A priority tree based coordination method for intelligent and connected vehicles at unsignalized intersections

Lv Dongxin1 | Wuniri Qiqige1 | Chu Wenbo1 | Yu Huilong3 | Du Xiaoping1,2

1 National Innovation Center of Intelligent and Connected Vehicles, Beijing, China
2 School of Software, Beihang University, Beijing, China
3 Department of Mechanical and Mechatronics Engineering, University of Waterloo, Waterloo, Ontario, Canada

Correspondence
Du Xiaoping, 37# Xueyuan Rd., Haidian District, Beijing 100191, China
Email: xpdu@buaa.edu.cn

Funding Information
Beijing Nova Program of Science and Technology(No. Z19110000119087), Beijing Municipal Science & Technology Commission(No. Z181100004618005)

Abstract
Intelligent and connected vehicles are believed to be the future solution to traffic management, especially in highly challenging areas such as intersections. In this paper, a priority tree based coordination method is proposed for intelligent and connected vehicles at unsignalized intersections. First, a dynamic scheduling method is used to generate the crossing order of the vehicles, considering the conflicting relationship, waiting time, and arrival time of the vehicles. Then a conflict resolution method is presented to handle the spatial and temporal conflicts among the vehicles inside the intersection. And the simulation results show that the method can generate collision-free traffic flows as well as improving the traffic performance.

1 | INTRODUCTION

Intersections are known as the bottleneck for better urban traffic performance due to the complexity of the traffic flow inside them. And the recent development of Intelligent Transportation System (ITS) has provided a new perspective to address this problem [1]. The wide use of vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication can also improve the safety and efficiency of the unsignalized intersections [2]. This paper focuses on the safe crossing for intelligent and connected vehicles (ICVs) at unsignalized intersections. ICV together with the intelligent roadside unit (RSU), has become a promising tool to manage traffic at intersections. The ICV system, as a centralized traffic controller, can resolve the conflicts between the vehicles and ensure safety. What’s more, through the coordination among vehicles, the implementation of the ICV system can also alleviate traffic congestion around the intersections, reduce the number of traffic accidents and improve traffic performance.

Autonomous intersection management (AIM) uses the reservation protocols at the unsignalized intersection in which the vehicles communicate with the manager to reserve a crossing time [3]. Several AIM systems are discussed to address the consensus coordination used in reservation protocol [4]. Reservation based methods can schedule the crossing sequence online, which makes it more robust in variable situations. But they have drawbacks in arrival time estimation and crossing order [5]. For instance, if vehicles on different legs make the reservation at the same time, the one with earlier arrival time may fail to make the reservation because of the temporal conflict, and the one with later arrival time can make the reservation successfully. Under this circumstance, the vehicle with a higher priority may pass the intersection much later than expected. Another way for cooperative intersection control is to convert the problem into a multi-objective optimization problem. In [6], Ding et al. proposed a method that aims to minimize the intersection delay, fuel consumption, and emission as well as the discomfort level. This method also contains a collision-avoidance model to ensure the safety of the vehicles.

Another way to manage the traffic flow at the unsignalized intersection is through the coordinated control of the ICV. One difficulty lies in the large-scale centralized control of ICV is that
the solution space of each vehicle is coupled with each other due to the constraints among the vehicles, and as the number of vehicles increases, the solution space will grow at an exponential rate, which makes it difficult to optimize in real time. Due to the high complexity of this task, the traffic control of the intersection is generally decomposed into two parts. The first part is to determine the arrival time or the crossing order of the vehicles on the fixed path. The second part consists of adjusting the speed of the vehicles in order to avoid collisions and to optimize traffic efficiency.

There are generally three ways to cope with this problem. The first kind including the heuristic methods and the numerical programming methods solve the traffic flow management problem directly. Rapidly-exploring Random Trees (RRT) is a typical heuristic method that is used in many works. In [7, 8], RRT was used to plan the motion of the vehicles at intersections to avoid collisions. The numerical programming method has also been used by many researchers. Fayazi et al. [9] used the Mixed Integer Linear Program (MILP) method to solve the vehicle-intersection coordination problem, and the method is proven to be effective and can eliminate the usage of the physical traffic light. Müller et al. [10] also used MILP to schedule the arrival time of the vehicles to minimize the vehicle delay at intersections. Hult et al. [11] used the Mixed Integer Quadratic Program (MIQP) method to compute an approximate solution of the crossing order of vehicles at an intersection. The second kind solves the problem sequentially, which divides the prime problem into sub-problems with a lower dimension. In this method, the priority order needs to be designed offline, and the vehicles will share the priority information at each iteration. In [12, 13], Qian et al. used the priority-based method to separate the problem into a priority assignment problem and a vehicle control problem under fixed priorities to ensure good properties like safety (collision-free trajectories, brake-safe control). In [14], Xu et al. considered the problem as a tree search problem and used Monte Carlo Tree Search (MCTS) and heuristic rules to generate a nearly optimal solution. In [15], Liu et al. used a tile-based reservation method to plan the entering time and velocity considering the different priorities of each vehicle, and the vehicles can pass the intersection area with a minimal delay following the planned collision-free trajectory. The third kind solves the problem parallelly, which divides the problem into a few sub-problems that can be processed at the same time, including the distributed sub-gradient method (DSM) and Alternating Direction Method of Multipliers (ADMM). Wang et al. [16] applied the ADMM to solve the centralized optimization problem of cooperative automation of large-scale connected vehicles in parallel.

