Transportation Research Part B 118 (2018) 429–456

Contents lists available at ScienceDirect

Transportation Research Part B

journal homepage: www.elsevier.com/locate/trb

A piecewise trajectory optimization model for connected automated vehicles: Exact optimization algorithm and queue propagation analysis

Xiaopeng Li[a,*], Amir Ghiasi[a], Zhigang Xu[b], Xiaobo Qu[c,*]

[a] Department of Civil and Environmental Engineering, University of South Florida, FL 33620, USA
[b] School of Information Engineering, Chang'an University, Xi'an 710064, China
[c] Department of Architecture and Civil Engineering, Chalmers University of Technology, Gothenburg SE-412 96, Sweden

ARTICLE INFO

Article history:
Received 20 September 2017
Revised 21 September 2018
Accepted 6 November 2018
Available online 16 November 2018

Keywords:
Connected automated vehicles
Trajectory optimization
Traffic smoothing
Exact algorithm
Shooting heuristic
Fuel consumption
Driving comfort
Signalized intersection
Non-stop intersection

ABSTRACT

This paper formulates a simplified traffic smoothing model for guiding movements of connected automated vehicles on a general one-lane highway segment. Adapted from the shooting heuristic proposed by Zhou et al. (2017) and Ma et al. (2017), this model confines each vehicle's trajectory as a piecewise quadratic function with no more than five pieces and lets all trajectories in the same platoon share identical acceleration and deceleration rates. Similar to the shooting heuristic, the proposed simplified model is able to control the overall smoothness of a platoon of connected automated vehicles and approximately optimize traffic performance in terms of fuel efficiency and driving comfort. While the shooting heuristic relies on numerical meta-heuristic algorithms that cannot ensure solution optimality, we discover a set of elegant theoretical properties for the general objective function and the associated constraints in the proposed simplified model, and consequentially propose an efficient analytical algorithm for solving this problem to the exact optimum. Interestingly, this exact algorithm has intuitive physical interpretations, i.e., stretching the transitional parts of the trajectories (i.e., parts with acceleration and deceleration adjustments) as far as they reach the upstream end of the investigated segment, and then balancing the acceleration and deceleration magnitudes as close as possible. This analytical exact model can be considered as a core module to a range of general trajectory optimization problems at various infrastructure settings. Numerical examples reveal that this exact algorithm has much more efficient computational performance and the same or better solution quality compared with the previously proposed shooting heuristic. These examples also illustrate how to apply this model to CAV control problems on signalized segments and at non-stop intersections. Further, we study a homogeneous special case of this model and analytically formulate the relationship between queue propagation and trajectory smoothing. One counter-intuitive finding is that trajectory smoothing may not always cause longer queue propagation but instead may mitigate queue propagation with appropriate settings. This theoretical finding has valuable implications to joint optimization of queuing management and traffic smoothing in complex transportation networks.

* Corresponding authors.
E-mail addresses: xiaopengli@usf.edu (X. Li), xiaobo@chalmers.se, drxiaoboqu@gmail.com (X. Qu).

https://doi.org/10.1016/j.trb.2018.11.002
0191-2615/

1. Introduction

Stop-and-go movements (Li et al., 2010) are almost inevitable experience in highway traffic due to intrinsic limitations in human driving behavior (Li and Ouyang, 2011; Li et al., 2012; Jiang et al., 2015) and information access (Li et al., 2014a). Stop-and-go traffic is linked to a number of adverse impacts from highway traffic, including excessive fuel consumption, extra safety hazards, and increased travel delay. Among a number of potential solutions to stop-and-go traffic (e.g., variable speed limits (Lu and Shladover, 2014), ramp metering (Hegyi et al., 2005), merging traffic control (Spiliopoulou et al., 2009), and signal coordination (Day et al., 2010), the connected automated vehicle (CAV) technology has received increasing attention recently due to its capability of controlling vehicle trajectories and modifying driving behavior (Ma et al., 2016). Various studies have been conducted to utilize CAV to improve traffic smoothness and throughputs on both uninterrupted freeways and signalized arterials.

Studies on the freeway side focus on guiding vehicle trajectories for minimum speed oscillations and minimum conflicts in lane changes and merges. Van Arem et al. (2006) investigates traffic stability and efficiency at a merge point. Ahn et al. (2013) proposed a rolling-horizon model for an individual CAV control strategy that minimizes fuel consumption and emissions at different grades. Yang and Jin (2014) studied a vehicle speed control strategy to reduce vehicle fuel consumption and emissions. Wang et al. (2014a,b) proposed optimal control models to determine optimal accelerations of a platoon of CAVs to minimize a variety of objective cost functions in a rolling horizon manner. Later, Wang et al. (2016) investigated distributed CAV acceleration control methods to mitigate formation and propagation of moving jams.

Studies on the signalized arterial side concern the problem of coordinating and scheduling vehicle trajectories to avoid conflicts at crossing points while improving traffic performance measures. Some studies focus on scheduling of vehicles arrival and departure times at an intersection and aim to minimize stops and delay at the intersection. Li and Wang (2006) studied CAV scheduling and trajectory planning for a two-lane intersection, using spanning tree and simulation techniques. Dresner and Stone (2008) investigated a similar non-stop intersection problem and proposed a heuristic control algorithm that processes vehicles as a queuing system. Lee and Park (2012) proposed a nonlinear optimization model to optimize trajectories for CAVs approaching and passing a non-stop intersection. Zohdy and Rakha (2016) proposed a nonlinear optimization model that integrates an embedded car-following rule and an intersection communication protocol for non-stop intersection management. Other studies consider how to control vehicle trajectories in compliance with existing traffic signal timing at intersections. Trayford et al. (1984a,b) proposed to use speed advice to reduce fuel consumption for vehicles approaching an intersection. Later studies further investigated car-following dynamics (Sanchez et al., 2006), in-vehicle traffic light assistance (Iglesias et al., 2008; Wu et al., 2010), multi-intersection corridors (Mandava et al., 2009; Guan and Frey, 2013; De Nunzio et al., 2013), scaled-up simulation (Tielert et al., 2010), and electric vehicles (Wu et al., 2015). These studies mainly concerned control of vehicle speeds but ignored acceleration detail, which however could cause significant errors in estimating fuel consumption and emissions and practical difficulties for real vehicles to follow these trajectories with speed jumps. To address this issue, Kamalanathsharma et al. (2013) considered acceleration detail in optimizing an individual vehicle trajectory. Li's team (Zhou et al., 2017; Ma et al., 2017) proposed a parsimonious shooting heuristic to simultaneously optimize trajectories of a stream of CAVs approaching an intersection.

Most studies on using CAVs to smooth traffic are essentially centered on a vehicle trajectory optimization problem. Simply speaking, this problem determines the optimal shapes for interdependent vehicle trajectories constrained by their boundary conditions, physical limits and safety risks. However, this problem in a general form is very complex and difficult to solve due to several computational challenges. First, each trajectory is essentially an infinite-dimensional object since every point along it can be a variable, and thus this problem deals with an infinite number of decision variables. Second, the optimization objective often involves highly non-linear components such as fuel consumption and emissions. Third, problem constraints can be quite complex due to vehicle interactions (e.g., two consecutive vehicles have to maintain a safe headway all the time) and boundary conditions (e.g., vehicles can only pass an intersection during a green light). Directly solving this problem, even a quite simple version, requires quite some computational resources and sophistication in algorithm design (Von Stryk and Bulirsch, 1992; Wei et al., 2016). Instead of solving the original trajectory problem, Li's team (Zhou et al., 2017; Ma et al., 2017) opted to investigate a reduced problem where a trajectory is broken into a small number of quadratic sections and only a few acceleration levels are used to control the overall smoothness of the stream of vehicle trajectories. Although this reduced problem may not necessarily solve the true optimal solution to the original problem, it can yield a stream of trajectories with appealing overall smoothness and performance measures that much outperform the benchmark case without trajectory smoothing. Further, this simplification enables discovery of elegant theoretical properties and development of an efficient sub-gradient-based optimization algorithm for real-time applications.

Despite the breakthroughs from this previous work, there still remain a number of fundamental challenges in CAV trajectory optimization. First, the trajectory optimization method based on the shooting heuristic still relies on a numerical algorithm that does not ensure solution optimality and may need many iterations to converge. Second, one may intuitively think that since trajectory smoothing always leads to longer acceleration and deceleration distances for vehicles (though with milder acceleration magnitudes), it shall always yield a longer queue propagation or spillback. However, this intuition has not been systematically and analytically verified. Third, the previous work only focuses on a signalized intersection and applications in other types of highway segments remain to be investigated.

This paper aims to address these challenges by investigating a further simplified trajectory optimization model. This simplified model confines each trajectory to consist of no more than five quadratic sections. This simplification is based on

the observation that a trajectory in the optimal shooting heuristic solution likely has a small number of quadratic sections, probably because more sections postulate more frequent accelerations, less smooth trajectories, and thus a sub-optimal solution. Further, this simplified model assumes that all vehicles arrive at the same speed. This is a reasonable assumption for cases when the upstream traffic is well controlled in a similar manner. While the new model preserves the main features of the shooting heuristic (e.g., yielding overall smooth trajectories) only with these minor simplifications, it has a number of appealing theoretical and algorithmic properties that were not found in the previous shooting heuristic. We discover elegant theoretical relationships between a general objective function and its associated variables and constraints. These findings enable development of an analytical solution algorithm that efficiently solves the exact solution to this simplified problem. This optimal solution has an elegant physical interpretation, i.e., stretching the trajectories as far as the deceleration parts of these trajectories are about to spill back upstream and making the acceleration and deceleration magnitudes as close as possible. This analytical exact algorithm could be consider as a core module to trajectory optimization problems on both freeway (e.g., speed harmonization, shock wave dampening, and merging control; see Ghiasi et al., 2017b; Wang et al., 2016; Yang and Jin, 2014; Ahn et al., 2013; Letter and Elefteriadou, 2017) and signalized arterial (e.g., joint signal and trajectory optimization or non-stop intersection control; see Zhou et al., 2017; Ma et al., 2017; Sun et al., 2017; Yang et al., 2016; Zohdy and Rakha, 2016; Guler et al., 2014; Li et al., 2014b; Kamalanathsharma et al., 2013) sides that have mostly relied on numerical and heuristic algorithms in the past. Numerical studies are conducted to verify the solution efficiency and quality compared with the existing approach and illustrate applications of the proposed model to signalized highways and non-stop intersections. Although the proposed longitudinal trajectory control can be directly implemented in future one-lane roadways (e.g., CAV managed lanes Ghiasi et al., 2017a), to demonstrate the robustness and applicability of the proposed model in multi-lane highway traffic, the algorithm is numerically tested on a multi-lane highway and on a stochastic vehicle arrival time setting. Further, to examine the intuitive conjecture that traffic smoothing leads to longer queue propagation, we investigate a homogeneous yet representative case and find analytical conditions for this conjecture to fail. Interestingly, we find that trajectory smoothing may not always cause longer queue propagation but instead may mitigate queue propagation with appropriate settings. This theoretical finding has valuable implications to joint optimization of queuing management and traffic smoothing in complex transportation networks. Note that the focus of this study is planning near-optimum trajectories for a platoon of vehicles with sufficient safety spacing, whereas real-time control (e.g., at the sub-second resolution) to ensure vehicles to follow the planned trajectories (while always maintaining sufficient safety gaps supported with a fail safe mechanism) is out of the scope of this study.

This paper is organized as follows. Section 2 describes the original trajectory optimization problem and proposes a simplified model. Section 3 investigates relevant theoretical properties of the simplified model and proposes an exact analytical solution algorithm accordingly. Section 4 presents numerical examples that test the solution algorithm efficiency and illustrate applications of this algorithm. Section 5 presents a homogeneous special case and analyzes the relationship between trajectory smoothing and queue propagation. Section 6 concludes this paper and briefly discusses future research directions.

2. Problem statement

2.1. Original formulation

This study investigates the pure CAV traffic scenario where all vehicles are controllable CAVs rather than mixed traffic including both CAVs and human-driven vehicles. Although the pure CAV traffic scenario is not likely prevailing soon, certain special infrastructures, such as managed lanes (Ghiasi et al., 2017a), may possibly be implemented at a local scale in the near future. Further, knowledge on pure CAV traffic performance provides a benchmark to the full potential of the CAV based management paradigm and sheds insights into the final goals and directions to pursue for even near future mixed traffic management. For convenience of the readers, the key symbols are listed in the notation table in Appendix A. Fig. 1 illustrates the studied problem. Consider a one-lane highway segment starting at location 0 upstream and ending at location L downstream. A number of N CAVs, indexed as $n \in \mathcal{N} = \{1, 2, \ldots, N\}$, consecutively arrive at location 0, drive through the segment and exit this segment at location L. This problem assumes that with advanced information system, vehicle n's arrival time at location 0 can be accurately estimated as t_n^-. Without much loss of generality, we assume that each vehicle arrives at a maximum cruising speed $\bar{v}$ for the maximum system throughput.[1] This problem also assumes that the traffic control protocol is given, and vehicle $n's$ departure at location L is scheduled at time t_n^+ in advance.[2] One example is that the CAV traffic control protocol is a fixed-timing traffic signal, and every vehicle is scheduled to depart as early as possible, which yields a fixed departure time for each vehicle. Another example is the first-in-first-out control policy at a non-stop

[1] The problem in Zhou et al. (2017) allows vehicles to arrive at different speeds, which however only affects the initial part of the trajectories but does not change much overall trajectory patterns. According to several studies (Zhou et al., 2017; Ma et al., 2017; Xu et al., 2018), relaxing this assumption would have marginal impacts to the overall trajectory shapes. Please see Xu et al. (2018) for the detailed revisions that adapt the proposed algorithm to heterogeneous arrival and departure speeds. Further, when all vehicles are CAVs, arrival vehicles at this segment shall be departure vehicles from the upstream segments, and the proposed control shall ensure that all vehicles exit from the upstream segments at speed $\bar{v}$, which is equivalent to arrival speed $\bar{v}$ at this segment.

[2] Note that the problem in Ma et al. (2017) does not fix a vehicle's exit time t_n^+ but rather lets the optimization result determine it. Nonetheless, the numerical experiments in Ma et al. (2017) show that exit time t_n^+ does not change much from the theoretical minimum value in the optimal results. Thus fixing the exit times will not much affect the generality of CAV trajectory optimization at the highway segment level.

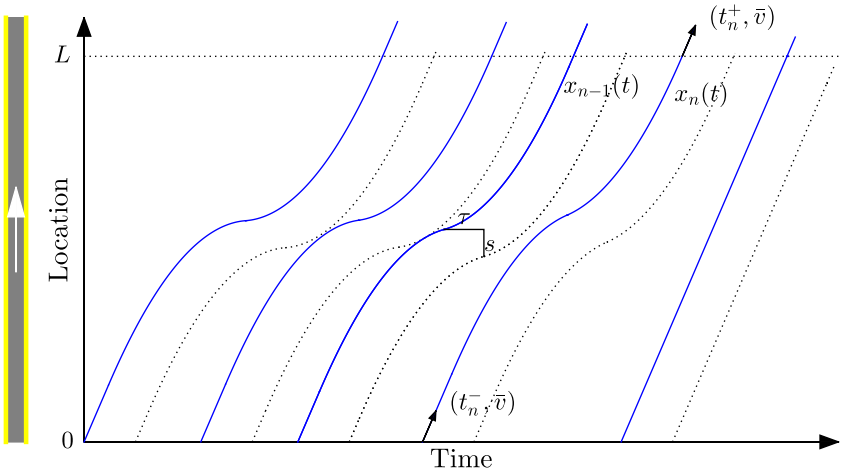

Fig. 1. Illustration of the constraints.

intersection (or a merge point), and each vehicle's departure time is essentially determined by the arrival times of all vehicles that have already arrived. To achieve the maximum system throughput, as discussed in Zhou et al. (2017), we postulate that each vehicle departs the segment at speed $\bar{v}$. We use $x_n(t), \forall t \in [t_n^-, t_n^+]$ to denote the trajectory of vehicle n, i.e., the location of vehicle n at every time point t. Define trajectory vector $\mathbf{x} = [x_n]_{n\in\mathcal{N}}$, which shall satisfy the following constraints.

