Multi-Lane Unsignalized Intersection Cooperation With Flexible Lane Direction Based on Multi-Vehicle Formation Control

Mengchi Cai, Student Member, IEEE, Qing Xu, Chaoyi Chen, Student Member, IEEE, Jiawei Wang, Student Member, IEEE, Keqiang Li, Jianqiang Wang, and Xiangbin Wu

Abstract—Unsignalized intersection cooperation of connected and automated vehicles (CAVs) is able to eliminate green time loss of signalized intersections and improve traffic efficiency. Most of the existing research on unsignalized intersection cooperation considers fixed lane direction, where only specific turning behavior of vehicles is allowed on each lane. Given that traffic volume and the proportion of vehicles with different turning expectation may change with time, fixed lane direction may lead to inefficiency at intersections. This paper proposes a multi-lane unsignalized intersection cooperation method that considers flexible lane direction. The two-dimensional distribution of vehicles is calculated and vehicles that are not in conflict are scheduled to pass the intersection simultaneously. The formation reconfiguration method is utilized to achieve collision-free longitudinal and lateral position adjustment of vehicles. Simulations are conducted at different input traffic volumes and turning proportion of incoming vehicles, and the results indicate that our method outperforms the fixed-lane-direction unsignalized cooperation method and the signalized method.

Index Terms—Connected and automated vehicles, flexible lane direction, formation control, multi-lane unsignalized intersections.

I. INTRODUCTION

INTERSECTION is one of the most common traffic scenarios where congestion happens. Numerous intelligent intersection management methods are proposed to reduce congestion and improve traffic efficiency [1]–[3]. Among the existing research about intersection management, signalized intersections and unsignalized intersections are the two main scenarios. At signalized intersections, researchers optimize traffic signal timing, control vehicles to adapt to the signal timing, and combine the two ideas to achieve cooperative optimization of both signal timing and vehicle control [4]–[6]. However, the stop-and-go behavior of vehicles at signalized intersections can’t be avoided due to the natural property of traffic signals, which may bring about green time loss and result in inefficiency for traffic management. With the help of Internet-of-things technologies, researchers have started to focus on methods for traffic control with connected vehicles and unsignalized intersection cooperation [7], [8].

In the field of unsignalized intersection management, reservation methods, planning methods, and optimization methods are three main categories. Reservation methods centralize and organize trajectories of vehicles where each vehicle reserves a block of space-time in the intersection and the centralized controller manages the reservations according to the “First Come, First Served” policy [9]–[11]. Reservation is a simple way to achieve multi-vehicle collision-free passing without traffic signal, but may not utilize the cooperation potential of vehicles and limit the improvement of traffic efficiency. Planning methods usually plan vehicles’ spatiotemporal trajectories to resolve conflict and avoid collision between vehicles. Passing order is the main issue that most of the planning methods consider, and some algorithms are designed to calculate passing order of vehicles, e.g., depth-first searching methods [12], Monte Carlo Tree Searching methods [13], and other graphical methods [14]–[16]. Optimization methods centrally model the behavior of vehicles and solve the optimization problem with collision-avoidance constraints. Some methods have been proposed to solve the centralized optimization problem, e.g., dynamic programming methods [17], model predictive control (MPC) methods [18], and other optimization methods [19], [20].

Among the above intersection management methods, fixed lane direction is commonly considered, where only specific turning is allowed on each lane. In fact, fixed lane direction is a legacy from the signalized intersections to guide vehicles with different turning expectation to distribute on different lanes and avoid collision when passing the intersection. However, considering the fluctuation of traffic input where proportion of vehicles with different turning expectation may change with time, the capacity of some lanes can’t be fully utilized in fixed-lane-direction intersections. Thus, removing the constraint of fixed lane direction and making full use of intersection capacity can further improve.
traffic efficiency with help of CAVs. Some research has focused on dynamic lane use and lane-allocation-free intersection management [21]–[28]. Dynamic lane use methods usually adjust the lane-use configuration of intersections according to the changing traffic inputs. Some research combines dynamic lane use and signal control to improve traffic efficiency at intersections [21], [22]. Some other research extends the idea of dynamic lane use and make some lanes reversible (allowing reverse traffic flow) to balance traffic flow demand at different time [23], [24]. A cooperation method for lane-allocation-free intersections is proposed in [25] which optimizes vehicles’ trajectories by mixed-integer linear programming. A lane-change-free cooperation method is proposed in [26] for future road intersections, where vehicles don’t change lane and adjust their longitudinal position to pass the intersection without collision. Other methods considering dynamic lane use include [27], [28].

Most of the existing intersection management methods ignore or simplify the lane changing behavior of vehicles, e.g., lane changing is assumed to have finished before vehicles enter the intersections in [15], [16], and is assumed to finish instantly in [25]. Success rate of lane changing has a great impact on the efficiency of intersections, and unsuccessful and inefficient lane changing is one of the main reasons that cause traffic congestion [29]. Some cooperative lane changing methods have been proposed in multi-lane scenarios, e.g., formation control methods, vehicle sorting methods, and priority-based methods.

Multi-lane formation control methods consider vehicles in a certain range as a group, and cooperatively control them to achieve formation forming, structure reconfiguration, and vehicle joining/leaving management [30]–[32]. Both formation control and vehicle sorting methods have focused on the problem of collision avoidance when a group of vehicles perform cooperative lane changing [33]–[35]. Vehicles’ preference on specific lanes is considered in [36], which is a key issue in intersection management because vehicles need to change to the lanes where their desired turning is allowed. A vehicle regrouping method based on multi-vehicle formation control is proposed in [37] to guide vehicles to change lanes in a multi-lane intersection. Some priority-based methods are combined with heuristic searching algorithms to plan trajectories for vehicles considering lane changing at ramps and lane-drop bottlenecks [38]. The above methods solve cooperative lane changing problems in multi-lane scenarios, and have the potential to be applied to multi-lane intersections.

