# Autonomous intersection management for connected and automated vehicles: a lane-based method 

Wei Wu, Yang Liu, Wei Liu, Fangni Zhang, Vinayak Dixit, S. Travis Waller


#### Abstract

Most existing studies on autonomous intersection management (AIM) often focus on algorithms to accommodate conflicts among vehicles by assuming that the entrance lane and the exit lane of vehicles are exogenous inputs. This paper shows that allowing entrance lanes and exit lanes to be optimized can significantly improve traffic efficiency. In particular, this paper proposes "all-direction" lanes, where left-turn, through, and right-turn traffic is all allowed at the same lane. We develop two methods for optimizing entering time (i.e., when to enter the intersection) and route choice decisions (i.e., entrance lane and exit lane), including the sliding-time-window-based global optimum (GO-STW) and the first-come-first-served method with optimal route choices (FCFS-R). The developed lane-based methods can be formulated as mixed integer linear programming (MILP) problems, which can be solved using the CPLEX solver. A heuristic is further adopted to solve the MILP model in a timely manner, which illustrates the potential real-time applicability of the proposed method. Numerical analysis is conducted to examine performance and effectiveness of the proposed methods and heuristic. We found that the optimization of lane/route choices is often more critical than entering time.


Index Terms-Autonomous intersection, connected and autonomous vehicles, lane-based method, sliding time window

## I. Introduction

Autonomous vehicles promise a fundamental revolution in mobility [1, 2], which brings enormous challenges and opportunities to planning and management of traffic systems [3]. By using connected vehicle technology [4, 5], autonomous vehicles (AVs) are currently being tested and examined in a variety of scenarios [6, 7]. How fully connected and autonomous vehicles will cross intersections, which is termed as autonomous intersection management (AIM), is one

[^0]of the most focused topics in recent years [8-11]. It is widely expected that autonomous vehicles can communicate and cooperate with each other, and traditional traffic signals may no longer be needed [12, 13].

As far as the authors know, Dresner and Stone are among the first to propose AIM, where the "First Come First Served" (FCFS) policy has been adopted to process vehicle requests [14]. After that, a large body of studies related to AIM have been proposed. Most of these studies fall into two categories: grid-based method and conflict point based method. For the grid-based method, the intersection is divided into an $n \times n$ grid of reservation tiles [14-16]. If an AV plans to cross the intersection, all the grids located on this AV's path must open to this vehicle successively. In order to deal with the situation that two or more AVs may request to use the same grid at the same time, FCFS-based method [17], platoon-based method [18], auction-based method [19, 20], and reservation method [21] have been examined. However, grid-based method often requires significant computation efforts. To reduce computing time, conflict point based method has been proposed, which only considers the vehicle prioritization at the conflict points [12, 22]. However, conflict point based methods ignore the physical size of vehicles and conflict types [8, 23, 24].

The aforementioned methods and studies often assume that the entrance lane and exit lane of all AVs at the intersection are exogenous inputs. Moreover, lanes are dedicated to certain traffic movements, e.g., through movement solely. Based on the predetermined entrance lane and the exit lane of an AV, existing studies considered the planning of vehicle trajectory inside the intersection for each AV to improve efficiency and avoid conflict. However, in a connected and autonomous intersection environment, highly automated vehicles can cooperate and coordinate with each other when crossing the intersection. In this context, it is not necessary to restrict lanes to certain traffic movements (i.e., a lane may accommodate all left-turn, through, and right-turn traffic, which is termed as "all-direction" lane hereinafter), and one can optimize the allocations of entrance lane and exit lane for all AVs. We further illustrate this with the following example.

Consider the left-turn movement with east entrance lane and the south exit lane in Fig. 1 (from east to south). At traditional intersections, the entrance lane and exit lane are often given, i.e., from lane 1 at east to lane 1 at south (please refer to Fig.1(a), where lane 2 at east is for through traffic and lane 2 at south is for through traffic). However, in a connected and automated traffic environment, AV can choose either lane 1 or lane 2 at east as the entrance lane, and either lane 1 or
lane 2 at south to make the left-turn (please refer to Fig.1(b)). Both entrance lanes (at east) in Fig.1(b) can be regarded as "all-direction" lanes. We can foresee that intersection operation as those in Fig.1(b) with traditional vehicles can lead to significant amount of conflict points and is unrealistic for human drivers. Thus, conventional methods assign leftside lanes to left-turn vehicles, and right-side lanes to rightturn traffic, and assign middle lanes (sometimes all lanes) to through traffic. However, the mechanism of autonomous intersection management (AIM) can be completely different from the traditional traffic signal lights with human-driven vehicles. Conflicts between AVs in the intersections can be avoided through vehicle coordination. The "all-direction" lanes in a fully connected and autonomous vehicle environment are as shown in Fig.1(b).


Fig.1. Comparison the lane choice between the previous and this research
We further illustrate the potential efficiency gain from alldirection lanes with a simple example (efficiency metrics are listed in Table I). We consider the through and left-turn traffic at an approach with two entrance lanes (other approaches are omitted here for simplicity). There are four lane setting schemes, where the lane markings are listed in Table I (Scheme 1 is the most inflexible one where each lane is dedicated to a single traffic movement, and Scheme 4 is the most flexible one with two "all-direction" lanes). Suppose the mean traffic arrival rate at the intersection is $\Phi$ and the arrival
rate follows the Poisson distribution, and the service rate of a lane is $\Psi$. The two rates $\Phi$ and $\Psi$ have the same unit, i.e., vehicles per second. This is similar to an M/D/c queuing model, where the system has c servers, a Poisson arrival process, a deterministic (fixed) service time (i.e., the server has a fixed capacity). For illustration, we consider five demand cases, as summarized in Table I. We can compute average delays in Table I for each lane setting scheme based on queuing theory. The delays in Table I are based on that the traffic is optimally assigned to the two lanes. For instance, when we have $75 \%$ through traffic, under Scheme 2, 25\% and $50 \%$ through traffic should be allocated to the first and second lanes, respectively, and $25 \%$ left-turn traffic is allocated to the first lane (the optimal allocation).

From Table I, it is evident that Scheme 4 (with two "alldirection" lanes) always yields the smallest delay (for the five demand cases), while Scheme 2 and Scheme 3 (with one "alldirection" lane) can yield the smallest delay for many cases with the help of one "all-direction" lane, and Scheme 1 with the smallest flexibility always yields the largest delay. This example illustrates the potential of "all-direction" lanes to improve the intersection traffic efficiency by appropriately allocating the entrance or exit lane to vehicles. Note that the results in Table I cannot be used to reflect the exact efficiency gain from the proposed method (based on "all-direction" lanes), which indeed should be examined case by case. In reality, the demand conditions will be more complicated than the demand cases considered in the example. There will be potential conflict points with the "all-direction" lane design, especially when traffic from different approaches is considered simultaneously. This study makes the first attempt to develop a lane-based AIM methodology with lane/route allocation or choices, which can more efficiently utilize the space of the intersection and improve traffic efficiency while avoiding any potential conflicts. We refer to the mentioned lane or route choices as "lane-based route choices".

TABLE I
Performance Comparison of All-Direction Lane Scheme with Other Schemes (minimum delays in bold)


|  | $0.25 \times \Phi$ | $0.25 \times \Phi$ |  |  |
| :---: | :---: | :---: | :---: | :---: |
| 25\% through-traffic and 75\% \% | $\overline{2 \Psi(4 \Psi-\Phi)}$ | $\overline{2 \Psi(4 \Psi-\Phi)}$ | $\Phi$ | $\Phi$ |
| left-turn traffic: average delay | $+\frac{0.75 \times 3 \Phi}{2 \Psi(4 \Psi-3 \Phi)}$ | $+\frac{0.75 \times 3 \Phi}{2 \Psi(4 \Psi-3 \Phi)}$ | $\overline{2 \Psi(2 \Psi-\Phi)}$ | $\overline{2 \Psi(2 \Psi-\Phi)}$ |
| $0 \%$ through-traffic and $100 \%$ leftturn traffic: average delay | $\frac{\Phi}{2 \Psi(\Psi-\Phi)}$ | $\frac{\Phi}{2 \Psi(\Psi-\Phi)}$ | $\frac{\Phi}{2 \boldsymbol{\Psi}(2 \boldsymbol{\Psi}-\Phi)}$ | $\frac{\Phi}{2 \Psi(2 \Psi-\Phi)}$ |

In particular, this paper develops methods to determine the time point that an $A V$ should enter the intersection (i.e., leave the lane and drive into the intersection area), through which lane the AV should enter the intersection, and through which lane the AV should leave the intersection, where all lanes are "all-direction" lanes. We propose two methods to determine the aforementioned decision variables. First, we propose the sliding-time-window-based global optimum (GO-STW) method, where the time horizon is discretized into multiple time intervals. At each time step, we solve the optimal entering time and lane/route choices for AVs within the given time interval. Adding the time-window helps control of computation time, where the computation time is related to the length of the time-window. Second, we propose the first-come-first-served method with optimal route choices (FCFSR ), where an earlier AV always has the priority to enter the intersection and we optimize the entering time (to cross the stop line at the intersection) and lane/route choices of the AVs. When optimizing the lane/route choices, we also set a time window to restrict the problem size. In the numerical study, we compare traffic efficiency and computation time of the proposed methods. We also compare them with a benchmark case where first-come-first-served principle is adopted and lane/route choice is not optimized.

This study also explicitly considers the physical size of a vehicle, which is more realistic than conflict point based methods (now we have "conflict areas"). In this context, we show that the proposed methods can be formulated as Mixed Integer Linear Programming (MILP) problems that can be solved by the CPLEX solver. Due to the NP-hard feature of proposed problems, a heuristic procedure is further developed to solve the proposed MILP in a timely manner.

The remainder of this paper is organized as follows. Section II presents basic descriptions and formulations for the intersection with all-direction lanes. Section III presents the methods to optimize the route choice/allocation and the intersection entering time for each upcoming vehicle, and then presents a heuristic procedure for solving the problems. Section IV conducts numerical analysis based on a large cruciform intersection with four entrance lanes and four exit lanes along each direction and illustrates the performance of the proposed methods. Section V concludes the paper.

## II. Intersections With All-Direction Lanes

## A. Problem Description, Assumptions and Notations

We focus on one single intersection. We consider all vehicles are autonomous and fully connected. Each entrance lane to the intersection is an "all-direction" lane (i.e., left-
turn/through/right-turn traffic is all allowed to use it), each exit lane can be used by vehicles from all other approaches. Each AV will be assigned an entrance lane, an exit lane, and an entering time (as well as the movement/trajectory within the intersection) and AVs will follow the assigned time and routes. When determining the entrance lane, the exit lane, and the entering time of each AV, we should ensure that conflicts will never occur. We further have the following assumptions.

Assumption 1: The vehicle speed is constant when crossing an intersection. This simplifies the travel time calculation when a vehicle traverses an intersection.

Assumption 2: Once a vehicle enters an intersection, it moves smoothly towards its exit lane and is not allowed to stop. This ensures that when a vehicle is allowed to enter an intersection, the right-of-way along its movement will be guaranteed under the entering time and route choice solution.

Assumption 3: Entering time at the intersection, entrance lane, exit lane information are provided to the AV in advance. AV will shift to the assigned entrance lane in advance. The lane-changing process (if any) is not explicitly considered.

Assumption 1 and Assumption 2 assume constant speed and smooth movement within the intersection, which simplifies the travel time calculation within an intersection. This simplification is adopted in the related literature to avoid tedious calculation of travel time, which is used in many AIM (autonomous intersection management) related studies [8, 2528]. However, acceleration and deceleration can be incorporated as long as the AV speed profile is defined. Extensions to incorporate time-varying speed is briefly discussed in Appendix.