- Entry boundary constraints: Vehicle n is cruising at speed $\bar{v}$ at location 0 at time t_n^-, i.e.,

$$x_n(t_n^-) = 0, \forall n \in \mathcal{N}, \tag{1}$$

$$\dot{x}_n(t_n^-) = \bar{v}, \forall n \in \mathcal{N}, \tag{2}$$

- Exit boundary constraints: Based on the CAV control protocol, each vehicle n shall exit location L at cruising speed $\bar{v}$ at a predetermined time t_n^+, i.e.,

$$x_n(t_n^+) = L, \forall n \in \mathcal{N}, \tag{3}$$

$$\dot{x}_n(t_n^+) = \bar{v}, \forall n \in \mathcal{N}. \tag{4}$$

- Speed constraints: Vehicle n cannot back up any time and cannot go beyond cruising speed $\bar{v}$, i.e.,

$$0 \le \dot{x}_n(t) \le \bar{v}, \forall t \in [t_n^-, t_n^+], n \in \mathcal{N}. \tag{5}$$

- Acceleration constraints: Vehicle n's acceleration is bounded between minimum acceleration (or maximum deceleration) $\underline{a} < 0$ and maximum acceleration $\bar{a} > 0$, i.e.,

$$\underline{a} \le \ddot{x}_n(t) \le \bar{a}, \forall t \in [t_n^-, t_n^+], n \in \mathcal{N}. \tag{6}$$

  We require $\ddot{x}_n(t)$ to be a piecewise second-order differentiable function. At a joint between two pieces, $\ddot{x}_n(t)$ is defined as the left differential.

- Safety constraints: For every two consecutive vehicles $n-1$ and n, their trajectories have to maintain certain safety headway to ensure

$$x_{n-1}(t-\tau) - x_n(t) \ge s, \forall t \in [t_{n-1}^- + \tau, t_{n-1}^+], \forall n \in \mathcal{N}\backslash\{1\}, \tag{7}$$

  where τ is the minimum time headway and s is the minimum space headway. The value of τ depends on CAV communication and control delay, and the value of s depends on the CAV size and the reserved safety buffer.

Each trajectory x_n is associated with an operational cost, formulated as the cost function below

$$c(x_n) := \int_{t_n^-}^{t_n^+} e(\dot{x}_n(t), \ddot{x}_n(t))dt, \forall n \in \mathcal{N} \tag{8}$$

where function $e(\dot{x}_n(t), \ddot{x}(t))$ measures the instantaneous cost of x_n at time point t. This study considers a general class of e functions as follows:

$$e(\dot{x}, \ddot{x}) = \sum_{p=1}^{P} A_p |\ddot{x}|^p + \sum_{q=1}^{Q} B_q \dot{x}^q + \sum_{r=-\infty}^{\infty} C_r \dot{x}^r \ddot{x}, \tag{9}$$

where power index P can be any positive integer, $Q \in \{1, 2, 3, 4\}$ and all coefficients A_p, B_q, $C_r \geq 0$. This equation does not limit the upper and lower bounds to r in the third term, which is just intended to capture all general cases that the proposed model can handle. In actual specific applications, r has to be set on a finite range, which is nonetheless equivalent to setting all C_r values to zero for all r's outside of this range. The reason we use this function form as the optimization objective is two-fold. First, it is a closed-form function suitable for analytical studies. More importantly, several commonly used instantaneous vehicle performance measures can be written as special forms of this function. For example, one special case is vehicle specific power (Frey et al., 2002), which is approximately propositional to vehicle fuel consumption and positively correlated with emissions, i.e.,

$$e^{\mathrm{VSP}}(\dot{x}, \ddot{x}) = \xi \dot{x}\ddot{x} + \psi \dot{x} + \zeta \dot{x}^3. \tag{10}$$

where ξ, ψ and ζ are positive coefficients. Another special case is squared acceleration (Smith et al., 1978), which has been frequently used to indicate driving comfort, i.e.,

$$e^{\mathrm{SA}}(\dot{x}, \ddot{x}) = \ddot{x}^2. \tag{11}$$

With formula (9), vehicle cost (8) can be rewritten as

$$c(x_n) = \sum_{p=1}^{P} A_p \int_{t_n^-}^{t_n^+} |\ddot{x}_n(t)|^p dt + \sum_{q=1}^{Q} B_q \int_{t_n^-}^{t_n^+} \dot{x}_n(t)^q dt. \tag{12}$$

Note that the third item is dropped because $\int_{t_n^-}^{t_n^+} \dot{x}_n(t)^r \ddot{x}_n(t) dt = \frac{1}{r+1} \dot{x}_n(t)^{r+1} \Big|_{t_n^-}^{t_n^+}$ is always equal to zero for any $r \in \mathbb{R}$ since $\dot{x}_n(t_n^-) = \dot{x}_n(t_n^+) = \bar{v}$. Then the system performance is measured by the average cost per vehicle

$$C(\mathbf{x}) := \sum_{n \in \mathcal{N}} c(x_n)/N.$$

Now the primary trajectory optimization problem can be formulated as follows:

$$\mathrm{PTO}: \min_{\mathbf{x}} C(\mathbf{x}) \tag{13}$$

subject to Constraints (1)–(7). Note that in this problem, we assume that all vehicles are subject to centralized control, and objective (13) aims to minimize the total cost for all vehicles. This is different from the decentralized control problem where each vehicle controls its own trajectory to minimize its own cost selfishly without considering its impact on the other vehicles.

In order for problem PTO to be feasible, parameters should satisfy the following conditions. Since vehicles shall satisfy safety constraints (7), then $\{t_n^-\}$ values shall satisfy

$$t_n^- - t_{n-1}^- \geq \tau + s/\bar{v}, \forall n \in \mathcal{N} \backslash \{1\}. \tag{14}$$

Since vehicles can only exit the highway segment consequentially, safety constraints (7) imply that $\{t_n^+\}$ should satisfy a similar relationship,

$$t_n^+ - t_{n-1}^+ \geq \tau + s/\bar{v}, \forall n \in \mathcal{N} \backslash \{1\}. \tag{15}$$

Further, since vehicles only have a limited speed $\bar{v}$, the following constraints shall be satisfied

$$t_n^+ - t_n^- \geq L/\bar{v}, \forall n \in \mathcal{N}. \tag{16}$$

2.2. Simplified model formulation

It is difficult to solve PTO to the exact optimal due to infinite-dimensional variables, highly nonlinear objective and vehicle dependency in the constraints. By adapting the simplification approach in Ma et al. (2017), instead of directly analyzing PTO, this study formulates a simplified model that restricts each trajectory to consist of no more than five quadratic segments and trajectories in each platoon (platoon will be defined in the next paragraph) have identical acceleration and deceleration magnitudes.[3] Although this restriction may slightly sacrifice the solution optimality since it reduces the feasible region of the trajectories, we believe that the restricted solution shall be close to the true optimum for the following two reasons. First, since a realistic vehicle cost function (12) shall be optimal with a smooth vehicle trajectory that does not frequently decelerate and accelerate, using piecewise quadratic approximation shall not bring too much error to the optimal trajectory shape. Second, since these vehicles closely follow each other in a platoon, their optimal trajectories shall have similar acceleration and deceleration levels, and thus assuming them sharing identical acceleration and deceleration magnitudes will not much compromise the optimality. While it is interesting to verify this conjecture with theoretical analysis

[3] Note that theoretically, such simplification causes unbounded jerk as the acceleration value may jump from one segment to another. To resolve this issue, we can revise the neighborhood around the conjunction of every two consecutive quadratic segments into a smooth curve with continuous acceleration change (e.g., a cubic segment). While the transitional segments may affect the solution optimality, due to their small sizes we believe that the effects would not be too much, and thus are ignored in the theoretical investigations.

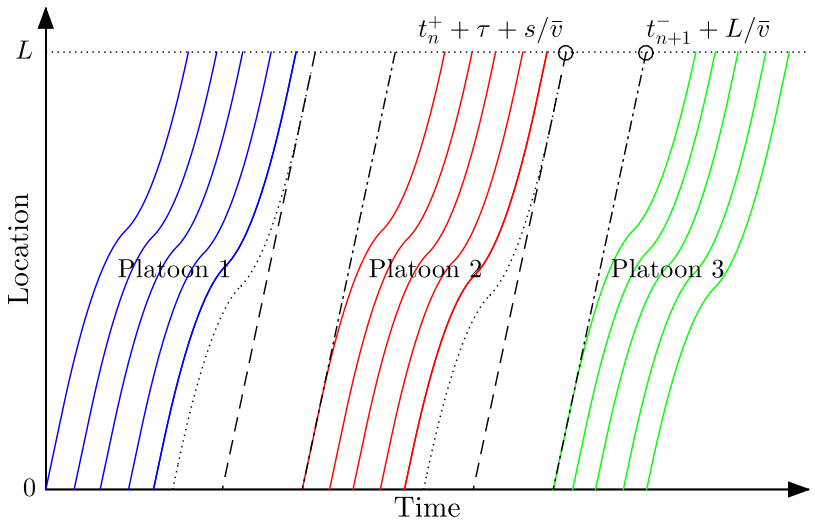

Fig. 2. Illustration of platoon decomposition.

and numerical experiments, it is beyond the scope of this study. This study will focus on formulating, analyzing and testing this simplified model, and this section will present the formulation of this model.

First, we want to note that safety constraints (7) will not be activated for two consecutive vehicles n and $n+1$ if vehicle n's departure time is not much later than vehicle $n+1$'s arrival time, i.e.,

$$t_n^+ + \tau + s/\bar{v} \le t_{n+1}^- + L/\bar{v}.$$

If this inequality holds, the optimization problem can be decomposed into two sub-problems, one for vehicles up to n and the other for vehicles from $n+1$ on. Such decomposition can be repeated for other vehicles satisfying this condition. Eventually, vehicles will be decomposed into a number of platoons such that only two vehicles in the same platoon can possibly activate safety constraints (7). This is illustrated in Fig. 2. Vehicle platoons can be identified by the following platooning algorithm (PA).

PA-0: Set platoon collection $\mathbb{M} = \emptyset$. Set current vehicle $n = 1$, and initialize current platoon vector $\mathcal{M} = [1]$.
PA-1: Check whether $t_n^+ + \tau + s/\bar{v} > t_{n+1}^- + L/\bar{v}$ holds. If yes, then vehicles n and $n+1$ belong to the same platoon, append $n+1$ to the end of $\mathcal{M}$. Otherwise, vehicles n and $n+1$ shall be in different platoons, and then add $\mathcal{M}$ to $\mathbb{M}$ and start a new platoon $\mathcal{M} = [n+1]$;
PA-2: If $n = N$, add $\mathcal{M}$ to $\mathbb{M}$ and end the algorithm. Otherwise, set $n = n+1$, got to Step PA-1.

With this, we can decompose the trajectory optimization into a set of subproblems, each for one platoon separately. Then the following analysis only focuses on the optimization of a generic non-trivial platoon (i.e., having two or more trajectories), for which we index vehicles with $\mathcal{M} = \{1, 2, \ldots, M\}$ without lose of generality.

The remainder of this section presents the simplified trajectory optimization model (STO) for vehicles in $\mathcal{M}$. Basically, STO restricts that each trajectory x_n has at most 5 consecutive quadratic sections, as illustrated in Fig. 3. Each section is with acceleration $-a^- \in [\underline{a}, 0]$, 0 or $a^+ \in [0, \bar{a}]$ where a^- and a^+ are the acceleration variables that determine the overall smoothness of the whole platoon. Let $t_{n1} \le t_{n2} \le t_{n3} \le t_{n4} \in \left[t_n^-, t_n^+\right]$ denote the joint time points between these sections. The first section of x_n during time interval $\left[t_n^-, t_{n1}\right]$ cruises at speed $\bar{v}$. The second section during time interval $(t_{n1}, t_{n2}]$ decelerates at a constant deceleration rate of $-a^-$. Note that the third section during time interval $(t_{n2}, t_{n3}]$ exists (i.e. $t_{n2} < t_{n3}$) only if x_n has to make a stop from time t_{n2} to t_{n3} (as illustrated in Fig. 3(a)). Otherwise, $t_{n2} = t_{n3}$, $\dot{x}_n(t) > 0, \forall t \in [t_n^-, t_n^+]$, and this third section does not exist (as illustrated in Fig. 3(b)). The fourth section during time $(t_{n3}, t_{n4}]$ accelerates at a constant acceleration rate of a^+. Note that these three intermediate sections form a reversed "S-shaped" transitional part that fits trajectory $x_n(t)$ for arrival time t_n^- and exit time t_n^+. Then the fifth section during time $\left(t_{n4}, t_n^+\right]$ cruises at speed $\bar{v}$ and reaches exit location L at exit time t_n^+.

Note that trajectory x_n in this piecewise quadratic form is determined by three variables, i.e., initial cruising time $\delta_{n1} := t_{n1} - t_n^-$ and deceleration magnitude a^- and acceleration magnitude a^+. Given these variables, each trajectory x_n can be formulated in the following way. For mathematical convenience, define $\Delta_n := t_n^+ - t_n^- - L/\bar{v}$ (note that Δ_n has to be positive or otherwise vehicle n itself is a trivial platoon), and $t_n^\Delta = t_n^+ - t_n^-$, as illustrated in Fig. 3. Note that t_n^Δ is the travel time for vehicle n on this segment, and Δ_n can be interpreted as the corresponding travel delay. Instead of investigating the two acceleration variables directly, we investigate two auxiliary variables $\phi := \frac{a^- a^+}{a^- + a^+}$ and $\lambda := a^-/(a^- + a^+)$, because these auxiliary variables much simplify the following formulations (which will be explained in a later remark). Although the physical meanings of these two auxiliary variables are not as intuitive, one can simply consider ϕ as an indication of the

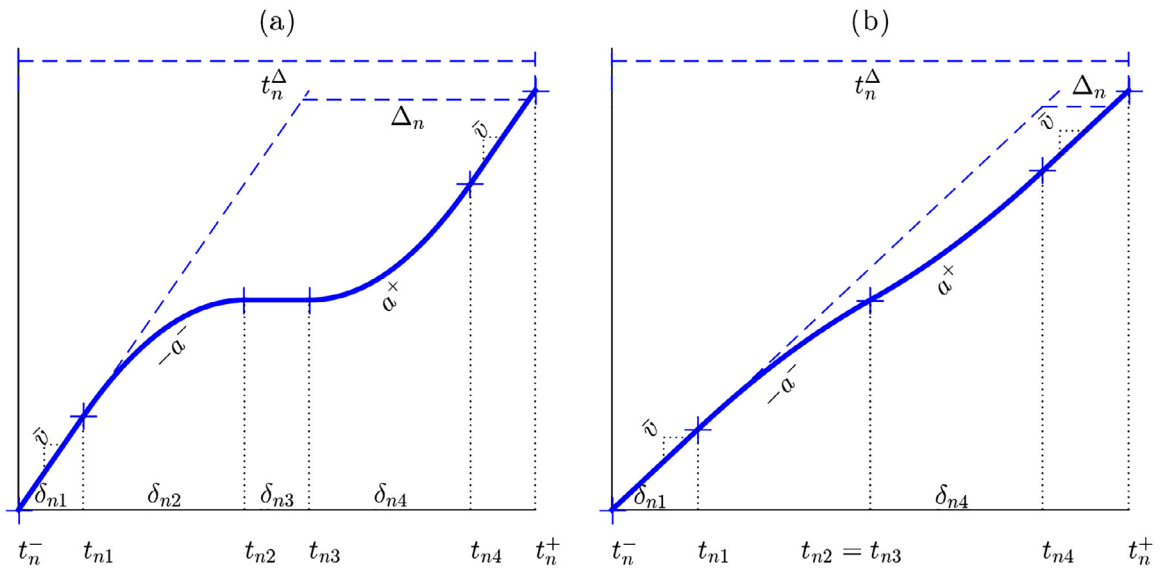

Fig. 3. Illustration of the piece-wise quadratic form of trajectory x_n (a) with a stop section and (b) without a stop section.

