Joint Optimization of Vehicle Trajectories and Intersection Controllers with Connected Automated Vehicles: Combined Dynamic Programming and Shooting Heuristic Approach

Yi Guo[a], Jiaqi Ma[a*], Chenfeng Xiong[b], Xiaopeng Li[c], Fang Zhou[d], Wei Hao[e]

a. Department of Civil and Architectural Engineering and Construction Management, University of Cincinnati, Cincinnati, OH 45221, USA
b. Department of Civil and Environmental Engineering, University of Maryland, College Park, College Park MD 20742 USA
c. Department of Civil and Environmental Engineering, University of South Florida, FL 33620, USA
d. Cincinnati Children's Hospital Medical Center (CCHMC), OH 45229, USA
e. Transportation Engineering Department, Changsha University of Science and Technology, Changsha, Hunan 410205, China

Abstract

Connected and automated vehicle (CAV) technologies offer promising solutions to challenges that face today's transportation systems. Vehicular trajectory control and intersection controller optimization based on CAV technologies are two approaches that have significant potential to mitigate congestion, lessen the risk of crashes, reduce fuel consumption, and decrease emissions at intersections. These two approaches should be integrated into a single process such that both aspects can be optimized simultaneously to achieve maximum benefits. This paper proposes an efficient DP-SH (dynamic programming with shooting heuristic as a subroutine) algorithm for the integrated optimization problem that can simultaneously optimize the trajectories of CAVs and intersection controllers (i.e., signal timing and phasing of traffic signals), and develops a two-step approach (DP-SH and trajectory optimization) to effectively obtain near-optimal intersection and trajectory control plans. Also, the proposed DP-SH algorithm can also consider mixed traffic stream scenarios with different levels of CAV market penetration. Numerical experiments are conducted, and the results prove the efficiency and sound performance of the proposed optimization framework. The proposed DP-SH algorithm, compared to the adaptive signal control, can reduce the average travel time by up to 35.72% and save the consumption by up to 31.5%. In mixed traffic scenarios, system performance improves with increasing market penetration rates. Even with low levels of penetration, there are significant benefits in fuel consumption savings. The computational efficiency, as evidenced in the case studies, indicates the applicability of DP-SH for real-time implementation.

Key Words

Connected and automated vehicle (CAV); dynamic programming (DP); shooting heuristic (SH); intersection control; signal timing and phasing; mobility and sustainability

1. Introduction

As science and technology advance, new solutions to the existing problems of the transportation system emerge, and automation is once again at the forefront of these advances. Connected and automated vehicle (CAV) technology, which includes connected vehicles (CVs) and automated vehicles (AVs), is one of the most promising solutions. CVs can communicate with each other (i.e., vehicle to vehicle [V2V]) and with infrastructure (i.e., vehicle to infrastructure [V2I]) to exchange critical real-time traffic safety and operations information (e.g., status data of CAVs [location, speed, acceleration, etc.], intersection geometry, traffic signal status, etc.). More importantly, these data are accurate, and their resolution is high (less than 1 second). These features of CV data provide a solid foundation for transportation agencies to improve transportation systems. One of the objectives of the U.S. Department of Transportation's Connected Vehicle Research Program is to develop innovative applications that take advantage of communications between V2I and V2V to improve safety, mobility, and environmental performances of multi-modal transportation systems. On the other hand, accomplishments in individual vehicle control (automated/autonomous vehicle technology) have laid the foundation for more advanced control that governs interactions among multiple CVs and can produce resultant effects on highway traffic performance. A higher level of controllability of each vehicle enables better traffic system control that aims to improve stability, traffic throughput, and energy and environmental impacts. Many applications using CAV technologies have been proposed in recent years, and this paper focuses on the integration of automated vehicle control with infrastructure-based control (e.g., traffic signals) to improve overall system performance.

1.1 Real-Time Intersection Control Based on CAV Technologies

To reduce conflicts between vehicles traveling in different directions, signalized arterials need to alternate between green and red lights. This causes traffic to frequently decelerate and accelerate along the signalized arterials. However, vehicles engaged in repeated stop-and-go movements are exposed to higher crash risks (Sen and Head, 1997), extra driver discomfort (Beard and Griffin, 2013), and excessive fuel consumption and emissions (Li et al., 2014). As a result, traffic signal control improvement remains a hot topic in the field of transportation. The ideal traffic signal control is to optimally allocate green time to serve traffic from different approaches to achieve the best system performance (e.g., minimum delay and maximum throughput). To this end, real-time, accurate, and high-resolution traffic data are the key. With the advent of CVs, especially the V2I technology, these data can be obtained. CV data enable a traffic signal application to acquire a much more complete picture of a signalized intersection (i.e., states of both the nearby vehicle and traffic infrastructure). As a result, traffic signal control models/algorithms based on CAV data can generate better results than those based on traditional traffic detectors' data.

Some studies in recent years have examined the benefits of using CV data for traffic signal control. These CV-based models could outperform start-of-the-practice traffic control modes and/or traffic signal optimization software. He et al. (2012) proposed a platoon-based arterial signal control algorithm in which Pseudo-platoons are identified with a headway-based platoon recognition

algorithm. A mixed-integer linear program is applied to optimize future signal plans, and this algorithm is adaptive to two different modes. Lee et al. (2013) developed a cumulative travel-time (CTR) responsive real-time intersection control algorithm, for which each vehicle's travel time have been measured or estimated. Both total delay and average speed can be improved by 34% and 36% respectively under the 100% CV market penetration. It also has been found that a minimum CV market penetration of 30% is essential to gain the benefits. Priemer and Friedrich developed a decentralized adaptive traffic signal control algorithm with V2I data (Priemer and Friedrich, 2009). According to simulation results, the presented algorithm could reduce average delay by 24% and increase average speed by 5%, compared with the fixed-time signal timing plan optimized by TRANSYT-7F. Feng et al. examined an approach to integrating real-time adaptive signal control in a connected vehicle environment assuming varied market penetration rates (Feng et al., 2015). This algorithm is based on the controlled optimization of the phases algorithm introduced by Sen and Head (1997). The authors found that, when minimizing total vehicle delay with a 100% market penetration rate under two demand levels, total delay decreased by 10.04% and 14.67%, respectively. Also, when minimizing queue length with a 100% market penetration rate, total delay decreased by 6.37% and 16.33%, respectively. Huang presented an arterial-level traffic progression optimization model (ALTPOM) for under/near-saturated conditions (Huang, 2016). According to the simulation study, under penetration rates of 25% and 50%, ALTPOM minimally reduced 26.0% control delay and increased 4.4% throughput for both directions of major and minor streets compared to signal timing plans optimized by TRANSYT-7F.

The above algorithms usually require at least a 10 percent CV penetration rate to estimate the status of signalized intersections. The algorithms' performance would improve as penetration rates of CV increase; however, as indicated by Goodall (2013), the connected vehicle penetration rate is anticipated to increase steadily to a near 100 percent rate in approximately 25 years in the United States. Therefore, estimation locations and/or trajectories of non-CVs are vital for CV-based traffic signal control applications for current conditions with low penetrations rates. Locations and/or trajectories of non-CAVs are usually estimated by the status of CVs (Feng et al., 2015; Goodall, 2013) or fusion data of traditional detectors and CV data (Huang, 2016).

1.2 Automated Vehicle Trajectory Control

Researchers are also investigating strategies to enhance transportation system performance by fully controlling vehicle trajectory according to real-time traffic conditions. Some CAV-based models/algorithms were developed to precisely control individual vehicle trajectories (e.g., Ahn, 2013; Wang et al., 2014; Wang et al., 2014). With precise trajectory control, vehicles can either adjust their driving according to existing intersection controller information (e.g., traffic signal timing plans) to smoothly pass the intersection during green phases. The majority of existing studies address individual trajectory control instead of the coordination of a stream of vehicles that interact with one another. Most control methods developed to date either seek algorithm efficiency by ignoring detailed acceleration tuning (e.g., allowing speed jumps) or rely on complex algorithms that may impede real-time applications. Seredynski and Khadraoui (2014) and Seredynski et al. (2015) proposed that TSP

could be complemented with in-vehicle systems such as Green Light Optimal Speed Advisory (GLOSA) and Green Light Optimal Dwell Time Advisory (GLODTA). They allow a bus to pass through signalized intersections during a green phase without modifying signal timings. Compared to the problems of controlling regular traffic, TSP focuses on controlling a small number of buses or special-use vehicles. Jiang et al. (2017) studied the CAV trajectory control problem under scenarios of less than 100% market penetration. While the simulation results show improvements across different penetration rates, the algorithm does not explicitly consider complex interactions between human-driven and CAV trajectories in the trajectory design stage. Zhou et al. (2017) and Ma et al. (2017) proposed a parsimonious shooting heuristic (SH) algorithm that can effectively smooth the trajectories of a stream of vehicles approaching a signalized intersection by controlling detailed acceleration profiles. The SH algorithm represents each infinite-dimensional vehicle trajectory with a few segments of analytical quadratic curves. Therefore, it efficiently constructs a large number of vehicle trajectories subject to physical limits, car-following safety, and traffic signal timing. Instead of using fixed parameters for trajectory construction, they also proposed to embed SH into an efficient optimization framework that aims to identify the optimum vehicle trajectories on a signalized highway segment to minimize multiple traffic performance measures (i.e., travel time, fuel consumption, and safety) simultaneously. These two studies are limited to fixed signal timing and phasing and only control vehicle trajectories.

Some studies coordinate different vehicle trajectories to cross an intersection without an explicit traffic light (visualize a school of fish) (Dresner and Stone, 2008; Lee and Park, 2012; Chai et al., 2018). These studies usually use reservation-type systems that pre-assign intersection spaces to vehicles. Apparently, these studies will require a 100% CAV market penetration, and this concept is out of the scope of this study. We still consider the necessity of traffic signals due to the existence of human-vehicles in the next decades.

To the best of our knowledge, there is only a limited number of studies on the combined optimization of traffic signal and vehicle trajectories. To address these research gaps, Li et al. (2014) is a pioneer study investigating joint trajectory and signal control algorithms. The trajectory control algorithm is based on simple vehicle kinematics and constructed with fixed acceleration and decelerations parameters. The signal timing optimization is based on pure enumeration. This works in simple case studies but is expected to be non-tractable when considering more complex scenarios. Pourmehrab et al. (2017) continued the study and used relatively simple rules (such as green extension) for signal control and did not consider the complex interactions between vehicle trajectory optimization and signal timing control. Therefore, the results may not be the optimal conditions for the given system. Guler et al. (2014) and Yang et al. (2016) analyzed the value of platooning, autonomous vehicle control and connected vehicle information for improving intersection performance through a large number of simulations (embedded with a bi-level automated trajectory planning model) on a simple two-approach intersection. Li and Zhou (2017) considered the interaction, but the study is for network planning, and the designed trajectories are not detailed trajectories at the subsecond level for actual implementation of vehicle control. Feng et al. (2018) proposed a spatiotemporal traffic control framework to optimize traffic signal and vehicle trajectories. Dynamic programming and non-linear programming are applied

in this framework for minimizing total delay and fuel consumptions and emissions under 100% CAV penetration rates. Yu et al. (2018) further proposed a mixed integer linear programming (MIIP) model to optimize traffic signal and vehicle trajectories under the 100% CAV scenario. All vehicle movements such as left-turn, right-turn and through are considered. Phase sequences, green start, duration of each phase, and cycle lengths are optimized together with vehicle lane-changing behaviors and vehicle arrival times for delay minimization. While these two studies present the latest developments of advanced traffic signal and trajectory control, there are still a few aspects to be improved, such as studying scenarios less than 100% and optimizing a large number of trajectories simultaneously while considering the complex interactions between human-driven vehicles and CAVs.

1.3 Contributions and Highlights

Overall, this paper aims to address the gaps mentioned above and make contributions to the literature by proposing an efficient DP-SH (dynamic programming with shooting heuristic as a subroutine) algorithm for the integrated vehicle trajectory and intersection control (ITIC) problem. The algorithm can simultaneously optimize CAV trajectories in a mixed traffic stream and an intersection controller (note that the complex interactions between the intersection control, CAV trajectory design, and human-driven vehicle behavior are explicitly considered in the proposed algorithm). We developed a two-step approach (DP-SH and trajectory optimization) to effectively obtain near-optimal intersection and trajectory control plans based on newly developed and enhanced previously developed algorithms (i.e., shooting heuristic for CAV trajectory design, human-driven vehicle trajectory construction, and a customized numerical gradient method for optimization over a composite performance measure).