After determining the crossing order, the vehicles need to adjust their speed to avoid potential collisions. In general, some vehicles need to brake to let others go first and some may need to accelerate to clear the space and improve traffic efficiency. Conflict resolution is also used in railway and air transportation systems [17, 18]. Hafner et al. [19] used a formal control theoretic method to regulate two vehicles at intersections which took the communication delay into consideration. The method can guarantee a collision-free system.

As mentioned before, the optimal control based methods can optimize certain performance metrics, such as energy-minimization or throughput maximization, but the computational cost makes this kind method hard to implement in real time. In this paper, we propose a heuristics method using the priority tree called priority tree based coordination (PTC) method. The method can plan the crossing order of vehicles near the intersection area and handle the conflicts among vehicles accordingly.

The contribution of this paper is as follows. First, the proposed PTC method fully considers the space inside the intersection, if two vehicles do not have conflict points on their route, they can enter the intersection simultaneously to improve the throughput of the intersection. Second, the PTC method can adjust the crossing order dynamically in different scenarios. For instance, if an emergency vehicle enters the intersection area, the method can give the vehicle the highest priority to let it pass the intersection area as quickly as possible. Thirdly, the PTC method can be applied to different intersections with different layouts, and the computation cost can meet the requirement of real-time application.

This reminder of this paper is organized as follows. In Section 2, we describe the problem in detail. In Section 3, we elaborate on the methods and algorithms used to solve the problem. Simulation results are shown in Section 4. Finally, the conclusions are presented in Section 5.

## 2 Problem Statement

Reservation-based methods are commonly used in the control of traffic flow at unsignalized intersections. These methods make sure that there is only one vehicle in the intersection at each time step. The vehicles which do not make the reservation successfully will have to stop at the stop line and wait until they get one. This method can make sure that the vehicles can cross the intersection safely, but it cannot improve transportation efficiency since it does not optimize the crossing order and speed profile of vehicles. In this paper, we developed a novel order planning strategy and a motion planning method to address the aforementioned problems.

Figure 1 shows a typical 4-leg unsignalized intersection, which has four entrances (labelled from A–D) and four exits (label from a–d). Each entrance and exit contain only one lane. The control area in this paper is defined as the intersection area combined with a portion of each of the four entrances within a distance $L_c$ upstream from the intersection, as shown in the grey area in Figure 1. And the length of the control area refers to 200 m to the stop line Vehicles will have three possible routes: through, left, or right. And according to the route the vehicles take, there are 12 possible paths that are indexed counter-clockwise with 1–12. And there are 16 crossing points and four converging points at the intersection, which make up the total 20 conflicting points. The vehicles with the same conflicting point on their route cannot pass the intersection at the same time. So it is important to decide the right way for vehicles especially for those that have conflicts with each other.
We assume that after choosing the route, the vehicles will follow the predefined paths inside the intersection and will report their state information (location, speed, and acceleration) at every time step, and we will only control the longitudinal velocity along its path. Also, no overtaking will occur once the vehicle enters the control area.

3 METHODOLOGY

In this section, we decompose the coordination problem into two subproblems: scheduling the crossing order and resolving the confliction.

3.1 Crossing order

One of the most commonly-used strategies to organize the crossing order is the First Come First Served (FCFS) rule. Although it is easy to implement, research has found that the FCFS rule cannot guarantee that the vehicle can come to a complete stop when it is not safe to enter the intersection, or when the vehicle needs to yield to the emergency vehicles or other vehicles with a higher right of way. Moreover, for vehicles whose routes do not have conflicts, they can enter the intersection area at the same time, which means they can share the same priority. Therefore, the crossing order can be considered as a tree structure called the priority tree, instead of a queue formed using the FCFS rule. A priority tree is designed to address the shared priority among vehicles in the intersection. Figure 2 shows a typical priority tree. The root node is on level 1. The vehicles on the same level in the priority tree have the same priority, and the higher the vehicles’ level is, the lower the priority is. And the vehicles will follow the vehicles on the upper level. For example, vehicle 1 and vehicle 3 can enter the intersection at the same time. In the priority tree, the vehicle might also have conflicts with other nodes on the same level (level_p) as its parent node. So in the priority tree, we define all the nodes on level_p as its ancestor nodes, and all the nodes on the same level are each other’s sibling nodes. Using the priority tree, the vehicles only have to check if it will have potential collisions with its ancestor nodes, which will reduce the computational cost significantly.

