element \( c_{ij} \) presents a special transition condition or null when \( \varepsilon_{ij} \) is zero.

\[
\begin{align*}
\dot{\alpha}_{s} &= \alpha_{s} + \rho_{i} \\text{action shift}, \\
\dot{\alpha}_{s} &= \alpha_{s} - \rho_{i} \\text{action shift}, \\
\dot{\alpha}_{s} &= \alpha_{s} \\text{transition action}, \\
\dot{\alpha}_{s} &= \alpha_{s} \\text{no action},
\end{align*}
\]

Consecutively, we jointly define four fundamental actions in Table II, which involves concrete target conditions, six uniform executing statuses \( \{ s_1, ..., s_5 \} \), condition matrix and two transition matrices, as shown in Formula(2) and (3). Among these actions, \( \alpha_0 \) is a virtual action for initializing the electronic system during each power-up time, rather than

---

**TABLE I**

<table>
<thead>
<tr>
<th>Para</th>
<th>Definition</th>
<th>Para</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lane and Path</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( \ell.id )</td>
<td>identification number.</td>
<td>( \ell.0 )</td>
<td>horizontal angle of current position.</td>
</tr>
<tr>
<td>( \ell.s )</td>
<td>coordinates of starting position.</td>
<td>( \ell.e )</td>
<td>coordinates of end-position.</td>
</tr>
<tr>
<td>( \ell.w )</td>
<td>width of lane.</td>
<td>( \ell.r )</td>
<td>limitation rules (e.g. maximum/minimum speed).</td>
</tr>
<tr>
<td>( \ell.f )</td>
<td>curve or straight line functions.</td>
<td>( \rho_{1} )</td>
<td>( j^{th} ) path, a series connected lanes.</td>
</tr>
<tr>
<td>Static Critical Section</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( \gamma.id )</td>
<td>identification number.</td>
<td>( \gamma.c )</td>
<td>coordinate vector of all vertices.</td>
</tr>
<tr>
<td>( \gamma.l )</td>
<td>length of section.</td>
<td>( \gamma.w )</td>
<td>width of section.</td>
</tr>
<tr>
<td>( \gamma.c_k )</td>
<td>path set.</td>
<td>( \gamma.s )</td>
<td>status of section: available or forbidden.</td>
</tr>
<tr>
<td>Vehicle Model</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( v.id )</td>
<td>identification number.</td>
<td>( v.l )</td>
<td>vehicle length.</td>
</tr>
<tr>
<td>( v.w )</td>
<td>vehicle width.</td>
<td>( v.p )</td>
<td>vehicle priority.</td>
</tr>
<tr>
<td>( v.x_c )</td>
<td>( x ) coordinate of current position.</td>
<td>( v.y_c )</td>
<td>( y ) coordinate of current position.</td>
</tr>
<tr>
<td>( v.\gamma )</td>
<td>current horizontal angle.</td>
<td>( v.\ell )</td>
<td>current lane.</td>
</tr>
<tr>
<td>( v.s )</td>
<td>current state.</td>
<td>( v.\alpha )</td>
<td>current action.</td>
</tr>
<tr>
<td>( v.v )</td>
<td>current velocity.</td>
<td>( v.a )</td>
<td>current acceleration.</td>
</tr>
<tr>
<td>( v.\omega )</td>
<td>current angle speed.</td>
<td>( v.\theta )</td>
<td>differential shift at ( x ) direction.</td>
</tr>
<tr>
<td>( v.\gamma_y )</td>
<td>differential shift at ( y ) direction.</td>
<td>( v.\omega )</td>
<td>differential shift of angle.</td>
</tr>
<tr>
<td>( v.\gamma_c )</td>
<td>current occupied section.</td>
<td>( v.\gamma )</td>
<td>required CS.</td>
</tr>
<tr>
<td>Action Model</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( \alpha_i )</td>
<td>( i^{th} ) action.</td>
<td>( \alpha.id )</td>
<td>identification number.</td>
</tr>
<tr>
<td>( \alpha.p )</td>
<td>the terminate condition.</td>
<td>( \alpha.\zeta_i )</td>
<td>( { s_0, ..., s_{k-1} } ), ( s_i ) is ((i+1)^{th}) status of this action.</td>
</tr>
<tr>
<td>( \alpha.M_k )</td>
<td>action transition matrix.( M_k = [\varepsilon_{ij}]_{k \times k} )</td>
<td>( \alpha.\zeta_i )</td>
<td>( { s_0, ..., s_{k-1} } ), ( s_i ) is ((i+1)^{th}) status of this action.</td>
</tr>
</tbody>
</table>

---

**Fig. 3.** S-CSs on Multiple Paths.
driving a vehicle. \(\alpha_1\) is an important action that involves synchronously cruising and section reservation, the operation is a necessary precondition to the subsequent one. \(\alpha_2\) enables a vehicle to follow its lane without any reservation. All similar actions, whether cruise, acceleration/deceleration, or a temporary stop (equal to a waiting), can be induced to \(\alpha_2\). \(\alpha_3\) is the passing-through action that can only be performed when the corresponding required CSs have been reserved successfully. Then, the dynamic features of any action can be represented formally by an event-based status-transition graph. For clarity, the condition \(\text{POWER\_UP}\) indicates the control system is ready; \(\text{PARA\_READY}\) represents that all required arguments are prepared; \(\text{FIN}\) and \(\text{!FIN}\) indicate wether this action is completed; \(\text{EME}\) and \(\text{!EME}\) show if there’s an emergency traffic situation; \(\text{CMD\_CCL}\) denotes a cancel command, through which each action is ensured as interventional and controllable; \(\text{SYS\_FAU}\) indicates a fault condition. Meanwhile, \(a \rightarrow b\) denotes that the left variable \(a\) is approximated to the right constant \(b\); \(a \leftrightarrow b\) denotes a big deviation; \(a \Rightarrow b\) indicates \(b\) is at the front of \(a\); \(a \Rightarrow \neg b\) shows \(a\) is not behind \(b\).