One highlight of the paper is the numerical experiments for scenarios of different traffic conditions and CAV market penetration rates using a realistic real-world 4-phase signal timing design to verify the performance of this proposed optimization framework and draw managerial insights. Also, in the case studies, we also investigated the computational efficiency of the algorithm by varying the number of vehicles, segment length, traffic saturation rate, and planning horizon, proving the algorithm's potential for online real-world applications.

Note that in this study we consider a realistic 4-phase signal design as introduced later. The nature of the proposed dynamic programming-based signal control makes it easy to extend the intersection control algorithm to any number of phases in the future. Additionally, when the CAV market penetration is 100%, and all vehicles can be automatically controlled, the algorithm proposed in this study is by nature the vehicle entry control to avoid potential conflicts – allocating the right-of-way at the intersection. It does not require physical traffic signals (red lights) to enforce vehicles to stop under full market penetration of CAVs. The proposed algorithms can also be applied to any intersection control: signaless intersection (similar to the concept in Dresner and Stone [2008]) and autonomous merging area coordination. In this paper, however, we use intersection control with physical signals to illustrate the proposed DP-SH algorithm. More importantly, this paper also considers the scenario of mixed traffic conditions with both CAVs and human-driven vehicles. These scenarios are particularly useful in the coming decades when the vehicle and highway system goes through the transition to full

automation. Traffic signals are necessary under these scenarios to ensure safety.

This paper is organized as follows. Section 2 states the studied CAV ITIC problem, including problem settings, constraints, and system objectives. Section 3 briefly reviews the SH algorithm developed in the author's previous work (Zhou et al., 2017; Ma et al., 2017) and proposes an ITIC Dynamic programming framework and the corresponding optimization method. Efficient methods of evaluating the system objectives are also discussed. Section 4 demonstrates the proposed optimization framework and tests its solution efficiency and related properties with numerical examples. Section 5 concludes this paper and proposes future research directions.

2. Problem Statement

This section states the integrated vehicle trajectory and intersection control (ITIC) problem under investigation in this paper. **Fig. 1** illustrates the problem with a simplified intersection with two highway segments of length L. The set of locations of this segment is $[0, L]$. There is a traffic signal at location L of both segments. This paper aims to solve the joint optimal design of signal timing and vehicle trajectory planning considering conflicts of different movements. Note that the paper aims to address general intersection configurations and the example in **Fig. 1** is for the purpose of problem illustration.

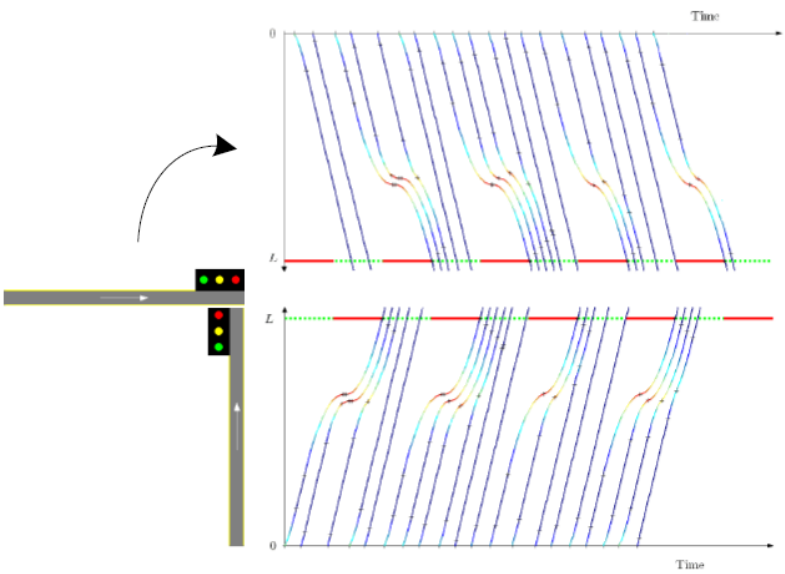


Fig. 1. Illustration of ITIC problem

2.1 Trajectory Control Problem

A set of consecutive vehicles $N = \{1,2,\dots,n\}$ travel from location 0 to L. The dynamics of these vehicles follow a number of constraints specified with the notion of trajectories. A vehicle trajectory is denoted with a second-order semi-differentiable function $p(t)$, $\forall t \in (-\infty, \infty)$, such that its first order differential (or velocity) $\dot{p}(t)$ is absolutely continuous, and its second-order right-differential (or acceleration) $\ddot{p}(t)$ is Riemann integrable. For notation convenience, we denote the whole

trajectory of function $p(t)$ with p, and the section of p over time interval $[t^-, t^+]$ by trajectory section $p(t^-, t^+), \forall -\infty \leq t^- \leq t^+ \leq \infty$. Let p_n denote the trajectory for vehicle n, $\forall n \in N$, and we call $P := [p_n]$, $n \in N$ a *trajectory vector* that contains all dynamic information of these vehicles. Trajectory vector P should satisfy the following constraints to be feasible.

Let v_n^- and t_n^- denote the speed and time when vehicle n enters location 0, $\forall n \in N$, and we consider pair (v_n^-, t_n^-) the entry boundary condition of vehicle n. We assume that all vehicles' boundary conditions can be exactly predicted (e.g., with the advanced sensing and tracking technology in the future transportation infrastructure). Then, each trajectory p_n should satisfy the *entry boundary constraint* $p_n(t_n^-) = 0$ and $\dot{p}_n(t_n^-) = v_n^-$. We assume all vehicles are identical, and each vehicle's acceleration is limited in $[\underline{a} < 0, \bar{a} > 0]$, and its speed range is $[0, \bar{v}]$. We say a trajectory section p is kinetically feasible if the following *kinematic constraint* is met.

$$0 \leq \dot{p}(t) \leq \bar{v}, \underline{a} \leq \ddot{p}(t) \leq \bar{a}, \forall t \in (-\infty, \infty) \tag{1}$$

A vehicle $n \in N \backslash \{1\}$ should always be at least a minimum space separation s (which is usually the summation of the length of a vehicle and a safety buffer) behind its preceding vehicle $(n-1)$'s location a reaction time τ (or a communication delay) ago. We call the trajectory obtained by translating any p rightwards by τ and downwards by s the shadow trajectory of p. This shadow trajectory is denoted by adding a superscript 's', i.e., p^s, denoting the shadow trajectory of p. Then, this *safety constraint* can be stated as

$$p_n(t) \leq p_{n-1}^s(t), \forall t \in (-\infty, \infty), \forall n \in N \tag{2}$$

We let r_m denote the start of next green time and define $G(t)$ as the function that finds the next r_m. $G(t) = t$ indicates that time t is in a green phase, or otherwise $G(t) > t$. Let $p^{-1}(l)$ denote the first time when trajectory $p(t)$ arrives at location l. Then, $p^{-1}(l)$ denotes the time when this vehicle exits the highway segment location l (i.e., passing the stop bar), and it has to be during a green time. We call this the *green exit time constraint*.

$$\mathrm{G}\big(p^{-1}(L)\big) = p^{-1}(L) \tag{3}$$

Any trajectory p_n, $\forall n \in N$ needs to satisfy constraints (1) – (3) simultaneously to be a feasible trajectory.

2.2 ITIC Problem

Unlike single approach trajectory control problems (e.g., Ma et al., 2017), at a signalized intersection, a traffic signal needs to assign the right-of-way to vehicles from different approaches such that the total delay, the number of stops or other performance measurements can be minimized. We consider an intersection and a set of possible vehicle movements as shown in **Fig. 2**. Combinations of non-

conflicting movements that allow safe passage through the intersection are called phases PH. The cardinality of this set is denoted as $|PH|$ and individual phase is denoted by $\emptyset$. In this study, $|PH| = 4$ and $PH = \{\emptyset_1, \emptyset_2, \emptyset_3, \emptyset_4\}$. Other phases are certainly possible including combinations of movements 1 and 2, 3 and 4, 5 and 6, and 7 and 8. We use γ to indicate the minimum green time and r to represent effective clearance time. It is assumed that $r \leq \gamma$.

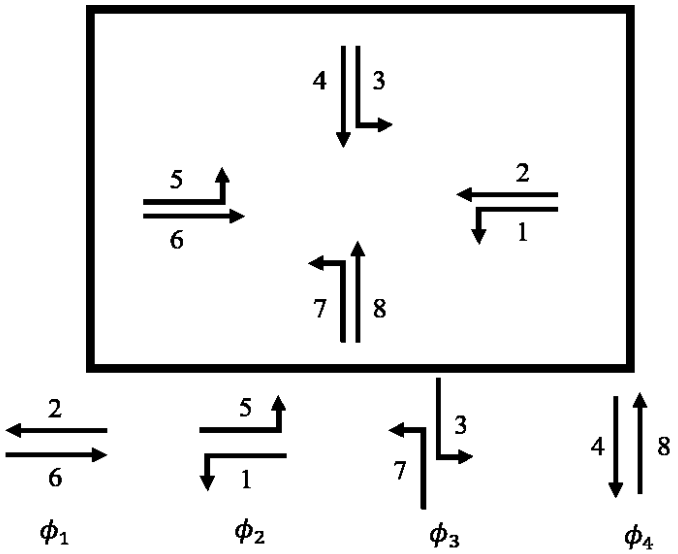


Fig. 2. Illustration of the 4-phase signal timing design considered in the paper

Performance measures are identified for the ITIC problem. The first measure is total intersection delay. In this study, vehicle n's delay is considered as the extra time vehicle n needs to traverse the segment as compared with the expected time traveling at entry speed without stopping at the red light $(t_n^- + \frac{L}{v_n^-})$. When trajectory vector $P = [p_n]_{n \in N}$ is given, the corresponding average vehicle delay can be formulated as:

$$D(P) = \frac{1}{N} \sum_{n \in N} (P_n^{-1}(L) - (t_n^- + \frac{L}{v_n^-})) \tag{4}$$

Note that $D(P)$ is usually a positive value for regular traffic because, after entering the segment, manually driven vehicles usually keep their current speed and are delayed when they have to stop at the red light or at the end of a queue. In this study, however, we will use the shooting heuristic algorithm (Ma et al., 2017) to plan CAV trajectories, and these vehicles can accelerate to the speed limit and aim at no stops at the intersection. Therefore, it is likely that $D(P)$ can take negative values, indicating generally early departure from the intersection. Alternatively, the speed limit can be used as the desired speeds to replace v_n^- in Equation (4), and in that case, the performance $D(P)$ will only take non-negative values.

Also, note that here we overload notation N to indicate the total number of vehicles on all movements that are assigned to a phase $\emptyset$. All the following performance measures should be calculated considering all vehicles controlled by a signal phase $\emptyset$ which may concern multiple movements as

illustrated in **Fig. 2**.

Similar to intersection delay, we can also use the total travel time of all vehicles to complete traversing the segments. When trajectory vector $P = [p_n]_{n \in N}$ is given, the corresponding average travel time (per vehicle) can be simply formulated as:

$$T(P) = \frac{1}{N} \sum_{n \in N} (t(p_n) \coloneqq p_n^{-1}(L) - t_n^{-1}) \tag{5}$$

where $t(p_n)$ is the travel time for vehicle n.

In many cases, reducing fuel consumption and pollutant emissions are also important goals for intersection control, particularly when trajectory control is possible. To measure consumption, the VT-Micro model (Ahn et al., 2013) was used, incorporating the speed profiles of the probe vehicles as shown in Equation (6).

$$e(\dot{p}_n(t), \ddot{p}_n(t)) = exp\left\{\sum_{i=0}^{3} \sum_{j=0}^{3} K_{ij}(\ddot{p}_n(t)) \left(\lfloor \dot{p}_n(t) \rfloor_0^{120\ km/h}\right)^i \left(\lfloor \ddot{p}_n(t) \rfloor_{-5\ km/h/sec}^{120\ km/h/sec}\right)^j\right\} \tag{6}$$

Where coefficient $K_{ij}(\ddot{p}_n(t))$ depends on the sign of $\ddot{p}_n(t)$, the type of vehicle, and the measure-of-effectiveness (MOE) (e.g., fuel consumption). Model parameters used in the calculation are listed in Ma et al. (2017).