This paper focuses on the problem of multi-lane unsignalized intersection cooperation with flexible lane direction. The main contributions of this paper include that:

1) A two-dimensional distribution calculation method is proposed to calculate the desired position of vehicles that includes both longitudinal (passing order) and lateral (lane occupation) position at intersections. Conflict relationship between different traffic movement is defined and available position is calculated for each vehicle considering achievability. The proposed method can be applied to multi-lane unsignalized intersections, either with flexible lane direction or with fixed lane direction.

2) The formation reconfiguration method using relative motion planning and multi-stage vehicle control is utilized for vehicles to achieve the calculated distribution. Collision-free motion of vehicles is planned in the relative coordinate system. Multi-stage trajectories are planned for vehicles and trajectory following with spatiotemporal constraints is modeled and solved as an optimal control problem. Compared with existing research that ignores or simplify lane changing behavior of vehicles, this method enables efficient lane changing at intersections.

3) The performance of the proposed method is verified in simulations at different input traffic volumes and different input turning proportion of vehicles. Simulations show that removing lane direction constraints can significantly improve passing capacity of intersections because more vehicles are allowed to pass the conflict zone of intersections simultaneously. The results indicate that the efficiency of the proposed method outperforms both the unsignalized cooperation method with fixed lane direction and the signalized method.

The rest of this paper is organized as follows. In Section II, the scenario studied in this paper is introduced and the three-step process of solving the multi-vehicle coordination problem at unsignalized intersections is formulated. In Section III, conflict behavior between vehicles is defined and the longitudinal and lateral scheduling algorithm is proposed. In Section IV, the formation reconfiguration framework is built, and the path planning, conflict resolution, and vehicle control methods are provided. Section V carries out simulations and Section VI gives the conclusion.

II. FRAMEWORK OF UNSIGNALIZED INTERSECTION MANAGEMENT

In this section, we introduce the scenarios studied in this paper and propose the framework for multi-lane unsignalized intersection management with flexible lane direction.

A. Scenarios

The scenario studied in this paper is the multi-lane unsignalized intersection with flexible lane direction. An example of three-lane intersection where all incoming lanes allow variable turning is shown in Fig. 1. A vehicle is allowed to turning right, going straight, and turning left on every incoming lane. The area where four arms of intersections overlap is conflict zone, shown as the area inside the dashed lines in Fig. 1. The conflict zone is also the area inside the stop line of intersections.

The incoming lanes are marked in gray. Lanes are indexed by integers from right to left beginning with one, from the perspective of vehicles. In order to simplify and regulate routes of vehicles, the assumption is made that vehicles can only travel to the lane with the same index when passing the conflict zone of an intersection, and the number of lanes of each arm is the same.

There are three types of traffic movements from one incoming arm, including right-turning movement, straight-going movement, and left-turning movement, and twelve movements in total from all the four arms. Three attributes are defined for each vehicle to describe their movement: $r$, $t$ and $l$. $r$ represents the arm where the vehicle comes from, and is set to 1, 2, 3,
...with the same constant speed, they can pass the conflict zone without collision.

2) Formation reconfiguration. In this part, all the vehicles driving on the same incoming arm are considered as a formation, and paths for them to travel to their desired position are calculated. A path consists of a sequence of points for a vehicle to follow, and guides vehicles to arrive at their desired position without collision.

3) Vehicle control. In order to realize reconfiguration of formation, vehicles plan their trajectories to pass through sequences of points, and calculate their control inputs with spatiotemporal constraints.

The distribution calculation part is realized by the proposed iterative grouping algorithm, and the conflicts between vehicles in the conflict zone are resolved. This part will be discussed in Section III. The formation reconfiguration and vehicle control problems are solved by relative motion planning and control of vehicles, and the conflicts between vehicles on the same arm of intersections are resolved. This part will be explained in detail in Section IV.

Remark 1: Unlike the existing research assuming that vehicles conduct lane changing behavior and longitudinal position adjustment separately, this paper integrates these two stages. Vehicles search for their paths in a two-dimensional map, whose size is limited by the number of lanes and vehicles’ distance towards the stopping line. The safety constraints are modelled as inputs (map size) for the path planning problem, and is accordingly guaranteed in the solution.

Remark 2: The proposed intersection management framework is applicable to unsignalized intersections either with or without flexible lane direction. The difference of applying it to intersections with and without flexible lane direction lies in the available position selection for vehicles in the distribution calculation part, by defining selectable lanes for vehicles to occupy when passing the conflict zone.

III. CONFLICT-FREE DISTRIBUTION CALCULATION

In this section, we define conflict relationship between vehicles and propose a conflict-free distribution calculation method to spatially arrange vehicles to avoid collision when passing the conflict zone.

A. Conflict Definition

A conflict represents that the motion of vehicles may lead to a collision. In this section, we focus on the conflict relationship between vehicles in the conflict zone of intersections. Previous research has defined conflicts in intersections with fixed lane direction [12]. In this paper, since lanes are set free for all kinds of turning, the conflict relationship is more complicated. As shown in Fig. 1, three types of conflict are defined:

1) Crossing conflict. If the trajectories of two vehicles in the conflict zone cross, the two vehicles are in crossing conflict. A crossing conflict may exist between any two types of traffic flow and will be used to check conflict between vehicles later in Section III-A.

B. Framework

In order to control vehicles to safely pass an intersection without collision with others, we need to determine two issues for every single vehicle: which lane to occupy, and when to pass the conflict zone. Assuming that vehicles pass the conflict zone with a constant speed [12], [15], the issue of determining when to pass the conflict zone is equal to determining the relative longitudinal desired position of vehicles. Then the key of controlling vehicles to safely pass an intersection becomes: calculating longitudinal and lateral relative position of vehicles, guiding them to drive to their desired position, and control them to pass the conflict zone with a constant and safe speed.

For realizing the above steps, we propose a framework for multi-lane unsignalized intersection management with flexible lane direction. The framework consists of three parts: distribution calculation, formation reconfiguration, and vehicle control.