\[
M_{s_0} = \begin{pmatrix}
0 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}
\tag{2}
\]

\[
M_{s_1} = \begin{pmatrix}
0 & 1 & 0 & 1 & 0 & 1 \\
0 & 1 & 1 & 1 & 1 & 1 \\
0 & 1 & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}
\tag{3}
\]

D. Typical PTI-related States

As mentioned earlier, an action can indicate current vehicular motion; however, all previous basic actions are insufficient to directly represent the entire behavior procedure, which includes special stages related to the decision-making progress. Hence, we introduce a higher-layer concept—vehicular state \(S\) to indicate the stage of vehicular behavior. Similar to the definition of actions, we formally define state \(S_j := \langle \text{id} = j, \zeta_i, M_S, M_C, \rangle\), where \(\zeta_i\) is the set of actions, \(\zeta_S^j\) is the set of possible target states, \(M_S\) and \(M_C\) are the global state transition matrix and the corresponding event matrix, separately. In this study, we define four necessary states: initial state \(S_0\), following state \(S_1\), passing-through state \(S_2\), and terminating state \(S_3\). \(S_0\) is the initial state that consists of \(\alpha_0\), and \(S_1\) is an \(\alpha_2\)-based frequent state switching among other states via event or action command \(\text{CMD\_ACT}\). \(S_2\) is a compound state that consists of \(\alpha_1\) and \(\alpha_3\), in which only the required CS has been reserved, thus allowing \(\alpha_3\) to continue. Through an event mechanism, a vehicular state can be automatically triggered to another by any newly generated event or command. Based on these definitions, PTI-related states and their transition relationships are further presented in Fig.4.

IV. STATE-DRIVEN PASSING-THROUGH MECHANISMS

A. A Universal ”Reserve in Advance, Act Later” Principle

Based on the aforementioned models, it’s clear that the success of all actions depends on the reservation of CSs. In this section, we present a ”Reserve in Advance, Act Later”(RAAL) principle, the main idea of which is to provide a universal passing procedure under the special precondition that all similar actions must obey despite having different features.

1) Precondition

a) Each intersection is supervised by a superintendent, called the \(\gamma\)-Agent;

b) All objects, including the vehicles and the \(\gamma\)-Agent, can sense and communicate with each others via real-time wireless messages;

c) Each reservation message from \(\upsilon_i\) includes at least one Passing-through Time Window \(PTW_i\), a tuple \([t_{i,s}, t_{i,p}]\), as defined in Table III;

d) The \(\gamma\)-Agent is the scheduling center in which several reservation queues \(Q\) are deployed to record reservation requests; \(Q_1, Q_2, ..., Q_m\) are for each lane of \(\upsilon_i\); A global queue \(Q^\gamma\) is employed to orderly store all requests of these queues.

2) A common passing procedure of a vehicle

a) When \(\upsilon_i\) is approaching a CS \(\gamma\), it changes its state to action \(\alpha_1\);

b) \(\upsilon_i\) sends a reservation message to \(\gamma\)-Agent and waits for a response within \(\alpha_1\); \(\gamma\)-Agent checks if such request is acceptable and then responds with a message;

c) \(\upsilon_i\) cruises along the current lane to the starting point of \(\gamma\), while autonomously following its predecessor within a safe distance until it gets an authorization called \(\gamma\)-token;

d) \(\upsilon_i\) passes through \(\gamma\) and releases this \(\gamma\)-token as soon as it leaves \(\gamma\).

Considering the basic concepts of RAAL, we clarified the kinematic statuses of a passing-through procedure. In Fig.5, all necessary special positions, velocities, accelerations, and their relationships are presented. All these parameters as described in Table III can be flexibly assigned for specific situations.

B. Expansion and Refinement of PTW

To describe clearly the spatiotemporal constraints on diverse CSs, the concept of \(PTW\) is redefined in our present work. In current studies, \(PTW\) is a predefined time window that is calculated once and always regarded as a constant.

\[
\begin{align*}
\text{PTW}_{\upsilon_i} &= \begin{cases}
\text{YES} & \text{if } t_{i,s} < t < t_{i,p} \\
\text{NO} & \text{otherwise}
\end{cases} \\
\end{align*}
\]

Fig. 5. Kinematic Models for a Vehicle during its Passing-through.
### TABLE II
DECLARATIONS OF ACTIONS AND CORRESPONDING RELATIONS

<table>
<thead>
<tr>
<th>α</th>
<th>Motion</th>
<th>α.p</th>
<th>α.ζ_α</th>
<th>α.M_α</th>
<th>α.M_δ</th>
</tr>
</thead>
<tbody>
<tr>
<td>α₀</td>
<td>Init</td>
<td>Power-up init, Self check, Bind a mission.</td>
<td>s₀: Start s₁: Run (Init) s₂: Suspend s₃: Cancel s₄: Finish s₅: Fault</td>
<td>M_α₀</td>
<td>c₁₀: POWER_UP c₁₁: FIN c₁₄: FIN c₁₅: SYS_FAU others: null</td>
</tr>
<tr>
<td></td>
<td>Cruise with reservation</td>
<td>&lt; ℓ₁, x₁, y₁, v₁, a₀₁, γ₁ &gt;</td>
<td>s₀: Start s₁: Run (cruise and reserve) s₂: Suspend s₃: Cancel s₄: Finish s₅: Fault</td>
<td>M_α₁</td>
<td>c₁₀: PARA_READY c₁₀, c₁₅, c₂₅: SYS_FAU c₁₁: FIN:=PARA_UPDATE</td>
</tr>
<tr>
<td>α₂</td>
<td>Cruise</td>
<td>&lt; ℓ₂, x₂, y₂, v₂, a₀₂ &gt;, &lt; x₂, y₂ &gt; is the end point.</td>
<td>s₀: Start s₁: Run (Cruise) s₂: Suspend s₃: Cancel s₄: Finish s₅: Fault</td>
<td>M_α₂ = M_α₁</td>
<td>c₁₀: PARA_READY c₁₀, c₁₅, c₂₅: SYS_FAU c₁₁: FIN:=PARA.UPDATE</td>
</tr>
<tr>
<td>α₃</td>
<td>Passing</td>
<td>&lt; ℓ₃, x₃, v₃, a₃ &gt;</td>
<td>s₀: Start s₁: Run (Pass) s₂: Suspend s₃: Cancel s₄: Finish s₅: Fault</td>
<td>M_α₃ = M_α₁</td>
<td>c₁₀: PARA_READY c₁₀, c₁₅, c₂₅: SYS_FAU c₁₁: FIN:=(γ₃ ⇒ v) c₁₂, c₂₂: EME c₁₃, c₁₄, c₂₃: CMD_CCL c₁₄: FIN:=(γ₃ ⇒ v) c₂₁: !EME others: null</td>
</tr>
</tbody>
</table>