Other performance measurements, such as safety as measured by surrogate safety measures (e.g., inverse time-to-collision) can also be applied. Ma et al. (2017) provides a detailed discussion of each of these measurements and how they can be effectively evaluated for real-time application without the need to evaluate fuel consumption and safety measures at discrete time points (e.g., 0.1 seconds). Due to space constraints, this paper will not discuss the expedited performance evaluation. Interested readers can refer to this work for details.

3. Methodology

This section introduces the main components of the ITIC methodology proposed in this paper. We first revisit the shooting heuristic (SH) previously proposed by the authors, and this algorithm is used to efficiently construct trajectories for a stream of traffic. Next, we introduce the core contribution of this paper, the DP-SH algorithm, which uses dynamic programming (DP) to obtain an optimal signal timing plan while considering vehicle movements as constructed by the SH algorithm. The last step is to optimize CAV trajectories given the optimal signal timing plan, and this optimization can be based on various performance measures (e.g., travel time, delay and/or fuel consumption).

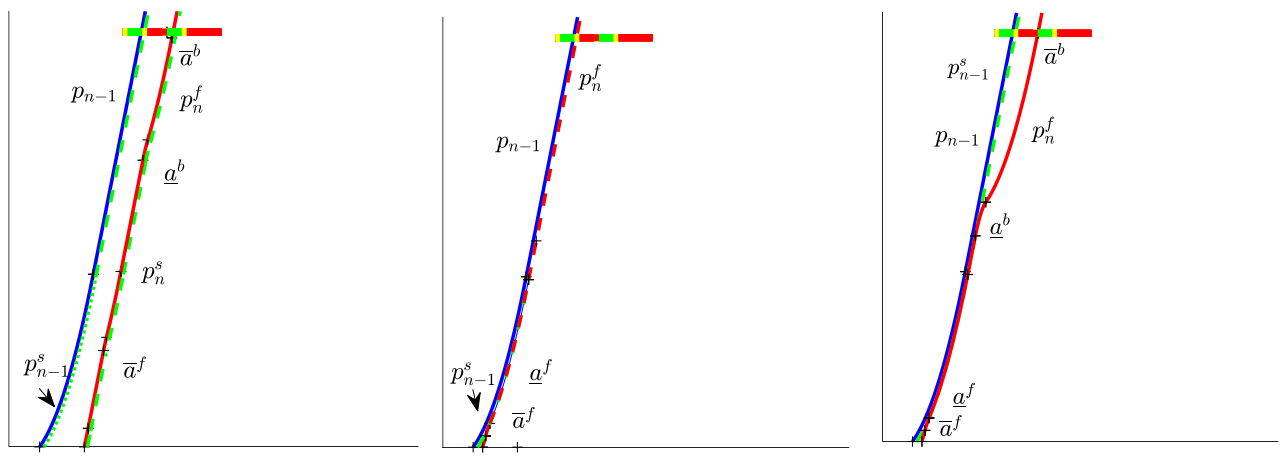
The proposed ITIC solution method is a two-step approach. Step 1 is to obtain an optimal signal timing plan that minimizes intersection delay (with consideration of the interactive relations between the designed trajectories and the signal timing plan). Step 2 is to design optimal trajectories for the signal

timing plan developed in Step 1 for various performance measures.

3.1 Revisit of Shooting Heuristic

This subsection briefly reviews the SH algorithm from Zhou et al. (2017), and interested readers can refer to it for detailed algorithmic descriptions and theoretical property analyses. The SH algorithm visits all N vehicles steering in the segment sequentially, and entry boundary condition (v_n^-, t_n^-) of each vehicle at location 0 is recorded. Based on the entry boundary condition and given parameter set $(\bar{a}^f, \underline{a}^f, \bar{a}^b, \underline{a}^b, v)$, a feasible trajectory can be designed using the SH algorithm for each vehicle as a piecewise quadratic function. Each two consecutive quadratic segments in the same trajectory are tangent to each other. Each trajectory is differentiable everywhere and does not contain any speed jumps. **Fig. 3**(a) and (b) show the forward shooting process (FSP), under two conditions, respectively. For each vehicle n, FSP shoots two sequential quadratic segments at first, as shown in **Fig. 3** (a). The first segment radiates from location 0 with entry boundary condition (v_n^-, t_n^-), and then reaches a *target cruising speed* $v \in [0, \bar{v}]$, where $\bar{v}$ is the maximum allowable cruising speed, with a forward acceleration $\bar{a}^f$. Note that in this paper, without the loss of generality, we assume $v = \bar{v}$ for through movements and $v = \vartheta\bar{v}$ for turning movements, where $\vartheta < 1$. It takes $\hat{t}_n^- = (v - v_n^-)/\bar{a}^f$ seconds to reach target speed at location $\hat{p}_n^- = v_n^-\hat{t}_n^- + \frac{1}{2}\bar{a}^f\hat{t}_n^-$. The second segment is simply cruise remaining with constant speed v until the vehicle exits location L at:

$$\hat{t}_n^+(v, \bar{a}^f) := t_n^- + \begin{cases} \frac{-v_n^- - \sqrt{(v_n^-)^2 + 2\bar{a}^f L}}{\bar{a}^f}, & \text{if } L \le \frac{v^2 - (v_n^-)^2}{2\bar{a}^f} \\ \frac{L}{v} + \frac{(v - v_n^-)^2}{2\bar{a}^f v}, & otherwise \end{cases} \tag{7}$$



(a) (b) (c)

Fig. 3. (a) Forward shooting process without activating safety constraint; (b) forward shooting process with activating safety constraint; and (c) backward shooting process

The safety constraint is tested against this candidate trajectory. If the safety constraint is not violated, i.e., the nth vehicle is always below the shadow trajectory of the preceding vehicle, this candidate

trajectory will be returned as the *forward shooting trajectory* of vehicle n denoted by p_n^f. Under this scenario, the candidate trajectory p_n^f is always below the shadow trajectory p_{n-1}^s of the preceding vehicle, which is shown using the red solid curve under the green dash curve in **Fig. 3**(a). Otherwise, if the candidate trajectory violates the safety constraint, as shown in **Fig. 3**(b), the candidate trajectory should smoothly merge into the previous shadow trajectory p_{n-1}^s at a forward deceleration $\underline{a}^f$ .This process will create a merging segment, and these three segments merge together and become a smooth curve as the forward shooting trajectory p_n^f. When the forward shooting trajectory p_n^f is constructed, the green time exit constraint will be tested against p_n^f. The SH algorithm compares the actual exit time of trajectory p_n^f and its corresponding time phase calculated using Equation (3). If the vehicle n exits during the green time, there is no need to revise p_n^f and it will be regarded as a feasible trajectory of vehicle n, noted as p_n. Otherwise, if the trajectory p_n^f runs into a red light, a backward shooting process (BSP) will be activated to revise p_n^f. This segment shifts rightwards horizontally to the start of the next green phase as the initial segment of a *backward shooting trajectory* p_n^b. Then, BSP shoots backwards along this initial part at a *backward acceleration* $\bar{a}^b$. When p_n^b gets close enough to p_n^f, BSP can shoot backwards a smooth merging segment at a *backward deceleration* $\underline{a}^b$. A merging segment will be constructed from p_n^b to p_n^f and then becomes tangent to p_n^f, as shown in **Fig. 3**(c). As a result, merging p_n^f and p_n^b generates feasible trajectory p_n. This loop of FSP and BSP is consecutively executed by the SH algorithm for all N vehicles, and a feasible trajectory vector $[p_n]_{n\in N}$ will be generated.

Note that the entire process of the SH algorithm deals with no more than $7n, \forall n \in N$ quadratic segments (Ma et al., 2017) in the construction of a single trajectory, and every segment is analytically solvable. This indicates the efficiency of the algorithm and makes it possible for large-scale practical applications. Further, only five variables, including four acceleration rates $\{\bar{a}^f, \underline{a}^f, \bar{a}^b, \underline{a}^b\}$ and the target speed v, need to be determined to control the overall smoothness and corresponding performance. Because the algorithm is efficient, parsimonious, and simple to implement, it is then possible that the proposed DP-SH optimization framework can use SH as a sub-routine. The steps of the SH algorithm are shown as the following pseudo code. The computational complexity of the Shooting Heuristic algorithm is $o(N^2)$, and the computing space complexity is also $o(N^2)$ (Ma et al., 2017).

Algorithm 1: Shooting Heuristic

Define segment length L, safety buffer s, reaction time τ as a global variable.
Initialize acceleration rates $(\bar{a}^f, \underline{a}^f, \bar{a}^b, \underline{a}^b)$ and target cruise speed v.
$N \leftarrow number\ of\ vehicle$.
$\left[v_N^-; t_N^-\right] \leftarrow entry\ boundry\ condition$.
$P \leftarrow \varnothing$ % Set trajectory vector
For $n = 1\ to\ N$
    If $\bar{v}_n < v$
        $\bar{v}_n^{next} \leftarrow v$

$t_n^{next} \leftarrow \hat{t}_n^-$

$p_n^{f(1)} \leftarrow PlotSegment\left(\bar{a}^f, \bar{v}_n^{next}, t_n^{next}\right)$ % Plot first segment of FSP

$t_n^{next} \leftarrow \hat{t}_n^+\left(v, \bar{a}^f\right)$

$p_n^{f(2)} \leftarrow PlotSegment\left(0, v, t_n^{next}\right)$ % Plot second segment of FSP

$p_n^f \leftarrow merge\left(p_n^{f(1)}, p_n^{f(2)}\right)$

Else

$p_n^f \leftarrow PlotSegment\left(0, v, \hat{t}_n^+\right)$

End

If $p_n^f\left(t_n^- : \hat{t}_n^+\right)$ *crosses or overlap* $p_{n-1}^s\left(t_n^- : \hat{t}_n^+\right)$ % If p_n^f violates safety constraint

$\bar{v}_{diff} \leftarrow Diff\left(v_{n-1}^s, \bar{v}_n\right)$ % Calculate the velocity gap between p_{n-1}^s and candidate p_n^f

$\hat{t}_{diff} \leftarrow MergeTime\left(t_n^-, \hat{t}_n^+\right)$ % Calculate the time range of merging

$p_n^{f(merge)} \leftarrow PlotSegment\left(\underline{a}^f, \bar{v}_{diff}, \hat{t}_{diff}\right)$ % Plot forward merging segment

$p_n^f \leftarrow merge\left(p_n^f, p_n^{f(merge)}\right)$

End

% Entire FSP has been executed and a forward shooting trajectory p_n^f has been constructed %

If $G\left(p_n^{-1}(L)\right) > p_n^{-1}(L)$

$p_n^b \leftarrow ShiftRightwards\left(p_n^f\left(\hat{t}_n^+ : \infty\right), G\left(p_n^{-1}(L)\right)\right)$

$d^b \leftarrow FindDistance\left(p_n^f, p_n^b\right)$

While $d^b > merge\ range$

If $\bar{v}_n < v$

$p_n^b \leftarrow PlotSegment\left(\bar{a}^b, \bar{v}_n, \hat{t}_n\right)$

Else

$p_n^b \leftarrow PlotSegment\left(0, v, \hat{t}_n\right)$

End

$d^b \leftarrow FindDistance\left(p_n^f, p_n^b\right)$

End

$p_n^{b(merge)} \leftarrow PlotSegment\left(\underline{a}^b, \bar{v}_{diff}, \hat{t}_{diff}\right)$ % Plot backward <u>merging</u> segment

$p_n^b \leftarrow merge\left(p_n^b, p_n^{b(merge)}\right)$

$p_n \leftarrow merge\left(p_n^f, p_n^b\right)$

Else

$p_n \leftarrow p_n^f$

End

$P(n) \leftarrow p_n$

End

3.2 Dynamic programming framework with SH as a subroutine

With a given signal timing plan or predictable timing plan, the SH algorithm can construct $\mathrm{P} = [p_n]_{n \in N}$ with parameter set $\{\bar{a}^f, \underline{a}^f, \bar{a}^b, \underline{a}^b, v\}$. Getting the signal timing plan, however, is a difficult problem itself because it is based on predicted vehicle movements in manual traffic or designed trajectories in automated traffic. Different signal plans may result in quite distinct automated vehicle trajectories because the signal timing optimization and trajectory optimization are two intertwined problems and need to be considered simultaneously. In this study, we use dynamic programming (DP) to optimize signal timing and phasing with the SH trajectory construction algorithm as a subroutine (DP-SH). The SH algorithm is extremely efficient as proved in Ma et al. (2017), and therefore the DP-SH algorithm can still be applied in real time.