overall acceleration/deceleration magnitude, and λ as a weight of deceleration magnitude. Since $a^- \in [0, -\underline{a}]$ and $a^+ \in [0, \bar{a}]$, then ϕ and λ shall fall in the following ranges: $\phi \in [0, \bar{\phi}]$ where $\bar{\phi} := \frac{-\underline{a}\bar{a}}{\bar{a}-\underline{a}}$ and $\lambda \in [\frac{\phi}{\bar{a}}, 1 + \frac{\phi}{\underline{a}}]$. Note that for given values of (ϕ, λ) except for trivial singular points where $\lambda = 0$ or 1, the corresponding acceleration values can be obtained uniquely as $a^- = \phi/(1-\lambda)$ and $a^+ = \phi/\lambda$. With this, we can formulate the time joints with variables $\{\delta_{n1}\}, a^-, a^+$ as follows:

$$t_{n1} = t_n^- + \delta_{n1}, \tag{17}$$

$$t_{ni} = t_{n(i-1)} + \delta_{ni}(\phi, \lambda), \forall i = 2, 3, 4 \tag{18}$$

where

$$\delta_{n2}(\phi, \lambda) = \begin{cases} \sqrt{\frac{2\bar{v}\Delta_n}{\phi}}(1-\lambda), & \text{if } \phi \le \frac{\bar{v}}{2\Delta_n}; \\ \frac{\bar{v}(1-\lambda)}{\phi}, & \text{otherwise,} \end{cases}$$

$$\delta_{n3}(\phi) = \begin{cases} 0, & \text{if } \phi \le \frac{\bar{v}}{2\Delta_n}; \\ \Delta_n - \frac{\bar{v}}{2\phi}, & \text{otherwise,} \end{cases}$$

and

$$\delta_{n4}(\phi, \lambda) = \begin{cases} \sqrt{\frac{2\bar{v}\Delta_n}{\phi}}\lambda, & \text{if } \phi \le \frac{\bar{v}}{2\Delta_n}; \\ \frac{\bar{v}\lambda}{\phi}, & \text{otherwise.} \end{cases}.$$

Then we obtain the time duration for the transitional part as,

$$\delta(\phi, \Delta_n) := \sum_{i=2}^{4} \delta_{ni}(\phi, \lambda) = \begin{cases} \sqrt{\frac{2\bar{v}\Delta_n}{\phi}}, & \text{if } \phi \le \frac{\bar{v}}{2\Delta_n}; \\ \Delta_n + \frac{\bar{v}}{2\phi}, & \text{otherwise.} \end{cases}$$

Note that function $\delta(\phi, \Delta_n)$ is differentiable with respect to ϕ and Δ_n; i.e.,

$$\frac{d\delta(\phi, \Delta_n)}{\phi} = \max\left\{-\sqrt{\frac{\bar{v}\Delta_n}{2}}\phi^{-1.5}, -\frac{\bar{v}}{2}\phi^{-2}\right\}$$

which is negative, increasing with ϕ and decreasing with Δ_n, and

$$\frac{d\delta(\phi, \Delta_n)}{\Delta_n} = \max\left\{\sqrt{\frac{\bar{v}}{2\phi\Delta_n}}, 1\right\}$$

which is positive and decreasing with Δ_n and ϕ. These results also suggest that $\delta(\phi, \Delta_n)$ decreases with ϕ and increases with Δ_n.

Remark. Note that time duration $\delta(\phi, \Delta_n)$ is essentially determined by variable ϕ but independent of λ. Thus the effects of variables ϕ and λ on the shape of x_n are separated: ϕ decreases with the duration of the transitional part, and λ affects the skewness of the transitional part between acceleration and deceleration. This separation, which cannot be achieved by original variables a^- and a^+, much facilitates the following analysis. This is the reason why we use auxiliary variables ϕ and λ instead of original variables a^- and a^+.

Now the corresponding acceleration function can be formulated as

$$\ddot{x}_n(t) = \begin{cases} 0, & \text{if } t \in [t_n^-, t_{n1}]; \\ -\frac{\phi}{1-\lambda}, & \text{if } t \in (t_{n1}, t_{n2}]; \\ 0, & \text{if } t \in (t_{n2}, t_{n3}]; \\ \frac{\phi}{\lambda}, & \text{if } t \in (t_{n3}, t_{n4}]; \\ 0, & \text{if } t \in (t_{n4}, t_n^+]. \end{cases} \tag{19}$$

The corresponding speed function is

$$\dot{x}_n(t) = \begin{cases} \bar{v}, & \text{if } t \in [t_n^-, t_{n1}]; \\ \bar{v} - \frac{\phi}{1-\lambda}(t - t_{n1}) & \text{if } t \in \left(t_n 1, t_{n2}\right]; \\ 0 & \text{if } t \in \left(t_{n2}, t_n 3\right]; \\ \bar{v} + \frac{\phi}{\lambda}(t - t_{n4}), & \text{if } t \in (t_{n3}, t_{n4}]; \\ \bar{v}, & \text{if } t \in (t_{n4}, t_n^+]. \end{cases} \tag{20}$$

The corresponding location function is

$$x_n(t) = \begin{cases} \bar{v}(t - t_n^-), & \text{if } t \in [t_n^-, t_{n1}]; \\ \bar{v}(t - t_n^-) - 0.5\frac{\phi}{1-\lambda}(t - t_{n1})^2 & \text{if } t \in (t_{n1}, t_{n2}]; \\ \bar{v}(t_{n2} - t_n^-) - 0.5\frac{\phi}{1-\lambda}\delta_{n2}^2(\phi, \lambda) & \text{if } t \in \left(t_{n2}, t_n 3\right]; \\ \bar{v}(t - t_{n3} + t_{n2} - t_n^-) - 0.5\frac{\phi}{1-\lambda}\delta_{n2}^2(\phi, \lambda) - 0.5\frac{\phi}{\lambda}(t - t_{n4})^2, & \text{if } t \in (t_{n3}, t_{n4}]; \\ \bar{v}(t - t_{n3} + t_{n2} - t_n^-) - 0.5\lambda\frac{\phi}{1-\lambda}\delta_{n2}^2(\phi, \lambda) - 0.5\frac{\phi}{\lambda}\delta_{n4}^2(\phi, \lambda), & \text{if } t \in \left(t_n^4, t_n^+\right]. \end{cases} \tag{21}$$

With this simplification, cost function (12) can be rewritten into a closed-form expression without integrals as follows:

$$\bar{c}_n(\phi, \lambda) = \sum_{p=1}^{P} A_p F_n^{\mathrm{A}}(\phi, \lambda, p) + \sum_{q=1}^{Q} B_q F_n^{\mathrm{V}}(\phi, q) \tag{22}$$

where

$$F_n^{\mathrm{A}}(\phi, \lambda, p) := \int_{t_n^-}^{t_n^+} |\ddot{x}_n(t)|^p dt = \min\left(\sqrt{2\bar{v}\Delta_n\phi}, \bar{v}\right) \cdot \left(\lambda^{1-p} + (1-\lambda)^{1-p}\right)\phi^{p-1} \tag{23}$$

and

$$F_n^{\mathrm{V}}(\phi, q) := \int_{t_n^-}^{t_n^+} \dot{x}_n^q(t) dt = L\bar{v}^{q-1} - \begin{cases} (q-1)\Delta_n\bar{v}^q + \frac{\bar{v}^{q+1}}{q+1}\sum_{i=3}^{q+1}\binom{q+1}{i}\left(-\sqrt{\frac{2\Delta_n}{\bar{v}}}\right)^i \phi^{i/2-1}, & \text{if } \phi \le \frac{\bar{v}}{2\Delta_n}; \\ \left(\frac{1}{2} - \frac{1}{q+1}\right)\bar{v}^q\frac{1}{\phi}, & \text{otherwise.} \end{cases} \tag{24}$$

Note that after this simplification, cost function c_n is only dependent on ϕ and λ but independent of $\{\delta_{n1}\}$.

Next, we investigate how to simplify the corresponding constraints. We denote the five sections of x_n in the following form: initial cruising section $x_n\left(t_n^- : t_{n1}\right)$, deceleration section $x_n(t_{n1}: t_{n4})$, stopping section $x_n(t_{n2}: t_{n3})$, accelerating section $x_n(t_{n3}: t_{n4})$, and final cruising section $x_n\left(t_{n4} : t_n^+\right)$ (where operator : separates the starting and ending time of a trajectory section). The length of initial cruising section is simply

$$\left|x_n\left(t_n^- : t_{n1}\right)\right| = \bar{v}\delta_{n1}.$$

Length $|x_n(t_{n1}: t_{n4})|$ is a function ϕ and Δ_n as follows:

$$|x_n(t_{n1} : t_{n4})| = \begin{cases} \bar{v}\left(\frac{\sqrt{2\bar{v}\Delta_n}}{\sqrt{\phi}} - \Delta_n\right), & \text{if } \phi \le \frac{\bar{v}}{2\Delta_n}; \\ \frac{\bar{v}^2}{2\phi}, & \text{otherwise.} \end{cases}$$

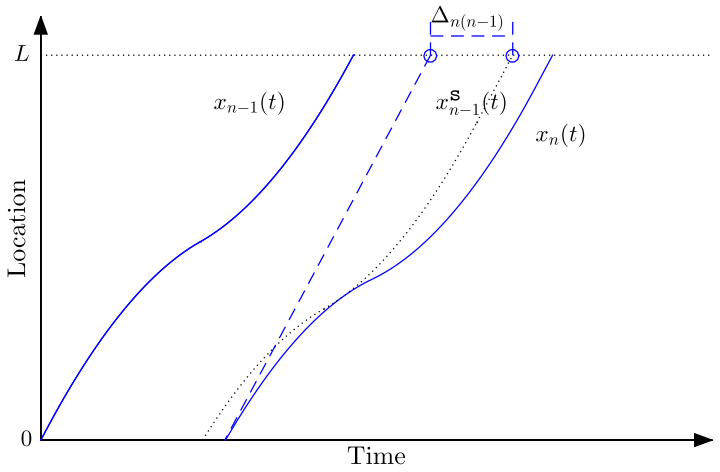

Fig. 4. Illustration of $\Delta_{n(n-1)}$.

Length $|x_n(t_{n4}\colon t_{n5})|$ is also a function of ϕ as follows,

$$|x_n(t_{n4}:t_{n5})| = L - \bar{v}\left(\delta_{n1} + \begin{cases} \frac{\sqrt{2\bar{v}\Delta_n}}{\sqrt{\phi}} - \Delta_n, & \text{if } \phi \le \frac{\bar{v}}{2\Delta_n}; \\ \frac{\bar{v}^2}{2\phi}, & \text{otherwise.} \end{cases}\right)$$

In order for x_n to satisfy constraints (1)–(6), we actually only need to impose $\left|x_n\left(t_n^- : t_{n1}\right)\right|, |x_n(t_{n4}:t_{n5})| \ge 0$, i.e.,

$$0 \le \delta_{n1} \le \tilde{\delta}_{n1}(\phi) := t_n^{\Delta} - \delta(\phi, \Delta_n). \tag{25}$$

Note that in order for $\tilde{\delta}_{n1}(\phi)$ to be no greater than 0 for all n, the value of ϕ should satisfy

$$\phi \ge \underline{\phi} := \begin{cases} \max_{n\in\mathcal{M}} \frac{2\bar{v}\Delta_n}{(L/\bar{v}+\Delta_n)^2}, & \text{if } \Delta_n \le \frac{2L}{\bar{v}}; \\ \frac{\bar{v}^2}{2L}, & \text{otherwise.} \end{cases} \tag{26}$$

where $\underline{\phi}$ can be taken as a lower bound for ϕ, which is tighter than 0.

Now we discuss how to select variables $a^-, a^+, \{\delta_{n1}\}_{\forall n}$ to comply with safety constraints (7). Define a shadow trajectory of $x_n(t)$ as

$$x_n^s(t) := x_n(t-\tau) - s.$$

Then safety constraints (7) essentially mean that x_n is always below or at maximum tangent to x^s_{n-1}. Note that this condition is equivalent to $x_n(t_{n1}\colon t_{n4})$ is always below or at maximum tangent to $x^s_{n-1}\left(t_{(n-1)1}+\tau : t_{(n-1)4}+\tau\right)$. We investigate the critical condition when $x_n(t_{n1}\colon t_{n4})$ gets tangent to $x^s_{n-1}\left(t_{(n-1)1}+\tau : t_{(n-1)4}+\tau\right)$, and we denote the corresponding critical δ_{n1} value as function $\hat{\delta}_{n1}\left(\delta_{(n-1)1}, \phi\right)$. For formulation convenience, define $\Delta_{(n-1)n} := t^+_{n-1} + \tau + s/\bar{v} - t^-_n - L/\bar{v}$ as illustrated in Fig. 4, which can be interpreted as the potential time headway conflict between vehicles n and $n-1$. Note that $\Delta_{(n-1)n} \le \Delta_{n-1}$ since $t^-_n \ge t^-_{n-1} + \tau + s/\bar{v}$, and $\Delta_{(n-1)n} \le \Delta_n$ since $t^+_{n-1} + \tau + s/\bar{v} \le t^+_n$. Then function $\hat{\delta}_{n1}\left(\delta_{(n-1)1}, \phi\right)$ can be formulated as

$$\hat{\delta}_{n1}\left(\delta_{(n-1)1}, \phi\right) := \delta_{(n-1)1} + \delta(\phi, \Delta_{n-1}) + t^-_{n-1} + \tau - t^-_n - \delta\left(\phi, \Delta_{n(n-1)}\right). \tag{27}$$

Then safety constraints (7) are essentially equivalent to $\delta_{n1} \le \hat{\delta}_{n1}\left(\delta_{(n-1)1}, \phi\right), \forall n \in \mathcal{M}\backslash\{1\}$. This together with Eq. (25) yields

$$0 \le \delta_{n1} \le \begin{cases} \tilde{\delta}_{n1}(\phi), & \text{if } n = 1; \\ \min\left\{\tilde{\delta}_{n1}(\phi), \hat{\delta}_{n1}\left(\delta_{(n-1)1}, \phi\right)\right\}, & \text{otherwise,} \end{cases} \forall n \in \mathcal{M}. \tag{28}$$

Now the simplified trajectory optimization model (STO) that this study investigates is formulated as

$$\text{STO:} \min_{\{\delta_{n1}\},\phi,\lambda} \bar{C}(\phi,\lambda) := \sum_{n=1}^{N} \bar{c}_n(\phi,\lambda)/N \tag{29}$$

where cost function $\bar{c}_n$ is defined in (22), subject to (28) and

$$\underline{\phi} \le \phi \le \bar{\phi}, \tag{30}$$

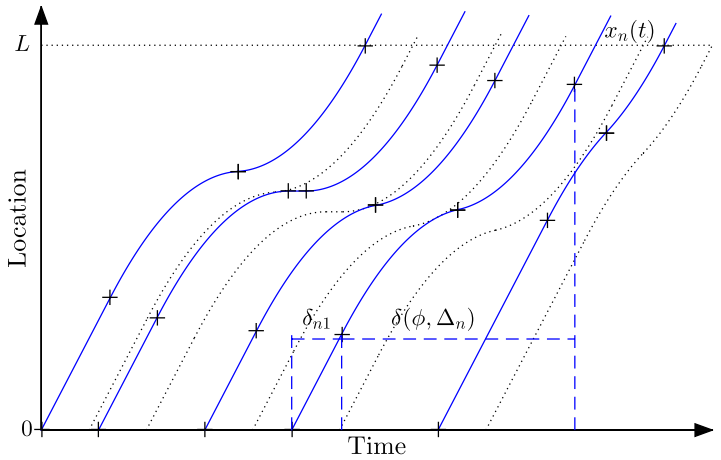

Fig. 5. Illustration of a feasible set of trajectories to Problem STO.

and

$$\frac{\phi}{\bar{a}} \le \lambda \le 1 + \frac{\phi}{\underline{a}}. \tag{31}$$

Note that Model STO only has $N+2$ independent variables, which is a dramatic simplification compared with PTO. Fig. 5 illustrates a set of feasible trajectories to STO.

