2018 IEEE Intelligent Vehicles Symposium (IV)
Changshu, Suzhou, China, June 26-30, 2018

Dynamic Space-time Resource Allocation for Signal-less Intersection Management in a Connected Autonomous Vehicle Environment

Nannan Wang[1], Xi Wang[1], Paparao Palacharla[1], Tadashi Ikeuchi[1]

Abstract— **In this paper, we consider the problem of dynamic space-time resource allocation for optimizing the movements of connected autonomous vehicles (CAVs) through intersections without traffic signals. We design a three-dimensional (3D) space-time resource model for maintaining the intersection resource information in both the two-dimensional (2D) space domain and the time domain. In the 3D resource model, the trajectory of a CAV through an intersection is assigned a specific parallelepiped resource that spans both 2D space and time domains. Moreover, the dynamic space-time resource allocation problem is simplified to a classic 3D container-packing problem. We propose a dynamic heuristic algorithm, Best Parallelepiped Fit (BPF), to maintain smooth traffic flow and maximize space-time resource usage by adjusting the speed and entry time of each approaching CAV through intersections. We evaluate the performance of the proposed algorithm under different traffic loads, and simulation results indicate that our algorithm can greatly reduce the average travel delay of CAVs.**

I. INTRODUCTION

The automotive industry is currently undergoing a significant transformation from human-driven to semi or fully autonomous vehicles (AVs). Several leading automakers and information technology companies around the world have announced their plans of deploying fully AVs by 2025 [1], [2], while, testing of driver-free AVs on public roads has started recently [3].

Meanwhile, intersection traffic management remains a challenging issue for intelligent transportation systems (ITS). In the US, intersection-related accidents account for 44.8% of all crashes and 21.5% of all traffic fatalities [4]. Moreover, intersections tend to become traffic bottlenecks, which lead to longer travel time [5]. Traditional intersection management methods, such as traffic lights and stop signs, cannot take full advantage of AVs. For example, AVs are capable of more accurate and reliable steering and speed control, which allow them to traverse intersections at a tighter inter-vehicle gap, thus improving overall traffic flow. In addition, AVs are expected to incorporate Vehicle-to-Everything (V2X) communication technologies (e.g., Dedicated Short Range Communication) for enhanced safety and traffic management [6]. We call these AVs the Connected Autonomous Vehicles (CAVs) hereafter. These new technologies provide a tremendous potential for coordinating the movements of CAVs through signal-less intersections that eliminate conventional stop & go traffic signs or signals. One such pioneering approach to autonomous intersection management is introduced by Dresner and Stone [7], in which a CAV can reserve a block of unique space-time resource for traversing an intersection, while maintaining safety and minimizing the waiting time. Compared with traditional traffic signal systems, autonomous intersection management (AIM) improves traffic throughput at intersections.

More recently, several other solutions [8], [9], [10], [11], [12] are proposed to implement autonomous intersection management for CAVs. In [8], a Cooperative Adaptive Cruise Control (CACC) system is designed to optimize the movements of CAVs in order to avoid collisions and minimize intersection delay. The work in [9] proposes a Cooperative Vehicle Intersection Control (CVIC) algorithm, which enables effective intersection operation and management under the connected vehicles environment. J. Wu et al. [10] introduce an algorithm of ant colony system to determine an optimized sequence of vehicles traversing intersections. An advanced traffic management system [11] based on multi-layer reservation policies is proposed to improve traffic flow and reduce fuel consumption. P. Lin et al. [12] propose a buffer-assignment based coordinated control method to guide CAVs through intersections cooperatively.

However, most existing solutions for autonomous intersection management do not leverage an efficient resource management method to optimize the movements of CAVs through intersections. In practice, each approaching CAV needs to reserve a specific block of space-time resource in order to traverse the intersection safely. Meanwhile, as large number of CAVs are entering and exiting the intersection constantly, the resource needs to be allocated in a dynamic and intelligent way so as to avoid unnecessary waiting time and maintain smooth traffic flow. Hence, it is essential to design a dynamic intersection space-time resource allocation algorithm to maximize intersection throughput while ensuring the safety of all CAVs.