---

Fig. 4. PTI-related States and Transition Relationships.

However, such definition can only be adapted to some ideal moments, such as vᵢ traveling from P_a to P_vᵢ without any interference or fault. This time window is unsuitable for all possible cases because a real traffic system is obviously a complex stochastic dynamic system wherein the worst PTW cannot be estimated exactly. Moreover, even if vᵢ has received a γ-token, determining whether it can pass within tᵢ,s still depends on whether the lane segment from P_γₑ to P_vᵢ,p is occupied, which is why we introduce γ-token. Hence, based on the relations in Fig.5, t₀ is assumed as the current reserving time; Δt(v₁, v₂) represents the time length required to increase velocity from v₁ to v₂; and Δd(v₁, v₂) is the corresponding travel distance, which is calculated via Formulas(4). Then we use Formulas(5) and Formulas(6) to...
TABLE III
IDENTIFICATIONS AND PARAMETERS DURING PASSING-THROUGH

<table>
<thead>
<tr>
<th>Para</th>
<th>Definition</th>
<th>Para</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v_{in}$</td>
<td>maximum cruising velocity on current lane.</td>
<td>$v_r$</td>
<td>maximum velocity during a reservation.</td>
</tr>
<tr>
<td>$v_r$</td>
<td>maximum velocity when passing through.</td>
<td>$a_a$</td>
<td>maximum acceleration.</td>
</tr>
<tr>
<td>$a_a$</td>
<td>absolute value of maximum deceleration.</td>
<td>$P_a$</td>
<td>position to adjust velocity to $v_r$.</td>
</tr>
<tr>
<td>$P_a$</td>
<td>position to send a reservation message.</td>
<td>$P_{ip}$</td>
<td>current position of $v_i$.</td>
</tr>
<tr>
<td>$P_b$</td>
<td>position from where $v_i$ should brake to stop, if it has no authorization to pass through.</td>
<td>$P_{ip}$</td>
<td>position where $v_i$ must stop for a $\gamma$-token.</td>
</tr>
<tr>
<td>$P_{ip}$</td>
<td>entering position of $\gamma$.</td>
<td>$P_{ic}$</td>
<td>end position of $\gamma$.</td>
</tr>
<tr>
<td>$P_{vi,p}$</td>
<td>left position of $v_i$ at $\gamma$.</td>
<td>$P_c$</td>
<td>possible special position where a vehicle switches from decelerate to accelerate.</td>
</tr>
<tr>
<td>$\Delta c$</td>
<td>minimum safe distance.</td>
<td>$\Delta c'$</td>
<td>minimum forward distance to $P_{ip}$.</td>
</tr>
<tr>
<td>$d_a$</td>
<td>distance for adjusting vehicle’s velocity from $v_{in}$ to $v_r$, $d_a \geq (v_{in}^2 - v_r^2)/(2a_a)$</td>
<td>$d_r$</td>
<td>distance for sending a reservation. $d_r \geq v_r \cdot \Delta t_c$</td>
</tr>
<tr>
<td>$d_{vi}$</td>
<td>distance between $P_{vi}$ and $P_b$.</td>
<td>$d_b$</td>
<td>braking distance. $d_b \geq (v_r^2)/(2a_d)$</td>
</tr>
<tr>
<td>$d_{\gamma}$</td>
<td>length of $\gamma$, and equal to $\gamma.l$.</td>
<td>$d_{(i-1)}$</td>
<td>distance between $v_i$ and $v_{(i-1)}$.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Passing-through Time Window</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$PTW_i$</td>
<td>$v_i$’s passing-through time window. $PTW_i = [t_{i,s}, t_{i,p}]$</td>
<td>$t_{i,s}$</td>
<td>expected minimum time for $v_i$ arriving at $P_{ip}$.</td>
</tr>
<tr>
<td>$t_{i,p}$</td>
<td>expected passing time length.</td>
<td>$t_{i,c}$</td>
<td>expected time $v_i$ leaves from $\gamma$.</td>
</tr>
<tr>
<td>$\Delta t_{i,s}$</td>
<td>real delay of $t_{i,s}$.</td>
<td>$\Delta t_{i,p}$</td>
<td>real delay of $t_{i,p}$.</td>
</tr>
<tr>
<td>$\Delta t_c$</td>
<td>maximum time for a Request-Response communication.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Queue and Delay Time</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$Q_i$</td>
<td>queue in the $\gamma$-Agent to record reservations on $\gamma$.</td>
<td>$Q^+$</td>
<td>global queue in the $\gamma$-Agent.</td>
</tr>
<tr>
<td>$\Delta t_{(i+1)i}$</td>
<td>time distance between $v_{(i+1)}$ and $v_i$.</td>
<td>$D_a$</td>
<td>ideal global delay time for $v_i$.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Evaluation</th>
<th>Formula</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>$PTW_i$ for $v_i$ when its acceleration is changed. If $(d_{vi} + d_b + \Delta c') \geq \Delta d(v_{i,v}, v_r)$, then Formula(5) is valid, otherwise Formula(6). The ideal $PTW_i$ can be calculated by using Formula(7), which can only be used for the first reservation. While traveling, $v_i$ recalculates and updates its $PTW$ at the moment when its acceleration is changed.</td>
<td>$\begin{cases} t_{i,s} = t_0 + \frac{v_i \cdot (v_{i,s})}{a_a} \ t_{i,p} = \Delta t(v_i, v_{(i,s)}, v_r) + \frac{d_{\gamma} + v_{i,l} - d_i(v_i, v_{(i,s)}, v_r)}{v_r} \ v_{i,v}(t_{i,s}) = \sqrt{v_{i,v}^2 + 2a_a \cdot (d_{vi} + d_b + \Delta c') - v_{i,v}} \end{cases}$</td>
<td></td>
</tr>
</tbody>
</table>