Assumption 3 defines the information availability for the autonomous vehicles, which relies on a fully connected and autonomous traffic environment [29, 30]. While lanechanging process is not explicitly modeled (Assumption 3), we consider that lane-changing may cause inconvenience, delay or cost to travelers and the system (as discussed in Section III-B1) and some lane-changing may not be allowed (as discussed in Section III-B2).

Before moving to the models, we list the main notations in Table II and some user-defined parameters in Table III (parameter values will be further discussed in the texts).

TABLE II
List of Main Notations

| Sets | Descriptions |
| :---: | :--- |
| $C$ | The set of all upcoming vehicles |
| $C_{A}$ | The set of sorted vehicles from direction $A$, where $A \in O$ |
|  | The subsets of $C$ (the full set of upcoming vehicles). Set $C_{b}$ |
| $C_{b}\left(C_{b}^{\prime}\right)$ | includes those vehicles with fixed entrance lanes. Set $C_{b}^{\prime}$ <br> includes those vehicles with fixed exit lanes. |
| $D$ | The set of exiting directions, where $D=\{E, W, S, N\}$ |


| G | The set of all grids of an intersection |
| :---: | :---: |
| 0 | The set of all approaching directions, where $O=\{E, W, S, N\}$ |
| $R$ | The set of all routes |
| Parameters | Descriptions |
| $\alpha, \beta, p, q$ | The parameters for defining an elliptic curve for turning movements |
| $\mu_{a b}^{r}$ | A binary variable, where $\mu_{a b}^{r}=1$ means that route $r$ traverses the grid $g_{a b}$, and zero otherwise, and $r \in R$. |
| $\omega$ | A weighting factor in the objective function, $\omega \in[0,1]$ |
| $B_{i}\left(B_{i}^{\prime}\right)$ | $B_{i}$ is the index of original approach lane of vehicle $i$, and $B_{i}^{\prime}$ is the index of original exit lane of vehicle $i$, where $i \in C$. |
| $d_{c}$ | The length of a vehicle (m) |
| $d_{g}$ | The side length of a grid (m) |
| $d_{r}$ | The lane width (m) |
| $E, W, S, N$ | The index of the four directions of an intersection. |
| $g_{a b}$ | The index for the gird, where $a$ and $b$ indicate the row and column, and $g_{a b} \in G$. |
| $I_{i}\left(I_{i}^{\prime}\right)$ | $I_{i}$ is the direction of the approach of vehicle $i$, and $I_{i}^{\prime}$ is direction of the exit of vehicle $i$, where $I_{i} \in O, I_{i}^{\prime} \in D, i \in C$. |
| $L_{j}^{K}$ | The index for the $j^{\text {th }}$ lane of the exit $K$, where $j \in$ $\left\{1,2, \ldots, m_{K}\right\}, K \in D$. |
| $l_{j}^{A}$ | The index for the $j^{\text {th }}$ lane of the approach $A$, where $j \in$ $\left\{1,2, \ldots, n_{A}\right\}, A \in O$. |
| $l_{r}\left(l_{r}^{\prime}\right)$ | $l_{r}$ is the index of the entrance lane of route $r, l_{r}^{\prime}$ is the index of the exit lane of route $r$, where $r \in R$ |
| $M, M_{1}, M_{2}$ | Large positive numbers |
| $m_{K}$ | The total number of lanes for the exit $K$, where $K \in D$ |
| $n_{A}$ | The total number of lanes for the approach $A$, where $A \in O$ |
| $T_{a b}^{r, 1}$ | The time between the vehicle entering the grid $g_{a b}$ along route $r$ and the vehicle crossing the stop line at the intersection, where $r \in R$. (sec) |
| $T_{a b}^{r, 2}$ | The time between the vehicle exiting the grid $g_{a b}$ along route $r$ and the vehicle crossing the stop line, where $r \in R$. (sec) |
| $t_{i}$ | $t_{i}$ is the planned arrival time at the stop line of the intersection for vehicle $i$. |
| $v$ | The travel speed of vehicle traversing an intersection (m/s) |
| Variables | Descriptions |
| $\lambda_{r}^{i}$ | A binary variable, $\lambda_{r}^{i}=1$ when vehicle $i$ chooses route $r$ and zero otherwise, where $i \in C, r \in R$ |
| $\rho_{a b}^{i, j}$ | A binary variable, $\rho_{a b}^{i, j}=1$ when vehicle $j$ traverses the grid $g_{a b}$ earlier than vehicle $i$ and zero otherwise, where $i, j \in C$ |
| $\sigma_{a b}^{i}$ | A binary variable, $\sigma_{a b}^{i}=1$ when vehicle $i$ traverses the grid $g_{a b}$ and zero otherwise, where $i \in C$ |
| $\tau_{a b}^{i}$ | The time point for a vehicle to enter the grid $g_{a b}, i \in C$. |
| $\varphi_{a b}^{i}$ | The time point for a vehicle to exit the grid $g_{a b}$, where $i \in C$. |
| $J_{i}\left(J_{i}^{\prime}\right)$ | $J_{i}$ is the index of chosen approach lane of vehicle $i, J_{i}^{\prime}$ is the index of chosen exit lane of vehicle $i$, where $i \in C$. |
| $t_{i}^{\prime}$ | The time point for vehicle $i$ to enter the intersection |

TABLE III
LIST OF USER-DEFINED PARAMETERS

| $Z$ | A positive integer to discrete the intersection, $Z=1$. |
| :---: | :--- |
| $M$ | A large positive number, $M=10$ |
| $M_{1}\left(M_{2}\right)$ | Two large positive numbers, $M_{2} \gg M_{1}, M_{1}=10^{3}, M_{2}=10^{6}$ |
| $\Delta t$ | The length of the sliding time window, $\Delta t=10 \mathrm{~s}$ |
| $\mu$ | A predetermined parameter for heuristic procedure, usually |
| $\eta_{j}^{i}$ | A binary variable. $\eta_{j}^{i}=0$ when $J_{j}<J_{i}, \eta_{j}^{i}=1$ when $J_{j} \geq J_{i}$ |
| $z_{1, i}, z_{2, i}$, | Four positive variables for model linearization. |
| $z_{1, i}^{\prime}, z_{2, i}^{\prime}$ | The cut down value for heuristic procedure in the $n^{t h}$ |
| $\theta^{(n)}$ | iteration. |

## B. Intersection Discretization and Conflict Area

We now discuss how to incorporate the physical size of vehicles and formulate the potential conflict areas within an intersection. First, we discretize the intersection into multiple square grids. Let $d_{r}$ denote the lane width and $d_{g}$ denote the length of the grid side. We let $d_{g}=d_{r} / Z$, where $Z$ is a positive integer. A grid can only be occupied by one vehicle at a time to avoid conflicts. The grid side length should not be too large to allow fine and efficient use of the intersection (e.g., the side length of a grid should be less than the vehicle length). However, the grid side length should not be too small in order to save computation time. A grid is labeled by its row and column, e.g., $g_{a b}$ indicates the grid of row $a$ and column $b$.

We now further discuss how to identify the grids where potential conflicts between vehicles may arise. When taking the physical size of vehicles into consideration, the conflicts between vehicles are significantly different from the situation where a vehicle is treated as a point. The comparison is shown in Fig.2. In Fig.2(a), vehicles are treated as points, and thus the potential conflict location for two vehicle trajectories is a point (see the red point in Fig.2(a)). However, when we consider the physical size of vehicles, to maintain safe clearance between vehicles, we consider a "trajectory band" that is occupied by a vehicle, with a width equal to the lane width (shown in Fig.2(b), Fig.3, and Fig.4). Conflicts between two vehicles can arise in an area (conflicts of two trajectory bands) rather than at a single point (indicated by the red grids in Fig.2(b)). Since a vehicle can be potentially assigned to any entrance lane and any exit lane, a large number of potential conflicts may arise. This raises computational challenges, especially when the number of vehicles in consideration is large. Fig.2(b) illustrates the conflict area between two vehicles (one vehicle from East to West and the other from South to West). The conflict area is apparently different from a conflict-point-based intersection modeling shown in Fig.2(a).


Fig.2. Comparison of conflicts between two different intersections
To identify the conflict area (e.g., the red grids in Fig.2(b)), we firstly should identify the trajectory band of the vehicles. We now discuss trajectory band (as well as routes) within an intersection and how to model the potential conflicts. For
given entrance lane and exit lane, a unique route (or "trajectory band") is determined, as shown in Fig.3. Generally speaking, there are two types of routes within an intersection, i.e., straight lines and non-straight curves. For example, from entrance lane $l_{2}^{S}$ to exit lane $L_{2}^{N}$ and from entrance lane $l_{2}^{S}$ to exit lane $L_{1}^{N}$ in Fig.3, the routes are straight lines, which may or may not be parallel the road. From entrance lane $l_{2}^{S}$ to exit lane $L_{1}^{W}$ for left-turn, the route is a non-straight curve. Moreover, to accommodate vehicle size, the route of a vehicle within the intersection corresponds to a trajectory band. The trajectory band has a left-boundary and right-boundary (as indicated in Fig. 3 and Fig.4), where the distance between two boundaries is assumed as the lane width (this ensures a safety guarantee that is similar to that for vehicles on roadways).

It is straightforward to define straight routes and their trajectory bands and how they intersect with the grids within the intersection, of which the details are omitted here. For the non-straight curves for turning movements, elliptic curve is adopted (similar to, e.g., [10]), i.e.,
$\frac{(x-\alpha)^{2}}{p^{2}}+\frac{(y-\beta)^{2}}{q^{2}}=1$
where $(x, y)$ is a point on the elliptic curve (turning vehicles should follow this curve), and $\alpha, \beta, p$ and $q$ are parameters for defining the elliptic curve.


Fig.3. The possible lane-based trajectory band within the intersection


Fig.4. An example for critical points along the trajectory band of a vehicle

We also define the curves for the left-boundary and rightboundary of the trajectory band within the intersection. For example, for the left-turn movement (from south to west in Fig.3), $\alpha, \beta$ are the same for both the left-boundary and rightboundary, which are given as follows:
$(\alpha, \beta)=\left(-d_{r} \cdot \max \left(l_{n}^{N}, L_{m}^{S}\right),-d_{r} \cdot \max \left(l_{n}^{W}, L_{m}^{E}\right)\right)$
while $p$ and $q$ for the left-boundary and right-boundary are different, i.e., for the left-boundary,
$p=d_{r} \cdot\left(\max \left(l_{n}^{N}, L_{m}^{S}\right)+l_{i-1}^{S}\right) \quad ; \quad q=d_{r} \cdot\left(\max \left(l_{n}^{W}, L_{m}^{E}\right)+\right.$
$\left.L_{j-1}^{W}\right)$
and for the right-boundary,
$p=d_{r} \cdot\left(\max \left(l_{n}^{N}, L_{m}^{S}\right)+l_{i}^{S}\right) ; q=d_{r} \cdot\left(\max \left(l_{n}^{W}, L_{m}^{E}\right)+L_{j}^{W}\right)$
where $l_{i}^{S} \in\left\{l_{1}^{S}, l_{2}^{S}, \ldots, l_{n}^{S}\right\}$ and $L_{j}^{W} \in\left\{L_{1}^{W}, L_{2}^{W}, \ldots, L_{m}^{W}\right\}$. The setting of $p$ and $q$ is consistent with the safety guarantee that a vehicle will occupy a width of $d_{r}$ (i.e., the lane width).