To address this problem, we proposed a novel Dynamic Scheduling Algorithm (DSA) based on an online formed priority tree. The nodes which have the same level share the same priority. And the parent node will enter the intersection right before its child node. The vehicles are categorized into three bins. The first bin with the highest priority contains vehicles that cannot stop at the stop line even when applying the maximum acceleration or emergency vehicles. The second bin contains the vehicles that have been waiting at the stop line for a while, and the vehicles that have waited longer have a higher priority. The third bin contains the rest of the vehicles. In this bin, the vehicles are sorted based on the traffic regulations and their estimated arrival time. Vehicles with earlier arrival time should have a higher priority in general. But for the sake of safety, if the two vehicles arrive at the intersection in a similar period, their priority should be determined under traffic regulations.

In a lot of countries, there are rules on how to regulate traffic operations in the unsignalized intersection. In China where vehicles keep right, for example, there are four rules: (a) The vehicle that is about to enter the intersection should yield to the ones that are already in the intersection; (b) the vehicles on the right side of the road have higher priority; (c) turning vehicles should give priority to vehicles going straight; (d) when driving in the opposite direction, a vehicle that turns right needs to let the vehicle that turns left go first. Figure 3 shows the rules from (a) to (d) respectively.

According to the aforementioned rules, we can get a global intersection priority table, as shown in Table 1. If there is a converging point between route R_i and route R_j, the corresponding absolute value of the cell is 1. If there is a crossing point, the absolute value is 2. If there is a diverging point, the absolute value is 3. If there is no conflicting point, the value is 0. Assuming that the row stands for the route of the host vehicle R_i, if R_j has a lower priority than R_i, the value will be negative. For the countries where the vehicle keeps left, the priority rules
are slightly different, and the intersection priority table can be generated using the same method.

Using the estimated arrival time, we can get a draft form of the crossing order for the vehicles in the third bin. Then, based on the intersection priority table, vehicle \( i \) can decide whether it has a higher priority than its parent node \( p_i \) using (1), where \( T_{\text{arrival}} \) is the threshold for the arrival time.

\[
\begin{align*}
1 & \quad \text{(change)}, \quad \text{if} \quad -3 < \text{priorityTable} [r_{p_i}] [r_i] < 0 \\
& \quad \text{and} \quad t_{\text{arr}}^i - t_{p_i}^\text{arr} < T_{\text{arrival}} \\
0 & \quad \text{(not change), else}
\end{align*}
\]

There are a lot of methods to estimate arrival time. In [3], it assumes that the vehicle will travel at its current speed if it has the highest priority. However, this assumption will fail to be practical under some circumstances. For example, the vehicles that are about to turn might have to slow down. And for those driving at a rather slow speed, they may need to accelerate to achieve better traffic flow. And also, when the traffic is heavy, it is almost impossible for vehicles to maintain their current speed approaching the intersection area. In [20], the vehicle either travelled at a constant speed or change its speed with a fixed acceleration or the maximal deceleration if it had to. This strategy makes it more flexible and can be applied to the changing situation. Under our assumption, having information on all the vehicles makes it possible to calculate the range of the arrival time.

There are two cases to estimate the arrival time of a given vehicle \( i \). We use \( v_i(t) \) to denote the current speed, \( v_d \) to denote the desired speed when entering the intersection area. Specifically, when the vehicle is going straight, \( v_d \) should equal to the speed limit of the road, which is \( v_{\text{max}} \), when the vehicle is about to turn, \( v_d \) should equal to \( v_{\text{turning}} \).

Figure 4(a) shows the first case. In this case, vehicle A has no leader in front of it. The vehicle would first change its speed from \( v_i(t) \) to \( v_d \) with a set acceleration and travel at \( v_d \) afterward. And the acceleration we choose in the paper is \( \pm 2 \text{ m/s}^2 \) which is a comfortable acceleration and most vehicles can meet the physical requirement for this acceleration.

Figure 4(b) shows the second case. In this case, vehicle A has a direct leader vehicle B on its current lane, since there is no overtaking in this area, vehicle A has to follow vehicle B. And the estimated arrival time will be the maximum between
The algorithm will maintain a virtual root node to keep the stability of the tree structure. This virtual root node has the highest priority and travels at the highest possible speed. In this way, the root node can stay constant in the priority tree.