3. Solution approach

This section analyzes the structure of STO and aims to find an exact solution approach to this problem. Section 3.1 investigates certain theoretical properties on how the variable values affect the STO objective and the constraints. Based on these theoretical results, Section 3.2 proposes an exact analytical algorithm to solve the optimal solution to STO.

3.1. Theoretical properties

Note that objective function (29) of STO is independent of the $\{\delta_{n1}\}$ values. Rather, the $\{\delta_{n1}\}$ values affect the feasible region of STO through constraints (28), (30) and (31). Therefore, to solve STO, we can first set $\{\delta_{n1}\}$ to values that are the least restrictive to the feasible region of variables ϕ and λ, which leads to the following proposition.

Proposition 1. *For given ϕ and λ values, if STO has at least one feasible solution to $\{\delta_{n1}\}$, then $\left\{\delta_{n1} = \delta^*_{n1}(\phi)\right\}$ must be feasible to STO as well, where*

$$\delta^*_{n1}(\phi) := \begin{cases} \bar{\delta}_{n1}(\phi), & \text{if } n = 1; \\ \min\left\{\bar{\delta}_{n1}(\phi), \hat{\delta}_{n1}\left(\delta^*_{(n-1)1}(\phi), \phi\right)\right\}, & \text{otherwise,} \end{cases} \forall n \in \mathcal{M}. \tag{32}$$

The proof is provided in Appendix B.

Proof. Note that in the above proposition, $\left\{\delta_{n1} = \delta^*_{n1}(\phi)\right\}$ essentially means that the transitional part of each trajectory is pushed downstream all the way until either $t_{n4} = t_n^+$ or safety constraint (7) is activated. The above proposition indicates that $\{\delta_{n1}\}$ values can be just fixed to $\left\{\delta^*_{n1}(\phi)\right\}$ without affecting the optimal objective of STO. Further, denote

$$\delta^*_1(\phi) := \min_{n \in \mathcal{M}} \{\delta^*_{n1}(\phi)\}. \tag{33}$$

Note that $\left\{\delta_{n1} = \delta^*_{n1}(\phi)\right\}$ are feasible to constraints (28) if and only if $\delta^*_1(\phi) \ge 0$. With this, STO essentially reduces to the following restricted STO (RSTO). □

$$\text{RSTO:} \quad \min_{\phi, \lambda} \bar{C}(\phi, \lambda) \tag{34}$$

subject to (30) and (31)

$$\delta^*_1(\phi) \ge 0. \tag{35}$$

Note that RSTO further reduces this problem to one with only two variables, ϕ and λ, which further simplifies the problem. The following analysis only investigates RSTO since RSTO's optimal solution also solves STO. Now we investigate the relationships between the cost objective function and the decision variables ϕ and λ.

Lemma 1. *For any $p \geq 1$, function $F_n^{\mathrm{A}}(\phi, \lambda, p)$ increases with $\phi > 0$. The proof is provided in Appendix B.*

Lemma 2. *For any $p \geq 1$, function $F_n^{\mathrm{A}}(\phi, \lambda, p)$ is symmetric with respect to $\lambda = 0.5$, decreasing with $\lambda \in (0, 0.5]$ and increasing with $\lambda \in [0.5, 1)$. The proof is provided in Appendix B.*

Lemma 3. *For $q \in \{1, 2, 3, 4\}$, function $F_n^{\mathrm{V}}(\phi, q)$ increases with $\phi > 0$. The proof is provided in Appendix B.*

These lemmas lead to the following relationship between the optimization objective and the decision variables.

Theorem 1. *Objective function $\bar{C}(\phi, \lambda)$ increases with $\phi > 0$, is symmetric with respect to $\lambda = 0.5$, decreases with $\lambda \in (0, 0.5]$ and increases with $\lambda \in [0.5, 1)$.*

The proof of this theorem directly follows Lemmas 1–3. With this property, the optimal solution to RSTO can be obtained as the following theorem states.

Theorem 2. *For RSTO, the optimal solution to ϕ, if existing, is*

$$\phi^* = \min_{\phi} \left\{ \phi \middle| \underline{\phi} \leq \phi \leq \bar{\phi}, \delta_1^*(\phi) \geq 0 \right\}, \tag{36}$$

and the optimal solution to λ, if existing, is

$$\lambda^* = \min\left(\max\left(0.5, \frac{\phi^*}{\bar{a}}\right), 1 + \frac{\phi^*}{\underline{a}} \right). \tag{37}$$

Proof. First, if ϕ^* exists and is given, based on the relationship between $\bar{C}(\phi^*, \lambda)$ and λ stated in Theorem 3 and constraints (31), it is easy to see that λ^* can be solved by (37). Note that with Eq. (37), as ϕ^* decreases, $\frac{\phi^*}{\bar{a}}$ shall decrease and $1 + \frac{\phi^*}{\underline{a}}$ shall increase. Therefore, as ϕ^* reduces, λ^* will be always feasible, and $|\lambda^* - 0.5|$ decreases as well. Therefore, based on Theorem 3, a further decrease of ϕ^*will not affect the feasibility of λ^* while improving the objective. Therefore, Eq. (36) holds. This completes the proof. □

The optimal solution stated in Theorem 2 can be intuitively interpreted as to stretch all trajectories as smooth as the transitional parts reach the upstream end of the investigated segment (or $\delta_1^*(\phi) = 0$) and the acceleration and deceleration magnitudes are maximally balanced (or λ^* gets as close to 0.5 as the feasibility allows). For most problem instances with realistic settings, the optimal acceleration and deceleration shall be mild and shall not activate their respective bounds. In this case, λ^* is just set to 0.5, indicating the same deceleration and acceleration magnitudes. Theorem 2 essentially further reduces RSTO into a one dimensional search problem where we only need to find the minimal $\phi \in \left[\underline{\phi}, \bar{\phi}\right]$ satisfying $\delta_1^*(\phi) \geq 0$. This result can be further narrowed as follows.

Corollary 1. *For RSTO, the optimal solution to ϕ, if existing, is*

$$\phi^* = \min_{\phi} \left\{ \phi \middle| \underline{\phi} \leq \phi \leq \bar{\phi}, \delta_1^*(\phi) = 0 \right\}. \tag{38}$$

The proof is provided in Appendix B.

3.2. Optimization algorithm

Theorem 2 and Corollary 1 indicate that the key to solving RSTO is finding the minimum feasible solution to $\delta_1^*(\phi) \geq 0$ in a finite range $\left[\underline{\phi}, \bar{\phi}\right]$. Although $\delta_1^*(\phi)$ may not have simple monotonicity that justifies a bisecting search algorithm, we notice that $\delta_1^*(\phi)$ is essentially a piece-wise quadratic function, and thus $\delta_1^*(\phi) = 0$ can be solved analytically at each piece with the following customized algorithm.

Define $\delta_{n1}^{\Delta}(\phi) := \tilde{\delta}_{n1}(\phi) - \hat{\delta}_{n1}\left(\delta_{(n-1)1}^*, \phi\right), \forall n \in \mathcal{M}$, and $\hat{\mathcal{M}}(\phi) := \{1\} \cup \left\{ n \middle| \delta_{n1}^{\Delta}(\phi) \leq 0, n \in \mathcal{M} \backslash \{1\} \right\}$. Note that $\delta_{n1}^{\Delta}(\phi) \leq 0$ actually indicates $t_{n4} = t_n^+$ while $\delta_{n1}^{\Delta}(\phi) > 0$ indicates $t_{n4} < t_n^+$. Index elements in $\hat{\mathcal{M}}(\phi)$ consecutively with $\left\{ \hat{n}_1(\phi) < \hat{n}_2(\phi) < \cdots < \hat{n}_{\hat{M}(\phi)}(\phi) \right\}$ where $\hat{M}(\phi) := \left| \hat{\mathcal{M}}(\phi) \right|$. For notation convenience, define $\hat{n}_{\hat{M}(\phi)+1}(\phi) = M + 1$. Note that vehicles in $\hat{\mathcal{M}}(\phi)$ are essentially the trajectories where safety constraints (7) are not activated for a given ϕ value. Then in the neighborhood of ϕ, $\delta_{\hat{n}_i(\phi)1}^*(\phi)$ does not depend on vehicle $\hat{n}_i(\phi) - 1$. Rather,$\delta_{n1}^*(\phi)$ depends on vehicles $\hat{n}_i(\phi), \hat{n}_i(\phi) + 1, \ldots, n - 1, \forall n \in [\hat{n}_i(\phi) + 1, \hat{n}_{i+1}(\phi) - 1]$. Therefore, each vehicle in $\hat{\mathcal{M}}(\phi)$ can be regarded as a lead vehicle. Specifically, we call $\hat{n}_i(\phi)$ the lead vehicle for all $n = \hat{n}_i(\phi), \hat{n}_i(\phi) + 1, \ldots, \hat{n}_{i+1}(\phi) - 1$, and denote this as

$$\hat{n}(n, \phi) := \hat{n}_i(\phi), \forall n = \hat{n}_i(\phi), \hat{n}_i(\phi) + 1, \ldots, \hat{n}_{i+1}(\phi) - 1.$$

Lead vehicles $\left\{ \hat{n}_i(\phi) \right\}$ are illustrated in Fig. 6. For a given $\phi > 0$, $\hat{\mathcal{M}}(\phi)$ can be easily identified with the following iterative algorithm (IA).

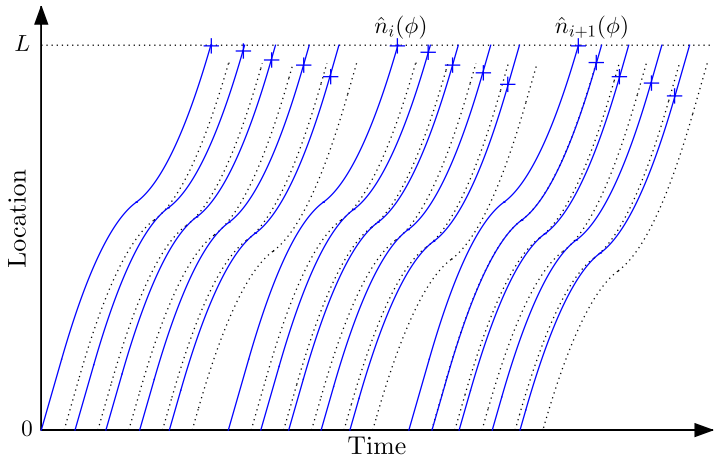

Fig. 6. Illustration of lead vehicles $\{\hat{n}_i(\phi)\}$, where each cross marks the time of t_{n4}.

IA-1: Set $\hat{n}_1(\phi) = \hat{n}(1,\phi) = 1$, $i = 2$ and $n = 2$.

IA-2: This step verifies whether $\hat{n}(n,\phi) = \hat{n}(n-1,\phi)$ holds. If it holds, then $\delta^{\Delta}_{n1}(\phi) = \hat{\delta}^{\Delta}_{n1}(\phi)$ as defined below.

$$\hat{\delta}^{\Delta}_{n1}(\phi) := t^{+}_{n} - \big(n - \hat{n}(n-1,\phi)\big)\tau - t^{+}_{n-1} + \sum_{n'=\hat{n}(n-1,\phi)+1}^{n} \Big(\delta\big(\phi, \Delta_{n'(n'-1)}\big) - \delta(\phi, \Delta_{n'})\Big) \tag{39}$$

Note that calculation of $\hat{\delta}^{\Delta}_{n1}(\phi)$ can be further expedited since $\hat{\delta}^{\Delta}_{n1}(\phi) = \hat{\delta}^{\Delta}_{(n-1)1}(\phi) + \delta(\phi, \Delta_{n-1}) - \delta\big(\phi, \Delta_{n(n-1)}\big)$.

IA-3: If $\hat{\delta}^{\Delta}_{n1}(\phi) > 0$, the above assumption holds, and then set $\hat{n}(n,\phi) = \hat{n}(n-1,\phi)$. Increase $n = n+1$ and go to Step IA-2. Otherwise, go to the next step.

IA-4: Set $\hat{n}(n,\phi) = n$ and $\hat{n}_i(\phi) = n$. If $n < M$, increase $i = i+1$, $n = n+1$, and go to Step IA-2. Otherwise, go to the next step.

IA-5: Return $\hat{\mathcal{M}}(\phi) = \{\hat{n}_1(\phi), \hat{n}_2(\phi), \ldots, \hat{n}_i(\phi)\}$.

Note that the computational complexity of the IA algorithm is $o(M)$. Once $\hat{\mathcal{M}}(\phi)$ is obtained, $\hat{\delta}_{n1}\Big(\delta^{*}_{(n-1)1}, \phi\Big)$ can be denoted as a closed form function $\hat{\delta}^{\text{lead}}_{n1}\big(\hat{n}(n,\phi^*), \phi\big)$ as defined below

$$\hat{\delta}^{\text{lead}}_{n1}\big(\hat{n}(n,\phi^*), \phi\big) := t^{+}_{\hat{n}(n,\phi^*)} + \big(n - \hat{n}(n,\phi^*)\big)\tau - t^{-}_{n} - \delta\big(\phi, \Delta_{n(n-1)}\big) \tag{40}$$

$$+ \sum_{n'=\hat{n}(n,\phi^*)+1}^{n-1} \Big[\delta(\phi, \Delta_{n'}) - \delta\big(\phi, \Delta_{n'(n'-1)}\big)\Big]. \tag{41}$$

Then based on Corollary 1, if there exists an optimal solution ϕ^*, it should be the minimum value satisfying the following conditions

$$\hat{\delta}^{\text{lead}}_{n1}\big(\hat{n}(n,\phi^*), \phi^*\big) \geq 0, \forall n \in \mathcal{M} \backslash \hat{\mathcal{M}}(\phi^*), \tag{42}$$

and constraint (30) for the feasible range of ϕ.

Then the sketch of the exact solution algorithm is to first identify $\hat{\mathcal{M}}(\phi^*)$ and then solve the above equations by analytically solving a set of piecewise quadratic functions. We first analyze the property of $\hat{\mathcal{M}}(\phi)$.

Proposition 2. *$\hat{\mathcal{M}}(\phi) = \{1\}$ as $\phi \to \infty$ and $\hat{\mathcal{M}}(\phi_1) \supseteq \hat{\mathcal{M}}(\phi_2), \forall \phi_1 < \phi_2 \in \mathbb{R}^{+}$. The proof is provided in Appendix B.*

Corollary 2. *$\delta^{\Delta}_{n1}(\phi)$ increases with $\phi > 0$. If $\delta^{\Delta}_{n1}(\phi^-) \leq 0$ for some $\phi^- > 0$, $\delta^{\Delta}_{n1}(\phi)$ strictly increases with ϕ from ϕ^- to some ϕ^+ with $\delta^{\Delta}_{n1}(\phi^+) > 0$. In this case, $\delta^{\Delta}_{n1}(\phi) = 0$ has a unique solution.*

We omit the proof to this corollary because this property is apparent following Proposition 2 and the formulation structure of $\delta^{\Delta}_{n1}(\phi)$. The above analysis indicates that as ϕ increases, the elements of $\hat{\mathcal{M}}(\phi)$ will only drop out but never grow. We can use the following algorithm to evaluate which index of $\hat{\mathcal{M}}(\phi)$ will first drop out as ϕ increases from a given value $\hat{\phi}$.

DROP-1: Given $\hat{\phi}$ and $\hat{\mathcal{M}}(\hat{\phi})$ (which could be obtained with the IA algorithm), set $i = 2$.