The DP algorithm is based on breaking a decision problem into manageable decision stages and computing an appropriate performance measure in a recursive manner. The recursion may be performed in a forward fashion, starting from the initial decision stage; or in a backward fashion, starting from the final decision stage. Similar to Sen and Head (1997), the DP-SH algorithm in this study uses a forward recursion. In this setting, a value function for any stage represents an accumulated measure of effectiveness for the current and all previous stages. In specifying a DP formulation, we must define the appropriate state variables, decision stages, and the recursion necessary to compute the value functions.

In this study, the recursion is based on using phases as stages. Starting with an initial phase, DP-SH treats each phase in a cycle as a stage and optimizes over a planning horizon to obtain an optimal plan. As introduced in Sen and Head (1997), DP-SH allows us to skip phases whenever it is advantageous to do so, and therefore has more flexibility for the optimization problem.

We use similar notations as in Sen and Head (1997) when necessary for comparison purposes and use new notations to cover additional elements for our new problem. We use j as the index for stages of the DP-SH. Let x_j be the control variable denoting the amount of green-time allocated to stage j; and s_j be the state variable denoting the total number of time-steps that have been allocated after the completion of stage j. Use $X_j(s_j)$ to denote the set of feasible control decisions, given state s_j; $f_j(s_j, x_j)$ to denote the performance measure at stage j, given state s_j and control x_j; and $v_j(s_j)$ to denote the value function (cumulative value of prior performance measures) given state s_j. Therefore, given the state variable s_j, the control variable x_j can assume the following values in set $X_j(s_j)$:

$$X_j(s_j) = \begin{cases} \{0\}, & s_j - r < \gamma \\ \{0, \gamma, \gamma + 1, \dots, s_j - r\}, & otherwise \end{cases} \tag{8}$$

The use of γ as the smallest non-zero value that can be assigned to the control variable x_j ensures the satisfaction of the minimum green-time requirement. We use the following relationship between two consecutive stages of the DP-SH due to the clearance time r:

$$s_{j-1} = s_j - h_j(x_j) \tag{9}$$

Where $h_j(x_j) = \begin{cases} 0, & if\ x_j = 0 \\ x_j + r, & otherwise \end{cases}$

In this study, we consider the average travel time as the performance function which is calculated using Equation (10). This establishes mobility as the main objective when designing signal timing and phasing. Other performance measures, such as fuel consumption and safety, can also be applied with increased levels of problem complexity. In order to ensure the algorithmic efficiency for real-time implementation, we leave fuel consumption and safety during the final optimization of vehicle trajectories when the signal control has been determined.

$$f_j(s_j, x_j) = \frac{1}{\sum_{PH=1}^{|PH|} N_{PH}} \left(\sum_{PH=1}^{|PH|} T(P)_{PH} \right) \tag{10}$$

where $T(P)_{PH}$ is calculated by Equation (5), representing the average travel time for each intersection movements with constructed trajectory sets $\{P_{N1}, P_{N2}, P_{N3}, P_{N4}\}$, respectively, with control x_j at stage s_j.

We now present the steps of the DP-SH algorithm. There are two parts of the algorithm: forward recursion and backward solution retrieval.

The use of the forward recursion is also in compliance with the problem nature because the trajectories are built sequentially. Thus, the subset of the solution to a problem at a later stage is still the optimal solution to the earlier subset of the problem. This is reflected in the forward recursion as shown in Equation (11). Note that at each stage, only trajectories that can pass the intersection during this stage are built, and the corresponding f is calculated. This ensures that the earlier constructed solutions are still solutions to the trajectory construction at later stages. Actually, if the computer storage space is available, the trajectories built at each early stage (j, s_j) can be stored along with $v_j(s_j)$ such that future stages can continue the trajectory construction without rebuilding the trajectories that have been constructed when calculating $v_j(s_j)$.

$$v_{j+1}(s_{j+1}) = \sum_{x_{j+1}} \left(v_j(s_j) + f_{j+1}(s_{j+1}, x_{j+1}) \right) \tag{11}$$

The Algorithm 2 below shows the pseudo code for the proposed DP-SH framework. We consider a planning horizon of H and step size ε can be indexed by $\varepsilon \in [1, H]$, which reflects the granularity of our algorithm.

Algorithm 2: DP-SH Framework

Define time horizon H as ga lobal variable.

$v_0(0) \leftarrow 0$

$j \leftarrow 1$

$s_0 \leftarrow 0$

For $s_j = r \ step\ \varepsilon \ to \ \left(H - s_{j-1}\right)$ % Start forward recursion

While $x_j \in X_j\left(s_j\right)$

$P \leftarrow ShootingHeuristic\left(x_j\right)$

$$f_j(s_j, x_j) = \frac{1}{\sum_{PH=1}^{|PH|} N_{PH}} \left(\sum_{PH=1}^{|PH|} T(P)_{PH} \right)$$

Loop

$v_j\left(s_j\right) \leftarrow \min_{x_j}\left(f_j\left(s_j, x_j\right) + v_{j-1}\left(s_{j-1}\right)\right)$

$x_j^*\left(s_j\right) \leftarrow \hat{x}_j$ % $\hat{x}_j$ is the corresponding control variable to the optimal solution in this stage

If $s_j = H - s_{j-1}$

break

Else if $\dfrac{\left(v_{j-1}(H) - v_j(H)\right)}{\left|v_{j-1}(H)\right|} < \sigma$

break

Else if $j > \lfloor |H| / r \rfloor$

break

Else

$j = j + 1$

End

End

$J_{complete} \leftarrow j$ % record complete stage j as J

For $j = J \ step\ -1 \ to\ 1$ % Start retrieval

$s_{j-1}^* = s_j^* - h_j\left(x_j^*\left(s_j^*\right)\right)$.

End

The later stages allow more phase changes for the same values of the state variable, and the value function always improves, i.e., $v_{j-1}(H) \geq v_j(H)$. Therefore, the forward recursion will stop when the percent change from $v_{j-1}(H)$ to $v_j(H)$ is less than a threshold value σ. In this study, we use $\sigma = 0.05$. Then, the DP-SH framework retrieves an optimal solution for determining the optimal trajectory of states and the associated optimal controls.

The fundamental difference of this method from the adaptive signal control system (Sen and Head, 1997) is $P \leftarrow ShootingHeuristic(x_j)$. They make assumptions on a vehicle arrival table based on certain short-term traffic predictive technology. The accuracy of the arrival table significantly affects the actual system performance. This study, however, designs trajectory in each substep in forward recursion, and trajectory design is a part of the algorithm itself. The trajectory design using the SH algorithm considers kinematic constraints, thus it is likely that vehicles' actual trajectories can be close to designed trajectories, increasing the actual effectiveness of the algorithm. Also, it is worth mentioning that any small changes in signal timing may change vehicle trajectories, and such interaction and the iterative process is only possible because of the extremely efficient SH algorithm.

Note that in this step, SH parameters are predetermined. While the next section will discuss methods for identifying these parameters for trajectory optimization, reasonable initial values of these parameters can be used in order to obtain an optimal signal timing and phasing plan. In fact, previous work (Ma et al., 2017) found an interesting distribution of the optimal parameters. For example, the most frequent optimal values of $\bar{a}^f$ are around 1 m/s^2.

3.3 Trajectory Optimization

The previous section uses average vehicle delay (calculated by Equation 3) as the performance function when optimizing signal phasing and timing. However, depending on actual conditions (e.g., traffic congestion, segment length, market penetration, cycle length), the input parameters may need to be optimized from case to case, particularly from the perspective of fuel consumption and safety.

After an optimal signal timing solution is obtained through DP-SH, we will run a customized numerical gradient-based approach with SH as a subroutine to optimize the five parameters $(\bar{a}^f, \underline{a}^f, \bar{a}^b, \underline{a}^b, v)$ for best system performance in terms of travel time, fuel consumption, and safety (Ma et al., 2017).

Fig. 4 illustrates how the SH-based vehicle trajectory optimization framework works. Basically, we first initialize a small set of control parameters and feed them to the SH algorithm to generate a feasible trajectory vector P. Then, system performance $M(P)$ is evaluated (potentially using expedited methods [Ma et al., 2017] based on the analytical properties of the SH solution). The evaluation result is then fed to the central optimization algorithm. This algorithm checks whether P is already optimal. If yes, it returns P as the optimal solution and sends back the corresponding parameter set $(\bar{a}^f, \underline{a}^f, \bar{a}^b, \underline{a}^b, v)$. Otherwise, it updates the control parameter values based on a customized numerical gradient based method. This loop will start from different start parameter sets and each loop generates an optimal solution. The central optimization algorithm compares all of the solutions and then returns the final optimum and its control parameters. For left-turn vehicles, a penalty parameter is added to the optimization objective function if the desired cruise speed v exceeds a given threshold (i.e. turning speed limit v^L). This penalty parameter ensures the left-turn vehicles do not travel too fast when they traverse the intersection, while still considering overall optimality in terms of travel time and fuel consumption. The purpose of using multiple starting points of the optimization is to avoid local optimal results because the objective function is non-convex. Please see Ma et al. (2017) for a detailed description of this algorithm and how to evaluate different performance measures in an expedited manner online.

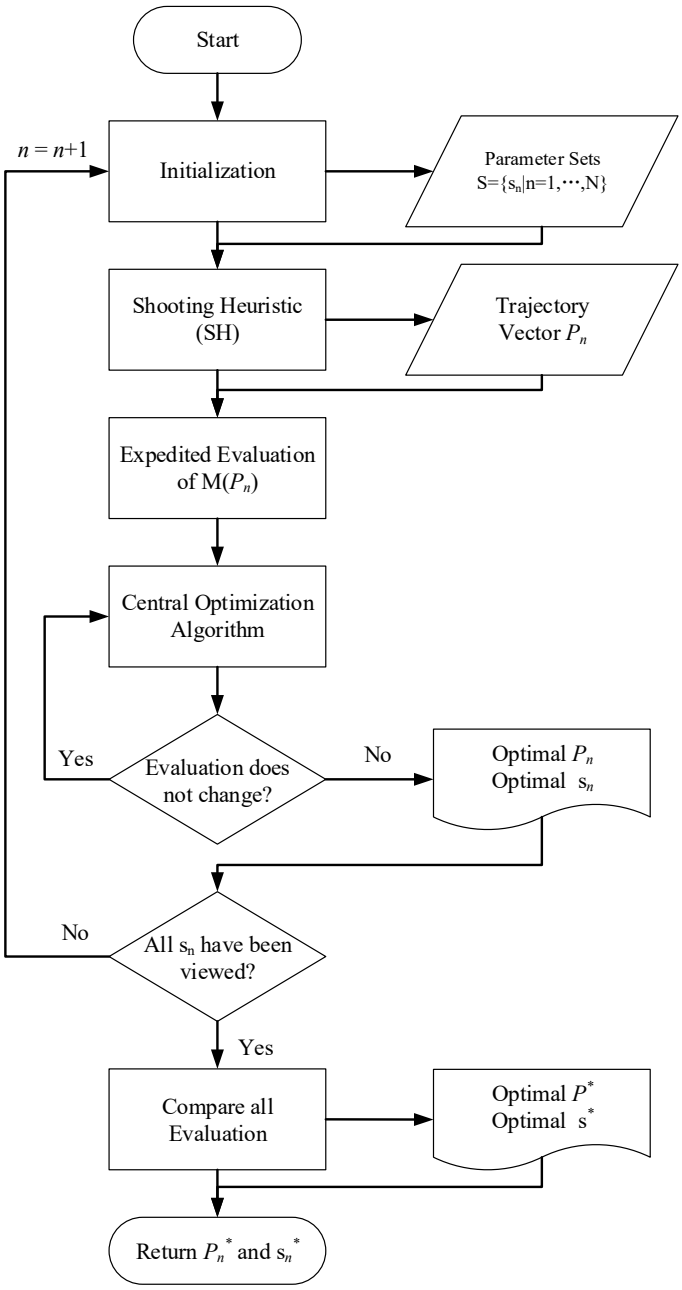


Fig. 4. SH-based vehicle trajectory optimization framework

3.4 Shooting Heuristic with Human-driven Vehicles

When CAVs and human-driven vehicles coexist in the traffic, the integrated trajectory and intersection control needs to take into account potential trajectories of human-driven vehicles and their impacts on and interactions with the signal control and CAV trajectories. In this study, we assume there are vehicle detectors upstream of the intersection at the beginning of the trajectory control section, so the information on the human-driven vehicle entry boundary information is known. If these human-driven vehicles are CVs, their information can be obtained at any time. Therefore, the human-driven vehicle trajectories need to be predicted efficiently based on the entry information. Traditional models such as Intelligent Driver Model simulate trajectories second by second, and the computational burden makes

it difficult to be implemented in real time with a large number of vehicles at the intersection. In this paper, we propose human-driven vehicle shooting heuristic (SH) to predict their trajectories under given entry boundary conditions.