\[
\begin{align*}
\Delta t(v_1, v_2) &= \frac{v_2 - v_1}{a_a} \\
\Delta d(v_1, v_2) &= \frac{v_2^2 - v_1^2}{2a_a} \\
\end{align*}
\]

\[
\begin{align*}
t_{i,s} &= t_0 + \Delta t(v_i, v_r, v_r) + \frac{d_{vi} + d_b + \Delta c' - \Delta d(v_i, v_r)}{v_r} \\
t_{i,p} &= \Delta t(v_r, v_r) + \frac{d_{\gamma} + v_{i,l} - \Delta d(v_r, v_r)}{v_r} \\
\end{align*}
\]  

In addition, vehicles on the same lane must travel in sequence and with a safe distance between them as prescribed by assistant systems [29], which have been also modeled as car-following models in traffic simulators [30]. Hence, competitions mainly exist among vehicles on different lanes. With the segment-based kinematic model as shown in Fig.5, we employ Rule(1) to guarantee a safe following. Then, assuming that all initial velocities and accelerations are equal to $v_r$ and zero, respectively, $v_i$ must begin to accelerate with $-a_d$ when the time length does not exceed $(d_{vi} - (\Delta c + \Delta c'))/v_r$. Meanwhile, we set Rule(2) as a precondition for vehicles on the same lane.
Rule 1: At any time, the distance between $v_i$ and its follower $v_j$ on the same lane, $d_{ij}$, should be at least $(\Delta c + \Delta c')$.

Rule 2: $\forall v_i, v_j \in Q_i(\forall v_j, v_j \in Q_i \& i \neq j \rightarrow PTW_i \cap PTW_j = \emptyset)$.

C. Potential Relationships in the Time Domain

Introducing PTW is meaningful to convert competition problems from the spatial domain to spatiotemporal domain. From a real traffic procedure, we know that potential time delays of $v_i$ are mainly caused by its direct predecessors on the same lane and indirect predecessors from neighbouring lanes. For direct predecessors, the pressure to maintain a safe distance results in some time delay because explicit overlaps among PTWs are induced when two vehicles are too close. To formulate such delay uniformly, we impor time distance $\Delta t_{ij}$ to indicate an explicit or potential competition between $v_j$ and its follower $v_i$ in $Q_i^+$. The ideal relative value of $\Delta t_{ij}$ is $(t_{i,s} - t_{j,e})$. Hence, any initial collision is evident and can be detected via Rule 3 as follows. For instance, Fig.6 shows an example of the PTW relationships of three crossed lanes. When $\Delta t_{31} < 0$, an overlap between $PTW_1$ and $PTW_3$, namely a collision between $v_1$ and $v_3$.

Rule 3: $\forall v_i, v_j \in Q_i(\exists v_j, v_j \in Q_k \& k \neq i \rightarrow \Delta t_{ij} < 0)$, then a collision occurs between $v_i$ and $v_j$.

The second important factor is the potential collision caused by the accumulation of time distances with gaps to all vehicles ahead of one vehicle in $Q^+$, called vehicular global delay time and indicated as $D$. Assuming that $v_0$ is the last vehicle left from $\gamma$ and at the front of vehicles in $Q^+$, the vehicles don't initially exist $v_0$, and $v_i$ is the $m^{th}$ and $j^{th}$ vehicle in queue $Q_i$ and $Q^+$ respectively, then $D_1$ can be calculated by Formula (8). If $D_1$ is positive, there's a potential collision between $v_i$ and its predecessors in $Q^+$, in this case $v_i$ can enter this applied section at the time $(t_{i,s} + D_1)$. If $D_1$ is negative, then the amount of free time intervals ahead of $v_i$ is sufficient to accommodate all overlapping requirements. Moreover, by considering stochastic environmental factors that lead to changing vehicular velocity and acceleration, we introduce two variables $\Delta t_{i,s}$ and $\Delta t_{i,p}$ to present the delays for $t_{i,s}$ and $t_{i,p}$ separately. Both real delays are countable, but not computable. Furthermore, we can employ $t'_{i,s}, t'_{i,p}$, and $D_i$ to indicate the real values of $t_{i,s}, t_{i,p}$ and $D_i$, as shown in Formula (9). This is valuable for both evaluating all probable delays in real time and to judge whether one vehicle is blocked.