1) Distribution calculation. Both the longitudinal (passing order and following distance) and lateral (lane occupation) distribution of vehicles are calculated in this part, considering achievability (whether a vehicle is able to catch up with a target position before reaching the stop line). The calculated distribution guarantees that if all vehicles travel...
2) Diverging conflict. If the trajectories of two vehicles in the conflict zone start from the same lane, they should be sorted at different longitudinal position, and the two vehicles are in diverging conflict, shown as the yellow circles in Fig. 1.

3) Merging conflict. If the trajectories of two vehicles in the conflict zone end at the same lane, the two vehicles are in merging conflict, shown as the yellow triangles in Fig. 1.

If two vehicles that are in conflict pass the conflict zone at the same time, a collision may happen. Based on the definition of movement of vehicles in (1), the conflict matrix is defined to check if vehicles with movement $i$ and $j$ are in conflict. The conflict matrix $C$, whose element on the $i$-th row and the $j$-th column represents the conflict relationship between movement $i$ and $j$, is defined as:

$$
C = [c_{i,j}] \in \mathbb{R}^{12n^L \times 12n^L},
$$

$$
c_{i,j} = \begin{cases} 
1, & \text{if } i, j \text{ are in conflict}, \\
0, & \text{otherwise}, 
\end{cases}
$$

$$
i, j = 1, 2, 3, \ldots, 12n^L. \quad (2)
$$

The number of rows and volumes of $C$ is the same as the number of movements. After defining conflict relationship, the key is to schedule vehicles in conflict to pass the conflict zone at different time to prevent collision.

**B. Distribution Calculation Method**

Based on the defined conflict relationship, we need to calculate longitudinal and lateral distribution of vehicles to let them pass the conflict zone without collision. In order to regulate motion of vehicles and simplify planning, we make the following assumptions:

1) Vehicles that are in conflict don’t exist in the conflict zone simultaneously.

2) Vehicles that are not in conflict and are planned to pass the conflict zone simultaneously arrive at the stop line at the same time.

Assumption 1) physically avoids collision between vehicles. Assumption 2) is achieved by the relative formation control method, which discrete and regulate vehicles’ position in the longitudinal direction (section IV-A). According to the above assumptions, vehicles are scheduled into layers. Vehicles that are in conflict are scheduled into different layers, and vehicles in the same layer have no conflict and pass the conflict zone simultaneously. The longitudinal gap between two layers is designed to let the former layer completely pass the conflict zone before the latter layer reach the stop line. Some methods have been proposed to calculate layers for vehicles in intersections with fixed lane direction, e.g., depth-first spanning tree method, and maximum matching method. In this paper, we proposed the Iterative Grouping Algorithm (IGA) to calculate layers for vehicles in intersections with flexible lane direction, shown as Algorithm 1.

The input of IGA is a sequence of vehicles with ID indexed from 1 to $N$, and the output is conflict-free longitudinal (layer assignment) and lateral (lane occupation) distribution of vehicles. IGA firstly calculates available layers for all the vehicles, considering their maximum acceleration ability and the distance towards the conflict zone. The detail of calculating available layers for vehicles will be discussed later in Section IV-A. Then, IGA iteratively assigns vehicles to layers and check their conflict relationship. A new vehicle is assigned to a layer if it is not in conflict with any other vehicles already in the layer, considering their lane occupation. Finally, IGA outputs the conflict-free longitudinal and lateral occupation results for vehicles. In IGA, the set $L$ represents the lane occupation of vehicles, where the $i$-th element $L_i$ represents the lane index that vehicle $i$ should occupy. From line 7 to line 11, IGA calculates lane occupation for vehicles. In line 10, the difference between two lane occupation results $L$ and $L'$ is calculated as follows:

$$
e = \sum_{i=1}^{N} (L_i - L'_i)^2. \quad (3)
$$

The lane occupation result that has the minimum difference with the current occupation is updated as the new lane occupation, in order to reduce unnecessary lane changes of vehicles.

**Theorem 1:** The IGA has the following properties:
1) Every vehicle will be assigned to a layer.
2) Vehicles that are assigned to the same layer are not in conflict with each other.
3) The algorithm will end in finite steps.

**Proof:** The three properties are proved as follows:

1) IGA iteratively assigns vehicles to layers and there are three situations:
   a) The layer that a vehicle is trying to be assigned to is empty, so that no conflict will exist and the vehicle is successfully assigned to the layer.
   b) The new vehicle is not in conflict with any other vehicles that are already in the layer, and the vehicle is successfully assigned to the layer.
   c) Conflict exists between the new vehicle and other vehicles that are already in the layer, and this vehicle will not be assigned to this layer. Now we consider the worst case that the vehicle is in conflict with all the other vehicles in the intersection, so that it can’t stay in the same layer with any other vehicle. Since the number of vehicles is finite in an intersection, the number of layers that the other vehicles are assigned to is also finite. After checking those finite number of layers, the new vehicle will be successfully assigned to the next available layer.

   This completes the proof of the first property.

2) This property is guaranteed by the assigning process. A vehicle will be assigned to a layer if:
   a) The layer is empty and available for this vehicle. In this case, only one vehicle exists in the layer, and no conflict exists.
   b) The vehicle is not in conflict with the vehicles that are already in the layer. In this case, no conflict exists between the new vehicles and the currently existing vehicles.
   Since vehicles are assigned to layers one by one, no conflict will exist between any two vehicles that are in the same layer.

   This completes the proof of the second property.

3) The algorithm will end if all the vehicles have been assigned to certain layers. As is provided in Proof (1), every vehicle will be assigned to a layer after finite steps, and the number of vehicles is finite. Thus, the total number of steps that are needed to assign all the vehicles to certain layers is also finite. This completes the proof of the third property.

   ■

**C. Example**

An example is provided to show the result of IGA. A sequence of ten vehicles is given as input, and their traffic movement types are shown in Fig. 2. Note that when calculating the distribution of vehicles for the example, we ignore the reachability of vehicles and all layers are available.