## C. Critical Points Along the Trajectory Band

We now discuss how a trajectory band will cross the defined grids. This process is critical since we need to ensure that a grid will be occupied by at most one vehicle at a time. By identifying how the trajectory band of a vehicle will cross a grid, we can determine entering/exiting times to/from the grid for a vehicle. We summarize the key steps to determine how the trajectory band will cross grids and how to determine the travel time associated with a grid in the following.

Step 1: Identify the critical points along the trajectory band. Based on the trajectory band identified by the curve functions for left-boundary and right-boundary (indicated in Fig.4), we can identify the critical points along the trajectory band. There are three groups of points: intersection points between grids and left-boundary curve (indicated with red dots in Fig.4); intersection points between grids and right-boundary curve (green dots); and vertices of grids that are covered by the trajectory band (blue dots). Identifying these critical points is sufficient for determining the entering and exiting times of a vehicle into/from a grid, which correspond to the "firstentering" and "last-exiting" points to be identified in Step 3.

Step 2: Identify the sequence of the critical points to be crossed/covered by the vehicle along the trajectory band. All critical points can be projected to the left-boundary or the right-boundary of the trajectory band (a few examples are depicted in Fig.4, where the critical points are projected to the right-boundary). The relative locations of the projected points along the right-boundary reflect the sequence of the corresponding critical points to be crossed by the vehicle. For example, for a left-turn movement from south entrance lane $l_{2}^{S}$ to west exit lane $L_{2}^{W}$ in Fig.3, there are around 30 critical
points to be projected to the right-boundary (with four entrance lanes and four exit lanes in each direction, a lane width of 3 meters, and a grid side length of 3 meters).

Step 3: Identify the "first" and "last" critical points for each grid and calculate the travel time. With the critical point set established in Step 1 and their sequence determined in Step 2, we can identify the "first" and "last" critical points for each grid, i.e., the first point of a grid encountered by the vehicle, and the last point of a grid that is occupied by the vehicle before it leaves the grid. The distance covered by the vehicle in the grid is equal to the vehicle length plus the curve length between the projected points of the "first" and "last" critical points. Thus, the travel time in the grid can be determined accordingly given the speed profile.

## III. Optimization Model

We now turn to present the formulations to optimize the lane-based "route choices" for AVs at the intersection. Section III-A starts with a basic model without detailed consideration of lane-changing behaviors, and Section III-B further extends the basic model by incorporating effects or constraints with respect to lane-changing. Furthermore, two methods to determine the entering times and route choices of vehicles are introduced in Section III-C. A heuristic to provide fast solutions is introduced in Section III-D.

## A. Laned-based Route Choice/Allocation Model

The overall objective of the basic model is to minimize the delay at the intersection, subject to constraints to be discussed below. The basic model assumes that AVs can always shift to their assigned routes from their original planned routes (which should be well planned in advance), and the feasibility and influences of lane-changing are not explicitly considered (these will be relaxed in Section III-B).

## 1) The objective function and decision variables

The objective of the route choice optimization problem is to minimize the total delay of all vehicles. The delay for a vehicle is defined as the difference between the time point that the vehicle is allowed to enter the intersection and the planned time point that the vehicle arrives at the intersection. The objective function can be written as:
$\min \sum_{i}\left(t_{i}^{\prime}-t_{i}\right), \quad i \in C$
where $C$ is the set of all upcoming vehicles from all approaches, $t_{i}^{\prime}$ is the time point that vehicle $i$ is allowed to enter the intersection, and $t_{i}$ is the original planned arrival time at the stop line of vehicle $i$.

The decision variables include: (i) the time points that each vehicle is allowed to enter the intersection, i.e., $t_{i}^{\prime}, i \in C$, and (ii) the route allocation for all vehicles, i.e., $\lambda_{r}^{i}$, where $\lambda_{r}^{i}=1$ if vehicle $i$ chooses route $r$ and zero otherwise, $i \in C, r \in R$. Note that each route is associated with an entrance lane and an exit lane, and is termed as the "lane-based route".

## 2) Problem constraints

(Arrival time) The time for a vehicle to enter the intersection should be no earlier than the planned arrival time at the stop line, i.e.,
$t_{i}^{\prime} \geq t_{i} \geq 0, \quad i \in C$
(Route choice) Each vehicle chooses exactly one route, i.e.,
$\sum_{r} \lambda_{r}^{i}=1, i \in C$
(Correspondence between vehicles and grids) Let $\mu_{a b}^{r}$ be a binary variable indicating correspondence between routes and grids, where $\mu_{a b}^{r}=1$ means that route $r$ crosses grid $g_{a b}$, and zero otherwise, where $r \in R, g_{a b} \in G$. The correspondence between vehicles and grids can be determined by
$\sigma_{a b}^{i}=\sum_{r} \lambda_{r}^{i} \mu_{a b}^{r}$
where $\sigma_{a b}^{i}$ is a binary variable, which equals one if vehicle $i$ crosses grid $g_{a b}$ and zero otherwise, $i \in C$, and $g_{a b} \in G$.
(Travel time) The time points of entering and exiting a grid for a vehicle, denoted by $\tau_{a b}^{i}$ and $\varphi_{a b}^{i}$, respectively, are
if $\sigma_{a b}^{i}=1$, then $\tau_{a b}^{i}=t_{i}^{\prime}+\sum_{r} \lambda_{r}^{i} T_{a b}^{r, 1}$
if $\sigma_{a b}^{i}=1$, then $\varphi_{a b}^{i}=t_{i}^{\prime}+\sum_{r} \lambda_{r}^{i} T_{a b}^{r, 2}$
where $T_{a b}^{r, 1}$ is the time length between a vehicle's entering time at the intersection (entering the stop line) and the entering time at grid $g_{a b}$ along route $r$ (associated with the trajectory band discussed in Section II), and $T_{a b}^{r, 2}$ is the time between a vehicle's exiting time from grid $g_{a b}$ along route $r$ and the entering time at the stop line, where $r \in R$. The calculation of these time durations is based on Section II-C.

The "travel time" constraints in Eqs. (9) and (10) can be rewritten into linear forms, i.e.,

$$
\begin{align*}
& t_{i}^{\prime}+\sum_{r} \lambda_{r}^{i} T_{a b}^{r, 1}-M\left(1-\sigma_{a b}^{i}\right) \leq \tau_{a b}^{i} \leq t_{i}^{\prime}+\sum_{r} \lambda_{r}^{i} T_{a b}^{r, 1}+ \\
& M\left(1-\sigma_{a b}^{i}\right)  \tag{11}\\
& t_{i}^{\prime}+\sum_{r} \lambda_{r}^{i} T_{a b}^{r, 2}-M\left(1-\sigma_{a b}^{i}\right) \leq \varphi_{a b}^{i} \leq t_{i}^{\prime}+\sum_{r} \lambda_{r}^{i} T_{a b}^{r, 2}+ \\
& M\left(1-\sigma_{a b}^{i}\right) \tag{12}
\end{align*}
$$

where $M$ is a large positive number. Take Eq. (11) for example, when $\sigma_{a b}^{i}=1$, Eq. (11) is equivalent to $t_{i}^{\prime}+$ $\sum_{r} \lambda_{r}^{i} T_{a b}^{r, 1} \leq \tau_{a b}^{i} \leq t_{i}^{\prime}+\sum_{r} \lambda_{r}^{i} T_{a b}^{r, 1}$, which is identical to $\tau_{a b}^{i}=$ $t_{i}^{\prime}+\sum_{r} \lambda_{r}^{i} T_{a b}^{r, 1}$. When $\sigma_{a b}^{i}=0$, Eq. (11) is automatically satisfied as $M$ is a large positive number.

For all vehicles, the time entering a grid $g_{a b}$ should be no later than the time exiting this grid, and should be no less than zero (zero is set as the starting time of the modeling period), i.e.,
$0 \leq \tau_{a b}^{i} \leq \varphi_{a b}^{i}$
(Queue sequence) If there are two or more vehicles in the same entrance lane, the vehicle in front (vehicle $i$ ) should enter the intersection earlier than the vehicle behind (vehicle $j)$, i.e.,

If $J_{i}=J_{j}$, then $t_{i}^{\prime} \leq t_{j}^{\prime}+d_{c} / v, \quad \forall i, j \in C_{A}, A \in O$
where vehicle $i$ is the vehicle in front and vehicle $j$ is the vehicle behind, i.e., $t_{i}<t_{j}$. $J_{i}$ and $J_{j}$ are indices of the chosen entrance lanes of vehicles $i$ and $j$, respectively, $t_{i}^{\prime}$ and $t_{j}^{\prime}$ is the entering times, $d_{c}$ is the vehicle length, and $v$ is the speed. Eq. (14) guarantees a minimal space between two successive vehicles in the same lane, i.e., $d_{c}$. If a larger space is needed due to safety consideration, one can increase the value of $d_{c}$. Moreover, for a vehicle $i, J_{i}$ can be written as:
$J_{i}=\sum_{r} \lambda_{r}^{i} l_{r}, \quad r \in R$
where $l_{r}$ is the entrance lane associated with route $r$.
The nonlinear constraint in Eq. (14) can be rewritten into linear form as follows:
$t_{i}^{\prime}-t_{j}^{\prime}-d_{c} / v-M_{1}\left(J_{i}-J_{j}\right) \leq M_{2} \eta_{j}^{i}$
$t_{i}^{\prime}-t_{j}^{\prime}-d_{c} / v-M_{1}\left(J_{j}-J_{i}\right) \leq M_{2}\left(1-\eta_{j}^{i}\right)$
where $M_{1}$ and $M_{2}$ are two large positive numbers, $M_{2} \gg M_{1}$ ( $M_{2}=10^{6}$ and $M_{1}=10^{3}$ in this paper), and $\eta_{j}^{i}$ is a binary variable (when $J_{j}<J_{i}, \eta_{j}^{i}=0$; and when $J_{j} \geq J_{i}, \eta_{j}^{i}=1$ ). When $J_{j}=J_{i}$, Eq. (17) is equivalent to $t_{i}^{\prime}-t_{j}^{\prime}-d_{c} / v \leq 0$, which is identical to Eq. (14), and Eq. (16) is equivalent to $t_{i}^{\prime}-t_{j}^{\prime}-d_{c} / v \leq M_{2}$, which is automatically satisfied. When $J_{j} \neq J_{i}$, no matter $J_{j}<J_{i}$ or $J_{j}>J_{i}$, Eqs. (16) and Eq. (17) will be automatically satisfied.

## B. Lane-changing Penalties and Restraints

In Section III-A, the impacts of lane-changing are not modeled and we consider that AVs can shift to the assigned entrance lane in advance. However, lane-changing may cause inconvenience/cost/delay and may be infeasible sometimes. In this section, we extend the model in Section III-A to incorporate impacts of lane-changing or constraints regarding lane-changing.