The remainder of this paper is organized as follows. In Section II, we describe the challenges, requirements, and overview of our intelligent signal-less intersection management system. We propose the design of a three-dimensional space-time resource model for maintaining the intersection resource information. In Section III, we analyze the space-time resource allocation problem, which is simplified to a classic 3D container-packing problem. A dynamic space-time resource allocation algorithm (Best Parallelepiped Fit) is proposed in Section IV. Section V presents and evaluates the simulation results. Conclusions are presented in Section VI.

[1]Nannan Wang, [1]Xi Wang, [1]Paparao Palacharla, and [1]Tadashi Ikeuchi are with Fujitsu Laboratories of America, Richardson, Texas 75082, USA. {nannan.wang, xi.wang, paparao.palacharla, tadashi.ikeuchi}@us.fujitsu.com

978-1-5386-4451-5/18/$31.00 ©2018 IEEE

II. PROBLEM FORMULATION

A. Problem Statement

Intersection management for CAVs is a new challenge for ITS. Traditional signalized intersection management methods (e.g., traffic lights, stop signs, etc.) are designed and optimized for human drivers, which may not be well suitable for CAVs. This is because CAVs can use Vehicle-to-Infrastructure (V2I) communications instead of visual signals as a more advanced means for exchanging information with intersection management systems. Accordingly, in this paper, we design a communication-based Intelligent Signalless Intersection Management (ISIM) system to guide each approaching CAV through intersections. We consider the following three main principles to guide the design of our ISIM system.

(1) *Safety*: intersection safety is a top priority issue for the ISIM system. Collisions often happen at intersections because they are resource-contending areas that are prone to conflicts. Without stop & go traffic signals, the ISIM system needs to allocate a sufficient and exclusive block of space-time resource for each approaching CAV to ensure its safety.

(2) *Efficiency*: intersections tend to become the bottlenecks of the transportation system as the traffic increases. The ISIM system needs to allocate intersection space-time resource to CAVs efficiently in order to reduce traffic congestion and travel delay at the intersection area. In this paper, we define the travel delay of a CAV crossing an intersection as the extra delay time caused by the slowing down or stopping of the CAV for safely traversing the intersection area, which is formally defined in Section V.

(3) *Fairness*: the traversing sequence of CAVs is scheduled by the ISIM system. In the same lane, the first coming CAV should be served firstly. Furthermore, each approaching CAV in any lane should be able to eventually traverse the intersection without experiencing excessive delay. A dynamic intersection resource allocation algorithm should be designed to provide fairness among all CAVs in all lanes.

B. ISIM System Introduction

The ISIM system is established at each intersection for managing the intersection traffic flow. The core component of the ISIM system is the intersection manager (IM), which is in charge of guiding each approaching CAV through the intersection safely and smoothly. The intersection and its surrounding areas controlled by the IM are divided into three zones, namely the request zone, the adjustment zone, and the intersection zone, as shown in Fig. 1. All CAVs within these three zones must follow the trajectories as instructed by the IM. The detail of the working procedure for CAVs to request a trajectory through an intersection is shown as follows:

(1) Through vehicular communication technologies, the IM requests that all CAVs within these three zones must be connected to the IM and follow its instructions. When a CAV realizes that it has entered the request zone of an upcoming intersection, it sends a traversing request (a V2I message) to the IM for requesting a block of exclusive space-time resource of the intersection zone. The V2I message

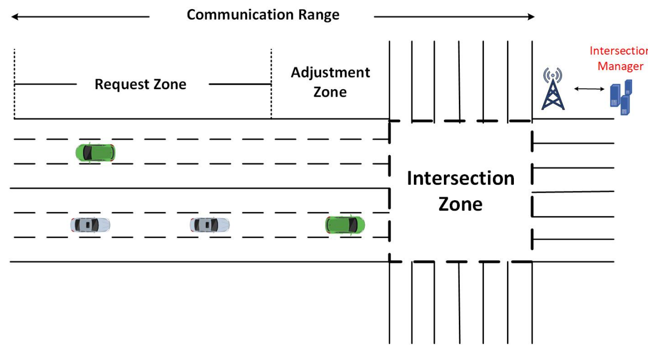


Fig. 1. Three zones controlled by the intersection manager

consists of the current vehicle speed and position, entry lane, departure lane, request failure count, and vehicle properties. In particular, the request failure count keeps track of the number of times that an intersection traversing request was denied; the vehicle properties include its width, length, maximal speed, acceleration and deceleration, etc.