In the intersection scenario with flexible lane direction, these vehicles are assigned into two layers, as shown in Fig. 3. In the first layer, seven vehicles including four right-turning vehicles, two straight-going vehicles, and one left-turning vehicle have no conflict and pass the conflict zone simultaneously. In the second layer, the other three vehicles, including two straight-going vehicles and one left-turning vehicle, have no conflict and pass the conflict zone simultaneously. In the intersection with fixed lane direction, where only one type of turning is allowed for each lane, four layers are needed to evacuate the ten vehicles, as shown in Fig. 4. This indicates that cooperation at intersections with flexible lane direction can achieve fewer layers, shorter passing time, and higher efficiency.

**Remark 3:** In the cases where the arms of an intersection have different number of lanes, the proposed method is also workable. In this kind of cases, the assumption that the vehicle should start and end at the lanes with the same index doesn’t hold, because the number of starting lanes and ending lanes may be different. If the starting lanes are more than the ending lanes, vehicles from some lanes may need to merge into the same lane. In this case, a merging conflict is formed and these vehicles should be considered for conflict resolution, as shown in Fig. 5. Other cases...
where new diverging conflict and crossing conflict happen can be treated by the same way. The new conflicts will change the results of the lane and longitudinal position calculation results, but the calculation method is the same.

IV. MULTI-VEHICLE FORMATION RECONFIGURATION AND CONTROL

After being assigned to layers and lanes, vehicles need to plan trajectories to drive to their desired position. Collision may happen between vehicles that are driving on the same arm. Since the starting position and desired position of vehicles are known, we can utilize multi-vehicle formation reconfiguration method [36] to solve the conflict-free motion planning problem, by considering vehicles on the same arm as a group.

A. Bi-Level Formation Control Framework

The bi-level formation control framework is adopted to plan conflict-free trajectories for vehicles to catch up with the planned distribution, where relative motion planning is performed in the upper level to avoid collision between vehicles, and trajectory planning with spatiotemporal constraints is conducted in the lower level to calculate control inputs for vehicles. The upper-level planning is performed in Relative Coordinate System (RCS), which is a dynamic coordinate system moving with vehicles. In RCS, position of vehicles is discretized by lanes laterally and by a constant gap $d_F$ longitudinally. Every relative point in RCS corresponds to a road point in Geodetic Coordinate System (GCS).

In order to prevent collision, time is also discretized by constant time intervals $T_F$. Vehicles occupy relative points in RCS with integer coordinates at fixed time points. During time period between two adjacent time points, vehicles can move from one relative point to another, or maintain their current relative position. If there are five available relative points for a vehicle at each time step (four points around and its current point), the motion map is four-connected. Given the relative position $(x_{i,j}^{r,v}, y_{i,j}^{r,v})$ of vehicle $i$ at step $j$, the position $(x_{i,j+1}^{r,v}, y_{i,j+1}^{r,v})$ at step $j + 1$ is limited by:

$$|x_{i,j+1}^{r,v} - x_{i,j}^{r,v}| + |y_{i,j+1}^{r,v} - y_{i,j}^{r,v}| \leq 1.$$  (4)

If there are nine available relative points for a vehicle at each time step (eight points around and its current point), the motion map is eight-connected, and the motion of vehicles is regulated by:

$$|x_{i,j+1}^{r,v} - x_{i,j}^{r,v}| + |y_{i,j+1}^{r,v} - y_{i,j}^{r,v}| \leq 1,$$  (5)

$$|y_{i,j+1}^{r,v} - y_{i,j}^{r,v}| \leq 1.$$  (6)

The choice between these two motion rules (four-connected and eight-connected) depends on the parameters of vehicles, formation, and road, including vehicle following gap, lane width, etc. The motion rules should be defined before path planning of vehicles.

Denote the distance from the current position of a vehicle towards the stop line of the intersection as $d_i$ and desired speed of the formation as $v_F$, the maximum number of layers $N_i^y$ that vehicle $i$ can accelerate to take forward before reaching the stop line is:

$$N_i^y = \left\lfloor \frac{d_i}{v_F T_F + d_F} \right\rfloor,$$  (7)

where $\lfloor \cdot \rfloor$ is used to calculate the nearest integer that is not bigger than a number. Layers that are behind the most forward achievable layer for a vehicle are its available layers. The theoretical most forward available layers for all the vehicles is the most forward layer for the vehicle that is the closest to the stop line. Thus, the point whose relative coordinate $x^r$ is 0 is set on this layer. Suppose that ID of the closest vehicle is 0, the relative coordinate $x_{i}^{r,v}$ for vehicle $i$ is calculated as:

$$x_{i}^{r,v} = \lfloor \frac{d_i - d_0}{d_F} \rfloor + N_0^y,$$  (8)

where $\lfloor \cdot \rfloor$ is used to calculate the nearest integer of a number. Note that different vehicles may have the same $x^{r,v}$ if they are close to each other longitudinally. In this case, they are forced to drive to identical relative points before the algorithm begins.

The relationship between relative states and real-world states are shown in Fig. 6, where an example of movement during one time interval is provided. The red arrow in the upper part of Fig. 6 shows an oblique move (if the relative motion map is eight-connected), where the vehicle moves from relative coordinate (2, 3) to (1, 2) to join the other four vehicles and forms an interlaced five-vehicle formation.
Given the current and desired position of vehicles, we need to calculate collision-free trajectories for those vehicles to travel to their target position. This process is divided into two parts: the first part is to calculate conflict-free relative paths for vehicles in RCS, and the second part is to plan trajectories for vehicles to pass through those points one by one with spatiotemporal constraints.

B. Relative Motion Planning

The input for the multi-vehicle relative motion planning is the initial and desired position of all vehicles in RCS. Besides, the movement of vehicles need to be regulated to avoid collision when vehicles are travelling to their desired points.