## 1) Lane-changing penalties

The index of the chosen entrance lane of vehicle $i$, i.e., $J_{i}$, can be computed by Eq. (15). The index of the chosen exit lane of vehicle $i$, i.e., $J_{i}^{\prime}$, can be computed similarly as follows.
$J_{i}^{\prime}=\sum_{r} \lambda_{r}^{i} l_{r}^{\prime}, \quad r \in R$
where $l_{r}^{\prime}$ is the index of the exit lane of route $r$. If the chosen entrance lane or chosen exit lane of a vehicle is different from its original entrance lane or exit lane, i.e., $J_{i} \neq B_{i}$ or $J_{i}^{\prime} \neq B_{i}^{\prime}$, penalties will be applied. In particular, different from Eq. (5), the objective function with lane-changing penalties is

$$
\begin{equation*}
\min \sum_{i}\left[(1-\omega)\left(t_{i}^{\prime}-t_{i}\right)+\omega\left(\left|J_{i}-B_{i}\right|+\left|J_{i}^{\prime}-B_{i}^{\prime}\right|\right)\right] \tag{19}
\end{equation*}
$$

where $t_{i}^{\prime}-t_{i}$ is the delay of vehicle $i, J_{i}$ is the index of the chosen approach lane and $B_{i}$ is the index of the original approach lane, $J_{i}^{\prime}$ is the index of the chosen exit lane and $B_{i}^{\prime}$ is the index of the original exit lane. Note that $J_{i}, B_{i}, J_{i}^{\prime}, B_{i}^{\prime}$ all take integer values, and $\left|J_{i}-B_{i}\right|$ is number of lane changes (in terms of approach lane), and $\left|J_{i}^{\prime}-B_{i}^{\prime}\right|$ is number of lanes changes (in terms of exit lane). For example, in Fig. 5, we have $B_{i}=1$ and $J_{i}=3$, and $J_{i}-B_{i}=2$ denotes the number of lane changes of vehicle $i$ for shifting from original lane $B_{i}=1$ to the chosen lane $J_{i}=3$.


Fig.5. One example for the original and chosen approach lanes of vehicle $i$
Therefore, $\left|J_{i}-B_{i}\right|+\left|J_{i}^{\prime}-B_{i}^{\prime}\right|$ is the total number of lanechanging of vehicle $i, \omega$ is a weighting factor and $\omega \in[0,1]$. When $\omega=0$, the objective function in Eq. (19) is equivalent to that in Eq. (5), and the negative effects of lane-changing are not considered. When $\omega=1$, only negative effects of lanechanging are in the objective function and no vehicles should change his or her lane in order to minimize the objective.

Eq. (19) contains absolute value, which cannot be solved directly by MILP-based algorithm. For each vehicle, we further define four intermediate positive variables, i.e., $z_{1, i}$, $z_{2, i}, z_{1, i}^{\prime}$ and $z_{2, i}^{\prime}$, where we have
$J_{i}-B_{i}=z_{1, i}-z_{2, i}, \quad i \in C$
$J_{i}^{\prime}-B_{i}^{\prime}=z_{1, i}^{\prime}-z_{2, i}^{\prime}, \quad i \in C$
$z_{1, i}, z_{2, i}, z_{1, i}^{\prime}, z_{2, i}^{\prime} \geq 0 \quad i \in C$

From Eqs. (20)-(22), one can further derive that
$\min \left|J_{i}-B_{i}\right|=\min \left(z_{1, i}+z_{2, i}\right), \quad i \in C$
$\min \left|J_{i}^{\prime}-B_{i}^{\prime}\right|=\min \left(z_{1, i}^{\prime}+z_{2, i}^{\prime}\right), \quad i \in C$
Eq. (19) then can be rewritten as
$\min \sum_{i}\left[(1-\omega)\left(t_{i}^{\prime}-t_{i}\right)+\omega\left(z_{1, i}+z_{2, i}+z_{1, i}^{\prime}+z_{2, i}^{\prime}\right)\right]$
Eq. (25) can be directly solved by MILP-based algorithms.

## 2) Lane-changing restraints

Some potential lane-changings may not be feasible, especially when some vehicles may have to stick with their original entrance lane and/or exit lane. These considerations can be incorporated by adding constraints on route choices.

We define two sets of vehicles, i.e., $C_{b}$ and $C_{b}^{\prime}$, which are both subsets of $C$ (the full set of upcoming vehicles). Set $C_{b}$ includes those vehicles with fixed entrance lanes. For these vehicles, we add a constraint $J_{i}=B_{i}, i \in C_{b}$, i.e., the chosen entrance lane is identical to the original entrance lane. Similarly, for any vehicle $i \in C_{b}^{\prime}$, its exit lane is fixed, where we have a constraint $J_{i}^{\prime}=B_{i}^{\prime}$, i.e., the chosen exit lane is identical to the original exit lane. The additional computation complexity arising from these linear constraints is marginal.

## C. Solution Schemes

For the optimization models in Section III-A and Section III-B, we consider two solution schemes (or methods) for vehicles' entering times and their routes in the intersection. The first method is termed as the "sliding-time-window-based global optimum (GO-STW)", where a time-window is added to restrict the problem size. The second method is termed as the "first-come-first-served method with optimal route choices (FCFS-R)", where a vehicle arriving earlier at the intersection will enter the intersection no later than a later vehicle. Note that the exact entering time will still be optimized.

## 1) Sliding-time-window-based global optimum (GO-STW)

The modeling time horizon is discretized into multiple time windows/intervals. The GO-STW method is to minimize the objective in Eq. (5) for each time window, where entering times of all upcoming vehicles at the intersection and the route choices are decision variables. The optimization problem is subject to constraints in Section III-A and the following constraints in Eq. (26) to ensure that no conflicts among vehicles will occur.

For any two vehicles that use the same grid, four different cases regarding their occupation of the same grid may arise (see Fig.6). There are conflicts in Situation 1 and Situation 3. To avoid Situation 1 and Situation 3, we set:

If $\sigma_{a b}^{i}=\sigma_{a b}^{j}=1$, then $\varphi_{a b}^{i}-\tau_{a b}^{j} \leq 0$ or $\varphi_{a b}^{j}-\tau_{a b}^{i} \leq$ $0, \forall i, j \in C$
where $\sigma_{a b}^{i}$ and $\sigma_{a b}^{j}$ are binary variables; $\sigma_{a b}^{i}=1$ if vehicle $i$ crosses grid $g_{a b}$ and zero otherwise; $\varphi_{a b}^{i}$ is the exiting time of
vehicle $i$ from grid $g_{a b} ; \tau_{a b}^{j}$ is the entering time of vehicle $j$ into grid $g_{a b} ; \varphi_{a b}^{j}$ and $\tau_{a b}^{i}$ are similarly defined; $g_{a b} \in G$. For a given vehicle, the entering and exiting times are calculated based on Eqs. (9) and (10).

Note that Eq. (26) can be rewritten into the linear form, i.e.,

$$
\begin{align*}
& \varphi_{a b}^{i}-\tau_{a b}^{j}-M\left(1-\sigma_{a b}^{i}\right)-M\left(1-\sigma_{a b}^{j}\right) \leq M \rho_{a b}^{i, j}  \tag{27}\\
& \varphi_{a b}^{j}-\tau_{a b}^{i}-M\left(1-\sigma_{a b}^{i}\right)-M\left(1-\sigma_{a b}^{j}\right) \leq M\left(1-\rho_{a b}^{i, j}\right) \tag{28}
\end{align*}
$$

where $\rho_{a b}^{i, j}$ is a binary variable that indicates the sequence of vehicle $i$ and $j$ entering the grid $g_{a b}$ when they both would use the grid, i.e., $\sigma_{a b}^{i}=\sigma_{a b}^{j}=1$. When $\sigma_{a b}^{i}=\sigma_{a b}^{j}=1$, Eq. (27) is equivalent to $\varphi_{a b}^{i}-\tau_{a b}^{j} \leq M \rho_{a b}^{i, j}$, Eq. (28) is equivalent to $\varphi_{a b}^{i}-\tau_{a b}^{j} \leq M\left(1-\rho_{a b}^{i, j}\right)$. Since $\rho_{a b}^{i, j}$ is a binary variable, if $\rho_{a b}^{i, j}=1$, Eq. (27) is automatically satisfied, and Eq. (28) is equivalent to $\varphi_{a b}^{i}-\tau_{a b}^{j} \leq 0$; if $\rho_{a b}^{i, j}=0$, Eq. (27) is equivalent to $\varphi_{a b}^{i}-\tau_{a b}^{j} \leq 0$, and Eq. (28) is automatically satisfied. These are identical to Eq. (26). When $\sigma_{a b}^{i} \neq 1$ or $\sigma_{a b}^{j} \neq 1$, Eqs. (27) and (28) will be automatically satisfied.


Fig.6. Four different situations for two vehicles to cross the same grid
As mentioned earlier, at each time window, we only solve the optimal entering time and lane-based route choices for AVs within the time window rather than the whole time horizon. The decisions from previous time windows are inputs for the optimization problem of next windows.

The length of the sliding time window is denoted by $\Delta t$. When $\Delta t \rightarrow \infty$, the GO-STW method is identical to optimizing the problem in the whole time horizon. When $\Delta t \rightarrow 0$, the GO-STW method can be considered as the first-come-first-served (FCFS) principle based method, which is identical to an extreme case of the FCFS-R method (when time window for FCFS-R is also set as zero) to be discussed.

## 2) The first-come-first-served method with route choices optimization (FCFS-R)

In the FCFS-R, a vehicle with an earlier planned arrival time at the intersection will enter the intersection earlier (or at least no later than a later vehicle). Thus, the entering sequence of vehicles is constrained by vehicles' planned arrival times at the intersection.

The constraints in Eq. (26) for the GO-STW method are no longer needed for the FCFS-R method. Instead, we have the following constraints in Eq. (29). Let $k$ indicate the $k^{\text {th }}$ vehicle based on the sorted sequence, let $n$ denote the number of all upcoming vehicles under consideration, then FCFS-R method requires that
$\varphi_{a b}^{k}-\tau_{a b}^{k+1} \leq 0, k \in\{1 \ldots n-1\}$
i.e., the exiting time from grid $g_{a b}$ for the $k^{t h}$ vehicle $\varphi_{a b}^{k}$ is no later than the entering time of the $(k+1)^{t h}$ vehicle $\tau_{a b}^{k+1}$, where $g_{a b} \in G$. The constraints in Eq. (29) for the FCFS-R method ensure no conflict among vehicles.

Again, the length of the time window is defined as $\Delta t$. When $\Delta t \rightarrow \infty$, it means that all vehicles over the time horizon is considered simultaneously, while the sequence of vehicles to enter the intersection is determined based on vehicle arriving time. When $\Delta t \rightarrow 0$, it is equivalent to GOSTW with a window length equal to zero, i.e., each vehicle minimizes its own delay when deciding entering time and route, while entering times and routes of vehicles before the concerned vehicle are taken as inputs/constraints.

## D. Heuristic procedure for MILP

The developed models (with two different solution schemes) are all mixed integer linear programming (MILP) problems, which can be solved through the CPLEX solver. However, when the number of vehicles increases (due to a higher demand or a longer time window), the computation time may increase sharply. This is because, the solution space and combinatorial complexity for computing each vehicle's routes and entering time increase exponentially with the number of vehicles. In order to reduce computation time, this paper adopts the heuristic proposed in [31]. The basic idea of the adopted heuristic is to find a feasible solution firstly and then look for a better solution near this feasible solution in the next round of optimization. Fig. 7 depicts the main steps of the adopted heuristic, which are summarized below.

Step 1: Find a feasible solution as an upper bound. In this paper, we adopt the "first-come-first-served method without route choices" (FCFS-WR) to produce an initial solution. FCFS-WR is similar to the FCFS-R introduced in Section IIIC, but without optimizing the route choices. In the numerical tests in Section IV-B, one can see that even with high-level demand, the average computation time of FCFS-WR method is only 0.03 seconds (a quick solution can be produced).