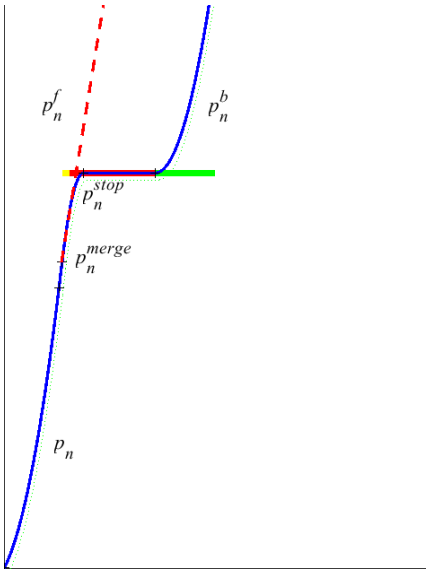


Fig. 5. Human-driven oriented backward shooting

As **Fig. 5** shows, the trajectory of a human-driven vehicle is also firstly constructed by the FSP. If the human-driven vehicle crosses the intersection during a green phase, its trajectory p_n^f will be kept as a feasible trajectory. Otherwise, if the existing trajectory p_n^f violates the green time exit constraints, as the red dash line shows in **Fig. 5**, a human-driven vehicle BSP (HD-BSP) will be activated to revise the existing infeasible trajectory p_n^f. The segment above the stop bar also shifts rightwards horizontally to the start of the next green phase and constructs the initial segment of the backward shooting trajectory p_n^b. But different from the regular BSP discussed in Section 3.1, the HD-BSP shoots horizontally from the start of the next green phase to a specific stop point p_n^{stop}, which is calculated by the HD-BSP along with the trajectory p_n^f at a forward deceleration $\underline{a}^f \in [\underline{a}, 0)$, and the original trajectory under the stop bar cannot be changed. Then HD-BSP shoots backwards from the stop point p_n^{stop} at a backward deceleration $-\underline{a}^f$ to merge into the p_n^f. As a result, a feasible trajectory p_n in **Fig. 5** can be generated by merging p_n^f and p_n^b, and the feasible trajectory p_n will be added into the feasible trajectory vector $[p_n]_{n\in N}$. **Fig. 6** shows the process of HD-BSP.

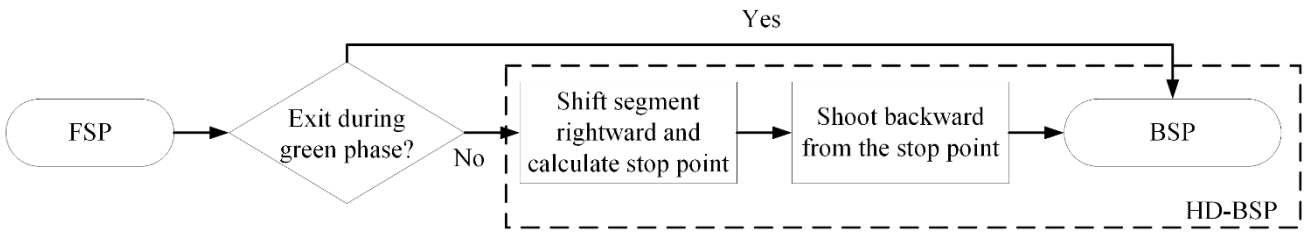


Fig. 6. Human-driven oriented backward shooting

If a CAV follows a human-driven vehicle, the CAV trajectory will be revised with an additional step in BSP, as shown in **Fig. 7**. The human-driven vehicle left the intersection at the point $p_{init}^{HD}(t_L, L)$. The point p_{max}^{HD} on the human-driven trajectory P_{HD} is the point that the human-driven vehicle reaches the target cruise speed v with the acceleration $\bar{a}^f$ after it crosses the intersection. The point p_{max}^{HD} can be depicted as a tuple $p_{max}^{HD}\left(t_L + \frac{v}{\bar{a}^{f\,a\,a}}, L + \frac{v^2}{2\bar{a}^f}\right)$. Then, the CAV can reach the target speed v at the point $p_{max}^{CAV}\left(t_L + \frac{v}{\bar{a}^f} + \tau, L + \frac{v^2}{2\bar{a}^f} - s\right)$ due to the car-following safety constraints. If the CAV follows the human-driven vehicle closely, it may end up following the human-driven vehicle's shadow trajectory, which may be less energy efficient and incur a higher level of safety risk (inverse time-to-collision). Instead, in our method, similar to regular SH, we still let the CAV to pass the intersection (stop bar) with the target cruise speed v, and then catch up and form a "platoon" with the human-driven vehicle with the constant speed v. The point p_{init}^{CAV} can be analytically solved as $p_{init}^{CAV}\left(t_L + \frac{v^2+2\bar{a}^f s}{2\bar{a}^f v} + \tau, L\right)$. Then the segment between points p_{max}^{CAV} and p_{init}^{CAV} will be used as the initial segment of the BSP. The wasted green time t_w, i.e., the distance between p_{init}^{HD} and p_{init}^{CAV}, can be calculated as $t_w = \frac{v^2+2\bar{a}^f s}{2\bar{a}^f v} + \tau$ and it is correlated with the cruise speed v and forward acceleration $\bar{a}^f$. In our DP-SH framework, the target cruise speed v and forward acceleration $\bar{a}^f$ are both optimization variables as mentioned. The proposed optimization process in Section 3.3 will eventually select the optimal values for v and $\bar{a}^f$ that minimize the system composite costs (i.e., travel time, fuel consumption and safety risk).

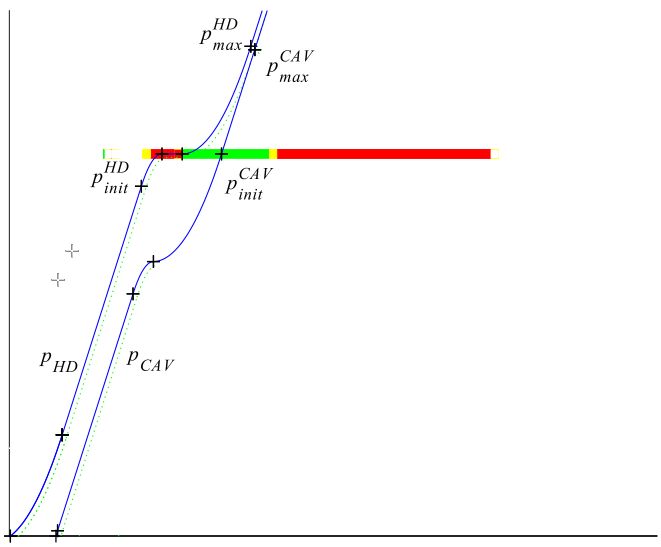


Fig. 7. Human-driven oriented backward shooting

4. Results and Discussions

4.1 Computational performance

The computational efficiency of the SH algorithm and the SH-based optimization framework are discussed in Ma et al. (2017), and both algorithms are applicable in real time. This subsection investigates the computational complexity of the proposed DP-SH algorithm. The default values for the input parameters $\{\bar{a}^f, \underline{a}^f, \bar{a}^b, \underline{a}^b, \bar{v}\}$ are shown in **Table 1**. For left-turn movements, we assume $\vartheta = 0.8$. Also note that without the loss of generality, we consider one movement in each of the four phases in the case study for presentation clarity. This does not affect the insights we obtained from the case studies. Particularly for computational efficiency analysis, since each movement can be independently evaluated, the actual additional computational time can be ignored when the computation for each movement in a phase is in parallel. The only difference is the addition of performance measures of different movements controlled by the same phase.

Table 1 Default input parameter values

Parameter	$\bar{a}^f(m/s^2)$	$\bar{a}^b(m/s^2)$	$\underline{a}^f(m/s^2)$	$\underline{a}^b(m/s^2)$	$\bar{v}(m/s^2)$
Value	1	1	-5	-5	30

We investigate the DP-SH algorithm's complexity with different input parameters $(L, N, f_s, \varepsilon, H)$. Table 2(a) shows computational performance results by varying L and N, which represents the segment length and number of vehicles in each segment correspondingly. It can be found intuitively from **Table 2**(a) and (b) that the number of vehicles N per phase does not affect the computational time significantly. That is because the DP-SH algorithm considers vehicles sequentially only for a specified planning horizon. It means that a limited number of vehicles will be calculated in DP-SH meaning vehicles that enter this segment after the current stage will be not considered in this stage. The algorithm, however, still needs to handle the vehicles that remain in the segment in the DP points and calculate the travel time of these vehicles, and therefore the computational time still increases to some extent. Segment length L also has a negligible impact on the computational time; however, the number of vehicles per time unit, which is reflected by saturation rate $f_s \in (0, C/G]$ (Ma et al., 2017), affects the computational time to some extent. That is because the more congested the road segment is, (i.e., higher values of f_s) , the more vehicle trajectories need to be constructed in each stage.

In contrast to other parameters, as shown in **Table 2**(c), the planning horizon H and step size ε significantly affect computational time, in both sparse and congested scenarios. Planning horizon H impacts computational time mainly because at each stage, DP-SH scans all possible values of s_j, and the algorithm evaluates all possible values of x_j (as shown in Equation 6) for each s_j. The extension of the planning horizon means more stages, and in each stage, larger sets of possible values of possible s_j and x_j will need to be considered. Thus, more vehicles can be taken into consideration, and more trajectories can be constructed and optimized. For the objective of reducing total average delay, extending H is effective; but, for the purpose of large-scale applications, a longer H consumes more time for computation and makes DP-SH infeasible. Based on a comprehensive consideration with these two objectives, we set the planning horizon at $H = 122\ s$. We use this planning horizon in reporting results in **Table 2**(a) and (b).

The computational times with the planning horizon of 122 seconds are about 9.75 seconds and 15.19 seconds for 10-vehicle-each-phase and 20-vehicle-each-phase scenarios, respectively, using a regular

office laptop (Intel Core i7-8550U CPU, 1.80 GHz processor frequency, and 8 GB DDR 4 RAM). This is close to or less than 10% of the planning horizon and can be considered as applicable in real time. It is reasonable to believe this algorithm can be much faster when using parallel computing on a powerful server.