(2) The IM attempts to reserve a trajectory in terms of an exclusive block of space-time resource to satisfy the CAV's traversing request, and sends either an approval reply or a rejection reply (in an I2V message) back to the CAV. An approval I2V message contains the reserved trajectory, which is prescribed as the specific time that the CAV enters the intersection zone (the entry time), the lanes to be traversed, and the speed of the CAV inside the intersection zone (the traversing speed). We assume that the IM can assign a traversing speed among several speed choices in order to support individual CAV's properties/constraints as well as improve the intersection resource usage. Each CAV must maintain a constant traversing speed as specified by the IM within the intersection zone.

(3) If the CAV receives an approval message, it stops sending additional traversing request and takes the responsibility to enter and traverse the intersection zone following the reserved trajectory. The CAV uses the adjustment zone to adjust its speed and entry time into the intersection zone. If the CAV receives a rejection message, the CAV increments its request failure count by one and repeatedly sends additional traversing requests to the IM until either the request is approved or the CAV enters the adjustment zone.

(4) If the CAV does not receive an approval message within the request zone, the CAV slows its speed in the adjustment zone so as to be able to fully stop before entering the intersection zone to avoid collision. Meanwhile, the CAV starts sending urgent requests with the request failure count set to infinity. The IM schedules such urgent requests with higher priority in order to minimize traffic slowdown.

C. Three-Dimensional (3D) Space-time Resource Model

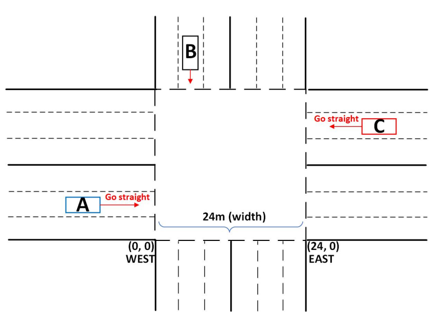
In order to traverse an intersection, each approaching CAV needs to get a reserved block of space-time resource inside the intersection zone. For this, we design a three-dimensional (3D) space-time resource model to represent the space-time resource information of the intersection zone. The

3D space-time resource model includes three dimensions: time dimension and two-dimensional space. The minimal resource allocation unit in the 3D space-time resource model is a small cube that is shaped by a granularity (length, width, and height). When one minimal resource allocation unit is reserved by one request, then this unit is not available for other requests.

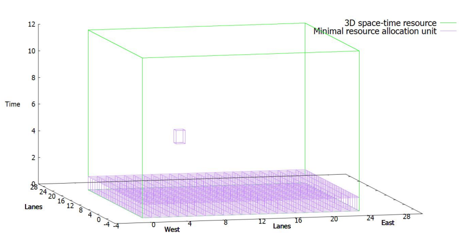
Fig. 2 shows a 3D space-time resource model for a 4-way intersection zone with 6 lanes on each side and a certain time domain. Fig.2 (a) shows the intersection zone, the length and width of which are 24 meters. The width of each lane is 4 meters. Fig.2 (b) shows the 3D space-time resource model, where the vertical axis denotes the time domain and the horizontal two-dimensional space denotes the intersection zone. We assume that the two-dimensional space coordinate of southwest corner of the intersection zone is (0, 0). The large green cube shows the 3D space-time resource, which indicates the resource condition of the intersection zone from time 0 to time 12 units. The small pink cube in the middle of the 3D space-time resource shows a minimal resource allocation unit, whose length, width, and height are all one unit. The 3D space-time resource can be divided into many minimal resource allocation units.