\[
D_i = D_1 + \sum_{j=1}^{i-1} (-\Delta t_{j+1}), \\
D_1 = \begin{cases} 
- (t_{i,s} - (D_0 + D_0)), & \text{when } D_0 \geq 0 \\
- (t_{i,s} - t_{e,c}), & \text{when } D_0 < 0 
\end{cases} \\
D_0 = \begin{cases} 
0, & \text{when } v_0 = v_k \\
D_k, & \text{when } v_0 = v_k 
\end{cases} \\
D_i' = D_i + \Delta t_{i-1,s} + \Delta t_{i-1,p}, \\
D_i' = \begin{cases} 
0, & \text{when } v_0 = v_k \\
D_k, & \text{when } v_0 = v_k 
\end{cases} \\
D_i'' = D_i + \Delta t_{i,s} + \Delta t_{i,p},
\] (8) (9)

D. Passing-through Mechanisms in $S_2$

According to the procedure in Fig.5, the passing-through procedure for $S_2$ can be presented as the following steps, which cover four motion stages and employ wireless messages defined in Table IV for consultations.

a) Velocity-adjusting: From $P_a$, $v_i$ adjusts its velocity from $v_m$ to $v_r$ with accelerations $a_{\alpha}$ or $-a_{\alpha}$;

b) $\gamma$-reserving: When arriving at $P_r$, $v_i$ sends a RES message to $\gamma$-Agent, while autonomously following its predecessor with a velocity not exceeding $\gamma_v$; $\gamma$-Agent then replies an acceptance message REG($\gamma$-token) or a rejection REJ according to the status of the requested section $\gamma$;

c) Braking: If $v_i$ doesn't receive a $\gamma$-token until arrived at $P_b$, it brakes with a deceleration not less than $-a_d$ to guarantee that $v_i$ will stop at the front of $P_{sy}$, and wait until be authorized;

d) Passing-through: When $v_i$ obtains a $\gamma$-token, it accelerates to the passing velocity $\gamma_v$ to pass through $\gamma$, sending PASS messages periodically; At the moment when $v_i$ leaves $P_{sy}$, it sends a REL message to release $\gamma$.

During this procedure, if any kinematic parameter that incurs a delay is changed, $v_i$ will send a REJ message to update its $PTW$. If one vehicle stops, it will broadcast WAIT messages periodically, warning its successors to avoid rear-end collisions. At worst, when a vehicular failure occurs, FAU messages will be broadcasted periodically. $\Delta p_1$ and $\Delta p_2$ represent variable sending periods of corresponding messages. $\Delta p_1$ is only effective before the sender obtains a token. In such a consultation procedure above, the communication protocol and procedure are also covered.

V. sPriorFIFO: A PRIORITY SCHEDULING ALGORITHM

FIFO is a common and classic scheduling method in current research. The flow of a centralized FIFO scheduling can be described as that $\gamma$-Agent queues all received requests in corresponding queues $Q_i$ and $Q^+$ with an ascending sequence of $t_{i,s}$, and then scheduling them in an orderly manner. Obviously, this method is effective in many situations. However, such method is significantly limited such that it can not function effectively under special situations, such as vehicles with different priorities or in traffic jams. Thus, to promote scheduling capability, we propose the new priority-based FIFO scheduling algorithm, called sPriorFIFO.
TABLE IV
DEFINITIONS OF COMMUNICATION MESSAGES

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Format</th>
<th>Type</th>
<th>Period</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>RES</td>
<td>&lt;v.id, v,x,v.y,v.p, v.γ,PTW, RES &gt;</td>
<td>V2A</td>
<td>ΔP1</td>
</tr>
<tr>
<td>2.</td>
<td>REN</td>
<td>&lt;v.id, v,x,v.y,v.p, v.γ,PTW, REN &gt;</td>
<td>V2A</td>
<td>Sporadically</td>
</tr>
<tr>
<td>3.</td>
<td>WAIT</td>
<td>&lt;v.id, v,x,v.y,v.p, WAIT &gt;</td>
<td>V2A</td>
<td>ΔP2</td>
</tr>
<tr>
<td>4.</td>
<td>REP</td>
<td>&lt;v.id, v,γ, REG</td>
<td>A2V</td>
<td>Once</td>
</tr>
<tr>
<td>5.</td>
<td>PASS</td>
<td>&lt;v.id, v,γ, PASS</td>
<td>V2A</td>
<td>Once</td>
</tr>
<tr>
<td>6.</td>
<td>REL</td>
<td>&lt;v.id, v,γ, REL</td>
<td>V2A</td>
<td>Once</td>
</tr>
<tr>
<td>7.</td>
<td>FAU</td>
<td>&lt;v.id, v,x,v.y,v.γ, FAU &gt;</td>
<td>V2A</td>
<td>Once</td>
</tr>
</tbody>
</table>

A. Priority Inheritance-based Decision-making in γ-Agent

For CS γ_i, γ-Agent owns its corresponding queues: Q_l(l ∈ 1, 2, ..., m) and Q^+. As constrained in Fig.5, all reservation messages will be submitted to γ-Agent only when vehicles have passed the position P_l, thus enabling γ-Agent to sequence all approaching vehicles into a corresponding queue according to their request moments or expected t_k,s. In response to each reservation, γ-Agent replies a REP message with REG. In addition, γ-Agent collects renewing messages continuously to update the PTWs of all coming vehicles in real time, thus reflecting an up-to-date traffic situation.

The main concept of the proposed sPriorFIFO is to reduce the delay times for all urgent vehicles as much as possible. To describe the design of sPriorFIFO, we first define Rule(4) to ensure that γ-Agent can select vehicles with higher priorities from all competitive vehicles to pass through. However, a pivotal problem exists, in which one vehicle with a high priority is always delayed by the vehicles with lower priorities in front of it, called the Vehicular Priority Inversion (VPI) traffic phenomena. To resolve this problem, we propose a Gradient Priority Inheritance (GPI) method as below.

Rule(4): For each queue Q_l(l ∈ 1, 2, ..., m), if there exists v_i, (i > 1), with a high priority and D_i > 0, then γ-Agent will repeatedly tackle all its predecessors from v_{i-1} one by one via Rule(5), until one vehicle with a zero delay or the head of Q_l is reached. In particular, when the distances between continuous neighboring vehicles at the front of v_i are all equal to (Δc + Δc̃), the priorities of these vehicles will be promoted simultaneously.