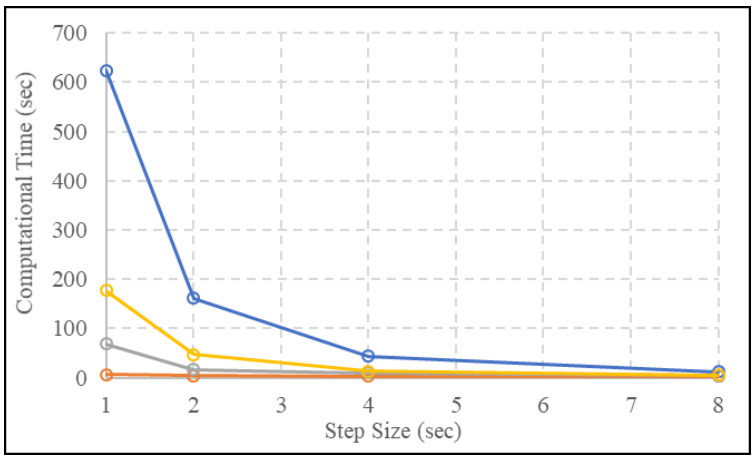


Fig. 8. Sensitivity of computational time to parameter step size ε

Table 2 DP-SH computational performance

(a) Computational Time (sec)			
	Segment Length $L(m)$		
	400	800	1200
N			
10	9.75	8.16	7.71
20	15.19	10.51	13.43
30	18.83	15.03	17.11
50	23.16	19.48	18.21

(b) Computational Time (sec)				
	Saturation Rate f_s			
	0.6	0.9	1.2	1.5
N				
10	11.39	9.72	9.88	9.75
20	12.57	16.56	15.83	15.19
30	16.64	18.06	16.51	18.83
50	18.01	17.59	23.96	23.16

(c) Computational Time (sec)										
	$f_s = 0.6$					$f_s = 1.5$				
	Planning Horizon (sec)									
	26	50	74	98	122	26	50	74	98	122
ε										

1	6.83	69.14	177.27	413.38	623.68	20.41	50.31	149.21	300.14	575.87
2	4.98	16.69	47.92	85.49	161.81	7.84	13.08	40.63	91.36	147.55
4	3.19	9.68	13.1	24.44	44.01	4.23	6.11	12.91	29.75	51.62
8	2.79	4.64	6.03	8.53	12.57	3.11	3.59	5.43	7.98	15.19

Clearly, larger step size ε can result in significantly less DP-SH evaluation scenarios and therefore reduce computational time effectively, but the final results may be suboptimal as a side-effect. This is because the use of a large time step size may not be able to scan the optimal solution due to the coarse resolution. As expected, as step size ε increases from 1 to 8 seconds, the computational time decreases almost exponentially, which as shown in **Fig. 8**.

Note that in this paper the shortest segment length we consider in the sensitivity analysis is 400 meters because no SH solutions can be guaranteed for shorter segments. This is because in some cases shorter segments may not allow vehicles to accelerate to the target speed v when they exist the intersection stop bar. The infeasibility usually occurs when the traffic is heavy (e.g., existing queues due to red lights). Then vehicles need to slow down to a complete stop, but they cannot accelerate to the target speed before the intersection because of the constraints (1) on maximum accelerations. As Zhou et al. (2017) indicated, for any given entry point $(0, v_0, t_0)$ and feasible state point (l_+, v_+, t_+), the set of feasible trajectories T_+ has a lower bound trajectory $\underline{p}$ (i.e., the vehicle having to decelerate to zero speed first). If we set the exit point $(L, v, \hat{t})$ as the feasible state point and let vehicle reach the target speed v at the intersection stop bar, a minimum segment length L_{min} can be proposed that ensure $T_+ \notin \emptyset$ for all vehicles. Under this scenario, the lower boundary trajectory is formulated as:

$$\underline{p}(\underline{t}) = \begin{cases} \frac{-v_0^2}{2\underline{a}} \quad , & \underline{t} \in \left[t_0, \frac{-v_0}{\underline{a}}\right]; \\ \frac{-v_0^2}{2\underline{a}} \quad , & \underline{t} \in \left[\frac{-v_0}{\underline{a}}, \hat{t} - \frac{v}{\bar{a}}\right]; \\ \frac{-v_0^2}{2\underline{a}} + \frac{v^2}{2\bar{a}} \quad , & \underline{t} \in \left[\hat{t} - \frac{v}{\bar{a}}, \hat{t}\right]. \end{cases} \tag{12}$$

in which the subject vehicle leaves the intersection at $\underline{t} = \hat{t}$. Therefore, the minimum length $L_{min} = \frac{-v_0^2}{2\underline{a}} + \frac{v^2}{2\bar{a}}$. Given a set of input parameters, L_{min} is positively correlated to the entry speed. When $v_0 = v$ and other parameters take the values in **Table 1**, $L_{min} = \frac{-v^2}{2\underline{a}} + \frac{v^2}{2\bar{a}} = \frac{-(30\ m/s)^2}{2\times(-5\ m/s^2)} + \frac{30\ m/s^2}{2\times(1.5\ m/s^2)} = 390\ m$.

Under these highly congested cases, the backward shooting component of the algorithm may find a start of the backward shooting segment upstream of the start point of the segment, and therefore not feasible. Then the results are not comparable to other scenarios with feasible SH solutions. The differences in effectiveness are not only caused by the segment length but also attributed to the fact that, due to traffic congestion, there are no feasible SH trajectories and the vehicles can only slowly

pass the intersection (with reduction of green time capacity). When urban signalized intersections are close-by, it is critical to consider optimization of trajectories through multiple traffic signals simultaneously. We leave this topic for future exploration. In this study, we consider three different segment lengths: $\{400, 800, 1200\}$ meters.

4.2 Scenario analysis

This section conducts a further simulation analysis to examine the algorithm's performance further. We first use the DP-SH algorithm to obtain the optimal signal timing plan that minimizes the average vehicle delay. Then, the corresponding timing plan is used in the numerical gradient-based optimization approach with SH as a subroutine to obtain optimal trajectories that minimize weighted average travel time and fuel consumption (refer to [Ma et al., 2017] for how to determine weights). In this case study, we use step size $\varepsilon = 8$ seconds, because this granularity also generates near-optimal results and significantly reduces computational burden. The results of DP-SH with default SH parameters and the results of final trajectory optimization will also be reported.

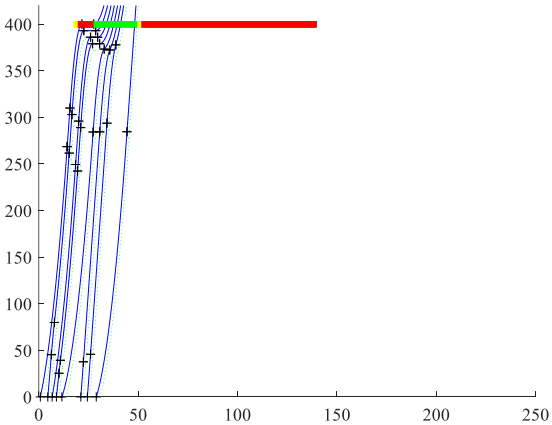
We aim to compare DP-SH with scenarios in which the intersection is controlled by adaptive signal control (ASC). The ASC algorithm used in the study is similar to the Controlled Optimization of Phases at an Intersection proposed by Sen and Head (1997).

Table 3 shows the simulation results of scenarios with different segment length L and saturation rate f_s. The segment length L has three levels, and saturation rate f_s has four levels, so there have twelve scenarios. As we can see, the SH-DP with default SH parameters can significantly improve both travel time (TT) and fuel consumption (Fuel). Compared with ASC results, the optimal trajectories reduce the travel time by 23.63% to 35.72% and fuel consumption by 11.75% to 31.5%. Even compared to the SH-DP results, there still have savings in fuel consumption by 10.13% to 17.55% with only limited sacrifice of travel time because the optimal case aims to smooth the trajectories more and avoid large accelerations. Note that it has been proved in Zhou et al. (14) that large acceleration parameters can result in best travel time performance. However, optimal trajectories can still achieve good travel times but significantly less fuel consumption.

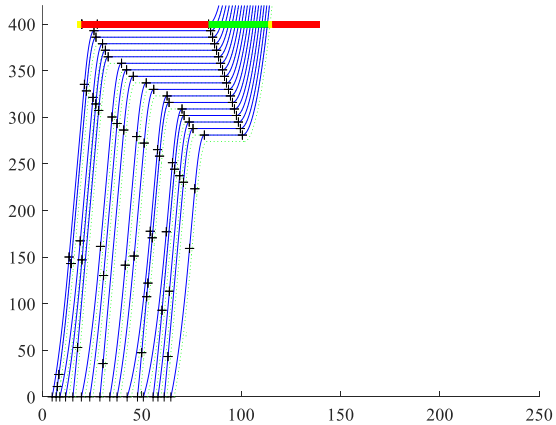
Table 3 Simulation results and benefits

		ASC		SH-DP		Optimal	
L (meter)	f_s	TT (second)	Fuel (L)	TT (second)	Fuel (L)	TT (second)	Fuel (L)
400	0.6	114.1036	0.1802	75.0557	0.1713	78.2417	0.1460
	0.9	114.9188	0.1969	75.9389	0.1671	76.3282	0.1420
	1.2	120.5917	0.1968	75.5210	0.1592	78.6325	0.1348
	1.5	122.3923	0.1901	75.6687	0.1603	78.6720	0.1355
800	0.6	138.6885	0.2563	94.4473	0.2126	98.8671	0.1821

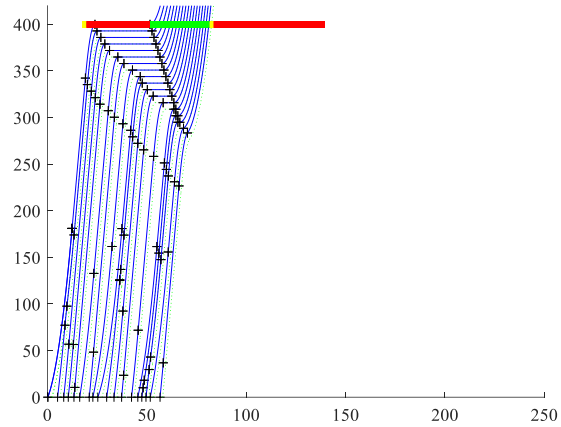
	0.9	115.3372	0.2431	79.2902	0.2188	82.5938	0.1804
	1.2	116.9924	0.2421	77.0059	0.211	79.071	0.1809
	1.5	114.7237	0.2413	76.3979	0.2125	80.2855	0.1829
1200	0.6	117.2056	0.2707	82.8195	0.269	87.7733	0.2389
	0.9	119.7548	0.2748	90.1766	0.2666	91.4602	0.2396
	1.2	128.9281	0.2832	92.037	0.2732	92.1662	0.2421
	1.5	130.7628	0.288	92.3627	0.2692	94.5745	0.2401
Percent Change				Optimal - ASC		Optimal - SH-DP	
		L (m)	f_s	TT	Fuel	TT	Fuel
		400	0.6	-31.43%	-18.98%	4.24%	-14.77%
			0.9	-33.58%	-27.88%	0.51%	-15.02%
			1.2	-34.79%	-31.50%	4.12%	-15.33%
			1.5	-35.72%	-28.72%	3.97%	-15.47%
		800	0.6	-28.71%	-28.95%	4.68%	-14.35%
			0.9	-28.39%	-25.79%	4.17%	-17.55%
			1.2	-32.41%	-25.28%	2.68%	-14.27%
			1.5	-30.02%	-24.20%	5.09%	-13.93%
		1200	0.6	-25.11%	-11.75%	5.98%	-11.19%
			0.9	-23.63%	-12.81%	1.42%	-10.13%
			1.2	-28.51%	-14.51%	0.14%	-11.38%
			1.5	-27.67%	-16.63%	2.39%	-10.81%