DROP-2: Then this algorithm checks at which ϕ value as $\hat{\phi}$ increases to, $\hat{n}_i(\hat{\phi})$ will be dropped out from $\hat{\mathcal{M}}(\hat{\phi})$, i.e., solving $\hat{\delta}^{\Delta}_{\hat{n}_i(\hat{\phi})1}(\phi) = 0$. Based on Corollary 2, since $\hat{\delta}^{\Delta}_{\hat{n}_i(\hat{\phi})1}(\hat{\phi}) < 0$, $\hat{\delta}^{\Delta}_{\hat{n}_i(\hat{\phi})1}(\phi) = 0$ has a unique solution in $[\hat{\phi}, \infty)$. For mathematical convenience, we equivalently investigate $F_{\hat{\phi}i}(\sqrt{\phi}) := \hat{\delta}^{\Delta}_{\hat{n}_i(\hat{\phi})1}(\phi) \cdot \phi = 0$. Note that $F_{\hat{\phi}i}(\sqrt{\phi})$ is essentially a piecewise quadratic function of $\sqrt{\phi}$, and its joint points between consecutive pieces can be obtained in the following way. Define

$$\phi_n^{\text{crit1}} := \frac{\bar{v}}{2\Delta_n}, \phi_n^{\text{crit2}} := \frac{\bar{v}}{2\Delta_{n(n-1)}}, \forall n. \tag{43}$$

Then define $\Phi_i^{\text{crit}}(\hat{\phi}) = \{\hat{\phi}, \infty\} \cup \{\phi_n^{\text{crit1}}, \phi_n^{\text{crit2}}\}_{n=\hat{n}_{i-1}(\hat{\phi}),\ldots,\hat{n}_i(\hat{\phi})-1}$. Then delete all elements in $\Phi_i^{\text{crit}}(\hat{\phi})$ less than $\hat{\phi}$ and then sort these elements in an ascending order. Denote the sorted elements as $\Phi_i^{\text{crit}}(\hat{\phi}) = [\phi_{i1}^{\text{crit}}, \phi_{i2}^{\text{crit}}, \ldots, \phi_{iK_i}^{\text{crit}}]$ where $K_i := |\Phi_i^{\text{crit}}(\hat{\phi})|$. Now $\Phi_i^{\text{crit}}(\hat{\phi})$ contains all the joints between consecutive pieces of $F_{\hat{\phi}i}(\sqrt{\phi})$. Then we iterate all these pieces, starting with $k = 1$.

DROP-3: This step makes a guess that the solution to $F_{\hat{\phi}i}(\sqrt{\phi}) = 0$ falls in $[\phi_{ik}^{\text{crit}}, \phi_{i(k+1)}^{\text{crit}}]$. Based on the definition, we know when $\phi \in [\phi_{ik}^{\text{crit}}, \phi_{i(k+1)}^{\text{crit}}]$, $F_{\hat{\phi}i}(\sqrt{\phi})$ is a quadratic function in the form of $A\phi + B\sqrt{\phi} + C = 0$, and coefficients A, B, C can be obtained in the following way. Initially, set $A = t^+_{\hat{n}_i(\hat{\phi})} - \left(\hat{n}_i(\hat{\phi}) - \hat{n}_{i-1}(\hat{\phi})\right)\tau - t^+_{\hat{n}_{i-1}(\hat{\phi})}$, $B = 0, C = 0$. Set $n = \hat{n}_{i-1}(\hat{\phi}) + 1$.

DROP-4: If $\phi_{i(k+1)}^{\text{crit}} \le \frac{\bar{v}}{2\Delta_n}$, update $B = B - \sqrt{2\Delta_n \bar{v}}$; otherwise, update $C = C - \bar{v}/2$, $A = A - \Delta_n$. If $\phi_{i(k+1)}^{\text{crit}} \le \frac{\bar{v}}{2\Delta_{n(n-1)}}$, update $B = B + \sqrt{2\Delta_{n(n-1)}\bar{v}}$; otherwise, update $C = C + \bar{v}/2$, $A = A + \Delta_{n(n-1)}$. If $n < \hat{n}_i(\hat{\phi})$, update $n = n + 1$ and repeat this step; otherwise, go to the next step.

DROP-5: Solve the roots to $A\phi + B\sqrt{\phi} + C = 0$. There should be no more than one solution to ϕ falling in $[\phi_{ik}^{\text{crit}}, \phi_{i(k+1)}^{\text{crit}}]$. If such a solution exists, then the guess at Step DROP-3 is correct, and we record it as $\phi^{\text{next}}_{\hat{n}_i(\hat{\phi})}(\hat{\phi})$ and go to the next step; otherwise, update $k = k + 1$, and go to Step DROP-3.

DROP-6: If $i < \hat{M}(\hat{\phi})$, update $i = i + 1$ and go to Step DROP-2. Otherwise, go to the next step.

DROP-7: Solve $n^{\text{next}}(\hat{\phi}) = \arg\min_{n \in \hat{\mathcal{M}}(\hat{\phi}) \setminus \{1\}} \phi_n^{\text{next}}(\hat{\phi})$. Then $n^{\text{next}}(\hat{\phi})$ will be the first element to be dropped from $\hat{\mathcal{M}}(\phi)$ as ϕ increases from $\hat{\phi}$ to $\phi^{\text{next}}_{n^{\text{next}}(\hat{\phi})}(\hat{\phi})$. Return $n^{\text{next}}(\hat{\phi})$ and $\phi^{\text{next}}_{n^{\text{next}}(\hat{\phi})}(\hat{\phi})$.

Note that the computational complexity of the DROP algorithm is $o(M^2)$. The DROP algorithm can help identify an interval $\left[\hat{\phi}, \phi^{\text{next}}_{n^{\text{next}}(\hat{\phi})}(\hat{\phi})\right]$ where $\hat{\mathcal{M}}(\phi)$ can be treated as the same $\hat{\mathcal{M}}(\hat{\phi})$[4]. We call such an interval a stationary interval. With this, we will search stationary intervals consecutively in an ascending order between $\underline{\phi}$ and $\bar{\phi}$. In each stationary interval, we try to find the minimum ϕ that satisfies conditions (42). This process is described in the following piece-wise search algorithm (PSA).

PSA-1: Initially, set $\phi^- = \underline{\phi}$, call the DROP algorithm to solve $\phi^+ = \phi^{\text{next}}_{n^{\text{next}}(\phi^-)}(\phi^-)$.

PSA-2: Call the IA algorithm to solve $\hat{\mathcal{M}}(\phi^-)$. Then this algorithm tries to find the region for constraints (42) to be feasible within $[\phi^-, \phi^+]$. We initially set the candidate feasible region $\mathcal{R}^{\text{feas}} = [\phi^-, \phi^+]$, then we narrow it down by iteratively checking feasible regions for all n in $\mathcal{M} \setminus \hat{\mathcal{M}}(\phi^-)$. We start the feasibility checking from the first index in $\mathcal{M} \setminus \hat{\mathcal{M}}(\phi^-)$, denoted by n.

PSA-2: Initially set the feasibility region $\mathcal{R}_n^{\text{feas}} = \emptyset$. Note that $\forall \phi \in [\phi^-, \phi^+]$, $\hat{\delta}^{\text{lead}}_{n1}(\hat{n}(n,\phi),\phi) = \hat{\delta}^{\text{lead}}_{n1}(\hat{n}(n,\phi^-),\phi)$. Then for mathematical convenience, we define $G_{n\phi^-}(\sqrt{\phi}) := \phi \cdot \hat{\delta}^{\text{lead}}_{n1}(\hat{n}(n,\phi^-),\phi)$, which is a piecewise quadratic function of $\sqrt{\phi}$. Similar to Step DROP-2, we will first identify the joints between consecutive pieces. Define $\Phi_n^{\text{crit}} = \{\phi^-, \phi^+\} \cup \{\phi_{n'}^{\text{crit1}}, \phi_{n'}^{\text{crit2}}\}_{n'=\hat{n}(n,\phi^-)+1,\ldots n}$ and drop all elements less than ϕ^- or greater than ϕ^+ from Φ_n^{crit}, where $\phi_{n'}^{\text{crit1}}, \phi_{n'}^{\text{crit2}}$ are defined in Eq. (43). Then sort all elements in Φ_n^{crit} in an ascending order, and denote them as $\Phi_n^{\text{crit}} = [\phi_{n1}^{\text{crit}}, \phi_{n2}^{\text{crit}}, \ldots, \phi_{nK_n}^{\text{crit}}]$

[4] Note that strictly speaking, $\hat{\mathcal{M}}(\phi^{\text{next}}_{n^{\text{next}}(\hat{\phi})}(\hat{\phi}))$ compared with $\hat{\mathcal{M}}(\hat{\phi})$ has one more element, $n^{\text{next}}(\hat{\phi})$, but deleting $n^{\text{next}}(\hat{\phi})$ does not change the validity of all proposed equations.

where $K_n = |\Phi_n^{\text{crit}}|$. Now Φ_n^{crit} contains all the joints between consecutive pieces of $G_{n\phi^-}\left(\sqrt{\phi}\right)$ during $[\phi^-,\phi^+]$ in an ascending order. Then will iterate through all these pieces, starting with $k = 1$.

PSA-3: This step will find the feasible region that $G_{n\phi^-}\left(\sqrt{\phi}\right)\geq 0$ (or $\hat{\delta}_{n1}^{\text{lead}}(\hat{n}(n,\phi^-),\phi)\geq 0$) during interval $\left[\phi_{nk}^{\text{crit}},\phi_{n(k+1)}^{\text{crit}}\right]$, where $G_{n\phi^-}\left(\sqrt{\phi}\right)$ is a quadratic function in the form of, where coefficients A, B, C can be obtained in the following iterative approach. Initially, set $A = t^+_{\hat{n}(n,\phi^-)} + (n-\hat{n}(n,\phi^-))\tau - t_n^-$, $B = 0$ and $C = 0$, then update these coefficients according to the following pseudo code:

For $n' = \hat{n}(n,\phi^-)+1$ to $n-1$
  If $\phi_{n(k+1)}^{\text{crit}} \leq \frac{\bar{v}}{2\Delta_{n'}}$
    Update $B = B + \sqrt{2\bar{v}\Delta_{n'}}$.
  Else
    Update $A = A + \Delta_{n'}$, $C = C + \bar{v}/2$.
For $n' = \hat{n}(n,\phi^-)+1$ to n
  If $\phi_{n(k+1)}^{\text{crit}} \leq \frac{\bar{v}}{2\Delta_{n'(n'-1)}}$
    Update $B = B - \sqrt{2\bar{v}\Delta_{n'(n'-1)}}$.
  Else
    Update $A = A - \Delta_{n'(n'-1)}$, $C = C - \bar{v}/2$.

PSA-4: It is easy to solve the subset $\mathcal{R}_{nk}^{\text{feas}}$ in $\left[\phi_{nk}^{\text{crit}},\phi_{n(k+1)}^{\text{crit}}\right]$ such that $G_{n\phi^-}\left(\sqrt{\phi}\right) = A\phi + B\sqrt{\phi} + C \geq 0$ with the following pseudo code:

If $A = 0$
  If $B = 0$
    If $C \geq 0$, return $\mathcal{R}_{nk}^{\text{feas}} = \left[\phi_{nk}^{\text{crit}},\phi_{n(k+1)}^{\text{crit}}\right]$.
      Else, return $\mathcal{R}_{nk}^{\text{feas}} = \emptyset$.
    Else if $B > 0$, return $\mathcal{R}_{nk}^{\text{feas}} = \left[\max\left\{\phi_{nk}^{\text{crit}}, \frac{C^2}{B^2}\right\},\phi_{n(k+1)}^{\text{crit}}\right]$.
    Else, return $\mathcal{R}_{nk}^{\text{feas}} = \left[\phi_{nk}^{\text{crit}},\min\left\{\phi_{n(k+1)}^{\text{crit}}, \frac{C^2}{B^2}\right\}\right]$.
  Else
    If $B^2 - 4AC < 0$, return $\mathcal{R}_{nk}^{\text{feas}} = \left[\phi_{nk}^{\text{crit}},\phi_{n(k+1)}^{\text{crit}}\right]$ if $A > 0$ or return $\mathcal{R}_{nk}^{\text{feas}} = \emptyset$ otherwise.
    Solve $\phi_{nk}^1 = \left(\frac{-B-\sqrt{B^2-4AC}}{2A}\right)^2, \phi_{nk}^2 = \left(\frac{-B+\sqrt{B^2-4AC}}{2A}\right)^2$.
    If $A > 0$, return $\mathcal{R}_{nk}^{\text{feas}} = \left[\phi_{nk}^{\text{crit}},\phi_{n(k+1)}^{\text{crit}}\right] \cap \{(-\infty,\min\{\phi_{nk}^1,\phi_{nk}^2\}] \cup [\max\{\phi_{nk}^1,\phi_{nk}^2\},\infty)\}$.
    Else, return $\mathcal{R}_{nk}^{\text{feas}} = \left[\phi_{nk}^{\text{crit}},\phi_{n(k+1)}^{\text{crit}}\right] \cap [\min\{\phi_{nk}^1,\phi_{nk}^2\},\max\{\phi_{nk}^1,\phi_{nk}^2\}]$.

Then update $\mathcal{R}_n^{\text{feas}} = \mathcal{R}_n^{\text{feas}} \cup \mathcal{R}_{nk}^{\text{feas}}$. If $k+1 < K_n$, update $k = k+1$ and go to Step PSA-3 to check the next piece. Otherwise, $\mathcal{R}_n^{\text{feas}}$ is all the feasible region for $G_{n\phi^-}\left(\sqrt{\phi}\right)\geq 0$ in $[\phi^-,\phi^+]$, and we go to the next step.

PSA-5: Set $\mathcal{R}^{\text{feas}} = \mathcal{R}^{\text{feas}} \cap \mathcal{R}_n^{\text{feas}}$. If n is not the last element in $\mathcal{M}\backslash\hat{\mathcal{M}}(\phi^-)$, update n to be the next element in $\mathcal{M}\backslash\hat{\mathcal{M}}(\phi^-)$ and go to Step PSA-2 to find the feasible region for this new vehicle. Otherwise, $\mathcal{R}^{\text{feas}}$ is exactly the feasible region in $[\phi^-,\phi^+]$ such that Constraints (42) holds, and then go to the next step.

PSA-6: If $\mathcal{R}^{\text{feas}} \neq \emptyset$, then the optimal solution is found and return $\phi^* = \min\{\phi \in \mathcal{R}^{\text{feas}}\}$. Otherwise, if $\hat{\mathcal{M}}(\phi^-) = \{1\}$, then there is no feasible solution. Otherwise, $|\hat{\mathcal{M}}(\phi^-)| > 1$, and we set $\phi^- = \phi^+$ and go to Step PSA-2.

The PSA yields the exact optimal solution ϕ^* to problem RSTO. Note that the computational complexity of the PSA is $o(M^3)$. This is because each vehicle n needs to check no more than M pieces, and at each piece, it takes no more than M steps to solve the feasible region for each $G_{n\phi^-}\left(\sqrt{\phi}\right)\geq 0, \forall n \in \mathcal{M}\backslash\hat{\mathcal{M}}(\phi^-)$. Note that the most complex operation is just to solve a quadratic equality. It is expected that the PSA can be very efficiently solved with modern computers.

4. Numerical examples

This section conducts numerical examples to test the solution efficiency of the proposed algorithm and the application of this trajectory optimization model. Section 4.1 reports the solution times of the PSA for different instances and concludes that this proposed algorithm has appealing computational efficiency for real-time applications. The proposed trajectory optimization model can actually be applied to a general highway segment under different control strategies. For illustration purposes, Sections 4.2 and 4.3 show its applications for signalized segments and non-stop intersections, respectively. Section 4.4 conducts a numerical experiment to demonstrate the applicability of the algorithm in multi-lane highway conditions. Finally, Section 4.5 demonstrates the robustness of the proposed trajectory optimization model to stochastic estimation errors.