The Conflict-based Searching (CBS) method is utilized to calculate collision-free relative paths for vehicles in RCS [36]. CBS method iteratively plan relative path for single vehicle, detect conflicts between vehicles, and re-plan paths for vehicles with constraints. A conflict exists between two vehicles and may lead to a collision. A constraint prevents a certain movement of vehicles. The searching process is shown in the upper part of Fig. 7, where each node represents a complete cycle of single-vehicle path planning for all the vehicles in the formation. Conflicts are detected for each node and are added as constraints for the child nodes. Each node is evaluated by the total cost for the child nodes. If there is a conflict for a child node, this node is claimed to be the optimal solution with the minimum cost.

For each node in the searching tree, individual path planning is performed for every vehicle, considering the constraints. The path planning is conducted without considering other vehicles, and may cause conflicts, which is then added as the attributes of the node. The edge between nodes represents the generating order of different nodes. Two nodes are called as parent node and child node if there is a directed edge connecting them and starting from the parent node to the child node. All the constraints of a parent node should also be added to its child node, and one of the conflicts of the parent node will be added as a new constraint to its child node. That is to say, the number of child nodes of a node is equal to the number of the parent node’s conflicts. Note that not every node is solid, because there may be conflicts and the corresponding paths calculated for the node will cause collision.

The tree starts from the root node $N_1$, which doesn’t have constraints since it has no parent node. By applying single-vehicle path planning method to the initial and target position of vehicles, a path set, a conflict set, and a cost is got. The single-vehicle path planning method utilized in this paper is the $A^*$ algorithm [39]. Child nodes are generated to the root node and the tree iterates the process until an optimal solution is found.

The CBS method is well known and widely used in the field of multi-robot path planning. The properties of CBS method including completeness, optimality, and time complexity have been proved and explained in numerous papers [40], [41], and we omit them in this paper.

C. Multi-Stage Vehicle Control

The CBS method calculates the order of relative points that a vehicle should pass at each $t_i$ in $T = \{t_i | t_i = iT, i = 0, 1, 2, 3, \ldots\}$ in RCS. The key for the control part is to ensure that vehicles arrive at their desired relative points at desired time. To do this, trajectories are first generated and lateral and longitudinal control inputs are then calculated for vehicles to conduct. Bézier curves are generated for the vehicles to pass through the corresponding road points in GCS. The trajectories of vehicles consist of several segments, and the inaccuracy during the trajectory following process may accumulate by time. To prevent severe following error which may lead to collision, a multi-stage motion control framework is proposed and vehicles will replan their control inputs after desired time intervals.

The 2-DOF vehicle model is utilized in this paper, where the center of the rear axle is chosen to represent the position of the vehicle. The yaw angle and the steer angle are represented as $\theta$ and $\delta$ respectively. $L$ represents the wheelbase of the vehicle. The state variable $z^v$ of the vehicle contains $x^v, y^v, v$ and $\theta$, where $v$ represents the velocity of the vehicle. Acceleration $a$ and steering angle $\delta$ are calculated as control inputs, and the state is calculated as:

$$
\dot{x}^v = \begin{bmatrix} v \sin(\theta) \\ v \cos(\theta) \\ a \end{bmatrix}, \quad z^v = \begin{bmatrix} x^v \\ y^v \\ \nu \tan(\delta) \end{bmatrix}. \quad (9)
$$

A linear feedback preview controller is designed for the lateral control of trajectory tracking. The preview point is chosen a distance ahead from the closest point. The vehicle looks ahead to the preview point and the closest point on the trajectory. The controller calculates the angle between the vehicle velocity and the desired velocity on the preview point, and the distance between the vehicle and the closest point as feedbacks.
As for the longitudinal control of trajectory tracking, the discretized state equation is given as:

$$\begin{align*}
\{s(k + 1) &= s(k) + v(k)dt, \\
v(k + 1) &= v(k) + a(k)dt,
\end{align*}$$

(10)

where $s(k)$, $v(k)$ and $a(k)$ represent the longitudinal position, speed and acceleration respectively. The time is discretized by sample interval $dt$. The distance that the vehicle will travel from the starting point to the final point is denoted as $S$. Since the vehicle moving in RCS should pass a sequence of relative points, $S$ consists of series of segments, whose length are denoted as $S_1, S_2, \ldots, S_N$, where $N$ is the number of segments. The time interval for the vehicle to cover each segment is $T_F$. Taking the control energy of the whole control process as the cost, the discretized longitudinal control can be described as:

$$\min \sum_{k=0}^{N^t-1} a^2(k),$$

(11)

s.t. $s(0) = 0,$

$$s(k_i) = \sum_{n=1}^{i} S_n, \quad k_i = iN^k, \quad i = 1, 2, 3, \ldots, N^t,$$

$\quad v(0) = v(N^tN^k), \quad v_F,$

$\quad v_{\min} \leq v \leq v_{\max},$

$\quad a_{\min} \leq a \leq a_{\max},$

where $N^k = \frac{T_F}{T_F}$ is the steps in one segment, $a(k)$ is the longitudinal control input (acceleration) of the vehicle at step $k$, $v(k)$ is the longitudinal speed, $v_F$ is the desired longitudinal speed of the formation, and $v_{\min}, v_{\max}, a_{\min}$ and $a_{\max}$ are the bounds of speed and acceleration.

We solve the optimal control problem in (11) and get the sequence of control input to guide the vehicle to travel desired distance at desired time. Since the linear feedback preview controller is designed for lateral control, the inaccuracy of lateral motion may cause deviation for longitudinal motion. In order to resolve the accumulated inaccuracy, the optimal problem is reformed and solved every time the vehicle completes the following of one segment at time $t = iT_F(i = 1, 2, 3, \ldots, N^t)$.

D. Example

An example is provided in Fig. 8 to show the process of formation reconfiguration. Six vehicles start from messy points and need to form a parallel three-lane formation. The initial and expected position of vehicles are shown in Fig. 8(b) and (c). The motion map for this example is eight-connected, and the reconfiguration process takes two cycles. The rectangles represent vehicles and the lines with the same color represent trajectories of certain vehicles. Fig. 8(a) shows the state and trajectories of vehicles in GCS, and the circles represent road points for vehicles to pass through. Fig. 8(b), (c), and (d) show the states and movements of vehicles in RCS, and the black circles represent discretized relative points.