In each time step, the algorithm will collect the real-time data of all vehicles in the control area, and update the waiting time and estimated arrival time of each vehicle. If a vehicle is detected for the first time in the control area, it will be added to the priority tree. It will first check all the nodes with the highest level, and if there exists, find the note whose route shares the same conflicting point and this node will be selected as the parent node. If there is no conflicting point among the routes of the current layer, the checking will move to the upper layer until they find one with conflicting points or it will choose the root node as its parent node.

Then, the priority tree will be updated to address the change in the priority if the estimated arrival time or the waiting time has changed. Also, the approaching of an emergency vehicle will cause big changes in the priority tree. The update will be done by the order of the three bins.

First, if an emergency vehicle enters the control area, it will move up to the highest possible priority. The emergency vehicle will find its path to the root node, and all the nodes on this path are called the ancestors of vehicle $e$. And the vehicles in front of vehicle $e$ on its current lane are called leaders. For vehicle $e$ to pass the intersection area as quickly as possible, the leaders should also have the highest possible priority. Leaders will determine if they can get higher priority than ancestors in sequence. Figure 5 shows the process for the adjustment in the case of the emergency vehicle. In this scenario, vehicle 9 is the direct leader for emergency vehicle 10. The red nodes represent the vehicle that should yield to the emergency vehicle. The blue nodes are the ones that can pass the intersection area before the emergency vehicles and their direct leaders. The white nodes are the ones that do not have conflicts with the emergency vehicle. First, vehicle 9 will find its highest possible priority, which
is higher than vehicle 6. Therefore, vehicle 9 will choose the parent among the nodes whose level is no higher than $\text{level}_6$, which is vehicle 2 in this case. So will vehicle 10. Note that after one vehicle changes its position in the tree and move to a new level $\text{level}_{\text{new}}$, the priority of vehicles whose original level is no less than $\text{level}_{\text{new}}$ might change. In this case, vehicle 7 is originally on level 5, and after vehicle 10 moves to level 5, vehicle 7 has to find its new parent among nodes that are not in its own subtree.

Secondly, if some vehicles’ waiting time has surpassed the threshold, it will gain the highest possible priority out of its ancestors, which means only the vehicles that cannot stop at the stop line can maintain their previous priority.

Finally, as the consequence of the update of the first two bins, the estimated arrival time will change, hence the change in the crossing order. If the difference between the adjacent two vehicles’ arrival time is less than the threshold, their priority will be determined by the intersection priority table. If the difference is higher, the vehicle with the earlier arrival time will have a higher priority.

Once the vehicle leaves the intersection, the corresponding node will be deleted from the tree. Based on the structure of the tree, only the vehicles on the second level are possible to leave the control area. If the vehicles on the second layer leave the control area at the same time step, then all the child nodes of these vehicles will be linked to the root node. If not, the child nodes of the leaving vehicles will be linked to the other nodes on the second layer or the root node according to the intersection priority table.

3.2 Speed control

In the previous section, we form the crossing order, and the vehicles need to adjust their speed profile accordingly. Unlike the car following situation, the safety margin inside the intersection is much more complex. For the cases where both vehicles are going straight, the conflict area could be deemed as a rectangle area, as shown in Figure 6(a). But for the cases where turning vehicles are involved, the conflict zone could be irregular, as shown in Figure 6(b).

![Figure 6](image)

**FIGURE 6** Different cases for conflict zones

To avoid colliding with each other inside the intersection and simplify the mathematical calculation, we divide the intersection area into four grids, as shown in Figure 7.

The gridded intersection can uncouple the spatial conflicts with the size of the vehicles and their state information. A pair of vehicles are considered to have spatial conflicts if their paths are crossed in a grid, as shown in Figure 8. In the first case, the crossing point is inside grid 4, and the conflict zone is grid 4. In the second case, the crossing point is on the side of both grid 1 and grid 3, and the conflict zone is the two grids combined. In the third case, there is no conflict zone.

Once two vehicles have spatial conflicts, we need to determine whether they have temporal conflicts, which means a pair of vehicles are in the same grid at the same time. We use $r_{\text{grid}}(t_{\text{arrival}}, t_{\text{departure}})$ to define the time range when vehicle $i$ stays in grid $k$, where $t_{\text{arrival}}$ is the arrival time and $t_{\text{departure}}$ is the departure time of a certain grid. For a given pair of vehicles, if two $r_{\text{grid}}$ overlap, that means they have both spatial and temporal conflicts, and the two vehicles are likely to collide with each other, and conflict resolution needs to be applied. To calculate the time range, the distances for arrival $S_a$ and departure $S_d$ need to be addressed first.

To calculate $S_a$ and $S_d$, the shape of the turning trajectory needs to be determined first. In a lot of studies, the turning trajectory is considered to be a transition curve to overcome the discontinuity of the radius of curvature along the turning trajectory. In [21], the vehicles’ trajectory is modeled using the optimal control theory to generate the fastest path or the smoothest
where velocity. The clothoid curve can be defined using Equation (2).

\[ s = l C \]  

where \( r \) is the radius of a certain point on the curve, \( l \) is the corresponding length measured along the spiral curve from its initial position, and \( C \) is a constant.