D. Parallelepiped Resource

We use the 3D space-time resource model to reserve and allocate space-time resource for each approaching CAV. The block of reserved space-time resource should be sufficient for a CAV to safely traverse the intersection zone through a specific trajectory. We first consider non-turning (straight) traffic in the intersection zone. We assume that the shape of all CAVs is a rectangle, and each CAV maintains a constant speed when traversing the intersection zone. Thus, the allocated space-time resource for a CAV's trajectory can

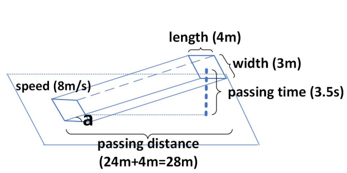


(a) Intersection zone

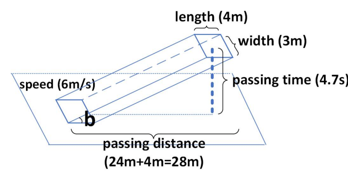


(b) 3D space-time resource model

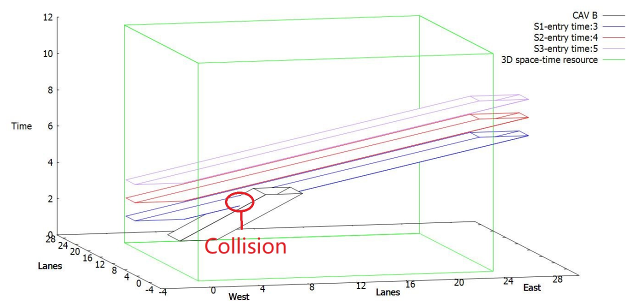
Fig. 2. An example of 3D space-time resource model



(a) Parallelepiped R



(b) Parallelepiped S



(c) Trajectories over different entry times for parallelepiped S

Fig. 3. An example of parallelepiped resource

be represented as a parallelepiped in the 3D space-time resource model. The IM has multiple speed choices for a CAV. Hence, the traversing request of a CAV can be supported by parallelepipeds with different slopes that correspond to the speed choices. Also, the different entry times of the CAVs can be represented by parallelepipeds with different elevations. In addition, the trajectory of turning traffic (left turns and right turns) can be represented by a smoothly curved 3D shape (a 'curved' parallelepiped), which can be approximated by a group of continuous parallelepipeds each with a slightly different turning angle.

For example, in Fig. 2(a), CAV A is going straight crossing the intersection. We assume that CAV A is 4 meters long and 3 meters wide, and has 8 m/s and 6 m/s speed choices for traversing the intersection. In Fig. 3(a) and Fig. 3(b), we show that two parallelepipeds (R and S) with different slopes can support CAV A with speeds of 8 m/s and 6 m/s. In Fig. 3(a), the base of parallelepiped R is a rectangle representing CAV A, the length and the width of which is 4 meters and 3 meters, respectively. The height of parallelepiped R is the traversing time (3.5 seconds), meaning that the CAV needs 3.5 seconds to traverse the intersection zone. The reciprocal of the slope ($\tan^{-1}(a)$) is the speed 8 m/s. Fig. 3(b) shows that 4.7 seconds is needed to traverse the intersection zone if the IM decides the 6 m/s ($\tan^{-1}(b)$) speed for CAV A.

This way, the trajectory of each approaching CAV can be determined by the reserved entry time and traversing speed. Fig. 3(c) shows three different trajectories based on parallelepiped S with speed 6 m/s over three different entry times. The entry time of the trajectories in blue, red, and purple is 3, 4, and 5, respectively. In Fig. 3(c), we assume that the black trajectory from north to south has been reserved for CAV B. The blue trajectory overlaps with the black trajectory, meaning expected collision between CAV A and CAV B, if entry time 3 and speed 6 m/s are allocated to CAV A.

Following the other two trajectories, however, CAV *A* can traverse the intersection safely. In this case, it is obvious that the red trajectory is better than the purple one because earlier entry time means less travel delay for CAV *A*.