Rule 4: If v_i and v_j are head vehicles of Q_l and Q_m respectively, then the vehicle with a higher priority will be authorized first when Rule(3) is satisfied.

Rule 5: Assume that v_i, (i > 1), is the first vehicle with a high priority found in Q_l, and v_{i-1} is its predecessor. If D_i > 0, then v_{i-1}.p will be promoted to v_i.p.

In particular, when v_i encounters a fault to stop, its priority will never be promoted until it recovers. And, if v_i holds a γ-token, it must give up its privilege. When a failure occurs, v_i broadcasts fault messages periodically to γ-Agent. In this case, γ-Agent will change PTW_{i} to (−1, −1), thus indicating v_i and all its successors on the same lane cannot be scheduled.

B. Single Authorization and Batch Authorization Policies

Based on the principle of sPriorFIFO above, new authorization policies can be further employed to schedule all queued vehicles. We design a Single Authorization Policy (SAP): When γ-Agent has a free γ-Token, it only grants this token to a suitable and schedulable vehicle according to FIFO and Rule(4). SAP can guarantee priority scheduling and safety because only one vehicle is allowed to enter a section at any time. However with regard to performance, such policy is insufficient because it introduces possible unnecessary delays. For example, m vehicles on the same lane are approaching a CS. Via SAP, one vehicle may decelerate, even stop, at the front of P_{γ} to wait for the unique γ-Token released by its predecessor. However, one vehicle should be able to start passing as long as Rule(1) is satisfied. In the worst-case scenario, when all subsequent vehicles stop to wait for a token, the introduced delay time will be at least (m−1)(d − Δc)/v_γ.

As an improvement, we propose a batch authorization policy αSAP as: (1) The continuous schedulable successors of the head vehicle in Q^+ can be authorized once as long as they are all on the same lane. (2) On each lane, all continuous schedulable vehicles can be authorized when all distances between neighbors don’t exceed (Δc + Δc̃), which is an adjustable factor for a short distance. Obviously, the main optimization of αSAP is trying best to reduce the stop-wait time for possible continuous vehicles on the same lane. Furthermore, to improve the flexibility of αSAP, we employ a variable Amax(αSAP ≥ 0) to adjust the maximum number of authorization at once time. When Amax is set to 1, αSAP degenerates to SAP. When Amax is greater than 1, all authorized vehicles will pass through autonomously under the constraint of Rule(1). For γ-Agent, merely when all these vehicles have passed through, it will launch the next authorization. A special situation, that is when one of the authorized vehicles is at fault, this vehicle must notify its successors that have been authorized to release their tokens. Fatally, when a failure happens to a vehicle in a passing state, a complete traffic break will occur at this section until the failure is removed manually.

VI. SIMULATION AND EXPERIMENTS

A. Scenarios and Parameters Set

To verify the proposed models and algorithms, we initially implement them within a scheduling simulator, in which vehicles are randomly generated with Poisson Distribution, described as Formula(10) [31], and the Krauss car-following model [30] is adopted. During experiments, the variable k is set to 1, t is not smaller than 1(second), and λ is assigned within [0.15, 0.2], all these values can support a higher vehicle density that is more meaningful to verify our proposed mechanisms than that with smaller λ, where the performance of sPriorFIFO will become indistinctive because urgent vehicles are few blocked. Meanwhile, the sequence of arriving vehicles are generated randomly by the constraints of Rule(1) and (2). Thus, any vehicle approaching an intersection will be first presented as a vehicular model with a PTW, which implies the relationship among d_{vi}, v_i.v and v_i.a, as shown in Formula(4)
TABLE V
EXPERIMENT PARAMETERS

<table>
<thead>
<tr>
<th>Para</th>
<th>Value</th>
<th>Para</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>lanes</td>
<td>2</td>
<td>priorities</td>
<td>H, M, L</td>
</tr>
<tr>
<td>v_L</td>
<td>5.6 m/s</td>
<td>v_m</td>
<td>10 m/s</td>
</tr>
<tr>
<td>a_L</td>
<td>5 m/s</td>
<td>a_m</td>
<td>10 m/s</td>
</tr>
<tr>
<td>v_L</td>
<td>4 m/s</td>
<td>v_m</td>
<td>12 m</td>
</tr>
<tr>
<td>d_1</td>
<td>4 m</td>
<td>d_5</td>
<td>6 m</td>
</tr>
<tr>
<td>d_L (&lt;= γ_L)</td>
<td>3.2 m, 6.4 m, 9.6 m</td>
<td>Δc + Δc'</td>
<td>4 m</td>
</tr>
<tr>
<td>p_h</td>
<td>&lt; 5%</td>
<td>p_m</td>
<td>&lt; 12%</td>
</tr>
<tr>
<td>p_l</td>
<td>≥ 83%</td>
<td>Δp_c</td>
<td>5s</td>
</tr>
</tbody>
</table>

To Formula(7). Assuming that any CS has two entry points and vehicles have two to three vehicular priorities, then we verify the passing-through performance of proposed methods with the initial parameters listed in Table V, where $a_a$ and $a_d$ are set to $3 m/s^2$ and $-4 m/s^2$ respectively [28]; $p_h$, $p_m$, and $p_l$ are introduced to represent the probabilities of vehicles with different priorities, and $Δp_c$ is the minimum interval among vehicles approaching $P_a$. By setting the parameters in Fig.5, different scenes will be established to verify the performance of the proposed mechanisms, particularly regard to the delay time and its corresponding change ratios for different vehicles.

$$P_k = (\lambda t)^k \times e^{-\lambda t} \times \frac{\lambda t}{k!}$$  \hspace{1cm} (10)