Figure 9 illustrates the clothoid curve in the coordinate system \( xoy \), where the coordinate of any point on curve \( s1 \) can be calculated using Equation (3) \([22]\). And the coordinate of the points on curve \( s2 \) can be calculated based on the principles of graphics transformation.

\[
\begin{align*}
\begin{cases}
  x = x_0 + \frac{l}{l_5} - \frac{l^5}{40l_5^2} & \\
  y = y_0 + \frac{l}{l_6} - \frac{l^6}{336l_6^3}
\end{cases}
\]  

In \([23]\), researchers have modelled the relationship between the size of the intersection and the critical radius \( R \). Using this information, \( S_a \) and \( S_d \) for each route of the given grid can be determined. Besides, the vehicle is assumed to follow the route alignment inside the intersection, so the velocity of the vehicles can be considered as the tangent to the curve in each time step. Under this assumption, we can calculate the velocity with a linear formula.

For vehicle \( i \), it will try to keep a safe distance with its parent node \( p_i \), while tracking the speed of vehicle \( p \). We use the car following model as in Equation (4). The model uses a modified optimal velocity model (OVM) \([24, 25]\) to plan the speed profile for the vehicles in the control area. The vehicle will consider its current leader on the road (actual leader) and its ancestor nodes in the priority tree (logical leaders). The vehicle will choose the smallest acceleration as its acceleration in the next time step.

\[
\frac{dv_i(t)}{dt} = \kappa \left[ V_{opt}(\Delta x_i(t),\Delta v_i(t)) - v_i(t) \right] + \lambda \Delta v_i(t) 
\]  

where \( \Delta x_i(t) \) is the distance between vehicle \( i \) and its leader \( l \), \( V_{opt}(\Delta x_i(t),\Delta v_i(t)) \) is the optimal velocity for vehicle \( i \), \( \kappa \) represents the sensitivity for the changes of the logical leaders, \( \Delta v_i(t) \) is the speed error between vehicle \( i \) and its leader \( l \), and \( \lambda \) is the feedback gain of the speed error.

The optimal velocity function \( V_{opt}(\cdot) \) is shown in Equation (5).

\[
V_{opt}(\Delta x_i(t)) = \frac{v_{max}}{2} \left[ \tan(\Delta x_i(t) - l_{safe}) + \tan(l_{safe}) \right] 
\]

where \( v_{max} \) is the maximal speed of the vehicle, and \( l_{safe} \) is the minimum safe distance. In this paper, \( l_{safe} \) is defined as the sum of the leading vehicle length and the minimal distance between the leading and the rear vehicles.

We also introduced a weighting factor for the optimal velocity in order to perform better braking performance. The weighting factor is defined in Equation (6).

\[
W(\Delta x_i(t),\Delta v_i(t)) = A \left[ 1 + \tanh B \left( \frac{\Delta v_i(t)}{\Delta x_i(t)} + C \right) \right] 
\]

And the new optimal velocity is shown in Equation (7).

\[
V_{opt'}(\Delta x_i(t),\Delta v_i(t)) = W(\Delta x_i(t),\Delta v_i(t)) V_{opt}(\Delta x_i(t)) 
\]

4 | SIMULATION RESULTS

To evaluate the effectiveness of the proposed method PTC, we conduct simulation tests in SUMO and compare it with some existing intersection management methods.

The parameters used in the simulation test are shown in Table 2. The approaching vehicles choose entrances randomly, and the arrival rate follows the Poisson distribution.

In the simulation, we first collected the position profiles of the vehicles under the control of PTC, as shown in Figure 10. The horizontal axes represent the location of vehicles, which are the corresponding approaching entrances and the distance to the intersection, and the vertical axis represents time in the simulation test. The vehicles can pass the intersection orderly. And some vehicles which enter the intersection at the same time have on conflicts with each other, which proves that our method can generate collision-free traffic flow.

Then, we compared our method PTC with the reservation-based method, the actuated intersection control (AIC) method, and the default intersection control method in SUMO with a static traffic light under different arrival rates. Figure 11 shows the results of average speed using different methods. When the arrival rate is 0.5 (pc/s), both PTC and the reservation-based method can maintain the average speed. The average speed
TABLE 2 Parameters and definitions

| Parameter  | Definition                                      | Value     |
|------------|-------------------------------------------------|-----------|
| $l_c$      | The length of the control area on each leg      | 200 (m)   |
| $l$        | The width of the lane                           | 3.2 (m)   |
| $v_{min}$  | The minimal speed                              | 0 (m/s)   |
| $v_{max}$  | The maximal speed when going straight           | 10 (m/s)  |
| $v_{turning}$ | The maximal speed when turning                   | 10 (m/s)  |
| $a_{acc}$  | The maximal acceleration                        | 1.5 (m/s²) |
| $a_{decel}$ | The minimal acceleration                        | -3 (m/s²) |
| $a_{emergency}$ | The comfortable deceleration                    | -1.5 (m/s²)|
| $\lambda$  | Arrival rate                                    | 0.5, 1, 1.5, 2 (pc/s) |
| $l_{safe}$ | Minimum safe distance                           | 5 (m)     |
| $\Delta t$ | Time step length                                | 0.1 (s)   |