III. Parallelepiped Packing Problem

The space-time resource reserved for a CAV's intersection traversing request can be denoted as a parallelepiped in the 3D resource model. For each CAV, different parallelepipeds with different speed choices and entry times can be considered. In addition, since CAVs arrive from all directions/lanes constantly, the IM needs to provide resource reservation for multiple CAVs at a time. Any placement of a group of non-overlapping parallelepipeds, which implies trajectories of multiple CAVs, is a solution for a group of approaching CAVs. Hence, the space-time resource allocation problem for approaching CAVs can be simplified to a classic 3D container-packing problem, which is a representative NP-hard problem [13]. It is essential for us to design a heuristic algorithm to find placement solutions for different parallelepipeds, such that the 3D resource model can accommodate as many parallelepipeds of requests as possible.

A. Sliding Time Window and Solution Space

We define the following two key concepts for investigating the space-time resource allocation problem: the sliding time window and the solution space.

The sliding time window: the sliding time window is a time frame during which a CAV is allowed to enter the intersection zone. The IM is able to calculate the earliest entry time of each CAV to the intersection zone based on its current speed and current position. For the sliding time window, we set up a time buffer, which is equal to 2 to the power of the request failure count. We define that the sliding time window is the time frame from the earliest entry time to the earliest entry time plus the time buffer.

The solution space: within the sliding time window, a CAV can be assigned a different entry time. We define that the solution space is the 3D space that consists of all possible placements of different parallelepipeds based on different speed choices over different entry times. If a CAV is given a bigger solution space, the CAV has a higher probability of obtaining the requested space-time resource.

For example, in Fig. 2(a), both CAV *A* and CAV *C* are going straight crossing the intersection zone. We assume that the earliest entry time of CAV *A* and CAV *C* is 2 and 1, respectively. The request failure count of CAV *A* and CAV *C* is 2 and 3, respectively. Hence, the sliding time window for CAV *A* is $[2, 2 + 2^2] = [2, 6]$. The sliding time window for CAV *C* is $[1, 1 + 2^3] = [1, 9]$. In addition, we assume that CAV *A* and CAV *C* each has only one speed choice: 12 m/s and 8 m/s, respectively. In Fig. 4, the black 3D space is the solution space for CAV *A*, and the red 3D space is the solution space for CAV *B*.

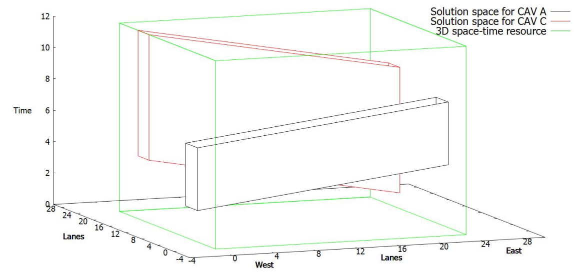


Fig. 4. Solution spaces for CAV A and CAV C

IV. Dynamic Space-time resource allocation algorithm

Leveraging the 3D space-time resource model, we design a heuristic algorithm, namely Best Parallelepiped Fit (BPF), to allocate a parallelepiped space-time resource for each CAV crossing the intersection. Based on the solution space discussed above, the BPF algorithm decides the placement of a group of parallelepipeds with the aim of maximizing the intersection traffic throughput. BPF consists of the following three phases: (1) filter the traversing requests from all approaching CAVs, and construct an initial set, (2) decide a processing sequence for the traversing requests in the initial set, (3) calculate the solution space, and find a best-possible placement solution for the traversing requests.

In Phase I, the IM collects the traversing requests (V2I messages) received from all approaching CAVs at each time step. To avoid starvation and maintain fairness, the IM only accepts the traversing request from the front-most CAV in each lane. Note that the relative position among CAVs can be deducted from the position information contained in the request message. All these traversing requests are stored in an initial set.

In Phase II, each approaching CAV updates its request failure count based on the reply from the IM. If the traversing request of a CAV is rejected by the IM, the CAV increments its request failure count by one and sends another traversing request at the next time step. Meanwhile, based on the descending order of failure counts, the IM constructs a processing sequence for all traversing requests in the initial set. The traversing request with the largest failure count is placed in the headmost position in this sequence, which will be served firstly. If two or more CAVs have a same failure count, their processing sequence is randomly decided.