V. Simulations

In this part, the simulation is conducted and the results are analyzed. The settings of the simulation are presented, and the benchmark methods are introduced. Then, numerical results and snapshots are presented to evaluate the proposed method.
Fig. 11. Snapshots of the west incoming arm at four continuous cycles of the simulation. The rectangles represent vehicles and their color shows their turning expectation, where red for right-turning, green for straight-going, and blue for left-turning. The full version video and simulation data is available online at the address: https://github.com/cmc623/flexible-lane-direction-intersection.

### Table I

Simulation Parameters

| Parameter                                      | Value  |
|-----------------------------------------------|--------|
| Safe one-lane following gap in formation      | 17.5 m |
| Safe one-lane following gap in conflict zone  | 35 m   |
| Formation switching cycle                     | 4 s    |
| Desired speed in formation                    | 10 m/s |
| Minimum speed in formation                    | 0 m/s  |
| Maximum speed of vehicle                      | 20 m/s |
| Minimum acceleration of vehicle               | -10 m/s² |
| Maximum acceleration of vehicle               | 5 m/s² |

### A. Simulation Settings

The simulation is implemented with MATLAB 2017b and SUMO 0.32.0 [42] on a personal computer with CPU Intel CORE i7-8700@3.2GHz. The average travelling time of vehicles is analyzed to evaluate the performance. The parameters chosen for this simulation are presented in Table I. The safe one-lane following gap in formation is the longitudinal distance between two adjacent points in RCS. The safe one-lane following gap in conflict zone is the following distance between vehicles when passing the conflict zone of an intersection. It is important to notice that the following distance of two layers \( d_c \) when passing the conflict zone is set twice as the safe following distance in formation. Hence, there is a transitional relative point between two layers for vehicles to temporally stay when moving in a formation. The optimization problem in (11) is solved by the GPOPS solver [43].

The scenario of simulation is a three-lane intersection with four incoming arms, as shown in Fig. 1. The length of each arm is 400 m. We conduct simulation in different input traffic volumes, from 1000 vehicle/hour to 5000 vehicle/hour, and vehicles enter the simulation scenario based on a binomial distribution, whose parameters including number of repetitions and probability of vehicle generating at each simulation step are set according to the simulation time and traffic volumes. The traffic volume of the simulation represents the volume of the whole intersection, which is counted for all the lanes of all the arms. Vehicles are generated randomly on different lanes. At each input traffic volume, we conduct simulation with four different input proportion of vehicles with different turning expectation, where the proportion of right-turning vehicles, straight-going vehicles, and left-turning vehicles is \((0.33, 0.33, 0.34), (0.25, 0.25, 0.5), (0.25, 0.5, 0.25), \) and \((0.5, 0.25, 0.25)\) respectively. At each input volume and turning proportion, we compare the results of three scenarios: unsignalized intersection with flexible lane direction, unsignalized intersection with fixed lane direction, and signalized intersection with fixed lane direction. The signal timing cycle is given in Fig. 9 and time for each phase is 20 s. In order to clear vehicles in conflict zone after each phase, a three-second clearing yellow-light phase is conducted between each of the four phases. Similar signal timing cycle design can be found in [5]. Random inputs are generated for each cycle of simulation, and the results are the average of ten cycles. The simulation time is set as 15 minutes for each cycle, so that the number of generated vehicles is about 250, 500, 750, 1000, 1250 respectively under traffic volume 1000, 2000, 3000, 4000, and 5000 vehicle/hour. Besides, the results of the first and the last 100 vehicles are removed to obtain the results of continuous traffic.

### B. Results

The results of average travelling time that vehicles take to travel on the incoming arms are shown in Fig. 10. The four figures are the results under four turning proportion of inputs.

The standard deviation under different random inputs is also
marked in the figures. From the figures we can see that our method at flexible-lane-direction scenarios achieves that best performance, under almost all the input volumes and turning proportion. The reason is that our method allows more vehicles to pass the conflict zone simultaneously than the other two methods. At low traffic volumes (from 1000 to 3000 vehicle/hour) the unsignalized cooperation methods outperform the signalized method, because vehicles don’t have to stop when arriving at the conflict zone and can maintain a higher speed in the whole journey. As the input volumes becomes higher, the performance of the unsignalized cooperation with fixed lane direction become worse than the signalized method (Fig. 10(a)), because vehicles have to slow down or even brake to achieve their desired layers to pass the intersection. In the unsignalized intersection, vehicles pass the conflict zone in layers, and the maximum number of vehicles in one layer is limited by the fixed lane direction. So when the traffic volume is high, vehicles may need to slow down to wait for catching up with their desired layers. And in the signalized intersection, although vehicles need to stop and wait for the signal phase, the maximum number of vehicles that can pass the conflict zone simultaneously is bigger, because vehicles don’t have to form layers, and can achieve a shorter following gap. This makes the performance of signalized method better than the unsignalized method in some cases. It is important to notice that the performance of the methods differs at different input proportion, but the proposed method always has the shortest travelling time results.

Snapshots of the west incoming arm at four continuous cycles are presented in Fig. 11, where the data of vehicles are recorded and presented using MATLAB. The rectangles represent vehicles and their color shows their turning expectation, where red for right-turning, green for straight-going, and blue for left-turning. The points represent trajectories of vehicles in the former 2.5 seconds. It shows the process of formation reconfiguration where vehicles adjust their position to catch up with their desired layers and lanes. Snapshots at input traffic volume 5000 vehicle/hour and turning proportion (0.33, 0.33, 0.34) of the three methods are provided in Fig. 12 to show the process of simulation. We also show the trajectories of vehicles when passing the conflict zone with lighter color. From the snapshots we can see that our method allows more vehicles to pass the conflict zone simultaneously than the other two methods, and
vehicles have to stop and wait for signal timing when using the signalized method.