FIGURE 10 Position profiles of vehicles in the control area

drops as the arrival rate increases, and PTC shows much better improvement than the reservation-based method, and the average speed only drops about 25% when the arrival rate is 2 (pc/s). AIC works better than SUMO since it can change the phase of the traffic light dynamically according to the density of each entrance.

Figure 12 shows the results of the average total waiting time using different methods under different arrival rates. Vehicles complied with PTC has nearly no waiting time when the arrival rate is under 1.5, when the arrival rate is higher, the waiting time is still very low. For the reservation-based method, the waiting time increases significantly when the arrival rate is higher than 1.0. This is because some vehicles have to wait until the intersection is empty, and the successful reservation made by a slow vehicle can cause junctions and make others wait for a long time. And AIC works better than the reservation-based method when the arrival rate is higher than 1.5.

In Figure 13, we compare the traffic flow rate of each method with the expected flow rate. The traffic flow rate in this paper is referred to as the number of vehicles that traverse the intersection area within an hour. When the arrival rate is under 1.0 (pc/s), PTC can catch up with the expected traffic flow rate, which means the vehicles can pass the intersection with nearly no delay. When the arrival rate is higher, the intersection will be much crowded, but PTC can still improve the traffic rate significantly. And AIC still shows a better performance than the reservation-based method when the arrival rate is higher than 1.5, which shows that when the traffic is crowded, the actuated traffic light control can achieve better traffic throughput than the commonly-used reservation-based intersection control method.
We also compared our method with some state-of-art methods, and the results are shown in Table 3. We compare the difference between the expected traffic flow rate and the actual traffic flow rate using different methods, and PTC shows the most improvement in terms of the traffic flow rate. When the arrival rate is under 1.0, CCIC (centralized cooperative intersection control) [6] and PTC can maintain the traffic flow rate to the expected one, and shows the least drop when the arrival rate increases, compared with MCTS (Monte Carlo Tree Search) [14] and TP-AIM (trajectory planning based autonomous intersection management mechanism) [15]. We also study the speed drop compare with the maximal speed under different methods, and PTC is also able to maintain a faster speed for vehicles entering the intersection, and the speed will only drop about 16.5% when the arrival rate is 1.14.

To evaluate the effectiveness of the emergency adjustment used in the update of the first bin, we carry out the simulation where the emergency vehicle is involved. Figure 14(a) shows the distance to the intersection of each vehicle, and Figure 14(b) shows the speed profiles. Vehicle 10 is the emergency vehicle,

![Figure 13](image1.png)  
**FIGURE 13**  Traffic flow rate under different arrival rates

![Figure 14](image2.png)  
**FIGURE 14**  Results in the case where emergency adjustment is applied. (a) Position profiles of each vehicle in the control area, (b) Speed profiles of each vehicle in the control area

| Methods | Arrival rate (pc/s) | Traffic flow rate | Expected traffic flow rate | Average speed | Max speed | Speed Drop |
|---------|---------------------|-------------------|---------------------------|---------------|-----------|------------|
| CCIC    | 0.56                | 2000              | 2000                      | 10.13         | 12.5      | 18.93%     |
| PTC     | 0.56                | 2000              | 2000                      | 9.41          | 10        | 5.9%       |
| CCIC    | 0.83                | 2988              | 3000                      | 8.02          | 12.5      | 28.13%     |
| PTC     | 0.83                | 3000              | 3000                      | 9.37          | 10        | 6.3%       |
| MCTS    | 0.5                 | 1766              | 1800                      | —             | —         | —          |
| PTC     | 0.5                 | 1800              | 1800                      | 9.46          | 10        | 5.4%       |
| MCTS    | 1.14                | 3897              | 4098                      | —             | —         | —          |
| PTC     | 1.14                | 4024              | 4098                      | 8.35          | 10        | 16.5%      |
| TP-AIM  | 0.55                | 1504              | 1980                      | —             | —         | —          |
| PTC     | 0.55                | 1980              | 1980                      | 9.43          | 10        | 5.7%       |
| TP-AIM  | 1.05                | 2872              | 3780                      | —             | —         | —          |
| PTC     | 1.05                | 3756              | 3780                      | 9.21          | 10        | 7.9%       |
and vehicle 9 is the leader of vehicle 10 on its current road (highest priority vehicles). Vehicles in blue are the ones that have conflict points with the highest-priority vehicles but don’t have to yield to them before they can leave the intersection area before the highest-priority vehicles arrive. Vehicles in black are the ones that are not on the path from the highest priority vehicles to the root. Vehicles in red are the ones that have to slow down or stop to let the highest-priority vehicles pass first, while other vehicles can maintain their maximum speed. After the highest priority vehicles can pass the intersection safely, other vehicles can speed up the traffic flow can be back to normal. Table 4 shows the comparison of the arrival time. The arrival time of emergency vehicle with emergency adjustment is applied is 7 s earlier than the case without the adjustment, showing that the adjustment can shorten the time needed for the emergency vehicle to pass the intersection.