In Phase III, based on the request failure count, the IM calculates the sliding time window for each traversing request in the sequence. In addition, the solution space of each traversing request is calculated based on the sliding time window and different speed choices. The traversing requests are then processed one by one in this sequence. For each traversing request, the IM finds a parallelepiped placement within the request's own solution space which will result in either *zero*, or if not possible, *minimal* overlapping with other pending traversing requests' solution spaces. By doing so, the subsequent requests will have a better chance of finding mutually non-overlapping solutions among their

Algorithm 1 Best Parallelepiped Fit Algorithm

Input: Traversing request set (*Tre*)
Output: Speed and entry time for CAVs

Begin
for each traversing request *tr* in *Tre* **do**
  if *tr* is coming from any lane's front-most CAV **then**
    Store *tr* in an initial set *IS*.
  end if
end for
Construct a processing sequence *PS* for all traversing requests in *IS* according to the descending order of the request failure count.
for each traversing request *tr* in *PS* **do**
  Calculate the sliding time window *stw* for *tr*.
  Based on the *stw* and different speed choices, calculate the solution space *ss* for *tr*.
end for
for each traversing request *tr* in *PS* **do**
  Find a parallelepiped placement for *tr* if the placement has *zero* or *minimal* overlapping space with other pending requests' solution spaces.
  Figure out the speed and the entry time of the found parallelepiped placement for *tr*.
end for
End

solution spaces. In case the overlapping of two or more parallelepiped placement choices has the same overlapping volume, the IM selects a parallelepiped placement that has the earliest entry time. Since each request is sequentially assigned a parallelepiped that is non-overlapping with all previous parallelepipeds, all traversing requests will get non-overlapping space-time resource in the end, thereby ensuring collision-free intersection traversing. Each parallelepiped placement decides both the traversing speed and the entry time of the CAV. The BPF algorithm is shown in Algorithm 1.

Time complexity analysis: we assume that n, m, and k are the numbers of traversing requests, entry time choices, and speed choices, respectively. In Phase I of BPF, the time complexity of filtering traversing requests is $\mathcal{O}(n)$. In Phase II, the time complexity of sorting all traversing requests is $\mathcal{O}(n\log n)$. In Phase III, each traversing request has $m{\cdot}k$ parallelepiped placements, any of which needs to be compared with other requests' solution spaces, so that the time complexity of finding a valid solution for the traversing requests is $\mathcal{O}(n^2mk)$. Hence, the time complexity of the BPF algorithm is $\mathcal{O}(n^2mk)$.

V. Simulation Results

A. Simulation Setup

The proposed dynamic BPF algorithm is numerically evaluated through simulations. We write a simulator implementing the intersection zone shown in Fig. 1, as well as its surrounding request zones and adjustment zones in all directions. The numerical values of the intersection are as described in Section II, which is a 4-way intersection with 24 meters width, 3 lanes per road, and 4 meters width per lane. Based on the same traffic load for each lane, we randomly generate five types of CAVs with different vehicle properties as shown in Table I. For simplicity, we assume that the width and the length of all CAVs are 3 meters and 4 meters, respectively. The middle lane of each road only generates non-turning traffic. The left-most lane generates non-turning traffic and left-turning traffic with a mix ratio of 1:1. In a similar way, the right-most lane generates non-turning traffic and right-turning traffic with a mix ratio of 1:1. Hence, the mix ratio among non-turning traffic, left-turning traffic and right-turning traffic is 4:1:1. We apply the classic vehicle following model [14] to the CAVs in our simulation, and the CAVs are not allowed to change lane within the request zone, the adjustment zone, and the intersection zone. For safety purpose, we follow a similar approach as [7] to add a constant space buffer around each CAV when making the space-time resource reservation, which ensures sufficient inter-vehicle gap to avoid potential collisions with nearby vehicles. We assume that V2I and I2V messages between the CAVs and the IM are sent and received through DSRC technology [15], which is able to support approximately 1 kilometer communication distance. In terms of travel delay, the performance of the proposed BPF algorithm is evaluated under different traffic loads ranging from 200 vehicles/hour to 1100 vehicles/hour per lane. We randomly generate 10,000 CAVs per simulation run for each traffic load. Two baseline algorithms: FCFS [7] and AVCM [12] are also simulated for comparison. FCFS allocates space-time resource for each CAV on a simple first-come, first-served basis. AVCM utilizes a three-segment linear speed profile to instruct CAVs through intersections.