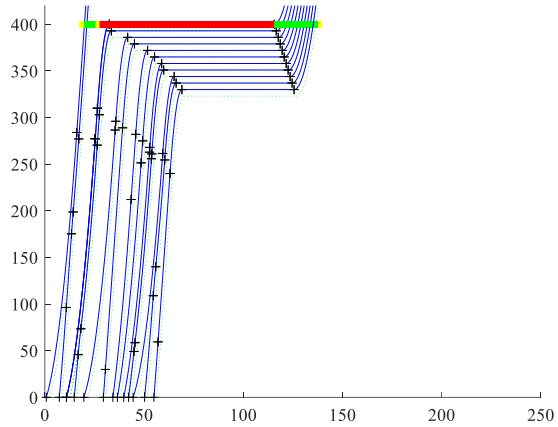
(a) Phase 1 – ASC



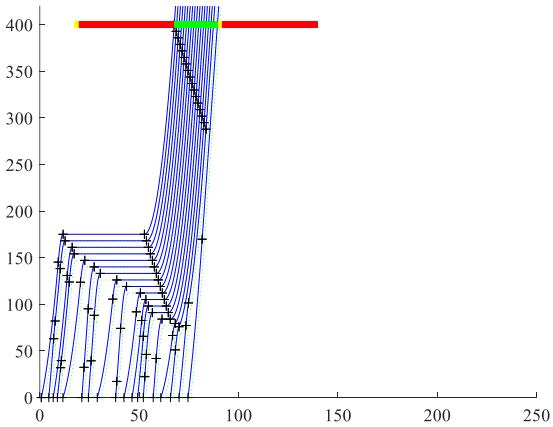
(b) Phase 2 – ASC



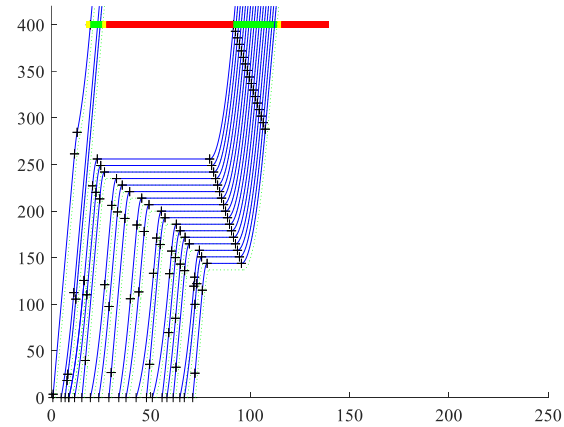
(c) Phase 3 – ASC



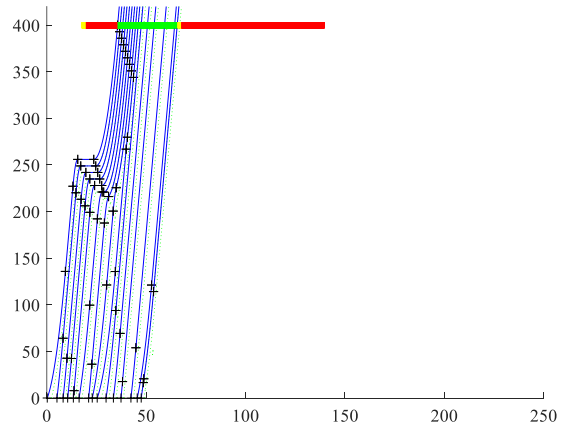
(d) Phase 4 – ASC



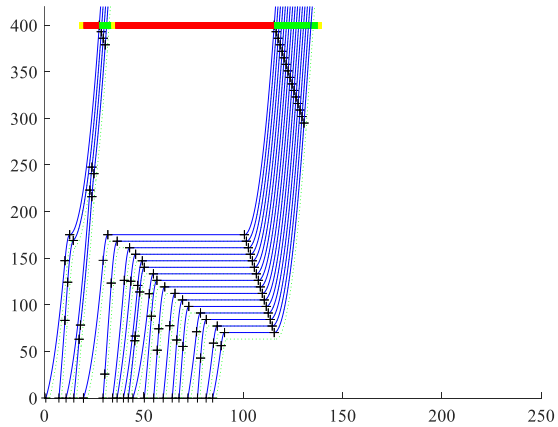
(e) Phase 1 – Optimal



(f) Phase 2 – Optimal



(g) Phase 3 – Optimal



(h) Phase 4 – Optimal

Fig. 9. Time planning and trajectory of scenario $L = 400m,\ f_s = 0.6$

Fig. 9 shows example trajectories and the optimal timing plan using our DP-SH approach under the scenario $L = 400\ m, f_s = 0.6$. **Fig. 9**(a) to (d) are movements under adaptive signal control scenario. Smoother trajectories in **Fig. 9**(e) to (h) are results from final trajectory optimization results and expected to generate less fuel consumption and safety concerns. More importantly, it can be seen from

all cases that DP-SH successfully distributes green times among the four phases when necessary, and green time assignment is well coordinated with the trajectory construction algorithm. This optimal algorithm allows vehicles to stop in the segment to generate optimal solutions. We see that the signal timing is adjusted such that multiple trajectories can pass the intersection at the beginning or the end of the green.

There may be concerns that the algorithm let vehicles stop in the middle of the segment, and the vehicles seem to be queuing to near the beginning of the segment, indicating a risk of spillback. Actually, this algorithm will prevent the spillback by increasing the acceleration values. For example, in **Fig. 9**(h), the fifth vehicle is commanded to stop at location 150 meters because the optimized acceleration after the stop is relatively small to save fuel and the algorithm predicts that there will be no spillback. In a hypothetical scenario when a possible spillback may occur, the algorithm will command a larger acceleration such that the fifth vehicle in **Fig. 9**(h) will stop downstream of location 150 meters.

Note again that the default SH parameters used in the DP-SH approach can be further optimized using the customized approach proposed in Ma et al. (2017), and this process can still generate more than 10% extra reduction of fuel consumption, though the travel time (or throughput) performance is similar. As our numerical examples show, this optimization process may be stuck at a local optimum, and the control parameter vector oscillates in a fairly small range. This is due to the non-convexity of the objective function. Therefore, alternative control parameter vectors are provided to the optimization algorithm as different start points, and the algorithm selects the final solution with optimal performance values.

4.3 Mixed Traffic Scenarios

This section shows the results under different CAV market penetration rates. A series of simulation runs with different CAV market penetration rates can help understand the robustness and effectiveness of DP-SH for a mixed traffic stream with both CAVs and human-driven vehicles. In this section, the market penetration varies from 0% to 100% with an increment of 10%. The 0% scenario reduces to all human-driven vehicles with adaptive signal control, similar to Sen and Head (1997). In **Table 5**, we present results obtained under two example scenarios of $L = 400\,m$, $f_s = 0.6$ and $L = 400\,m$, $f_s = 1.2$, which represent common urban congested and uncongested scenarios. The sequences of human-driven vehicles in the traffic steam are randomly assigned for 10 simulation runs under each scenario.

The HD-BSP assumes that the constructed analytical human-driven vehicles' trajectories can approximate the actual trajectories that are obtained using simulation to reduce computational burden. In order to numerically show the approximation accuracy, we obtain simulation trajectories using microscopic traffic simulation. Several microscopic simulations were conducted with the same speed limit and signal phasing and timing. In **Fig. 10**, the orange dotted line is the human-driven vehicle trajectory constructed using the intelligent driver model (IDM) with the default parameter set $\{a = 1.44, b = 1.67, \delta = 4, s_0 = 2, T = 1, v_0 = 30\}$ (Treiber et al., 2000). The three dashed lines are

the human-driven vehicle trajectories constructed using SH with different parameter sets. These three parameter sets are $\{\bar{a}^f = 1.50, \underline{a}^f = -5.00\}$, $\{\bar{a}^f = 1.76, \underline{a}^f = -6.60\}$ and $\{\bar{a}^f = 1.31, \underline{a}^f = -3.97\}$, which are selected from the former research results in Ma et al. (2017) to represent the most typical cases of good trajectories. All other six solid lines are human-driven vehicle data collected from VISSIM simulations. It can be found that these segments of trajectories overlap with each other well. We generated 15 human-driven trajectories stochastically with the random process in VISSIM. In **Table 4**, the differences between the three SH human-driven trajectories and 6 example simulated trajectories are reported. Also, the average percent differences between the three SH trajectories and all 15 simulated trajectories are all less than 5% on average. Note that the main difference actually comes from the variation of VISSIM desired cruise speeds and different types of vehicles (e.g., different desired accelerations), though VISSIM car-following parameters are set to similar to what SH scenarios imply. This indicates that SH human-driven trajectories can well approximate the IDM trajectories and VISSIM human-driven trajectories and can meet the requirement for a trajectory planning algorithm. More importantly, constructing SH trajectories analytically are much more efficient than the simulation, more applicable for online applications.

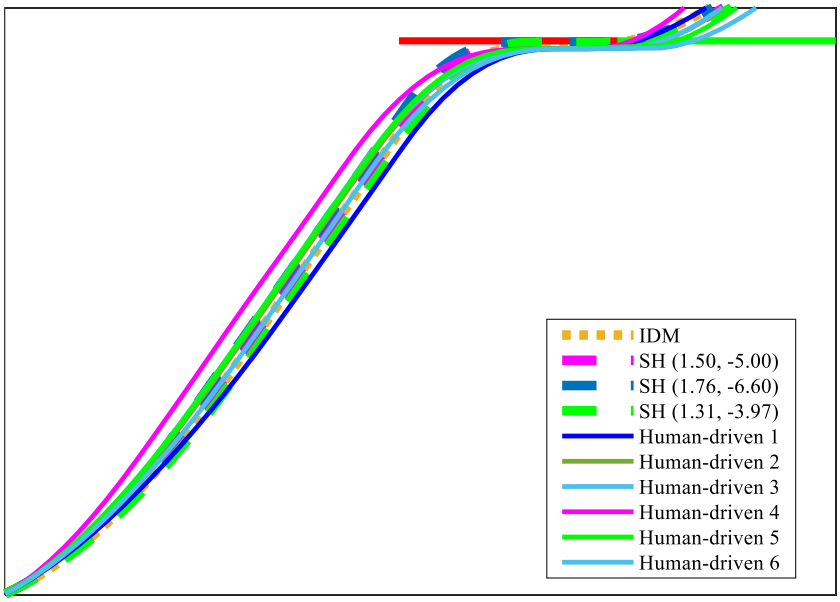


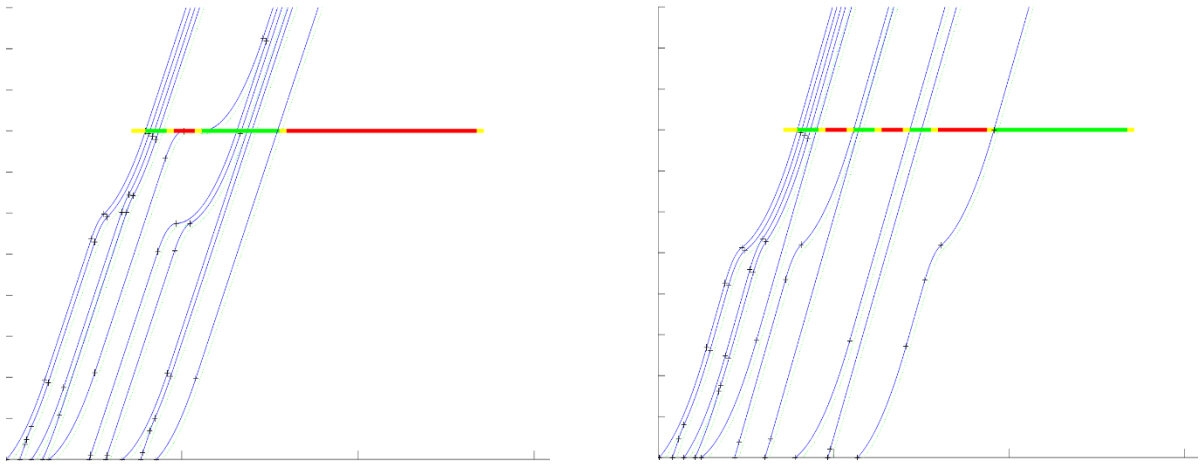
Fig. 10. Trajectories Built by IDM, SH and Collected Human-Driven Data

Table 4 Difference Between SH Built Trajectories and Human-driven Trajectories

	SH(1.50, -5.00)	SH(1.76, -6.60)	SH(1.37, -3.97)
Human-driven trajectory 1	0.70%	3.09%	2.7%
Human-driven trajectory 2	1.21%	2.30%	3.46%
Human-driven trajectory 3	4.30%	7.40%	1.51%
Human-driven trajectory 4	2.13%	2.83%	4.43%
Human-driven trajectory 5	1.45%	2.03%	3.65%

Human-driven trajectory 6	3.78%	6.86%	1.21%
…	…	…	…
Average of 16 human-driven trajectories	3.84%	4.25%	4.54%

The human-driven vehicle trajectories can have a major or minor impact on the following CAV's trajectory and the signal phasing and timing developed using DP-SH, and the impacts also depend on the distances between the human-driven vehicle and the following CAV. **Fig. 11** shows two example trajectories of the 90% market penetration scenario. In **Fig. 11**(a), the fifth vehicle is human-driven, and others are CAVs. It is expected the human-driven vehicle stops at the intersection during the red light and slowly start up afterward. The following CAV slows down upstream of the intersection, as commanded by SH, and then accelerates to and passes the stop bar with the target speed v. In **Fig. 11**(b), the human-driven vehicle is also the fifth vehicle in this stream, and in this case, it does not affect the following CAV and this situation can be regarded as the "best situation" under this scenario. In some other situations where the human-driven vehicle are the first vehicle in a traffic stream and it stops at the intersection for some time, all the following CAV trajectories may be affected, and they can be considered as the "worst situation". The DP-SH framework, however, can optimize traffic signal to avoid stops of human-driven vehicles and CAVs adaptively based on their actual arrival patterns (i.e., entry boundary conditions).