VI. CONCLUSION

This paper proposes a multi-lane unsignalized intersection cooperation method that considers flexible lane direction. The two-dimensional distribution of vehicles is calculated and vehicles that are not in conflict are scheduled to pass the intersection simultaneously. The formation reconfiguration method is utilized to achieve cooperative lane changing and longitudinal position adjustment of vehicles. Simulations are conducted at different input traffic volumes and turning proportion of vehicles, and the results indicate that: (1) our method is able to cooperatively control vehicles to avoid collision and pass unsignalized intersections with flexible lane direction, and (2) the performance of our method is better than the unsignalized cooperation method with fixed lane direction and the signalized method.

Accelerating the solution-finding speed of CBS is the future direction of this research. Candidate solutions include that utilizing the Graph Neural Networks (GNN) to learn the process of CBS, and return a near-optimal solution after real-data training [44].

REFERENCES

[1] Y. Bian, S. E. Li, W. Ren, J. Wang, K. Li, and H. X. Liu, “Cooperation of multiple connected vehicles at unsignalized intersections: Distributed observation, optimization, and control,” IEEE Trans. Ind. Electron., vol. 67, no. 12, pp. 10744–10754, Dec. 2019.
[2] C. Chen, J. Wang, Q. Xu, J. Wang, and K. Li, “Mixed platoon control of automated and human-driven vehicles at a signalized intersection: Dynamical analysis and optimal control,” Transp. Res. Part C: Emerg. Technol., vol. 127, 2021, Art. no. 103138.
[3] Q. Ge et al., “Real-time coordination of connected vehicles at intersections using graphical mixed integer optimization,” IET Intell. Transport Syst., vol. 15, no. 6, pp. 795–807, 2021.
[4] P. Mirchandani and L. Head, “A real-time traffic signal control system: Architecture, algorithms, and analysis,” Transp. Res. Part C: Emerg. Technol., vol. 9, no. 6, pp. 415–432, 2001.
[5] B. Xu et al., “Cooperative method of traffic signal optimization and speed control of connected vehicles at isolated intersections,” IEEE Trans. Intell. Transp. Syst., vol. 20, no. 4, pp. 1390–1403, Apr. 2019.
[6] N. J. Goodall, B. L. Smith, and B. Park, “Traffic signal control with connected vehicles,” Transp. Res. Rec., vol. 2381, no. 1, pp. 65–72, 2013.
[7] K.-H. N. Bui and J. J. Jung, “Internet of agents framework for connected vehicles: A case study on distributed traffic control system,” J. Parallel Distrib. Comput., vol. 116, pp. 89–95, 2018.
[8] K.-H. N. Bui and J. J. Jung, “Aco-based dynamic decision making for connected vehicles in IoT system,” IEEE Trans. Ind. Informat., vol. 15, no. 10, pp. 5646–5655, Oct. 2019.
[9] K. Drosner and P. Stone, “A multiagent approach to autonomous intersection management,” J. Artif. Intell. Res., vol. 31, pp. 591–656, 2008.
[10] J. Lee and B. Park, “Development and evaluation of a cooperative vehicle intersection control algorithm under the connected vehicles environment,” IEEE Trans. Intell. Transp. Syst., vol. 13, no. 1, pp. 81–90, Mar. 2012.
[11] S. Huang, A. W. Sadek, and Y. Zhao, “Assessing the mobility and environmental benefits of reservation-based intelligent intersections using an integrated simulator,” IEEE Trans. Intell. Transp. Syst., vol. 13, no. 3, pp. 1201–1214, Sep. 2012.
[12] B. Xu et al., “Distributed conflict-free cooperation for multiple connected vehicles at unsignalized intersections,” Transp. Res. Part C: Emerg. Technol., vol. 93, pp. 322–334, 2018.
[13] H. Xu, Y. Zhang, L. Li, and W. Li, “Cooperative driving at unsignalized intersections using tree search,” IEEE Trans. Intell. Transp. Syst., vol. 21, no. 11, pp. 4563–4571, Nov. 2020.
[38] H. Xu, Y. Zhang, C. G. Cassandras, L. Li, and S. Feng, “A bi-level cooperative driving strategy allowing lane changes,” Transp. Res. Part C: Emerg. Technol., vol. 120, 2020, Art. no. 102773.

[39] P. E. Hart, N. J. Nilsson, and B. Raphael, “A formal basis for the heuristic determination of minimum cost paths,” IEEE Trans. Syst. Sci. Cybern., vol. 4, no. 2, pp. 100–107, Jul. 1968.

[40] G. Sharon, R. Stern, M. Goldenberg, and A. Felner, “The increasing cost tree search for optimal multi-agent pathfinding,” Artif. Intell., vol. 195, pp. 470–495, 2013.

[41] G. Sharon, R. Stern, A. Felner, and N. R. Sturtevant, “Conflict-based search for optimal multi-agent pathfinding,” Artif. Intell., vol. 219, pp. 40–66, 2015.

[42] P. A. Lopez et al., “Microscopic traffic simulation using sumo,” in Proc. 21st Int. Conf. Intell. Transp. Syst., 2018, pp. 2575–2582.

[43] M. A. Patterson and A. V. Rao, “Gpops-II: A matlab software for solving multiple-phase optimal control problems using hp-adaptive gaussian quadrature collocation methods and sparse nonlinear programming,” ACM Trans. Math. Softw., vol. 41, no. 1, pp. 1–37, 2014.

[44] Q. Li, F. Gama, A. Ribeiro, and A. Prorok, “Graph neural networks for decentralized multi-robot path planning,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., 2020, pp. 11785–11792.

Mengchi Cai (Student Member, IEEE) received the B.E. degree from Tsinghua University, Beijing, China, in 2018. He is currently a Ph.D. candidate in mechanical engineering with the School of Vehicle and Mobility, Tsinghua University. He was the recipient of the Comprehensively Excellent Scholarship of Tsinghua University. His research interests include connected and automated vehicles, multi-lane formation control, and multi-vehicle cooperation at intersections.