TABLE I
Simulation parameters

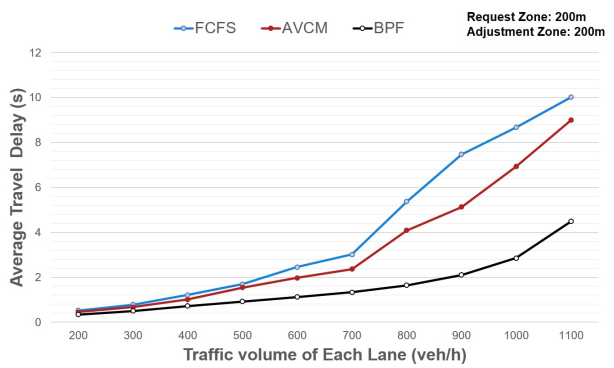
Vehicle Type	Max Speed (km/h)	Max Acceleration (km/h/s)	Max Deceleration (km/h/s)
A	50.0	15.0	30.0
B	55.0	17.5	32.0
C	60.0	20.0	34.0
D	65.0	22.5	36.0
E	70.0	25.0	38.0

B. Travel Delay

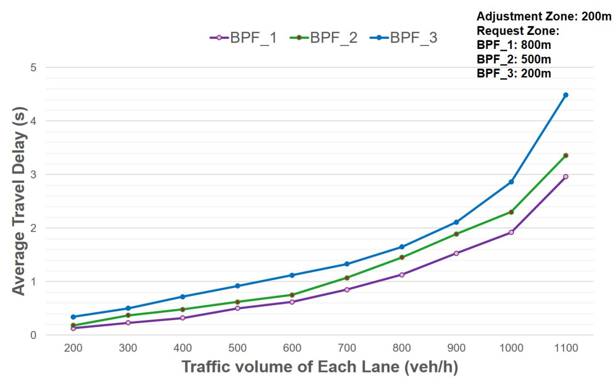
Based on the reserved entry time and traversing speed, we define the travel delay for each CAV crossing the intersection as shown in the formula below.

$$T = At - Et + \frac{Td}{Rs} - \frac{Td}{Ts} \tag{1}$$

in which T is the travel delay, Et and At are the earliest entry time and the reserved entry time for entering the intersection zone, respectively. Td is the traversing distance in the intersection zone with the CAV's length included. Rs and Ts are the reserved speed and the target speed within the intersection zone, respectively. As mentioned previously, the IM has several speed choices for each CAV. We define



(a) Travel delay among BPF, AVCM, and FCFS



(b) Travel delay for different request zone lengths

Fig. 5. Simulation results on travel delay

that the target speed is the maximal speed among all speed choices. Therefore, the reserved speed is equal or less than the target speed. According to this definition, if a CAV can enter the intersection at its earliest possible entry time, in other words, maintaining its original speed without slowing down at the adjustment zone, and can also traverse the intersection at the highest possible speed, the CAV will experience zero travel delay for crossing the intersection.

In Fig. 5(a), the average travel delay is evaluated among BPF, AVCM, and FCFS based on different traffic loads. The lengths of the request zone and the adjustment zone are all set to 200 meters. We assume that the IM has two speed choices: 50 km/h and 55 km/h. Compared to the other two algorithms, we confirm that BPF achieves a lower average travel delay. The main reason for this result is that the proposed BPF algorithm efficiently allocates space-time resource in the 3D resource model to each approaching CAV. Furthermore, Fig. 5(b) compares the average travel delay for BPF with different lengths of the request zone. The length of the request zone for BPF_1, BPF_2, and BPF_3 are 800, 500, and 200 meters, respectively. We find that BPF_1 with the longest request zone achieves the lowest travel delay. This is because the longer the request zone, the earlier a CAV can send the request. Consequently, the IM can plan further ahead in the 3D space-time resource model, with more chances of finding more optimized allocations of space-time resource for upcoming CAVs.