Step 2: Add a constraint where the objective function value from the initial solution or from last iteration minus a certain
cut-off value is set as the upper-bound for the objective function in the next iteration. We take the objective function in Eq. (5) as an example. For a given objective function value obtained in the $n^{\text {th }}$ iteration $t_{i}^{\prime(n)}$ and $t_{i}^{(n)}$, we add the following constraint
$\sum_{i}\left(t_{i}^{\prime}-t_{i}\right) \leq \sum_{i}\left(t_{i}^{\prime(n)}-t_{i}^{(n)}\right)-\theta^{(n)}, \quad i \in C$
In Eq. (30), the objective function in the original MILP should be less than the objective value at the $n^{\text {th }}$ iteration by at least $\theta^{(n)} \geq 0$, where $\theta^{(n)}$ is further formulated in Eq. (31).

How to generate a value of $\theta^{(n)}$ is critical to the success of the proposed heuristic procedure. If $\theta^{(n)}$ is too large, one may not be able to obtain a feasible solution in the next iteration within the time length of a time window. However, if $\theta^{(n)}$ is too small, the improvement of solution for the next iteration is small. This paper adopts the following:
$\theta^{(n)}=\frac{1}{n+1}\left(\mu \sum_{i}\left(t_{i}^{\prime(n)}-t_{i}^{(n)}\right)\right), \quad i \in C$
where $\mu>0$ is a predetermined parameter and should be chosen case by case, and usually $\mu=1$ as suggested in [31]. As $\sum_{i}\left(t_{i}^{\prime(n)}-t_{i}^{(n)}\right)$ and $\frac{1}{n+1}$ both decrease with $n, \theta^{(n)}$ will decrease gradually with $n$.

Step 3: Define a new objective function by utilizing the Hamming distance. In Step 2, the original objective (total delay) is incorporated in the constraint in Eq. (30). A new objective function is developed according to the Hamming distance, which is defined between the solution to be optimized and the current solution as follows:
$\Delta\left(\lambda_{r}^{i}, \lambda_{r}^{i(n)}\right)=\sum_{i \in C, r \in R: \lambda_{r}^{i(n)}=0} \lambda_{r}^{i}+\sum_{i \in C, r \in R: \lambda_{r}^{i(n)}=1}\left(1-\lambda_{r}^{i}\right)$
We then solve the following
$\min \Delta\left(\lambda_{r}^{i}, \lambda_{r}^{i(n)}\right)$
where $\lambda_{r}^{i}$ is the binary decision variable, $\lambda_{r}^{i}=1$ when vehicle $i$ chooses route $r$ and zero otherwise. Eq. (33) means that we try to find a new solution close to the current solution (this often saves computation time) while a delay reduction is ensured through the constraint in Eq. (30).

Step 4: Solve the new optimization problem. With the new objective function in Eq. (33) and the new constraints in Eq. (30)-(32), the updated MILP can be solved by CPLEX solver. In order to ensure that we can obtain a solution in a given time frame, we set up a total computation time limit. In the numerical tests, we use $\Delta t$ as the computation time limit, where $\Delta t$ is the length of the time window in the GO-STW and FCFS-R methods. Note that an even smaller time limit can be readily adopted.

Step 5: Collect the solution including the total delay and the route choices for each vehicle and define them as the $(n+1)^{t h}$ solution, i.e., $\sum_{i}\left(t_{i}^{(n+1)}-t_{i}^{(n+1)}\right)$ and $\lambda_{r}^{i(n+1)}$. These will be the inputs for the $(n+2)^{t h}$ iteration. Then go
to Step 2 for the next iteration of optimization until the computation time limit in Step 4 is reached or the solution cannot be further improved (or improvement is too small).

## IV. Numerical Examples

This section presents some numerical tests to illustrate the efficiency of the lane-based methods and shows the efficiency of the proposed heuristic to provide a solution in real-time. Section IV-A presents the basic setting. Section IV-B compares the efficiency and computation time for different methods. Section IV-C presents the computation efficiency of the proposed heuristic.


Fig.7. Basic idea and the flowchart of the heuristic

## A. Basic Setting

The numerical setting is based on the parameters of the intersection between Shanmuchong East Road and Wanfu North Road in Changsha City, China. This intersection has four entrance lanes and four exit lanes in each direction. The lane width is 3 meters. The intersection is discretized into 64 square grids with a side length of 3 meters. Each vehicle is 4.5 -meter long and 2.5 -meter wide. There are 192 trajectory bands in total and more than 5000 critical points associated with the trajectory bands. The number of potential conflicts between vehicles is much larger than these numbers. The numerical example considers a time duration of one-minute. This time duration is discretized into 6 time windows with a length of 10 seconds. Table IV summarizes the basic setting. By using these parameters, for through traffic, the minimum time headway is 0.75 seconds, that is, when a vehicle enters a
grid, it clears after 0.75 seconds and other vehicles can enter. For turning traffic, the time headway is similar given the same speed. As reported in [32], autonomous vehicles can maintain a time gap as small as 0.6 seconds ( 1.5 seconds for conventional non-automated vehicles).

We tested three different mean demand levels, i.e., 720 $\mathrm{pcu} / \mathrm{h}$ per entrance lane (high-level), $540 \mathrm{pcu} / \mathrm{h}$ per entrance lane (medium level), $360 \mathrm{pcu} / \mathrm{h}$ per entrance lane (low-level). With high-level demand, the total demand rate is $11520 \mathrm{pcu} / \mathrm{h}$ (the mean demand rate). The arrival is assumed to follow the Poisson distribution. We assume equal proportions for the through-traffic, left-turn traffic and right-turn traffic, i.e., 1/3 each. This evenly distributed demand case provides a performance benchmark. For example, if there is an overwhelming left-turn traffic, since left-turn movements generally have more conflict points with other traffic, it is expected that less delays can be saved when compared with the benchmark. If there is more right-turn traffic, as right-turn movements in general have less conflict points with other traffic, more delay savings are expected.

Moreover, each vehicle has its original planned entrance lane and exit lane. The original planned entrance lanes and exit lanes for vehicles in the numerical tests are based on the following principles. In terms of the entrance lane, left turn vehicles are distributed to the two lanes on the leftmost with an equal probability (i.e., 50:50), right turn vehicles are assigned to the two lanes on the rightmost with an equal probability (i.e., 50:50), through vehicles are distributed to the four entrance lanes with an equal probability. In terms of the exit lane, the planned exit lane is compatible with the planned entrance lane, e.g., if a left-turn vehicle's entrance lane is $l_{2}^{S}$, then its exit lane should be $L_{2}^{W}$ (please refer to Fig.3). However, under the proposed "all-direction" lane framework, neither the GO-STW nor FCFS-R method requires vehicles to follow the original entrance or exit lanes.

The numerical tests are conducted on a platform with a Win-7 64-bit operating system and $\operatorname{Intel}(\mathrm{R})$ Core(TM) i53470 CPU 3.20 GHz , 8G RAM.

TABLE IV
Basic Numerical Setting

| Parameters | Specification |
| :---: | :---: |
| Vehicle size | Length: $4.5(\mathrm{~m})$, width: $2.5(\mathrm{~m})$ |
| Lane width | $d_{r}=3(\mathrm{~m})$ |
| Vehicle speed | $v=10(\mathrm{~m} / \mathrm{s})$ |
| Side lengths of a grid | $d_{g}=3(\mathrm{~m})$ |
| Number of grids | 64 |
| Number of approaches | 4 approaches |
| Lanes on each approach | 4 entrance lanes and 4 exit lanes |
| Number of routes | 192 routes |
| Number of available routes for a vehicle | 16 routes |
| High/Medium/Low-level demand | 720/540/360 (pcu/h per entrance lane) |
| The time window length | 10 (s) |

## B. Comparison of Different Methods

We examine the traffic efficiency and the computation efficiency of the proposed decision methods, i.e., the sliding-time-window-based global optimum (GO-STW) method and the first-come-first-served method with optimal route choices (FCFS-R) method. For comparison purpose, two special cases are added, i.e., the global optimum method without route choices (GO-WR), which is similar to GO-STW but with given routes (i.e., routes are not optimized but follow those specified in Section IV-A), and the first-come-first-served method without route choices (FCFS-WR), which is similar to FCFS-R but again with given route choices that are specified in Section IV-A.

This section presents the performance benchmark solved by the CPLEX solver, which will be further compared with solutions from the proposed heuristic in Section IV-C. We set 300 seconds (five minutes) as the computation time limit for all the methods to be compared. Note that the five-minute computation time limit is chosen based on our extensive tests, where solution improvement after five minutes is often small.

## 1) Comparison of computation time and average delay

Table V summarizes the computation times and traffic delays under four different methods (GO-STW, FCFS-R, GOWR, FCFS-WR) with different demand levels. We discuss the observations from Table V in the following.

Firstly, under three demand levels, GO-STW always yields the smallest average delay, while the computation time is relatively large. It provides a traffic efficiency upper bound for different methods. In contrast, FCFS-WR, where first-come-first-served principle is adopted and routes of AVs are given, can provide a traffic efficiency lower bound. It is evident that the proposed GO-STW and FCFS-R can yield very competitive efficiency performance (average delay per vehicle is much lower than that under FCFS-WR).

TABLE V
Comparison of the Four Different Methods

| COMPARISON OF THE FOUR DIFFERENT METHODS |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: |
| Travel <br> demands | Methods | Average <br> computation <br> time(s) | Total <br> delays(s) | Average <br> vehicle <br> delay(s) |
| High-level | GO-STW | 300.00 | 64.82 | 0.35 |
|  | FCFS-R | 300.00 | 245.05 | 1.34 |
|  | GO-WR | 18.67 | 284.00 | 1.55 |
|  | GO-STW | FCFS-R | 300.00 | 5727.84 |
| level | GO-WR | 300.00 | 39.90 | 0.31 .30 |
|  | FCFS-WR | 0.30 | 79.24 | 0.47 |
|  | GO-STW | 28.22 | 2696.15 | 0.60 |
| Low-level | FCFS-R | 14.73 | 4.04 | 0.43 |
|  | GO-WR | 0.11 | 15.42 | 0.04 |
|  | FCFS-WR | 0.01 | 37.02 | 0.40 |

Secondly, we compare GO-STW and GO-WR. The difference in average delays can be regarded as the benefits from optimizing lane-based route choices (entering time is optimized under both methods while route choices are
optimized for GO-STW). We focus on the high-level demand case for comparison. As can be seen in Table V, the average delays are 0.35 and 1.55 under GO-STW and GO-WR, respectively, indicating an evident saving by optimizing the route choices.

Thirdly, by comparing FCFS-R and FCFS-WR, we can identify the benefits from lane-based route choices optimization under the FCFS policy. For the high-level demand case, the average delays are 1.34 and 31.3 under FCFS-R and FCFS-WR, respectively. This indicates a significant saving from route choice optimization.

Fourthly, FCFS-R dominates GO-WR in terms of traffic efficiency under all demand levels. This indicates that optimizing route choices but with FCFS-based entering time is more efficient than optimizing the entering time freely but with unoptimized route choices. This further implies that optimization of lane-based route choices can be more critical than the optimization of entering time.

We also highlight a few more numerical observations in relation to the advantages of "all-direction" lanes here. In the high demand case, within 6 time windows, 6 left-turn vehicles use the right-most approach lane and the right-most exit lane under the solution of GO-STW. At the same time, 3 right-turn vehicles use the left-most approach lane and the left-most exit lane; 9 left-turn vehicles use the right-most lane either for entering or exiting; 14 right-turn vehicles use the left-most lane either for entering or exiting. This means that the solution of GO-STW takes advantage of the flexibility of the "alldirection" lanes. We further verify that under GO-STW with all-direction lanes, the occupancy rate of all grids is $41.53 \%$ against $38.35 \%$ under GO-WR (which is like there is no "alldirection" lane). These results further verify that the proposed all-direction lane framework helps to improve the space utilization of an intersection and decrease vehicle delays.