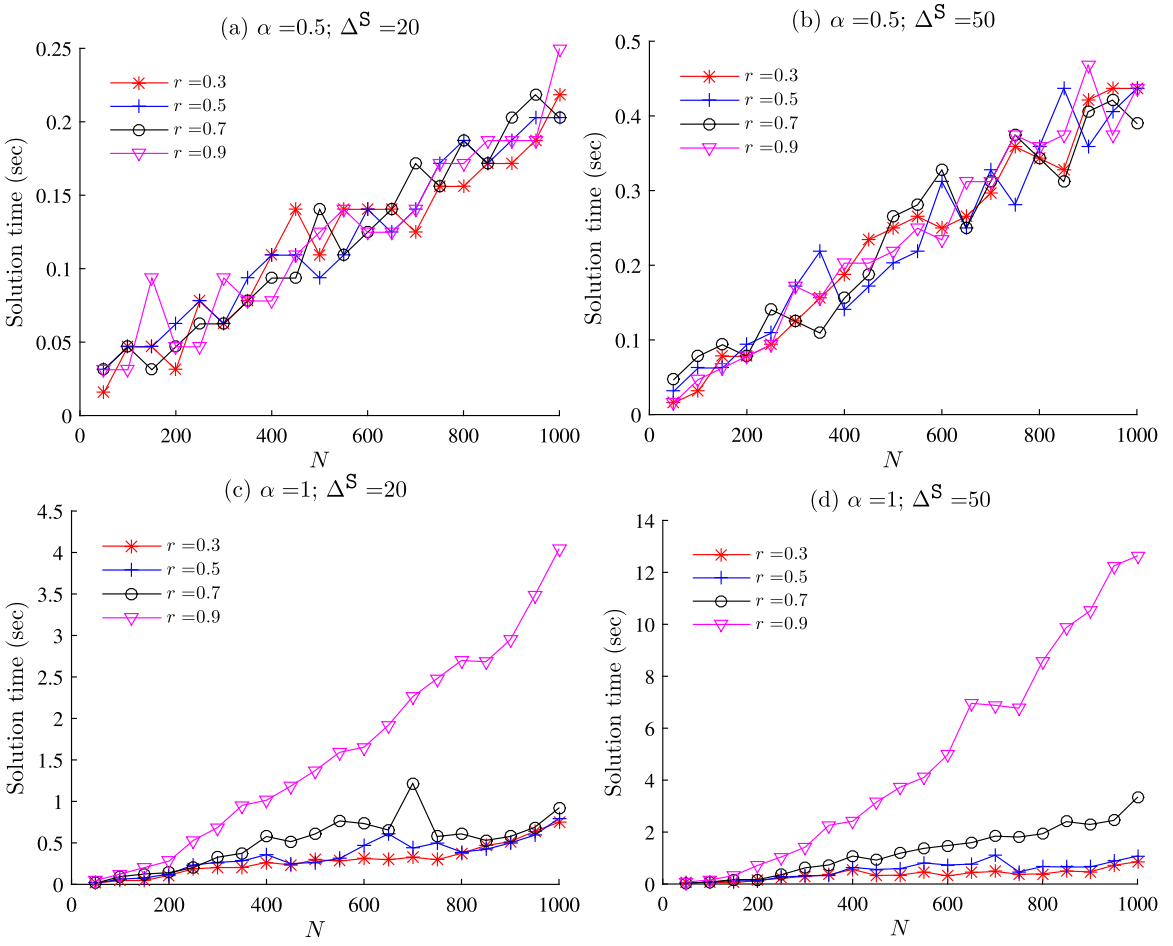

Fig. 7. Solution times for different problem instances.

4.1. Algorithm performance

The computation experiments are conducted on a PC with 2.6 GHz CPU and 16 GB RAM. The parameters are set in the following way. The vehicle arrival times are generated as

$$t_n^- = \begin{cases} 0, & \text{if } n = 1; \\ t_{n-1}^- + (\tau + s/\bar{v})\left(1 + \left(\frac{1}{r} - 1\right)[(1-\alpha) + \alpha\xi_n]\right), & \text{otherwise,} \end{cases} \tag{44}$$

where $r \in (0, 1]$ indicates traffic saturation rate, parameter $\alpha \in [0, 1]$ controls the dispersion of arrival time headway (greater α indicates higher dispersion), ξ_n is an uniformly distributed random number over [0,2], and ξ_n values are independent across different n values. We generate arrival times in this way so that they are feasible to Eq. (14) and we can control traffic volume with r and arrival randomness with α. Similarly, we set vehicle departure times as

$$t_n^+ = \begin{cases} L/\bar{v} + \Delta^S, & \text{if } n = 1; \\ \max\left\{t_{n-1}^+ + (\tau + s/\bar{v})\left(1 + \left(\frac{1}{r} - 1\right)\left[(1-\alpha) + \alpha\xi_n'\right]\right), t_n^- + L/\bar{v}\right\}, & \text{otherwise,} \end{cases}$$

where $\Delta^S \geq 0$ is a time shift (e.g., due to a downstream bottleneck) and ξ_n' again is a uniformly and independently distributed random number over [0,2]. This formulation ensures feasible conditions (15) and (16) hold. In the experiments, we purposefully set L to an large value, 8000 m. Although this value may not be realistic for all applications, it ensures that each tested problem instance is feasible (i.e., there exist solutions not causing queue spillback) even for large N values. The examples in this section set $\underline{a} = -3.5\,\text{m/s}^2$, $\bar{a} = 2\,\text{m/s}^2$, $\bar{v} = 16\,\text{m/s}$ ($\approx 35\,\text{mph}$), $s = 7\,\text{m}$ and $\tau = 1.5\,\text{s}$. To test instances of different input sizes, we vary N between 50 and 1000. Further, we try $r \in \{0.3, 0.5, 0.7, 0.9\}$, $\alpha \in \{0.5, 1\}$ and $\Delta^S \in \{20, 50\}$ to create different scenarios. The solution times of all these instances are reported in Fig. 7. We see that the solution times are less than 3 s for most instances and less than 10 s for all these instances, which are suitable for real-time engineering practices where a few hundred vehicles would take tens of minutes to arrive. Overall, as the increase of r, α or Δ^S, the solution

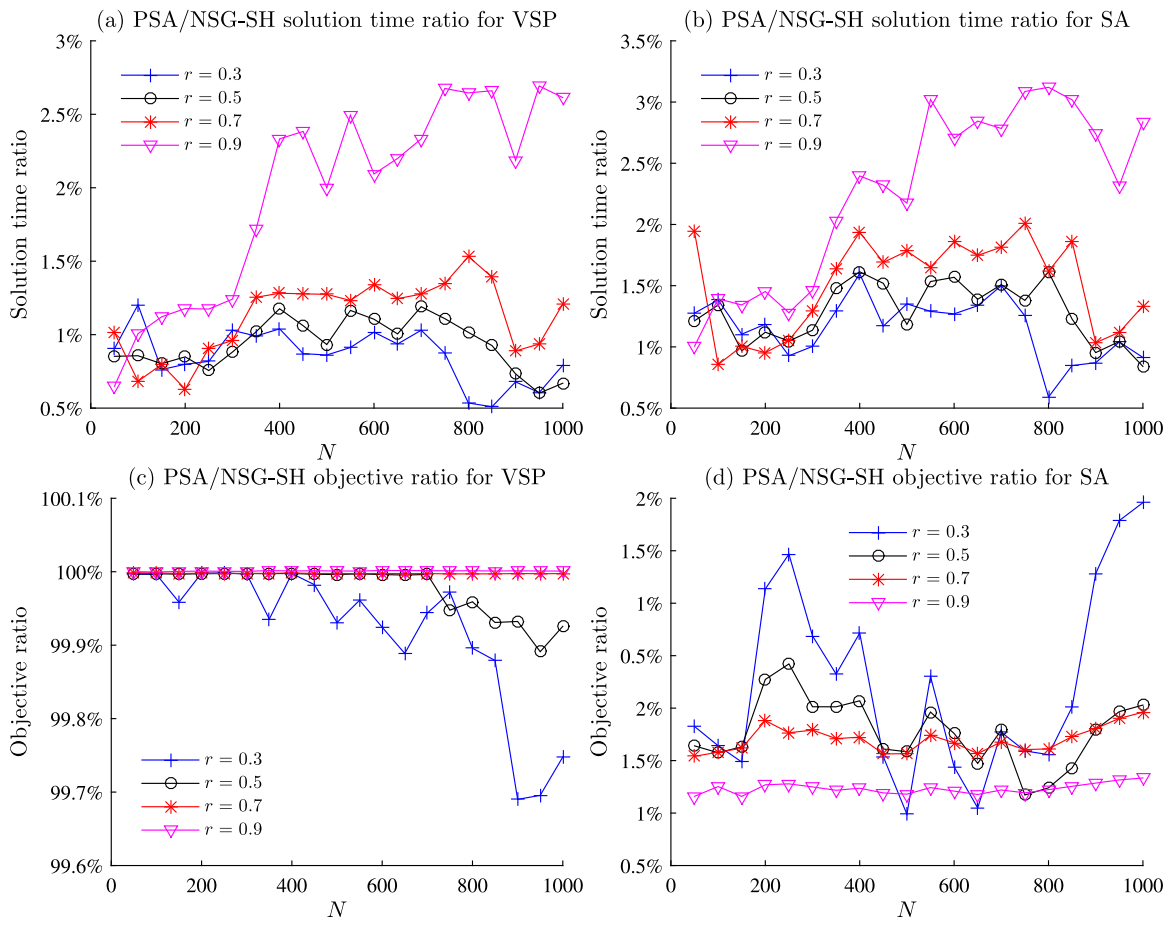

Fig. 8. Performances of PSA and NSG-SH.

time in general increases. This is because when these variable increases, the average size of a platoon shall increase and the interactions between consecutive trajectories become stronger. Therefore, the PSA likely needs to check more pieces and thus the solution time generally increases. For most instances, the solution time increases almost linearly with the instance size (or the N value). For some instances with relatively large r and α, the solution time increase exhibits a super-linear trend when N gets large. This is probably due to the increased interactions between consecutive trajectories as mentioned above. Yet the super-linear increasing trend looks less than that of a cubic function, and thus the actually solution times are likely less than the theoretical cubic time complexity bound as discussed in the end of Section 3.2.

To further investigate the performance of PSA, the following analysis compares the results from PSA and those from the numerical sub-gradient algorithm (NSG) with an slightly adapted shooting heuristic (SH)[5] proposed by Ma et al. (2017). Basically, compared with the proposed trajectory construction model (21), a feasible SH solution also contains piece-wise quadratic trajectories, which however may have more than five pieces. Compared with the analytical PSA approach, the NSG-SH approach contains more acceleration variables, but it is numerical and may not guarantee to find the optimal solution. For the algorithm detail, please refer to Ma et al. (2017). Fig. 8 shows the comparison results with the same parameter values as Fig. 7 (d). Fig. 8 (a) and (b) show the ratio of the PSA solution time over the NSG-SH solution time for the VSP and SA objectives, respectively, as N increases at different r values. We can see that for all instances, the PSA solution time is less than 3.5% of the NSG-SH solution time. This ratio generally drops as r decreases. This verifies that the analytical PSA algorithm is much more efficient than the numerical NSG-SH algorithm, though the latter's solution time is already reasonable for practical applications. Fig. 8 (c) and (d) show the ratio of the PSA objective value over the NSG-SH objective value for VSP and SA, respectively. We see that for VSP, both PSA and NSG-SH have very close objective values while most PSA objectives are slightly less than the NSG-SH objectives. Whereas for SA, the PSA objectives are much better than the NSG-SH objectives: all PSA objectives are less than 2% of their NSG-SH counterparts, and the ratio in general further drops

[5] We made the following minor adaptations to SH to fit our problem. The backward shooting process for each vehicle n now starts at its fixed exit time t_n^- instead of being regulated by signal timing. The objective function is VSP (10) or SA (11). The speed variable v is fixed to speed limit $\bar{v}$.

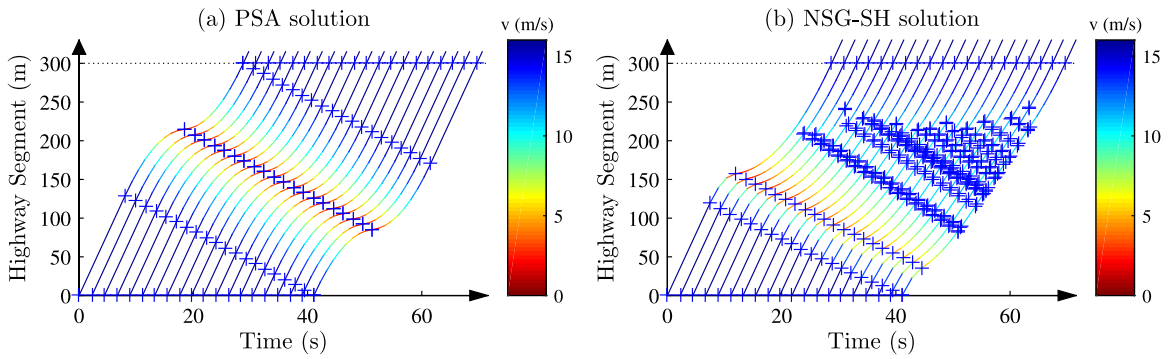

Fig. 9. Trajectories produced by PSA and NSG-SH (the crosses separate quadratic pieces).

as r increases. These results indicate that the PSA solutions, though generated from a more restrictive model with fewer quadratic pieces and less variables, are no worse than those from the NSG-SH objectives. Overall, we see that compared with the previously proposed NSG-SH approach, the PSA approach much improves the solution efficiency without compromising the solution quality (actually the solution quality is improved for most instances).

Next we compare the shapes of trajectories produced by PSA and NSG-SH. For clarity of the plot, we investigate a shorter segment with less vehicles, where we set $L = 300$ m, $N = 20$, $r = 0.9$, $\alpha = 0.5$ and $\Delta^S = 10$ s and keep the remaining settings the same. Fig. 9 compares the trajectory solutions from PSA and NSG-SH for this instance, where crosses mark the ends of quadratic pieces. We can see that the PSA solution contains less quadratic pieces and appears smoother compared with the NSG-SH solution. and therefore the PSA trajectories may be easier to implement in real-time control. Whereas as N increases, a NSG-SH trajectory could contain quite a number of quadratic pieces including repeated deceleration-acceleration cycles, though at mild acceleration magnitudes. This would add some control difficulty and slightly compromise driving comfort.

4.2. Signalized segment

This section investigates a highway section where a fixed-timing signal controls exit location L.[6] Assume that the effective green starts at time 0 and has a duration of G, and the effective red time has a duration of R. For illustration purposes, we set $G = R = C/2$ in the following experiments, where $C = R + G$ is the cycle length. The vehicle arrival times are again generated by Eq. (44). Further, we apply Proposition 1 in Ma et al. (2017) to obtain exit times t_n^+ as the earliest time when vehicle n can exit this segment, formulated as follows:

$$t_n^+ = \begin{cases} \mathcal{G}(t_n^- + L/\bar{v}), & \text{if } n = 1; \\ \mathcal{G}\left(\max\left\{t_n^- + L/\bar{v}, t_{n-1}^+ + \tau + s/\bar{v}\right\}\right), & \text{otherwise,} \end{cases}$$

where function

$$\mathcal{G}(t) := \begin{cases} t, & \text{if } \mod(t, R+G) \in [0, G) \text{ (or the light is green at time } t); \\ \lceil t/(R+G) \rceil \cdot (R+G), & \text{otherwise,} \end{cases}$$

which pushes time t to the beginning of the next green phase if it is in a red phase. We set the default parameter values as: $L = 500$ m, $C = 60$ s, $N = 50$, $\underline{a} = -3.5\,\text{m/s}^2$, $\bar{a} = 2\,\text{m/s}^2$, $\bar{v} = 16\,\text{m/s}$ (≈ 35 mph), $s = 7$ m, $\tau = 1.5$ s, $\alpha = 0.5$ and $r = 0.4$. With this, all the input parameters are ready. The trajectory optimization approach first breaks the traffic stream into independent platoons with the PA algorithm and then applies the PSA algorithm to each platoon to smooth the corresponding trajectories.

Fig. 10 shows the trajectory results in the time-space diagram for two cases. The first case (Fig. 10(a)), referred as the extreme acceleration (EA) case, is a feasible solution to RSTO where the acceleration variables are set to their extreme values, i.e., $a^- = -\underline{a}$ and $a^+ = \bar{a}$. This solution is regarded as the benchmark without optimally smoothing the trajectories. The second case (Fig. 10(b)), referred as the optimal trajectory case, is the optimal trajectories (OT) obtained with the PSA algorithm. We can see that the trajectories in the EA case have relatively sharp accelerations and decelerations, and vehicles are forced to stop before passing this intersection. Whereas the OT result exhibits smooth trajectories and completely eliminates stops. Therefore we expect that the OT result has better performance compared with the benchmark EA case.