VI. CONCLUSION

In this paper, we propose a vehicular communication-based intelligent signal-less intersection management system to guide connected autonomous vehicles through intersections. We design a three-dimensional space-time resource model for maintaining the intersection resource information, and propose a dynamic heuristic algorithm called BPF to allocate collision-free intersection resource to autonomous vehicles' traversing requests. We compare the proposed BPF algorithm with other algorithms under different traffic loads, and confirm that BPF can improve the intersection traffic flow through intelligent resource allocation, resulting in lower average travel delay and fuel consumption for autonomous vehicles crossing signal-less intersections.

REFERENCES

[1] D. Fagella, "Self-driving car timeline for 11 top automakers," https://venturebeat.com/2017/06/04/self-driving-car-timeline-for-11-top-automakers, Jun. 2017.
[2] A. J. Hawkins, "Gm will make an autonomous car without steering wheel or pedals by 2019," https://www.theverge.com/2018/1/12/16880978/gm-autonomous-car-2019-detroit-auto-show-2018, Jan. 2018.
[3] A. Aupperlee, "Waymo announces driver-free av testing in arizona," http://www.govtech.com/fs/transportation/Waymo-Announces-Driver-Free-AV-Testing-in-Arizona.html, Nov. 2017.
[4] J. Shaw, "Intersection safety issue briefs: the national intersection safety problem," https://www.theverge.com/2018/1/12/16880978/gm-autonomous-car-2019-detroit-auto-show-2018, Nov. 2009.
[5] D. Schrank, B. Eisele, and T. Lomax, "Tti's 2012 urban mobility report," *Texas A&M Transportation Institute. The Texas A&M University System*, Dec. 2012.
[6] K. Dar, M. Bakhouya, J. Gaber, M. Wack, and P. Lorenz, "Wireless communication technologies for its applications," *IEEE Communications Magazine*, vol. 48, no. 5, pp. 156–162, May 2010.
[7] K. Dresner and P. Stone, "A multiagent approach to autonomous intersection management," *Journal of artificial intelligence research*, vol. 31, pp. 591–656, Mar. 2008.
[8] I. H. Zohdy, R. K. Kamalanathsharma, and H. Rakha, "Intersection management for autonomous vehicles using icacc," in *Proceedings, IEEE International Conference on Intelligent Transportation Systems (ITSC)*, Anchorage, AK, Sep. 2012.
[9] 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, Jan. 2012.
[10] J. Wu, A. Abbas-Turki, and A. El Moudni, "Cooperative driving: an ant colony system for autonomous intersection management," *Applied Intelligence*, vol. 37, no. 2, pp. 207–222, Oct. 2012.
[11] Q. Jin, G. Wu, K. Boriboonsomsin, and M. Barth, "Advanced intersection management for connected vehicles using a multi-agent systems approach," in *Proceedings, IEEE Intelligent Vehicles Symposium (IV)*, Madrid, Spain, Jun. 2012.
[12] P. Lin, J. Liu, P. J. Jin, and B. Ran, "Autonomous vehicle-intersection coordination method in a connected vehicle environment," *IEEE Intelligent Transportation Systems Magazine*, vol. 9, no. 4, pp. 37–47, Oct. 2017.
[13] S. Martello, D. Pisinger, and D. Vigo, "The three-dimensional bin packing problem," *Operations Research*, vol. 48, no. 2, pp. 256–267, Apr. 2000.
[14] M. Treiber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," *Physical review E*, vol. 62, no. 2, p. 1805, Aug. 2000.
[15] Y. Jijun, E. Tamer, Y. Gavin, R. Bo, H. Stephen, K. Hariharan, and T. Timothy, "Performance evaluation of safety applications over dsrc vehicular ad hoc networks," in *Proceedings, ACM International Workshop on Vehicular Ad-hoc Networks*, Philadelphia, PA, Oct. 2004.