B. Experiment Results Analysis

1) GPI Scheduling Sequences

To verify the GPI mechanism when a VPI problem occurs, a CS with two lanes is used. Fig.7 shows the scheduling sequences of vehicles with two priorities. During the simulation, we inserted one vehicle(H) at the head position. Under such situation, vehicle(H) is only delayed at most by one conflicting vehicle on another lane; thus, the sequence of sPriorFIFO is equal to that of FIFO, as shown in Fig.7(a), and GPI processing is not conducted. When vehicle(H) is located at the sixth from the end of its queue, after scheduling by sPriorFIFO, vehicles(L) ($v_{20}, v_{22}, v_{24}$ in $Q^+$) on $\ell_1$ are promoted to vehicles(H), as shown in Fig.7(b). Moreover, these promoted vehicles on $\ell_1$ are scheduled before vehicles ($v_{19}, v_{21}, v_{23}$) on $\ell_2$ to guarantee prior passing of $v_{24}$.

Fig.8(a) shows that when multiple vehicles(H) occur on both lanes concurrently, ($v_{27}, v_{28}, v_{31}$) and ($v_{29}, v_{30}$) inherit a high priority from $v_{32}$ and $v_{33}$, separately. In this situation, vehicles(H) in $Q^+$ are scheduled only sequentially, which means a reversion from sPriorFIFO to FIFO. Fig.8(b) shows a complex scheduling procedure when vehicles with three priorities occur on both lanes, where $v_2$ and $v_{27}$ on $\ell_1$ are vehicles(M), $v_3$ and $v_{26}$ on $\ell_2$ are vehicles(M), and $v_{24}$ on $\ell_1$ has the highest priority. Vehicle(L) $v_1$ is not promoted because $D_3$ is not positive. When vehicle(H) $v_{24}$ arrives, $v_{20}$ and $v_{22}$ are promoted to vehicles(H). Although, the arriving of $v_{26}$ touches off a promotion of $v_{19}$, $v_{21}$, $v_{23}$ and $v_{25}$ to vehicles(M), all these vehicles(M) are delayed by the vehicles(H) ahead. Similar to that in Fig.8(a), sPriorFIFO will revert to FIFO when scheduling these collided vehicles(M). Via sPriorFIFO, the final optimized scheduling sequence of $v_{19}$ to $v_{24}$ will be ($v_{20}, v_{22}, v_{24}, v_{19}, v_{21}, v_{23}$). During the experiments, we also find that some vehicles(L) are always promoted two times from vehicles(M) to vehicles(H). It also expresses an optimized efficiency of sPriorFIFO when vehicles(H) arrives after vehicles(M) arrived on the same lane.

In addition, Renew messages are designed to update changing PTWs of vehicles to $\gamma$-Agent as soon as their accelerations or priorities changed, which is an enhancement for both FIFO and sPriorFIFO algorithms. When such messages are employed, the scheduling sequences always change according to the changing traffic situations. For example, corresponding to Fig.7(b), the priority scheduling sequence without Renew messages of vehicles $v_{18}$ to $v_{25}$ is ($v_{18}, v_{20}, v_{22}, v_{24}, v_{25}, v_{19}, v_{21}, v_{23}$), whereas when the Renew mechanism is activated, the sequence will be ($v_{20}, v_{22}, v_{24}, v_{25}, v_{18}, v_{19}, v_{21}, v_{23}$) because the arriving sequence in $Q^+$ is updated dynamically along with the change in PTWs. From the experiment results, we make sure that although the full scheduling sequence has changed, the sequence of vehicles(H) remains the same, and the scheduling performance is guaranteed.

2) Scheduling Performances

Fig.9 shows the scheduling performance of vehicles on two lanes and with two priorities. From this figure, the delay time of vehicle(H) $v_{25}$ on $\ell_1$ via FIFO is 10.51s. Through sPriorFIFO, vehicles(L) ($v_{20}, v_{22}, v_{24}$) are promoted to high priority, and the corresponding delay time of the seven vehicles(H) is respectively reduced from (9.49s, 11.5s, etc.)
and \( \gamma \) lane inherit a high priority that will result in a block.

When vehicle(H) exists, the delay time of vehicles with lower priorities on other lanes always increases, such as that of \( v_{21} \) and \( v_{23} \). This phenomenon is normal mainly because when VPI occurs, vehicles(L) in front of vehicles(M) on another lane inherit a high priority that will result in a block.

From the experiments with SAP policy, \( A_{max} = 1 \), the passing-through performance sometimes becomes even better than expected. The reason for this finding is that when \( d_{r} \) is 3.2m, it frequently makes \( (t_{i,s} + t_{i,p}) \) of the passing-though vehicle \( v_{i} \) earlier than the time when its follower \( v_{i+1} \) in \( Q^{+} \) arrives at \( P_{b} \), thus guaranteeing that \( v_{i+1} \) will eventually acquire a \( \gamma \)-Token and pass through smoothly without decelerating, which is consistent with Fig.5. This effect is similar to that of batch authorization. Furthermore, to verify such authorization policies, we valued \( d_{r} \) with different typical lengths: 3.2m, 6.4m and 9.6m. The experiment results show that when \( d_{r} \) is assigned with the first two lengths, the increase in \( A_{max} \) only slightly affects delay time because passing-through velocity is relatively high. However, when \( d_{r} \) is set to 9.6m, the effect of \( \alpha \)SAP becomes significant. From Fig.12, we can observe that the delay time with \( \gamma \)-PriorFIFO scheduling is reduced markedly when \( A_{max} = 2 \). However, we also learn that the reduction of delay time will not increase more when \( A_{max} \) is bigger than 3 for the reason above.