TABLE VI
The Proportions of Vehicles with Route Adjustment under GO-STW AND FCFS-R METHODS

| Travel demand | Route adjustment or lane change | GO-STW | FCFS-R |
| :---: | :---: | :---: | :---: |
| High-level | Entrance lane only | $73.22 \%$ | $69.95 \%$ |
|  | Exit lane only | $72.13 \%$ | $66.67 \%$ |
|  | Entrance and/or Exit lane(s) | $81.97 \%$ | $74.86 \%$ |
| Medium level | Entrance lane only | $68.18 \%$ | $60.61 \%$ |
|  | Exit lane only | $65.91 \%$ | $62.88 \%$ |
|  | Entrance and/or Exit lane(s) | $78.03 \%$ | $74.24 \%$ |
| Low-level | Entrance lane only | $71.74 \%$ | $59.78 \%$ |
|  | Exit lane only | $71.74 \%$ | $59.78 \%$ |
|  | Entrance and/or Exit lane(s) | $80.43 \%$ | $68.48 \%$ |

The significance of the proposed lane-based route choice optimization is further verified by the proportion of vehicles that have been reallocated to a new route against the original planned route (i.e., either entrance or exit lane is adjusted), as shown in Table VI. As can be seen, under the two more efficient methods, i.e., GO-STW and FCFS-R, significant proportions of vehicles have been reallocated to a new route.
2) Variation of individual vehicle delays

The distributions of individual vehicles delay under the four methods (GO-STW, FCFS-R, GO-WR, FCFS-WR) are shown in Fig.8, where the numerical setting is the same as that in Section IV-A1.

In Fig.8, there are two main observations. Firstly, as demand increases, there is an increasing variation level of individual delays under each of the four methods. This is indicated by lengths of blue boxes, which represent the $20 \%$ $80 \%$ range of the delays. Secondly, under different demand levels, the variation of individual vehicle delays follows GOSTW < FCFS-R < GO-WR < FCFS-WR, i.e., the GO-STW method produces the smallest variation (as well as the smallest average delay, as indicated in both Table V and Fig.8) and FCFS-WR method produces the largest variation (as well as the largest average delay, as indicated in both Table V and Fig.8).


Fig.8. Box-plot of individual vehicle delays ( $x$-axis: $1 \rightarrow$ GO-STW, $2 \rightarrow$ FCFSR, $3 \rightarrow$ GO-WR, $4 \rightarrow$ FCFS-WR)
(Note: the values for " $4 \rightarrow$ FCFS-WR" is one third of true values (scaled down) for a more compact display for the figures)

## 3) The impacts of lane-changing

(Lane-changing penalties) We now further examine the case where adjustments of planned routes are penalized (to reflect the inconvenience/cost/delay), as discussed in Section III-B, where the objective in Eq. (25) combines delays and lane-changing penalties. Note that for illustration purpose and to save space, we only present the results for the GO-STW method under high-level demand (as specified in Section IVA). When varying the weight for lane-changing penalties $\omega$ from zero to one, the average delay of all vehicles and the total number of lane-changing, are shown in Fig.9.

There are two main observations from Fig.9. Firstly, $\omega=0$ and $\omega=1$ are two special cases. When $\omega=0$, no penalty is adopted for lane-changing. The average vehicle delay is found to be 0.35 seconds, which is exactly equal to the average delay in Table V (the GO-STW method). The total number of lane-changing is the largest ( 419 in this example), when compared with other methods. When $\omega=1$, the objective is
to minimize the number of lane-changing. In this case, it is optimal for all vehicles to stick with the planed entrance and exit lanes. However, the average delay is high when compared with that under $\omega=0$. This indicates the large efficiency gains by optimizing the route choices. Secondly, the average vehicle delay increases while the total number of lanechanging decreases with the increase of $\omega$. This is because with a larger $\omega$, lane-changing is more heavily penalized and less preferable, and the proposed lane-based AIM will be less efficient in terms of decreasing the average delay defined.


Fig.9. Variation of average vehicle delay and total number of lane-changing against $\omega$
(Lane-changing constraint) As discussed in Section III-B, not all potential lane-changing may be feasible. We further investigate the performance of the proposed methods when adding constraints on some vehicles' entrance lane and/or exit lane, i.e., some vehicles' entrance lane and/or exit lane cannot be adjusted. For illustration purpose, here we test and present the cases where a proportion of vehicles' entrance lane and exit lane cannot be adjusted. Also, to save space, we only show the cases with high demand level defined in Section IVA and present the results based on the GO-STW model.

In particular, we vary the proportion of vehicles with fixed entrance lane and exit lane, and examine how the average vehicle delay changes, which is show in Fig.10. Note that we assume that traffic of each movement has the same proportion of vehicles with fixed routes.

There are two major observations in Fig.10. Firstly, as the proportion of vehicles with fixed route increases from zero to 0.7 , the average vehicle delay increases; and when that proportion further increases beyond 0.7 , the average vehicle delay does not vary too much. This indicates that, in general, fewer vehicles have the fixed route, more efficiency gain can be generated by the lane-based route optimization. Also, when there are too many vehicles with fixed route (more than $70 \%$ in this example), the efficiency gain from optimizing the lanebased route is limited. This suggests that the proposed lanebased AIM model would generate benefits where there is sufficient flexibility in vehicles' routes. Secondly, there is a significant delay reduction when comparing the average
vehicle delays in the two extreme cases, i.e., "no vehicles with fixed routes" and "all vehicles with fixed routes". This reflects the potential benefit of the proposed lane-based AIM model if all vehicles are flexible to change their entrance and exit lanes.


Fig.10. The average vehicle delay against the proportion of vehicles with fixed routes

## 4) Comparison with traditional lane assignment

We define a benchmark case for comparison where left-turn vehicles always use the left-most lane and right-turn vehicles always use the right-most lane, which is termed as the sliding-time-window-based global optimum with fixed lane-marking (GO-STW-FLM). The vehicle delays under GO-STW-FLM and the proposed GO-STW is summarized in Table VII. The results in Table VII clearly show that the proposed GO-STW outperforms GO-STW-FLM under different demand levels.

TABLE VII
THE COMPARISON OF VEHICLE DELAYS BETWEEN GO-STW and GO-STW-

| $c$ <br> wind <br> ow |  |  |  |  |  | High-level demand |  | Medium level <br> demand |  | Low-level demand |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  | GO- | GO-STW- | GO- | GO-STW- | GO- | GO-STW- |  |  |  |  |  |
|  | STW | FLM | STW | FLM | STW | FLM |  |  |  |  |  |
| 1 | 7.10 | 20.46 | 0.92 | 2.42 | 0.00 | 2.40 |  |  |  |  |  |
| 2 | 19.71 | 22.69 | 5.58 | 12.24 | 1.29 | 1.61 |  |  |  |  |  |
| 3 | 9.57 | 17.54 | 3.86 | 6.77 | 0.08 | 5.84 |  |  |  |  |  |
| 4 | 7.18 | 14.99 | 16.47 | 12.01 | 0.09 | 0.87 |  |  |  |  |  |
| 5 | 12.12 | 30.35 | 0.49 | 1.30 | 1.39 | 5.68 |  |  |  |  |  |
| 6 | 9.13 | 32.88 | 12.59 | 11.15 | 1.19 | 3.90 |  |  |  |  |  |

Table VIII summarizes vehicle delays for different movements under high-level demand. As can be seen, the delay reduction (i.e., GO-STW against GO-STW-FLM) for left-turn traffic is most significant. This is in line with the fact that left-turn vehicles often experience more delays before introducing the all-direction lanes (due to more conflicts with other traffic).

TABLE VIII
The comparison of Vehicle delays for different movements BETWEEN GO-STW AND GO-STW-FLM

| Right turn |  |  |  |  |  | Left turn |  | Through |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  | GO- <br> STW | GO-STW- | FLM | STW | GO-STW- | GL- |  |  |  |
| STM | STW | GO-STW- |  |  |  |  |  |  |  |
| West <br> bound <br> East | 0.00 | 3.11 | 8.43 | 26.24 | 5.84 | 8.36 |  |  |  |
| bound | 0.78 | 1.13 | 13.04 | 30.75 | 5.96 | 11.75 |  |  |  |
| North <br> bound | 1.17 | 0.37 | 7.79 | 22.21 | 5.21 | 5.34 |  |  |  |
| South <br> bound <br> Total | 0.59 | 1.38 | 9.78 | 20.15 | 6.22 | 8.12 |  |  |  |

## C. The Heuristic: Vehicle Delay and Computation time

In Section IV-B, GO-STW and FCFS-R are directly solved through CPLEX solver with a 5-minute computation time limit. It means that demand and vehicle arrival information should be available to the intersection operator at least 5 minutes in advance (in practice, additional data processing and communication time is needed). This may be feasible since all AV trips may have to be booked in advance and the trajectories of AV trips can be more predictable. We now further test the proposed heuristic and examine the trade-off between traffic efficiency and computation time. In particular, we ensure that the solution for a time window can be provided within the length of a time window.

We focus on the GO-STW method for illustration purpose and test the three demand levels defined in Section IV-A. The delays and computation times are shown in Tables IX, X and XI for the three demand levels, respectively. Note that in Table IX, X and XI, "delay" is the total vehicle delay for vehicles within a time window, "time" is the computation time for the current iteration of calculation, "W1" to "W6" denote the first to the sixth time window, respectively. Each time window is 10 seconds (specified in Table V) and the solution for each time window is provided within 10 seconds. Note that the initial solution (Iteration 1) is from the FCFSWR method, which is negligible (please refer to Step 1 in Section III-D). As can be seen, the vehicle delay decreases over iterations in the proposed heuristic procedure. However, the decreasing speed of delay slows down over iterations. This is in line with that $\theta$ in Eq. (30) decreases over iterations.

Table XII further summarizes the delays and computation times under GO-STW method and FCFS-WR method directly solved by CPLEX solver (the results are time-window specific, where Table V only provides the aggregate results over all time windows) and those under GO-STW method based on the heuristic solution procedure (these are consistent with those in Table IX, X and XI).