Table 1 compares the EA objective values with the optimal OT objective values. Nine different instances are tested, and in each instance at most one parameter value is changed and the remaining parameters stay at their default values. Both

[6] We would like to note that although the signal timing is set to be fixed in this example, the proposed trajectory optimization method in general is not limited to a fixed-timing setting. Rather, as long as the signal timing in the optimization horizon is determined, even if each cycle's setting is variable, the proposed method is still applicable. In the case of dynamic signal timing plan, the proposed method can be applied repeatedly in a rolling window manner as long as the signal timing in the current window is determined with the estimated upstream demand information (e.g., each vehicle's arrival time t_n^-) before trajectory optimization.

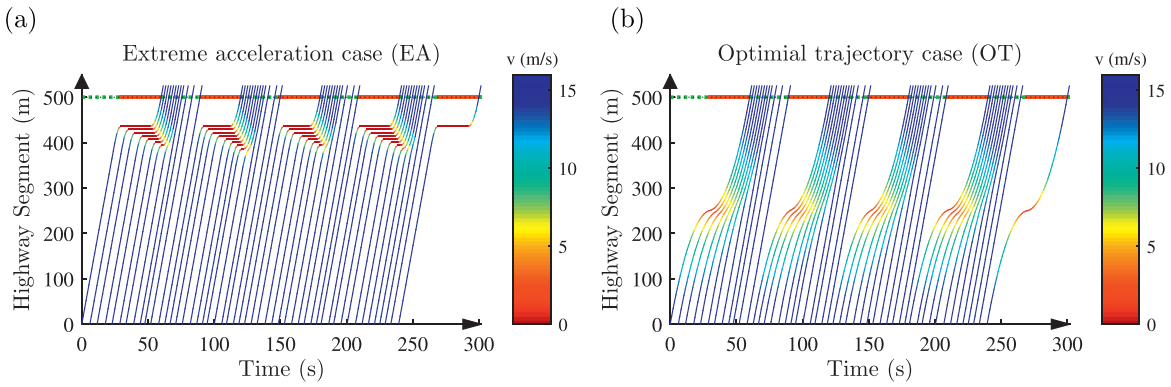

Fig. 10. Signalized segment results: (a) trajectory plot for extreme accelerations (EA); and (b) optimal trajectory (OT) plot.

Table 1
Comparison of objective values for signalized segments.

Parameter values	C_{EA}^{VSP} (kJ/ton)	C_{OT}^{VSP}(kJ/ton)	ε^{VSP}	C_{EA}^{SA}(m^2/s^4)	C_{OT}^{SA}(m^2/s^4)	ε^{SA}
Default	547.9	468.8	14%	63.2	7.9	88%
$N = 25$	549.1	473.1	14%	60.9	7.6	88%
$N = 75$	546.6	462.2	15%	65.9	8.4	87%
$L = 250$ m	256.1	230.6	10%	66.6	25.5	62%
$L = 750$ m	836.4	710.0	15%	66.8	4.2	94%
$C = 30$ s	549.0	498.3	9%	58.3	3.9	93%
$C = 90$ s	545.6	449.0	18%	69.1	11.8	83%
$r = 0.2$	552.4	477.7	14%	55.1	7.2	87%
$r = 0.6$	535.4	514.2	4%	89.7	49.7	45%

VSP function (10) and SA function (11) are tested in the objective. For VSP function (10), we set $\xi = 5.5043$, $\psi = 0.2953$ and $\zeta = 0.00338$ by converting the coefficients from Frey et al. (2002) to fit the metric units. Denote the EA objectives for VSP and SA with C_{EA}^{VSP} and C_{EA}^{SA}, respectively, and denote the OT objectives for VSP and SA with C_{OT}^{VSP} and C_{OT}^{SA}, respectively. The corresponding objectives are compared between EA and OT. The improvement from EA to OT for the VSP objective is denoted by $\varepsilon^{VSP} := \left(C_{EA}^{VSP} - C_{OT}^{VSP}\right)/C_{EA}^{VSP}$, and the improvement for the SA objective is $\varepsilon^{SA} := \left(C_{EA}^{SA} - C_{OT}^{SA}\right)/C_{EA}^{SA}$. We see from Table 1 that for both VSP and SA objectives, the OT results yield significantly better performance than the benchmark EA results. We see both ε^{VSP} and ε^{SA} are insensitive to vehicle number N. Note that at the default values, the vehicle arrival rate is less than the intersection capacity, and thus no queue remains at the end of a green phase. This indicates vehicles arriving in different cycles shall belong to different platoons and the number of arrival vehicles does not much affect the average platoon size. This explains why ε^{VSP} and ε^{SA} are insensitive to N. As L increases, ε^{VSP} does not change much, which indicates that the saving of fuel consumption from trajectory smoothing is not much affected by the segment length. But ε^{SA} increases significantly as L increases, this is because a longer segment provides more space for trajectory smoothing and thus shall further reduce acceleration magnitudes. We see ε^{VSP} increases with C. This is probably because a longer signal cycle may force EA trajectories to stop for a longer time and thus cause more fuel consumption, while the OT trajectories may still have room to glide through without full stops. However, ε^{SA} decreases as C increases, which indicates a longer cycle increases the acceleration magnitudes of the smoothed trajectories. As r goes above 0.5, both ε^{VSP} and ε^{SA} decrease dramatically. Note when $r > 0.5$, the intersection capacity is less than the arrival vehicle rate, and thus more and more vehicles will be queued over cycles. An increasing queue occupies much of the segment space and diminishes the room for trajectory smoothing, and thus the trajectory smoothing effect is not as salient in this case.

4.3. Non-Stop intersection

This section investigates a one-lane non-stop intersection where trajectories of approaching vehicles are coordinated such that they all can pass the intersection without stops (Li and Wang, 2006; Dresner and Stone, 2008). We consider this intersection has two identical approaches of vehicles crossing at the intersection. The segment length of each approach is identical to L, and each approach has N vehicles and their arrival times are again generated by Eq. (44). We rank these $2N$ vehicles from both approaches by their arrival times in an ascending order, denoted as $n_1, n_2, \ldots, n_{2N}$. Assume the exit times of vehicles from both approaches follow a first-in-first-out (FIFO) protocol, i.e., $t_{n_1}^+ < t_{n_2}^+ < \cdots < t_{n_{2N}}^+$, and all vehicles enter the intersection at speed $\bar{v}$ so as to maximize the intersection throughput. Then for every two consecutive vehicles n_{i-1} and n_i, if they are from the same approach, then the separation of their exit times should be no less than $\tau + s/\bar{v}$ due to safety constraints (7). Otherwise, if n_{i-1} and n_i are from different approaches, we assume the separation of their exit times should be no less than a minimum switching headway h^S for safe crossing. With this FIFO protocol, vehicle exit times can be fixed

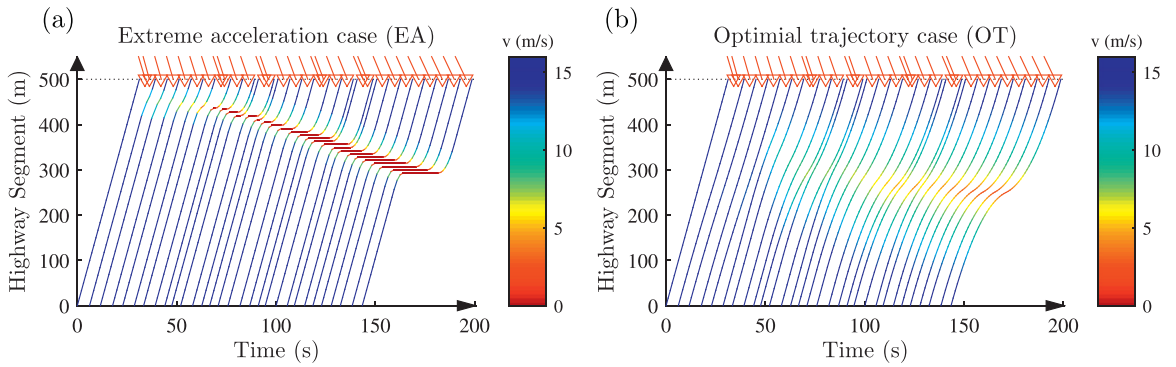

Fig. 11. Non-stop intersection results: (a) trajectory plot for extreme accelerations (EA); and (b) optimal trajectory (OT) plot (red triangles mark the exit times of vehicles from the other approach). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 2
Comparison of objective values for non-stop intersections.

Changed parameter	C_{EA}^{VSP} (kJ/ton)	C_{OT}^{VSP}(kJ/ton)	ε^{VSP}	C_{EA}^{SA}(m^2/s^4)	C_{OT}^{SA}(m^2/s^4)	ε^{SA}
Default	545.3	441.0	19%	81.0	8.5	90%
$N = 15$	548.3	481.5	12%	72.1	3.5	95%
$N = 45$	544.8	436.8	20%	83.1	13.1	84%
$L = 250$ m	255.3	245.5	4%	81.0	57.0	30%
$L = 750$ m	835.5	697.3	17%	81.0	4.0	95%
$r = 0.2$	578.8	575.9	0.5%	9.5	0.04	99.5%
$r = 0.6$	543.9	446.5	18%	86.1	21.1	75%

as

$$t_{n_i}^{+} = \begin{cases} 0, & \text{if } i = 1; \\ \max\left\{t_{n_i}^{-} + L/\bar{v},\ t_{n_{i-1}}^{+} + \tau + s/\bar{v}\right\}, & \text{if vehicles } i \text{ and } i-1 \text{ are from the same approach;} \\ \max\left\{t_{n_i}^{-} + L/\bar{v},\ t_{n_{i-1}}^{+} + h^{S}\right\}, & \text{if vehicles } i \text{ and } i-1 \text{ are from different approaches.} \end{cases}$$

The default parameters are set the same as those in the previous section except for $N = 30$ (which is for the clearance of trajectory plots in Fig. 11), and in addition, h^S is set to 3s. For illustration purposes, we only investigate one approach. Fig. 11 compares the trajectories between the benchmark EA case and the OT case. We see that the OT trajectories are much smoother than the EA trajectories and minimize the need for stops.

Table 2 compares results between the benchmark EA case and the OT case for problem instances with different parameters. The setting follows Table 1. For most instances, the improvement from EA to OT is significant for both the VSP and SA objectives. We see that as N increases, ε^{VSP} increases yet ε^{SA} decreases. Note that with the default saturation rate, the queue grows with the vehicle number and thus more upstream trajectories in the EA case would have higher speed variations or longer stop sections. The improvement of these upstream queued vehicles shall dominate ε^{VSP} and ε^{SA} values as N increases. As L grows, ε^{VSP} first increases and then decreases, which indicates there would be some intermediate segment length range best for fuel consumption saving. ε^{SA} consistently increases as L grows, which again is because a longer segment provides more room for smoother trajectories. When r is very low, the improvement of VSP is not apparent because traffic is anyway close to free flow. As r increases to a certain level at which traffic gets congested, ε^{VSP} seems to be insensitive as r further grows. Whereas ε^{SA} increases as r drops, which is because lower r provides more room to smooth trajectories closer to straight lines with near-zero accelerations.

4.4. Multi-Lane highway

Although the proposed trajectory optimization model focuses on a one-lane highway segment, it can be easily extended to a multi-lane highway. To demonstrate the applicability of this algorithm to a multi-lane highway, this section investigates a segment of a two-lane highway that merges into one lane. Let L^{up} and L^{down} denote the lengths of the upstream two-lane and downstream one-lane highway segments, respectively. Two cases are considered in this experiment: the optimal trajectory case and the benchmark case. In the optimal trajectory case, the vehicle trajectories on the two lanes in the upstream are controlled to merge into the one-lane downstream segment without any conflict. In the benchmark case, it is assumed that the vehicles are not controlled. In both cases, we assume that N vehicles arrive at each lane on the upstream segment with the arrival times generated by Eq. (44). Similar to the previous section, we rank these vehicles by

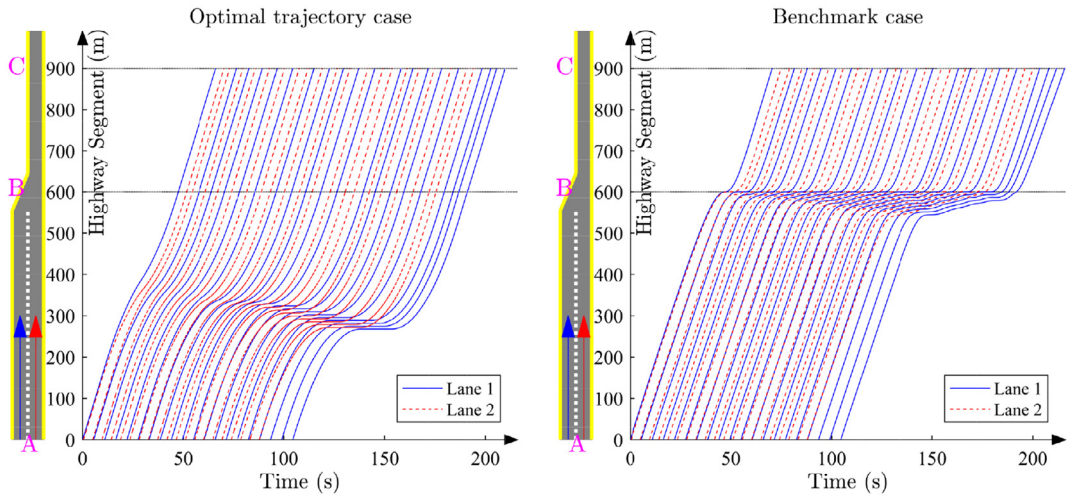

Fig. 12. Highway merge application results.

their arrival times in an ascending order. Then, the exit times from the first upstream segment are determined by the FIFO protocol. These exit times are basically the times when vehicles merge into the one-lane downstream segment. According to safety constraint (7), the separation between these exit times for each consecutive pair of vehicles should be no less than $\tau + s/\bar{v}$. In the optimal trajectory case, it is assumed that all vehicles arrive at the upstream end of the upstream segment (e.g., location 0 m in Fig. 12), merge into the upstream end of the downstream segment (or the downstream end of the upstream segment, e.g., location 600 m in Fig. 12, which is also the merging bottleneck), and also exit the downstream end of the downstream segment (e.g., location 900 m in Fig. 12) with the maximum speed of $\bar{v}$ (e.g., to maximize the system throughput). Note that we let the slow-down or stopping segments of the vehicle trajectories shift slightly upstream of the bottleneck (instead of right at the bottle neck) and let the vehicles pass the bottleneck right at the maximum speed in order to improve bottleneck throughput and trajectory smoothness (Ma et al., 2017; Zhou et al., 2017; Ghiasi et al., 2017b). In the benchmark case, vehicle trajectories are simulated using the Intelligent Driver Model (IDM) car-following model, which describes the dynamics of semi-automated vehicles in addition to that of human-driven vehicles (Treiber and Kesting, 2013). This model is formulated as

$$a = \bar{a}\left(1 - \left(\frac{v}{\bar{v}}\right)^{\eta} - \left(\frac{s^*}{s_0}\right)^2\right),$$

$$s^* = s_0 - s^{\text{car}} + \max\left(0, vh + \frac{v \cdot \Delta v}{2\sqrt{-\bar{a}\underline{a}}}\right),$$

where v and a are the vehicle speed and acceleration, respectively, s^* is the desired space gap, s^{car} is the vehicle length, h is the time gap, Δv is the speed difference between the preceding and the current vehicle, and η is the acceleration exponent.