(a) (b)

Fig. 11. The Influence of a Human-driven Vehicle on CAVs

Table 5 The improvement of DP-SH with different market penetrations

MP	$L = 400m, fs = 0.6$			$L = 400m, fs = 1.2$		
	Average improvement (%)	Improvement under best situation (%)	Improvement under worst situation (%)	Average improvement (%)	Improvement under best situation (%)	Improvement under worst situation (%)

	TT	Fuel	TT	Fuel	TT	Fuel	TT	Fuel	TT	Fuel	TT	Fuel
0%	0	0	0	0	0	0	0	0	0	0	0	0
10%	3.09	6.23	15.46	17.15	0	0	7.60	8.79	25.93	16.67	3.02	6.96
20%	6.19	9.82	15.46	17.15	0	4.94	12.19	12.04	25.93	17.89	3.02	8.13
30%	12.47	13.12	15.46	17.15	0	6.16	21.75	14.99	34.23	19.26	3.02	8.59
40%	15.56	16.15	31.43	18.98	0	7.66	24.27	15.40	34.23	20.02	3.02	9.20
50%	18.76	16.84	31.43	18.98	0	10.16	28.21	23.48	34.79	31.50	3.02	10.98
60%	21.95	17.55	31.43	18.98	0	12.76	30.03	24.54	34.79	31.50	3.02	12.70
70%	21.98	18.11	31.43	18.98	0.15	15.59	30.86	28.96	34.79	31.50	3.02	14.38
80%	26.07	18.70	31.43	18.98	4.61	17.59	31.59	30.69	34.79	31.50	9.66	18.90
90%	28.08	18.81	31.43	18.98	4.61	17.59	33.21	30.95	34.79	31.50	11.49	20.68
100%	31.43	18.98	31.43	18.98	31.43	18.98	34.79	31.50	34.79	31.50	34.79	31.50

In **Table 5**, we report the average, worst-case, and best-case improvements. To obtain these results, we randomly selected 10 sets of possible positions of human-driven vehicles in the mixed traffic stream. In this table, we use ASC as the base case for comparison. The results show that DP-SH can significantly reduce travel time and fuel consumption on average, even at a low CAV market penetration rates. The average travel time decreases from 3.09% to 34.79%, and the fuel consumptions reduction ranges from 6.23% to 31.5%. We can see that even under low market penetration scenarios, the fuel consumption savings due to the improved smoothness of the trajectories are still quite significant compared with ASC, though the travel time improvements in those scenarios are limited. Also, the travel time savings under relatively congested traffic ($f_s = 1.2$) are much higher than the savings under less congested traffic ($f_s = 0.6$), particularly when the market penetration is below 60%. This is because the less congested traffic can be well handled by regular adaptive signal control, which can be limited during rather congested conditions. Trajectory control offers the possibility to modify vehicle arrival patterns at the intersection stop bar and further reduce system delays. Meanwhile, there are dramatic differences in fuel savings between congested and less congested cases when the market penetration rate is high. This is also because DP-SH can better handle the congested scenarios than adaptive signal control and therefore the trajectories of vehicles traversing the intersections can be smooth instead of having to coming to complete stops. As expected, both fuel consumption and travel time improvements increase as the market penetration rate becomes higher. The differences in improvements between the worst and best situations are quite dramatic, and it implies that the sequence of human-driven vehicles in a traffic stream affects the system performance significantly. Overall, the results prove that the DP-SH framework can effectively handle the mixed traffic situations without 100% CAV market penetration.

5. Conclusions and Future Research

Vehicle trajectory control and intersection control are two intertwined problems that need to be considered simultaneously. Advanced connected and automated vehicle (CAV) technologies offer

better solutions for these problems by both adjusting the intersection controllers and optimizing vehicle trajectories. This paper applies the joint control at intersections with physical signal controllers (i.e., optimizing traffic signal phases and timing) to illustrate the algorithm. Based on former research results in Ma et al. (2017), the Shooting Heuristic (SH) algorithm can construct near-optimal trajectories for a platoon of vehicles for one direction with a given traffic signal phasing and timing plan. By adopting this parsimonious SH algorithm and its numerical sub-gradient-based framework (NG-SH) as a sub-routine, this paper proposes a framework of dynamic programming with shooting heuristic as a subroutine (DP-SH). DP-SH can coordinate signal phases and timing and platoons of vehicles in different directions at an intersection section while optimizing comprehensive performances of travel time, fuel consumption, and safety. As results show, DP-SH has feasible computational time and solution optimality performance, which enables DP-SH to be applied in practical real-time scenarios. DP-SH generates a signal timing plan with given parameters and entry boundary information, and then constructs trajectories for each platoon of vehicles with NG-SH based on the signal timing plan. In this paper, numerical examples illustrate that all solutions significantly outperform the benchmark and adaptive signal control cases for all performance measures. The proposed DP-SH algorithm, compared to the adaptive signal control, can reduce the average travel time by up to 35.72% and save the consumption by up to 31.5%. In mixed traffic scenarios, system performance improves with increasing market penetration rates. Overall, the DP-SH enables this intertwined and infinite-dimensional optimization problem to be solved simultaneously and efficiently, and it can be considered suitable for real-time application when related technologies are deployed.

This study reveals great potential for combined signal timing control and vehicle trajectory optimization. This study can be further extended in several aspects. First, this paper controls intersections with physical signals. The proposed algorithms can also be applied and tested for intersection control for signaless intersections and autonomous merge areas. Second, this paper focuses on an isolated signalized intersection. It is interesting to apply and adapt the method to controlling busy physical urban signalized corridors. Besides, for real-time application, parallel computation algorithms, feedback control, and distributed computation need to be taken into account for future real-time implementation.

Acknowledgments

This research is supported in part by the U.S. Department of Transportation Federal Highway Administration through Grant DTFH61-12-D-00020 and DTFH61-12-D-00030, by University of Cincinnati Office of Research, and by the U.S. National Science Foundation through Grants CMMI CAREER #1453949 and CMMI #1541130. The work presented in this paper remains the sole responsibility of the authors.

References

Ahn, K., Rakha, H., and Park, S. 2013. Ecodrive application: Algorithmic development and preliminary testing. Transportation Research Record (2341), 1–11.

Beard, G., and Griffin, M. 2013. Discomfort during lateral acceleration: Influence of seat cushion and

backrest. Applied Ergonomics 44(4), 588–594.

Chai, L., Cai, B., ShangGuan, W., Wang, J. and Wang, H., 2018. Connected and autonomous vehicles coordinating approach at intersection based on space–time slot. Transportmetrica A: Transport Science, pp.1-23.

Feng, Y., Head, L., Khoshmagham, S., and Zamanipour, M. 2015, A real-time adaptive signal control in a connected vehicle environment. Transportation Research Part C 55, 460-473. https://doi.org/10.1016/j.trc.2015.01.007.

Feng, Y., Yu, C. and Liu, H.X., 2018. Spatiotemporal intersection control in a connected and automated vehicle environment. Transportation Research Part C: Emerging Technologies, 89, pp.364-383.

Gipps, P.G. , 1981. A behavioural car-following model for computer simulation. Transp. Res. Part B 15 (2), 105–111 .

Goodall, N.J., 2013. Traffic Signal Control with Connected Vehicles. Ph.D. dissertation. University of Virginia.

Guler, S.I., Menendez, M. and Meier, L., 2014. Using connected vehicle technology to improve the efficiency of intersections. Transportation Research Part C: Emerging Technologies, 46, pp.121-131.

He, Q., Head, K.L. and Ding, J., 2012. PAMSCOD: Platoon-based arterial multi-modal signal control with online data. Transportation Research Part C: Emerging Technologies, 20(1), pp.164-184.

Hoffmann, E., and Mortimer, R. 1994. Drivers' estimates of time to collision. Accident Analysis and Prevention 26 (4), 511–520. 8

Huang, Z. 2016. Traffic Signal Control at Connected Vehicle Equipped Intersections. Ph.D. dissertation. Mississippi State University.

Jiang, H., Hu, J., An, S., Wang, M. and Park, B.B., 2017. Eco approaching at an isolated signalized intersection under partially connected and automated vehicles environment. Transportation Research Part C: Emerging Technologies, 79, pp.290-307.

Lee, J. and Park, B. 2012. Development and evaluation of a cooperative vehicle intersection control algorithm under the connected vehicles environment. IEEE Transactions on Intelligent Transportation Systems, 13, 81–90. http://dx.doi.org/10.1109/TITS.2011.2178836.

Lee, J., Park, B.B. and Yun, I., 2013. Cumulative travel-time responsive real-time intersection control algorithm in the connected vehicle environment. Journal of Transportation Engineering, 139(10).

Li, X., Cui, J., An, S., and Parsafard, M. 2014. Stop-and-go traffic analysis: Theoretical properties, environmental impacts and oscillation mitigation. Transportation Research Part B 70, 319–339. https://doi.org/10.1016/j.trb.2014.09.014.

Li, Z., Elefteriadou, L. and Ranka, S., 2014. Signal control optimization for automated vehicles at isolated signalized intersections. Transportation Research Part C: Emerging Technologies, 49, pp.1-18.

Ma, J., Li, X., Zhou, F., Hu, J. and Park, B.B., 2017. Parsimonious shooting heuristic for trajectory design of connected automated traffic part II: Computational issues and optimization.

Transportation Research Part B: Methodological, 95, pp.421-441.

Priemer, C. and Friedrich, B. 2009. A decentralized adaptive traffic signal control using V2I communication data. Proceedings of the 12th International IEEE Conference on Intelligent Transportation Systems. pp. 1–6. http://dx.doi.org/10.1109/ITSC.2009.5309870.

Sen, S. and Head, K.L., 1997. Controlled Optimization of Phases at an Intersection. Transportation Science (31)1, 5–17. http://dx.doi.org/10.1287/trsc.31.1.5.

Seredynski, M. and Khadraoui, D., 2014, October. Complementing transit signal priority with speed and dwell time extension advisories. In Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on (pp. 1009-1014). IEEE.

Seredynski, M., Khadraoui, D. and Viti, F., 2015, October. Signal phase and timing (SPaT) for cooperative public transport priority measures. In Proc. 22nd ITS World Congress.

Treiber, M., Hennecke, A. and Helbing, D., 2000. Congested traffic states in empirical observations and microscopic simulations. Physical review E, 62(2), p.1805.

Wang, M., Daamen, W., Hoogendoorn, S.P., and van Arem, B. 2014a. Rolling horizon control framework for driver assistance systems. Part i: Mathematical formulation and non-cooperative systems. Transportation Research Part C 40, 271–289.

Wang, M., Daamen, W., Hoogendoorn, S.P., and van Arem, B. 2014b. Rolling horizon control framework for driver assistance systems. Part ii: Cooperative sensing and 40 cooperative control. Transportation Research Part C, 290–311.

Williams, B.M., Rouphail, N.M., Kim, S. and Song, T.J., 2016. Incident Management Assistance Patrols–Assessment of Benefits/Costs, Route Selection, and Prioritization (No. Research report number: FHWA/NC/2014-12). University of Kentucky Transportation Center.

Yang, K., Guler, S.I. and Menendez, M., 2016. Isolated intersection control for various levels of vehicle technology: Conventional, connected, and automated vehicles. Transportation Research Part C: Emerging Technologies, 72, pp.109-129.

Yu, C., Feng, Y., Liu, H.X., Ma, W. and Yang, X., 2018. Integrated optimization of traffic signals and vehicle trajectories at isolated urban intersections. Transportation Research Part B: Methodological, 112, pp.89-112.

Zhou, F., Li, X. and Ma, J., 2017. Parsimonious shooting heuristic for trajectory design of connected automated traffic part I: Theoretical analysis with generalized time geography. Transportation Research Part B: Methodological, 95, pp.394-420.