TABLE IX
The Total Delay and Computation Time under High-Level Demand ("-"": no solution within the time limit)

| Time window | FCFS-WR (Iteration 1) |  | Iteration 2 |  | Iteration 3 |  | Iteration 4 |  | Iteration 5 |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  | delay(s) | time(s) | delay(s) | time(s) | delay(s) | time(s) | delay(s) | time(s) | delay(s) | time(s) |


| W1 | 299.35 | 0.02 | 39.14 | 1.78 | - | - | - | - | - | - |
| :--- | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| W2 | 563.36 | 0.05 | 113.31 | 1.44 | 66.30 | 1.65 | 45.29 | 1.70 | - | - |
| W3 | 865.61 | 0.03 | 179.22 | 1.06 | 72.86 | 1.86 | - | - | 1.03 | 49.60 |
| W4 | 966.85 | 0.02 | 161.80 | 0.69 | 90.86 | 1.05 | 62.75 | 1.97 |  |  |
| W5 | 1375.41 | 0.02 | 471.88 | 1.00 | 160.17 | 1.23 | 118.69 | 1.78 | 92.69 | 1.89 |
| W6 | 1657.26 | 0.02 | 360.33 | 1.19 | 240.00 | 2.32 | - | - | - |  |

TABLE X
The Total Delay and Computation Time under Medium-Level Demand ("-": no solution within the time limit)

| Time window | FCFS-WR (Iteration 1) |  | Iteration 2 |  | Iteration 3 |  | Iteration 4 |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  | delay(s) | time(s) | delay(s) | time(s) | delay(s) | time(s) | delay(s) | time(s) |
| W1 | 60.18 | 0.02 | 23.66 | 0.50 | 10.57 | 0.47 | 7.50 | 0.84 |
| W2 | 248.12 | 0.02 | 84.70 | 0.73 | 23.73 | 0.73 | 17.15 | 0.72 |
| W3 | 358.80 | 0.00 | 26.36 | 0.37 | 14.02 | 0.45 | 10.49 | 0.44 |
| W4 | 645.60 | 0.02 | 147.27 | 0.70 | 43.31 | 0.80 | 31.83 | 0.72 |
| W5 | 609.07 | 0.02 | 304.50 | 0.50 | 202.67 | 0.50 | 36.66 | 0.50 |
| W6 | 774.38 | 0.03 | 14.91 | 0.48 | - | - | - | - |
| Time window | Iteration 5 |  | Iteration 6 |  | Iteration 7 |  | Iteration 8 |  |
|  | delay(s) | time(s) | delay(s) | time(s) | delay(s) | time(s) | delay(s) | time(s) |
| W1 | 6.00 | 8.05 | , | ( | , | ( | ( | (s) |
| W2 | - | - | - | - | - | - | - | - |
| W3 | 8.00 | 7.55 | - | - | - | - | - | - |
| W4 | - | - | - | - | - | - | - | - |
| W5 | 28.27 | 0.53 | 20.87 | 0.50 | 16.67 | 0.66 | 14.00 | 3.99 |
| W6 | - | - | - | - | - | - | - | - |

TABLE XI
The Total Delay and Computation Time under Low-Level Demand ("-"": NO SOLUTION within the time Limit)

| Time window | FCFS-WR (Iteration 1) |  | Iteration 2 |  | Iteration 3 |  | Iteration 4 |  | Iteration 5 |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  | delay(s) | time(s) | delay(s) | time(s) | delay(s) | time(s) | delay(s) | time(s) | delay(s) | time(s) |
| W1 | 9.81 | 0.00 | 3.70 | 0.86 | 2.47 | 0.78 | 1.85 | 0.61 | 1.48 | 1.12 |
| W2 | 11.29 | 0.02 | 4.85 | 0.27 | 3.23 | 0.58 | 2.35 | 2.70 | 1.88 | 0.50 |
| W3 | 33.73 | 0.03 | 13.86 | 0.33 | 7.51 | 0.33 | 5.63 | 3.17 | 4.09 | 0.33 |
| W4 | 104.41 | 0.00 | 14.18 | 0.31 | 7.40 | 0.30 | 5.46 | 0.31 | 4.32 | 1.01 |
| W5 | 156.11 | 0.02 | 41.00 | 0.38 | 16.12 | 0.38 | 11.95 | 0.36 | 9.60 | 0.56 |
| W6 | 203.00 | 0.00 | 10.54 | 0.33 | 7.00 | 0.67 | 4.91 | 0.44 | 3.90 | 1.06 |
| Time window | Iteration 6 |  | Iteration 7 |  | Iteration 8 |  | Iteration 9 |  | Iteration 10 |  |
|  | delay(s) | time(s) | delay(s) | time(s) | delay(s) | delay(s) | time(s) | delay(s) | time(s) | delay(s) |
| W1 | 1.23 | 1.47 | 1.05 | 2.04 | 0.92 | 1.64 | - | - | - | - |
| W2 | 1.57 | 0.58 | 1.34 | 0.69 | - | - | - | - | - | - |
| W3 | 3.32 | 0.70 | 2.23 | 0.67 | 0.79 | 1.05 | 0.68 | 0.87 | 0.51 | 0.42 |
| W4 | 1.72 | 0.31 | 1.45 | 0.89 | 1.04 | 1.08 | 0.92 | 0.61 | 0.65 | 0.31 |
| W5 | 8.00 | 2.59 | 6.86 | 3.14 | - | - | - | - | - | - |
| W6 | 2.93 | 1.08 | 2.41 | 0.62 | 1.87 | 0.37 | 1.60 | 0.80 | 1.44 | 0.66 |

TABLE XII
The Total Delays and Total Computation Times: GO-STW and FCFS-WR (without using heuristic); GO-STW(HP) Using the Proposed Heuristic

|  | High-level demand |  |  |  |  |  | Medium-level demand |  |  |  |  |  | Low-level demand |  |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  | GO-STW |  | $\begin{gathered} \text { GO- } \\ \text { STW(HP) } \end{gathered}$ |  | FCFS-WR |  | GO-STW |  | STW(HP) |  | FCFS-WR |  | GO-STW |  | $\begin{gathered} \text { GO- } \\ \text { STW(HP) } \end{gathered}$ |  | FCFS-WR |  |
|  | delay | Time | delay | time | delay | time | delay | time | delay | time | delay | time | delay | time | delay | time | delay | time |
| W1 | 7.10 | 300.00 | 39.14 | 1.79 | 299.35 | 0.02 | 0.92 | 300.00 | 6.00 | 9.88 | 60.18 | 0.02 | 0.00 | 6.67 | 0.92 | 8.52 | 9.81 | 0.00 |
| W2 | 19.71 | 300.00 | 45.29 | 4.83 | 563.36 | 0.05 | 5.58 | 300.00 | 17.15 | 2.20 | 248.12 | 0.02 | 1.29 | 5.98 | 1.34 | 5.32 | 11.29 | 0.02 |
| W3 | 9.57 | 300.00 | 72.86 | 2.95 | 865.61 | 0.03 | 3.86 | 300.00 | 8.00 | 8.81 | 358.80 | 0.00 | 0.08 | 9.05 | 0.51 | 7.89 | 33.73 | 0.03 |
| W4 | 7.18 | 300.00 | 49.60 | 4.74 | 966.85 | 0.02 | 16.47 | 300.00 | 31.83 | 2.23 | 645.60 | 0.02 | 0.09 | 10.26 | 0.65 | 5.13 | 104.41 | 0.00 |
| W5 | 12.12 | 300.00 | 92.69 | 5.91 | 1375.41 | 0.02 | 0.49 | 300.00 | 14.00 | 7.19 | 609.07 | 0.02 | 1.39 | 104.51 | 6.86 | 7.41 | 156.11 | 0.02 |
| W6 | 9.13 | 300.00 | 240.00 | 3.53 | 1657.26 | 0.02 | 12.59 | 300.00 | 14.91 | 0.51 | 774.38 | 0.03 | 1.19 | 32.85 | 1.44 | 6.02 | 203.00 | 0.00 |

In Table XII, "HP" stands for heuristic procedure, GOSTW(HP) means that the GO-STW method is solved by the proposed heuristic, "delay" denotes the total vehicle delay associates with a certain time window, and "time" denotes the total computation time to obtain the solution associates with a certain time window. Both "delay" and "time" are presented in seconds. GO-STW (solved directly by CPLEX
solver) provides a traffic efficiency upper bound but requires the longest computation time. In contrast, FCFSWR provides a traffic efficiency lower bound and requires the shortest computation time. The results in Table XII show that while GO-STW(HP) is less efficient than GOSTW, GO-STW(HP) only requires a computation time less than 10 seconds for each time window. Moreover, GO-

STW(HP) yields substantial efficiency gains against the FCFS-WR. The above results illustrate the potential of the proposed method to be applied in real-time applications. Future research may further examine more efficient algorithms to solve the proposed models.

## V. Conclusions

This paper proposed a lane-based intersection control model for connected and automated vehicles, where vehicle's physical size is incorporated. Specifically, we develop a "trajectory band" based approach to accommodate the vehicle size, which may be adapted in other similar scenarios with and without AVs. We demonstrate that with AVs, lane-based route choices should be considered to improve intersection traffic efficiency.

We develop two methods for optimizing vehicle entering time (i.e., when to enter the intersection) and route choice decisions, including the sliding-time-window-based global optimum (GO-STW) and the first-come-first-served method with optimal route choices (FCFS-R). The developed lanebased models can be transformed into mixed integer linear programming (MILP) problems, which can be solved by CPLEX solver. We also investigate the impacts of lanechanging through two aspects, i.e., adding a lane-changing penalty term into the objective function and adding constraints on travelers' entrance lane and exit lane choices.

To improve the computation efficiency, a heuristic is developed to solve the proposed MILP model. Numerical tests show that the adopted heuristic can provide an effective solution in a given time frame, which illustrates the potential of the proposed methods to be further developed for real-time application. Our numerical studies also reveal that the optimization of lane/route choices is often more critical than the optimization of entering time. This highlights the importance of this paper to consider lane-based route choices.

This paper mainly focuses on optimization of entering time and route choices of AVs inside an intersection. The study scope can be expanded from an intersection to the intersection combined with its connecting roadways. In this case, the speed and the full trajectory of each vehicle on the roadways and within the intersection can be optimized jointly. The proposed lane-based model for the intersection can be integrated with modeling of vehicle's lane-changing and car-following behavior. We indeed conducted preliminary simulations and found it feasible to integrate the lane-based model for intersection management and lanechanging and speed adjustment along the approach lanes. Moreover, future studies may also examine both timeefficient and energy-efficient control strategies when optimizing the speed and trajectory profile of vehicles. Furthermore, this study only considers private AVs. In a future study, we may consider autonomous buses' priority at the intersections, and examine automated intersection in a dynamic and multimodal context. Last but not least, it is of
our interest to examine prediction-based models with a certain prediction time window to facilitate real-time applicability of the proposed method.

## Acknowledgment

We are grateful to the reviewers for their useful comments, which helped improve the paper substantially. This research was supported by the National Natural Science Foundation of China (No. 61773077), the Research Foundation of Education Bureau of Hunan Province (No. 19A023), and the Australian Research Council (DE200101793).

## APPENDIX: TRAVEL TIME CALCULATION WITH TIMEVARYING SPEEDS AND ACCELERATIONS.

This appendix discusses how vehicles' travel times within grids can be obtained where the speed might vary with time. For turning movement, i.e., left-turn or right-turn, elliptic curve for vehicle movement can be adopted, which can be written as:
$\left\{\begin{array}{l}x=p \cos \eta \\ y=q \sin \eta\end{array}\right.$
$\left\{\begin{array}{l}x=q \sin \eta\end{array}\right.$
where $p, q$ and $\eta$ are parameters for defining the elliptic curve. The point that a vehicle enters the intersection is denoted by $z_{1}\left(x_{1}, y_{1}\right)$, the point it enters a given grid is $z_{2}\left(x_{2}, y_{2}\right)$, which can be obtained from Section II-C. The travel time from point $z_{1}\left(x_{1}, y_{1}\right)$ to $z_{2}\left(x_{2}, y_{2}\right)$ can be calculated by
$\eta_{1}=\arccos \frac{x_{1}}{p}$
$\eta_{2}=\arccos \frac{x_{2}}{p}$
where $\eta_{1}$ and $\eta_{2}$ are the eccentric angles of the ellipse at points $z_{1}\left(x_{1}, y_{1}\right)$ and $z_{2}\left(x_{2}, y_{2}\right)$, respectively.