Since PTC in this paper is designed to be used in real-time applications, we also evaluate the computation cost of PTC, and the results are shown in Figure 15. When the arrival rate is 2 pc/s, the computational cost of the proposed method is less than 5 ms in most cases. This latency is acceptable in our current simulation test. Considering that update frequency for intelligent vehicles in high-priority safety applications is no more than 50 Hz [26], the proposed method PTC can be applied to real-time applications.

Finally, this method can be used in different intersection layouts by changing the intersection priority table accordingly. We also applied this method in a T-junction and an intersection with three approaching lanes in each direction. And the results in Table 5 shows that compared with the AIC method, the coordination among vehicles can help to improve the traffic performance in different intersection layouts.

### 5 | CONCLUSIONS

This paper proposed a coordination method for the intelligent and connected vehicles at unsignalized intersections, which is aimed at generating collision-free traffic flow at the unsignalized intersections, as well as improving the overall traffic performance. The simulation results show that the method we proposed not only can make sure that the vehicles can pass the unsignalized intersections without any collision, but also can improve the traffic performance in terms of average speed, waiting time and traffic throughput. When the traffic flow is heavy with the arrival rate of more than 1 pc/s, the proposed method can improve the traffic flow rate by over 30% and manage to keep the speed drop under 30% when the arrival rate reaches 2 pc/s. And in the case where the emergency vehicle is involved, the emergency adjustment process can effectively reduce the time the emergency vehicle needs to pass the intersection area by around 20%. Also, the speed control method can reduce the influence of the emergency vehicle on the traffic flow shortly after the emergency vehicle leaves the intersection area. And we also prove that the method can be applied to different layouts.
of the intersections including T-junctions and intersections with multiple approaching lanes in each direction.

In the future, we will work on the vehicle platooning at road intersections, the regulation of traffic flow in a larger scale road network with multiple intersections, taking into consideration the network communication imperfection and measurement errors.

**NOMENCLATURES**

- \( L_{c} \): Length of the control area in each approaching lane
- \( T_{\text{arrival}} \): Threshold for the arrival time
- \( T_{\text{waiting}} \): Threshold for the waiting time
- \( a_{\text{acc}} \): Acceleration used when a vehicle speeds up
- \( a_{\text{dec}} \): Deceleration used when a vehicle slows down
- \( a_{\text{emergency}} \): Deceleration used when a vehicle needs to slow down for emergency
- \( a_{i}(t) \): Acceleration of vehicle \( i \) at time \( t \)
- \( d_{i}(t) \): Distance to the stop line from the front bumper of vehicle \( i \) at time \( t \)
- \( l_{i} \): Length of the vehicle
- \( l_{a,fe} \): Minimum safe distance
- \( r_{i} \): Route index of vehicle \( i \)
- \( t_{\text{headway}} \): The headway time between a vehicle and its leader
- \( t_{\text{waiting}}^{i,\text{arr}} \): Time when vehicle \( i \) is expected to arrive at the stop line
- \( t_{\text{waiting}}^{i,\text{exit}} \): Time when vehicle \( i \) is expected to leave the conflict zone with respect to a given vehicle
- \( v_{i}(t) \): Velocity of vehicle \( i \) at time \( t \)
- \( v_{\text{max}} \): Maximum speed of a vehicle
- \( v_{\text{turning}} \): Maximum speed of the vehicle when turning
- \( x_{i}(t) \): Position of the vehicle \( i \) along the lane at time \( t \)
- \( \Delta t \): Time step length

**Variables**

- \( l_{\text{level}} \): Definition level of the vehicle \( i \) in the priority tree

**ORCID**

Lv Dongxin: https://orcid.org/0000-0001-5041-8964
Du Xiaoping: https://orcid.org/0000-0002-1370-5902