Fig.13 shows the experimental results for FIFO and \( \gamma \)-PriorFIFO scheduling on 100 random generated traffic queues when no vehicle with higher priority exists, and the corresponding average delay time is set to zero. By comparing Fig.13(a) with Fig.13(b), it’s clear that \( \gamma \)-PriorFIFO frequently guarantees a decrease in the delay time of vehicles(H), disregarding a few exceptions. For example, when two vehicles that have or inherit the same high priority are arriving from different lanes and competing the same CS, \( \gamma \)-PriorFIFO degrades to FIFO, and the vehicle arriving later will be delayed. From Fig.13(c), the average delay of vehicles(L) is reduced also by \( \gamma \)-PriorFIFO. After analyzing experimental data, we find that the main reason for such phenomenon is that vehicles with low priorities on one lane are always scheduled continuously via \( \gamma \)-PriorFIFO, which saves much time caused by shifting a \( \gamma \)-Token among vehicles on different lanes. Fig.14 indicates 100 scheduling results on randomly generated vehicles with three priorities. From Fig.14(a) and Fig.14(b), prior scheduling of vehicles(H) is clearly guaranteed to be better than that of vehicles(M), which, in turn, is superior to that of vehicles(L). Fig.14(c) also shows that the delay time of vehicles(M) may sometimes worsen because vehicles(L) in front inherited a high priority have blocked vehicles(M).

Different experimental results have shown that the proposed model and algorithm have a good stability to scheduling vehicles with highest priority in prior even if kinematic parameters are set to various values, which corresponds to different types of vehicles or road surfaces under certain weather (dry, wet, snow etc.). It agrees well with our expectations on this study.
the proposed models and algorithms in a scheduling analysis simulator. After performing extensive simulations under different scenarios, we conclude that modeling standard reservation-oriented actions and states is effective in presenting vehicular behaviors uniformly. The proposed mechanisms can solve the VPI problem efficiently and guarantee that vehicles with high priorities will be scheduled in prior; The aSAP policy and Renew message can further enhance the adaptability of our proposed mechanism.

Ongoing and future studies on this topic is mainly focused on passing-through multiple critical sections and integration with classic traffic simulators. Moreover, we also apply such thoughts in this paper to study the distributed cooperation problem among vehicles with different priorities.

**REFERENCES**


**VII. CONCLUSIONS**

In this study, we propose a novel state-driven scheduling method to solve the TPI problem of driverless vehicles with different priorities. Focusing on this topic, various PTI-related traffic objects, involving critical sections, driverless vehicles, and vehicular actions and states, are modeled by fusing the physical and kinematic features. After mapping spatiotemporal collision relationships into an expanded time-distance model, a state-driven RAAL principle and a priority scheduling algorithm sPriorFIFO are proposed. In addition, we implement...


Kailong Zhang received his B.S., M.S., and Ph.D. in computer science and technology, with a specialization in embedded computing and system design, from the School of Computer, Northwestern Polytechnical University, Xi’an, China. He has been connected with Networked Embedded Computing and Technology Laboratory in Northwestern Polytechnical University since 2003, and has been conducting intelligent embedded computing studies, covering novel computation platforms and simulation technologies for autonomous embedded systems. From 2012 to 2014, he was a post-doctor with the CyberCars program of the Center of Robotics in Mines ParisTech, Paris, France, mainly working on the cooperative mechanisms of multiple driverless vehicles by using state-driven automatic control and vehicular ad-hoc networks. His research interests include the architecture, mechanisms, and design methods for autonomous embedded system, cooperation among such systems, and corresponding simulation technologies.

He has won several awards for his research, including a HP scholarship(2006) and the Chinese Ten Youths Award in Embedded System Domain (2011).

Xiao Wu received her B.S. in industrial automation from Zhengzhou University, Zhengzhou, China, in 1981, and her M.S. in computer science and technology from Department of Computer, Northwestern Polytechnical University, Xi’an, China, in 1987.

From 1987, she was connected with the Department of Bioengineering, the Fourth Military Medical University, Xi’an, China, working on intelligent data analysis and simulation in the bioinformatics domain. She has worked with the Networked Embedded Computing and Technology Laboratory in Northwestern Polytechnical University, conducting networked embedded system studies and covering adaptive control technologies for intelligent embedded systems. Her research interests include design and simulation of autonomous embedded systems.

Arnaud de La Fortelle respectively received his Engineer and Ph.D., with a specialization in applied mathematics, from the French Ecole des Ponts et Chaussées and the French Ecole Polytechnique, Paris, France.

He first studied the theoretical properties of probability distributions with application to queuing networks in 1997, and then applied this knowledge to vehicular networks, with a special focus on CyberCars. From 2003 to 2005, he investigated communications for cooperative systems and the architecture required in distributed systems at INRIA. Since 2006, he has been the director of the Joint Research Unit LaRA(La Route Automatise) of INRIA and Mines ParisTech, and has also served as the director of the Center of Robotics(CAOR) in Mines ParisTech since 2008. He has managed several French and European projects(Puvame, Prevent/Intersafe, REACT, COM2REACT) and has been one coordinator of the European project GeoNet and the French project AROS, which has received the French award for enhancing industry competitiveness in 2011. His main topic of interest is cooperative systems(communication, data distribution, control, mathematical certification) and their applications(CyberCars, collective driverless taxis).

He has been an active member of the Board of Governors of IEEE Intelligent Transportation System Society since 2009 and the Board of the French Automotive Engineers Society. He is also a member of several technical program committees for conferences, and vice president of the French ANR evaluation committee for sustainable transport and mobility.

Jean Grégoire received his M.S. in science and executive engineering from Mines ParisTech, Paris, France, in 2011. Since 2011, he has been a Ph.D. candidate at the Center of Robotics, Mines ParisTech, Paris, France. In 2013, he spent three months as a visiting Ph.D. student at the Singapore-MIT Alliance for Research and Technology, Singapore.

His research interests are multiple robot coordination applied to autonomous vehicles and adaptive traffic signal control.