The arc length of the ellipse from $z_{1}\left(x_{1}, y_{1}\right)$ to $z_{2}\left(x_{2}, y_{2}\right)$, i.e., $L_{z_{1} z_{2}}$, can be specified as
$L_{z_{1} z_{2}}=p \int_{\eta_{1}}^{\eta_{2}} \sqrt{1-e^{2} \cos ^{2} \theta} d \theta$
where $e$ is the eccentricity of the ellipse, which is
$e=\frac{\sqrt{p^{2}-q^{2}}}{p}, 0<e<1$
The arc length for a vehicle traveling along the ellipse can also be calculated by its time-varying speed and acceleration, which can be specified as
$L_{z_{1} z_{2}}=\int_{t_{z_{1}}}^{t_{z_{2}}}\left(v_{z_{1}}+\int_{t_{z_{1}}}^{t_{z_{2}}} a(t) d t\right) d t$
where $t_{z_{1}}$ is the time for the vehicle entering the intersection, $t_{z_{2}}$ is the time for the vehicle entering the grid. $v_{z_{1}}$ is the speed the vehicle entering the intersection. $a(t)$ is the acceleration rate at time $t$. By combining Eqs. (34)-(39), one can calculate the travel time from stop line to a given grid, i.e., the value of $t_{z_{2}}$.

For through movements, their trajectories are straight lines. The arc length for a vehicle traveling along the straight line can be written as
$L_{z_{1} z_{2}}=\sqrt{\left(x_{2}-x_{1}\right)^{2}+\left(y_{2}-y_{1}\right)^{2}}$
By combining Eq. (39) and Eq. (40), $t_{z_{2}}$ can be obtained accordingly.

## REFERENCES

[1] J. Meyer, H. Becker, P. M. Bösch, and K. W. Axhausen, "Autonomous vehicles: The next jump in accessibilities?," Research in Transportation Economics, vol. 62, pp. 80-91, 2017.
[2] W. Liu, "An equilibrium analysis of commuter parking in the era of autonomous vehicles," Transportation Research Part C-Emerging Technologies, vol. 92, pp. 191-207, 2018.
[3] X. Zhang, W. Liu, S. T. Waller, and Y. Yin, "Modelling and managing the integrated morning-evening commuting and parking patterns under the fully autonomous vehicle environment," Transportation Research Part B: Methodological, vol. 128, pp. 380407, 2019.
[4] F. Zhu and S. V. Ukkusuri, "An optimal estimation approach for the calibration of the car-following behavior of connected vehicles in a mixed traffic environment," IEEE Transactions on Intelligent Transportation Systems, vol. 18, no. 2, pp. 282-291, 2016.
[5] F. Zhu and S. V. Ukkusuri, "Modeling the Proactive Driving Behavior of Connected Vehicles: A Cell - Based Simulation Approach," Computer - Aided Civil and Infrastructure Engineering, vol. 33, no. 4, pp. 262-281, 2018.
[6] Z. B. Chen, F. He, Y. F. Yin, and Y. C. Du, "Optimal design of autonomous vehicle zones in transportation networks," Transportation Research Part B-Methodological, vol. 99, pp. 44-61, 2017.
[7] Z. B. Chen, F. He, L. H. Zhang, and Y. F. Yin, "Optimal deployment of autonomous vehicle lanes with endogenous market penetration," Transportation Research Part C-Emerging Technologies, vol. 72, pp. 143-156, 2016.
[8] M. W. Levin and D. Rey, "Conflict-point formulation of intersection control for autonomous vehicles," Transportation Research Part CEmerging Technologies, vol. 85, pp. 528-547, 2017.
[9] A. I. M. Medina, N. van de Wouw, and H. Nijmeijer, "Cooperative Intersection Control Based on Virtual Platooning," Ieee Transactions on Intelligent Transportation Systems, vol. 19, no. 6, pp. 1727-1740, 2018.
[10] C. H. Yu, W. L. Sun, H. X. Liu, and X. G. Yang, "Managing connected and automated vehicles at isolated intersections: From reservation- to optimization-based methods," Transportation Research Part B-Methodological, vol. 122, pp. 416-435, 2019.
[11] S. Le Vine, X. Liu, F. Zheng, and J. Polak, "Automated cars: Queue discharge at signalized intersections with 'Assured-Clear-DistanceAhead'driving strategies," Transportation Research Part C: Emerging Technologies, vol. 62, pp. 35-54, 2016.
[12] A. Mirheli, L. Hajibabai, and A. Hajbabaie, "Development of a signal-head-free intersection control logic in a fully connected and autonomous vehicle environment," Transportation Research Part C: Emerging Technologies, vol. 92, pp. 412-425, 2018.
[13] J. Q. Ma, X. P. Li, F. Zhou, and W. Hao, "Designing optimal autonomous vehicle sharing and reservation systems: A linear programming approach," Transportation Research Part C: Emerging Technologies, vol. 84, pp. 124-141, 2017.
[14] K. Dresner and P. Stone, "A multiagent approach to autonomous intersection management," Journal of Artificial Intelligence Research, vol. 31, pp. 591-656, 2008.
[15] Y. Wu, H. Chen, and F. Zhu, "DCL-AIM: Decentralized coordination learning of autonomous intersection management for connected and automated vehicles," Transportation Research Part C: Emerging Technologies, vol. 103, pp. 246-260, 2019.
[16] L. Chen and C. Englund, "Cooperative intersection management: a survey," IEEE Transactions on Intelligent Transportation Systems, vol. 17, no. 2, pp. 570-586, 2016.
[17] Z. X. Li, M. V. Chitturi, D. X. Zheng, A. R. Bill, and D. A. Noyce, "Modeling Reservation-Based Autonomous Intersection Control in VISSIM," Transportation Research Record, no. 2381, pp. 81-90, 2013.
[18] M. Bashiri and C. H. Fleming, "A platoon-based intersection management system for autonomous vehicles," in Intelligent Vehicles Symposium (IV), 2017 IEEE, 2017, pp. 667-672: IEEE.
[19] D. Carlino, S. D. Boyles, and P. Stone, "Auction-based autonomous intersection management," in Intelligent Transportation Systems-
(ITSC), 2013 16th International IEEE Conference on, 2013, pp. 529-534: IEEE.
[20] M. W. Levin and S. D. Boyles, "Intersection Auctions and Reservation-Based Control in Dynamic Traffic Assignment," Transportation Research Record, no. 2497, pp. 35-44, 2015.
[21] W. Liu, H. Yang, and Y. Yin, "Efficiency of a highway use reservation system for morning commute," Transportation Research Part C: Emerging Technologies, vol. 56, pp. 293-308, 2015.
[22] J. Lee and B. Park, "Development and Evaluation of a Cooperative Vehicle Intersection Control Algorithm Under the Connected Vehicles Environment," IEEE Transactions on Intelligent Transportation Systems, vol. 13, no. 1, pp. 81-90, 2012.
[23] S. Krishnan, R. G. Aadithya, R. Ramakrishnan, V. Arvindh, and K. Sivanathan, "A Look at Motion Planning for AVs at an Intersection," in 2018 21st International Conference on Intelligent Transportation Systems (ITSC), 2018, pp. 333-340: IEEE.
[24] F. Zhu and S. V. Ukkusuri, "A linear programming formulation for autonomous intersection control within a dynamic traffic assignment and connected vehicle environment," Transportation Research Part C: Emerging Technologies, vol. 55, pp. 363-378, 2015.
[25] Z. He, L. Zheng, L. Lu, and W. Guan, "Erasing lane changes from roads: A design of future road intersections," IEEE Transactions on Intelligent Vehicles, vol. 3, no. 2, pp. 173-184, 2018.
[26] B. Liu, Q. Shi, Z. Y. Song, and A. El Kamel, "Trajectory planning for autonomous intersection management of connected vehicles," Simulation Modelling Practice and Theory, vol. 90, pp. 16-30, 2019.
[27] D. Rey and M. W. Levin, "Blue phase: Optimal network traffic control for legacy and autonomous vehicles," Transportation Research Part B: Methodological, vol. 130, pp. 105-129, 2019.
[28] C. Yu, Y. Feng, H. X. Liu, W. Ma, and X. Yang, "Corridor level cooperative trajectory optimization with connected and automated vehicles," Transportation Research Part C: Emerging Technologies, vol. 105, pp. 405-421, 2019.
[29] Q. Lu and K. D. Kim, "A Mixed Integer Programming Approach for Autonomous and Connected Intersection Crossing Traffic Control," in 2018 IEEE 88th Vehicular Technology Conference: VTC2018Fall, 2018, pp. 1-6.
[30] S. A. Fayazi and A. Vahidi, "Mixed Integer Linear Programming for Optimal Scheduling of Autonomous Vehicle Intersection Crossing," IEEE Transactions on Intelligent Vehicles, vol. 3, no. 3, pp. 287-299, 2018.
[31] M. Fischetti and M. Monaci, "Proximity search for 0-1 mixedinteger convex programming," Journal of Heuristics, vol. 20, no. 6, pp. 709-731, 2014.
[32] D. Chen, S. Ahn, M. Chitturi and D.A. Noyce, "Towards vehicle automation: Roadway capacity formulation for traffic mixed with regular and automated vehicles," Transportation Research Part B: Methodological, vol. 100, pp. 196-221, 2017.


Wei WU is an Associate professor at Changsha University of Science and Technology. His primary research interest is in the area of traffic control and management within connected and autonomous vehicles environment.


Yang LIU is a Ph.D. candidate at the Changsha University of Science and Technology. He received his master degree from Changsha University of Science and Technology in June 2020. His research interest covers traffic control and intersection management.


Wei LIU is an Associate Professor at The Hong Kong Polytechnic University. He mainly works in data-driven decision analytics; large-scale transport system modeling, optimization and computing; aviation and rail system modeling; transport and aviation economics.


Fangni Zhang is an Assistant Professor at The University of Hong Kong. Her research interests include multimodal transport system, shared and automated transport management, high-speed rail and air transport interactions, and wider economic impacts of transport.


Vinayak Dixit is a Professor in the School of Civil and Environmental Engineering at the University of New South Wales, Sydney. His key research interest lies in studying risk in the transportation infrastructure system as it relates to highway safety, travel time uncertainty.

S. Travis Waller is the Advisian Chair of Transport Innovation and Head of Civil and Environmental Engineering at the University of New South Wales, Sydney. Prof. Waller received his BS in Electrical Engineering from The Ohio State University and his MS/PhD degrees in Industrial Engineering and Management Sciences from Northwestern University. Before, Prof. Waller was on the faculty at University of Texas at Austin in the US.


[^0]:    This work was supported by the National Natural Science Foundation of China under Grant No. 61773077, the Research Foundation of Education Bureau of Hunan Province under Grant No. 19A023, and Australian Research Council (DE200101793) (Corresponding author: Wei Liu)
    W. Wu and Y. Liu are with the Key Laboratory of Highway Engineering of Ministry of Education, Changsha University of Science \& Technology, Changsha 410114, Hunan, China (e-mail: jiaotongweiwu@csust.edu.cn; liuyang@stu.csust.edu.cn).
    W. Liu is with the Department of Aeronautical and Aviation Engineering, The Hong Kong Polytechnic University, Hong Kong SAR, China (e-mail: wei.w.liu@polyu.edu.hk).
    V. Dixit and T. Waller are with the Research Centre for Integrated Transport Innovation, School of Civil and Environmental Engineering, University of New South Wales, Sydney, NSW 2052, Australia (e-mail: v.dixit@unsw.edu.au; s.waller@unsw.edu.au);
    F. Zhang is with Department of Industrial and Manufacturing Systems Engineering, University of Hong Kong, Hong Kong, China (e-mail: fnzhang@hku.hk).