**REFERENCES**

1. Younes, M.B., Boukereche, A.: An intelligent traffic light scheduling algorithm through vanets. In: 39th Annual IEEE Conference on Local Computer Networks Workshops, Edmonton, Alberta, Canada (2014)
2. Zhang, N., et al.: Unsignalized intersections: Can its offer improved efficiency and safety? In: 2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC), Washington, DC (2011)
3. Dressner, K.M., Stone, P.: A Multiagent approach to autonomous intersection management. J. Artif. Intell. Res. 31, 591–656 (2008)
4. Wuthishuwong, C., Traechtler, A.: Stability of the consensus in the network of multiple autonomous intersections management. In: 2014 20th International Conference on Automation and Computing, Cranfield, UK (2014)
5. Bellhouche, F.: Collaboration and optimal conflict resolution at an unsignalized intersection. IEEE Trans. Intell. Transp. Syst. 20(6), 2301–2312 (2019)
6. Ding, J., et al.: Centralized cooperative intersection control under automated vehicle environment. In: 2017 IEEE Intelligent Vehicles Symposium (IV), Los Angeles, California (2017)
7. Mehta, S.S., et al.: Human-assisted RRT for path planning in urban environments. In: 2015 IEEE International Conference on Systems, Man, and Cybernetics, Hong Kong (2015)
8. Kuwata, Y., et al.: Motion planning for urban driving using RRT. In: 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, Nice, France (2008)
9. Fayazi, S.A., Vahidi, A., Luckow, A.: Optimal scheduling of autonomous vehicle arrivals at intelligent intersections via milp. In: 2017 American Control Conference (ACC), Seattle (2017)
10. Müller, E.R., Carlson, R.C., Junior, W.K.: Intersection control for automated vehicles with Milp. IFAC-PapersOnLine 49(3), 37–42 (2016).
11. Hult, R., et al.: An Miep-based heuristic for optimal coordination of vehicles at intersections. In: In: 2018 IEEE Conference on Decision and Control (CDC), Miami Beach, Florida (2018)
12. Qian, X., et al.: Decentralized model predictive control for smooth coordination of automated vehicles at intersection. In: In: 2015 European Control Conference (ECC), Linz, Austria (2015)
13. Qian, X., et al.: Priority-based coordination of autonomous and legacy vehicles at intersection. In: 17th International IEEE Conference on Intelligent Transportation Systems (ITSC), Qingdao, China (2014)
14. Xu, H., et al.: Cooperative driving at unsignalized intersections using tree search. IEEE Transactions on Intelligent Transportation Systems, 21(11), pp. 4563–4571 (2019).
15. Liu, B., et al.: Trajectory planning for autonomous intersection management of connected vehicles. Simul. Model. Pract. Theory 90, 16–30 (2019).
16. Wang, Z., et al.: Parallel optimal control for cooperative automation of large-scale connected vehicles via Admm. In: In: 2018 21st International Conference on Intelligent Transportation Systems (ITSC), Maui, Hawaii (2018)
17. Min, Y.-H., et al.: An appraisal of a column-generation-based algorithm for centralized train-conflict resolution on a metropolitan railway network. Transp. Res. Part B Methodol. 45(2), 409–429 (2011)
18. Visintini, A.L., et al.: Monte Carlo optimization for conflict resolution in air traffic control. IEEE Trans. Intell. Transp. Syst. 7(4), 470–482 (2006)
19. Hafner, M.R., et al.: Cooperative collision avoidance at intersections: Algorithms and experiments. IEEE Trans. Intell. Transp. Syst. 14(3), 1162–1175 (2013)
20. Huang, S., et al.: Assessing the mobility and environmental benefits of reservation-based intelligent intersections using an integrated simulator. IEEE Trans. Intell. Transp. Syst. 13(3), 1201–1214 (2012)
21. Zhao, J., et al.: Two-dimensional vehicular movement modelling at intersections based on optimal control. Transp. Res. Part B Methodol. 138, 1–22 (2020)
22. Vazquezmenendez, M.E., Casal, G.: The clothoid computation: A simple and efficient numerical algorithm. J. Surv. Eng. 142(3), 04016005 (2016)
23. Bando, M., et al.: Dynamical model of traffic congestion and numerical simulation. Phys. Rev. E 51(2), 1035–1042 (1995)
24. Lazar, H., Rhoalanski, K., Rahmani, M.D.: A review analysis of optimal velocity models. Period. Polytchn. Transp. Eng. 44, 123–131 (2016)
25. The CAMP Vehicle Safety Communications Consortium: Vehicle Safety Communications Project: Task 3 Final Report: Identify Intelligent Vehicle Safety Applications Enabled by DSRC, National Highway Traffic Safety Administration, US Department of Transportation, Washington DC, pp. 809–859 (2005).

**How to cite this article:** Dongxin Lv, Qiqige W, Wenbo C, Huilong Y, Xiaoping D. A priority tree based coordination method for intelligent and connected vehicles at unsignalized intersections. IET Intell. Transp. Syst. 2021;15:1053–1063. https://doi.org/10.1049/itr.212082