Jiawei Wang (Student Member, IEEE) received the B.E. degree in 2018 from Tsinghua University, Beijing, China, where he is currently working toward the Ph.D. degree in mechanical engineering with the School of Vehicle and Mobility. His research interests include connected automated vehicles, distributed control and optimization, and data-driven control. He was the recipient of the National Scholarship in Tsinghua University. He was also the recipient of the Best Paper Award at the 18th COTA International Conference of Transportation Professionals.

Keqiang Li received the B.Tech. degree from the Tsinghua University of China, Beijing, China, in 1985, and the M.S. and Ph.D. degrees in mechanical engineering from the Chongqing University of China, Chongqing, China, in 1988 and 1995, respectively. He is currently a Professor with the School of Vehicle and Mobility, Tsinghua University. His main research interests include automotive control system, driver assistance system, and networked dynamics and control. He is leading the national key project on CAV's Intelligent and Connected Vehicles in China. He was the recipient of the Changjiang Scholar Program Professor, National Award for Technological Invention in China.

Qing Xu received the B.S., M.S., and Ph.D. degrees in automotive engineering from Beihang University, Beijing, China, in 2006, 2008, and 2014, respectively. During the Ph.D. research, he was a Visiting Scholar with the University of Illinois at Urbana–Champaign, Champaign, IL, USA. From 2014 to 2016, he had his postdoctoral research with Tsinghua University. He is currently an Assistant Research Professor with the School of Vehicle and Mobility, Tsinghua University. His main research focuses on the decision and control of intelligent vehicles.

Jianqiang Wang received the B.Tech. and M.S. degrees from the Jilin University of Technology, Changchun, China, in 1994 and 1997, respectively, and the Ph.D. degree from Jilin University, Changchun, China, in 2002. He is currently a Professor with the School of Vehicle and Mobility, Tsinghua University, Beijing, China. His research interests include intelligent vehicles, driving assistance systems, and driver behavior. He was the recipient of the Changjiang Scholar Program Professor in 2017, Distinguished Young Scientists of NSF China in 2016, and New Century Excellent Talents in 2008.

Chaoyi Chen (Student Member, IEEE) received the B.E. degree from Tsinghua University, Beijing, China, in 2016, and the M.S. degree from Tsinghua University, and RWTH Aachen University, Aachen, Germany, in 2019. He is currently working toward the Ph.D. degree in mechanical engineering with the School of Vehicle and Mobility, Tsinghua University. His research interests include vehicular network, control theory, and cooperative control. He was the recipient of the Scholarship of Strategic Partnership RWTH Aachen University and Tsinghua University.

Xiangbin Wu received the M.S. degree on computer science from North-Western Polytechnic University, Xi’an, China, in 2000. He is currently an Intel Principal Engineer, Senior Research Director of Autonomous Systems Lab of Intel Labs China, and Principal Investigator of Intel Collaborative Research Institute on Intelligent and Automated Connected Vehicles. His main research interests include autonomous driving, smart transportation, edge computing, and robotics.

Jiawei Wang (Student Member, IEEE) received the B.E. degree in 2018 from Tsinghua University, Beijing, China, where he is currently working toward the Ph.D. degree in mechanical engineering with the School of Vehicle and Mobility. His research interests include connected automated vehicles, distributed control and optimization, and data-driven control. He was the recipient of the National Scholarship in Tsinghua University. He was also the recipient of the Best Paper Award at the 18th COTA International Conference of Transportation Professionals.

Keqiang Li received the B.Tech. degree from the Tsinghua University of China, Beijing, China, in 1985, and the M.S. and Ph.D. degrees in mechanical engineering from the Chongqing University of China, Chongqing, China, in 1988 and 1995, respectively. He is currently a Professor with the School of Vehicle and Mobility, Tsinghua University. His main research interests include automotive control system, driver assistance system, and networked dynamics and control. He is leading the national key project on CAV's Intelligent and Connected Vehicles in China. He was the recipient of the Changjiang Scholar Program Professor, National Award for Technological Invention in China.

Qing Xu received the B.S., M.S., and Ph.D. degrees in automotive engineering from Beihang University, Beijing, China, in 2006, 2008, and 2014, respectively. During the Ph.D. research, he was a Visiting Scholar with the University of Illinois at Urbana–Champaign, Champaign, IL, USA. From 2014 to 2016, he had his postdoctoral research with Tsinghua University. He is currently an Assistant Research Professor with the School of Vehicle and Mobility, Tsinghua University. His main research focuses on the decision and control of intelligent vehicles.

Jianqiang Wang received the B.Tech. and M.S. degrees from the Jilin University of Technology, Changchun, China, in 1994 and 1997, respectively, and the Ph.D. degree from Jilin University, Changchun, China, in 2002. He is currently a Professor with the School of Vehicle and Mobility, Tsinghua University, Beijing, China. His research interests include intelligent vehicles, driving assistance systems, and driver behavior. He was the recipient of the Changjiang Scholar Program Professor in 2017, Distinguished Young Scientists of NSF China in 2016, and New Century Excellent Talents in 2008.

Chaoyi Chen (Student Member, IEEE) received the B.E. degree from Tsinghua University, Beijing, China, in 2016, and the M.S. degree from Tsinghua University, and RWTH Aachen University, Aachen, Germany, in 2019. He is currently working toward the Ph.D. degree in mechanical engineering with the School of Vehicle and Mobility, Tsinghua University. His research interests include vehicular network, control theory, and cooperative control. He was the recipient of the Scholarship of Strategic Partnership RWTH Aachen University and Tsinghua University.

Xiangbin Wu received the M.S. degree on computer science from North-Western Polytechnic University, Xi’an, China, in 2000. He is currently an Intel Principal Engineer, Senior Research Director of Autonomous Systems Lab of Intel Labs China, and Principal Investigator of Intel Collaborative Research Institute on Intelligent and Automated Connected Vehicles. His main research interests include autonomous driving, smart transportation, edge computing, and robotics.