This experiment sets $N = 20$, $r = 0.4$, $L^{\text{up}} = 600$ m, $L^{\text{down}} = 300$ m, $s^{\text{car}} = 5$ m, $h = 1.5$ s, $\eta = 4$, and the other parameters are set at their default values at Section 4.2. Fig. 12 plots the vehicle trajectories in the longitudinal direction, in which the optimal and benchmark cases are shown in the left and right-hand sides, respectively. In the optimal trajectory case, we see that on the upstream segment where the vehicles are separate across the two lanes, the trajectories may intersect with each other since vehicles on different lanes can pass each other freely. In this case, the trajectory shapes on the upstream segment are designed such that the vehicles can merge into the downstream segment according to the FIFO order. After merging, the vehicles are coordinated to follow one another to exit the downstream segment properly. As a result, the vehicles merge into the downstream section at the maximum possible throughput (i.e.,$1/(\tau + s/\bar{v})$) without speed or capacity drop, which ensures the maximum throughput and consequentially, the minimum travel delay and a relatively short queue length. Further, the vehicle trajectories upstream of the bottleneck are controlled to have mild accelerations so as to reduce fuel consumption (e.g., in terms of the VSP objective) and improve driving comfort (e.g., in terms of the SA objective). On the other hand, in the benchmark case, speed and capacity drop when vehicles pass the bottleneck, which results in smaller throughput, longer vehicle delay and a longer queue. Further, stop and go traffic occurs, which causes extra fuel consumption and driving discomfort.

Table 3 quantitatively compares the objective values of relevant measures in both cases, where C^{VSP} and C^{SA} denote the VSP and SA objectives, respectively, and Q, D and T denote the maximal queue length (the maximum occupied space of stopped or slowly moving vehicles of all simulation time points), the total vehicle delay (as apposed to free flow travel time), and the average downstream throughput, respectively. Further let ε^{VSP}, ε^{SA}, ε^{Q}, ε^{D}, ε^{T}, denote the percentage improve-

Table 3
Comparison of objective values for highway merge.

	C^{VSP} (kJ/ton)	C^{SA}(m^2/s^4)	Q (m)	D (s)	T (veh/hr)
Optimal trajectory	833.3	13.5	32	1282	978
Benchmark	1662.3	84.4	56	1496	966
	$\varepsilon^{VSP} = 50\%$	$\varepsilon^{SA} = 84\%$	$\varepsilon^{Q} = 43\%$	$\varepsilon^{D} = 14\%$	$\varepsilon^{T} = 1\%$

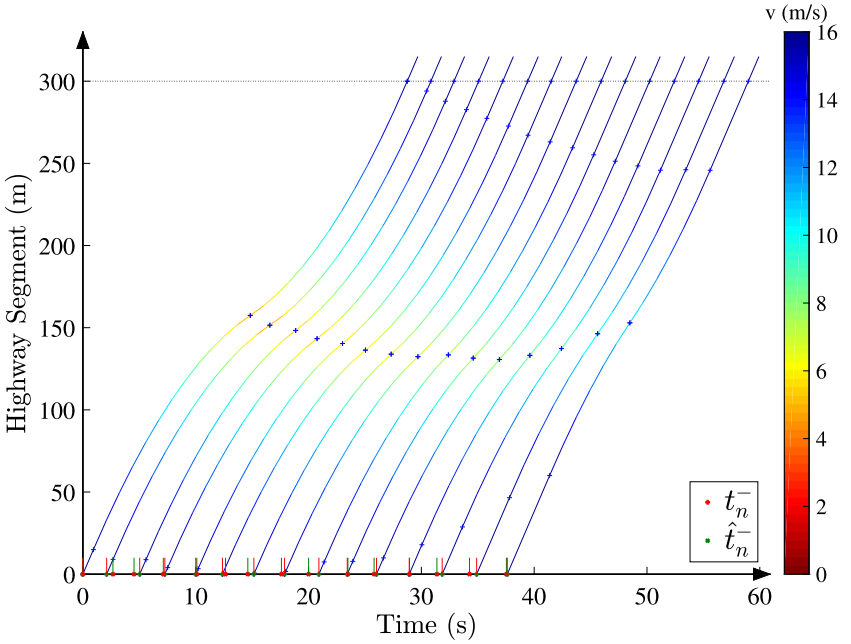

Fig. 13. Simulation results with the stochastic arrival time. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

ment of C^{VSP}, C^{SA}, Q, D and T from the benchmark case to the optimal trajectory case. We see that our control algorithm significantly improves all of these measures compared to the benchmark case. This result demonstrates that the proposed trajectory control algorithm can be effectively extended to a multi-lane highway with simple yet realistic adjustments.

Note that this example assumes that no interfering traffic is flowing into the upstream trajectory smoothing section forming a second bottleneck within a few hundreds of meters upstream. This is consistent with common freeway geometries since two consecutive on-ramps are often miles apart. However, cautions need to be taken in applying this control method for cases (e.g., on arterials) where two bottlenecks are close and may interfere with each other. In this case, future investigation effort is needed to balance the trade-off between the extra upstream space needed for smoothing trajectories at the downstream bottleneck and the interference with the upstream bottleneck.

4.5. Stochastic demand

The previous numerical experiments are based on deterministic vehicle arrival time settings. To test the robustness of the proposed algorithm, this section performs a numerical experiment with stochastic arrival times. This experiment is conducted on a one-lane highway segment with the length of L and N vehicles. We assume that with the connected vehicle technology, a set of arrival times are estimated, denoted by $t_1^- < t_2^- < \ldots < t_N^-$. However, due to potential uncertain conditions, the observed vehicle arrival times may differ from the estimated times. Let $\hat{t}_1^- < \hat{t}_2^- < \ldots < \hat{t}_N^-$ denote the observed arrival times. The estimated and observed arrival times in this experiments are generated by Eq. (44) with different random seeds. To capture the stochastic setting, in addition to trajectory planning initially based on estimated arrival times $\{t_n^-\}$, when each vehicle n arrives at the highway segment (and thus $\hat{t}_n^-$ is just realized), the optimization problem is solved again dynamically to adjust the trajectories of vehicle n and the following vehicles in the platoon. We set $N = 20$, $r = 0.9$, $L = 300$ m, and the other parameters at their default values. The simulation results are shown in Fig. 13, in which the estimated and observed arrival times are shown with red and green dots, respectively. We see that despite descrepancies between estimated $\{t_n^-\}$ and observed $\{\hat{t}_n^-\}$, the final trajectories remain smooth and thus will yield desired performance. This experiment indicates that the proposed algorithm can handle uncertainties that may occur in vehicle arrival times.

5. Queuing propagation analysis

Intuitively, it may be easy to arrive at a conjecture that traffic smoothing would cause vehicles queued (or slowing down) at more upstream locations, or even cause further queue spillback. This section will investigate this conjecture by rigorously analyzing a special case of the studied problem with homogeneous settings. We assume that the entry headway and the exit headway between every two vehicles is the same, i.e.,

$$t_1^1 = 0, t_n^- = (n-1)(\tau + s/\bar{v})/r, \forall n \in \mathcal{M}\backslash\{1\}, \tag{45}$$

$$t_1^+ = L/\bar{v} + \Delta^S, t_n^+ = t_1^+ + (n-1)(\tau + s/\bar{v})/r, \forall n \in \mathcal{M}\backslash\{1\}. \tag{46}$$

where again parameter $r \in (0, 1]$ is the traffic saturation rate and $\Delta^S \geq 0$ is the phase shift (e.g., due to being blocked by a red light or coordination with the downstream segment). For the conciseness of the formulations, define $\gamma := \tau + s/\bar{v}$. With this, we obtain.

$$\Delta_n = \Delta^S, \forall n \in \mathcal{M},$$

$$\Delta_{n(n-1)} = \Delta^S - \gamma\left(\frac{1}{r} - 1\right), \forall n \in \mathcal{M}\backslash\{1\}.$$

Note that for all vehicles in $\mathcal{M}$, we have $\Delta_{n(n-1)} > 0$. For each vehicle n, since the impact of trajectory smoothing starts at time $\delta^*_{n1}(\phi)$, we will investigate how $\delta^*_1(\phi)$, which marks the end of the trajectory-smoothing-induced queue, changes as ϕ varies. If the above-mentioned conjecture is true, $\delta^*_1(\phi)$ shall always decrease (or the trajectories always get smoother) as ϕ drops. This is to say, $\delta^*_1(\phi)$ increases with ϕ, $\forall \phi \in (0, \infty)$. The following analysis will check whether this is true.

Define

$$\hat{\sigma}_n(\phi, \Delta^S) = \hat{\delta}_{n1}\left(\delta^*_{(n-1)1}(\phi), \phi\right) - \delta^*_{(n-1)1}(\phi) = \alpha + \delta(\phi, \Delta_{n-1}) - \delta\left(\phi, \Delta^S\right).$$

where $\alpha := -\tau\left(\frac{1}{r} - 1\right) - \frac{s}{\bar{v}r}$ for the conciseness of the formulation. This can be expanded as

$$\hat{\sigma}(\phi, \Delta^S) = \alpha + \begin{cases} \sqrt{\frac{2\bar{v}}{\phi}}\left(\sqrt{\Delta^S} - \sqrt{\Delta^S - \gamma\left(\frac{1}{r} - 1\right)}\right), & \text{if} \phi \leq \frac{\bar{v}}{2\Delta^S}; \\ \Delta^S + \frac{\bar{v}}{2\phi} - \sqrt{\frac{2\bar{v}}{\phi}}\sqrt{\Delta^S - \gamma\left(\frac{1}{r} - 1\right)}, & \text{if } \frac{\bar{v}}{2\Delta^S} < \phi \leq \frac{\bar{v}}{2\Delta^S - \gamma\left(\frac{1}{r} - 1\right)}; \\ \gamma\left(\frac{1}{r} - 1\right), & \text{if } \phi > \frac{\bar{v}}{2\Delta^S - \gamma\left(\frac{1}{r} - 1\right)}. \end{cases} \tag{47}$$

Note that $\hat{\sigma}_n(\phi, \Delta^S)$ indicates the difference between $\delta^*_{(n-1)1}(\phi)$ and $\delta^*_{n1}(\phi)$ as

$$\delta^*_{n1}(\phi) - \delta^*_{(n-1)1}(\phi) = \begin{cases} \hat{\sigma}_n(\phi, \Delta^S), & \text{if } \hat{\sigma}_n(\phi, \Delta^S) < 0; \\ 0, & \text{otherwise.} \end{cases}$$

With this formulation, we see that when $r = 1$, $\hat{\sigma}_{(\phi, \Delta_{n-1}) = -s/\bar{v}, \forall \phi,}$ and $\delta^*_1 = t_n^\Delta - \delta_{(\phi, \Delta^S) - (M-1)\underline{d}/\bar{v},}$ which shall always increase with ϕ. Thus the conjecture trivially holds for this case. The following analysis will investigate the non-trivial case when $r \in (0, 1)$. In this case, we see that $\hat{\sigma}(\phi, \Delta^S)$ decreases with ϕ, $\lim_{\phi \to 0} \hat{\sigma}(\phi, \Delta^S) = \infty$ and $\hat{\sigma}(\phi, \Delta^S) = -s/\bar{v}, \forall \phi \geq \frac{\bar{v}}{2\Delta^S - \gamma\left(\frac{1}{r} - 1\right)}$. Since $\hat{\sigma}(\phi, \Delta^S)$ is apparently continuous with ϕ, then there must exist an $\phi^0 < \frac{\bar{v}}{2\Delta^S - \gamma\left(\frac{1}{r} - 1\right)}$ such that $\hat{\sigma}(\phi, \Delta^S) > 0, \forall \phi \in (0, \phi^0)$ and $\hat{\sigma}(\phi, \Delta^S) \leq 0, \forall \phi > \phi^0$. As ϕ increases from 0 to ϕ^0, note that $\hat{\sigma}(\phi, \Delta^S) \geq 0$ and thus $\delta^*_1(\phi) = \delta^*_{11}(\phi) = L/\bar{v} + \Delta^S - \delta(\phi, \Delta^S)$, which shall increase with ϕ. Then we consider two cases:

Case-1: If $\phi^0 < \frac{\bar{v}}{2\Delta^S}$. When $\phi \in \left[\phi^0, \frac{\bar{v}}{2\Delta^S}\right]$,

$$\delta^*_1(\phi) = t_n^\Delta - \delta\left(\phi, \Delta^S\right) + (M-1)\hat{\sigma}(\phi, \Delta^S) = t_n^\Delta - (M-1)\alpha + \beta_1 \phi^{-0.5},$$

where

$$\beta_1 := \left[(M-2)\sqrt{2\bar{v}\Delta^S} - (M-1)\sqrt{2\bar{v}\left(\Delta^S - \gamma\left(\frac{1}{r} - 1\right)\right)}\right]. \tag{48}$$

If $\beta_1 \leq 0$, $\delta^*_1(\phi)$ continues to increase with $\phi \in [\phi^0, \frac{\bar{v}}{2\Delta^S}]$. Otherwise if $\beta_1 > 0$, $\delta^*_1(\phi)$ decreases with ϕ. Next, when $\phi \in [\frac{\bar{v}}{2\Delta^S}, \frac{\bar{v}}{2\left(\Delta^S - \gamma\left(\frac{1}{r} - 1\right)\right)}]$,

$$\delta^*_1(\phi) = t_n^\Delta - (M-1)\alpha + \beta_2(\phi).$$

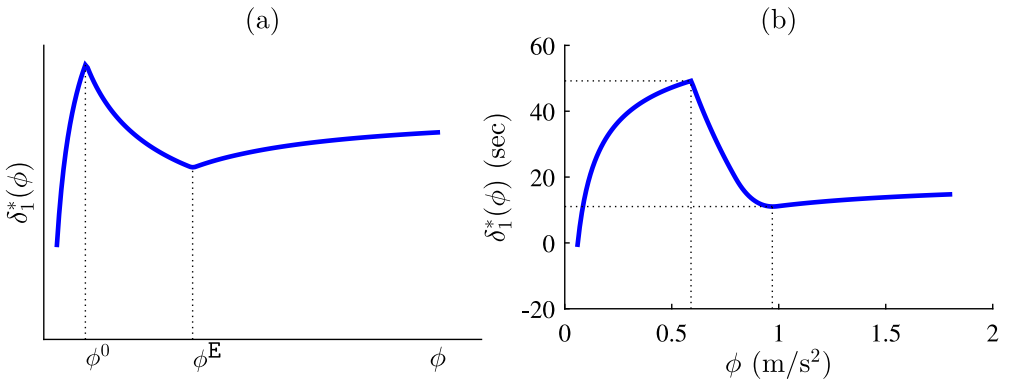

Fig. 14. (a) Illustration of $\delta_1^*(\phi)$ vs. ϕ when $\beta_1 > 0$; (b) $\delta_1^*(\phi)$ vs. ϕ for the default instance.

where $\beta_2(\phi) := (M-2)(\Delta^S + \frac{\bar{v}}{2\phi}) - (M-1)\sqrt{2\bar{v}(\Delta^S - \gamma(\frac{1}{r}-1))}\phi^{-0.5}$. We can obtain

$$\frac{d\beta_2(\phi)}{d\phi} = \left[-(M-2)\frac{\bar{v}}{2\sqrt{\phi}} + (M-1)\sqrt{\frac{\bar{v}\left(\Delta^S - \gamma\left(\frac{1}{r}-1\right)\right)}{2}}\right]\phi^{-1.5}$$

Therefore, if $\beta_1 \le 0$, it is easy to see that $\frac{d\beta_2(\phi)}{d\phi} \ge 0$, and $\delta_1^*(\phi)$ will continue to increase with ϕ. Otherwise if $\beta_1 > 0$, we shall have $\frac{d\beta_2(\frac{\bar{v}}{2\Delta^S})}{d\phi} > 0$ and $\frac{d\beta_2(\frac{\bar{v}}{2\Delta^S - \gamma(\frac{1}{r}-1)})}{d\phi} < 0$. Therefore, there exists a $\phi^E \in [\frac{\bar{v}}{2\Delta^S}, \frac{\bar{v}}{2(\Delta^S - \gamma(\frac{1}{r}-1))}]$ such that $\delta_1^*(\phi)$ decreases with $\phi \in [\frac{\bar{v}}{2\Delta^S}, \phi^E]$ and increases with $\phi \in [\phi^E, \frac{\bar{v}}{2(\Delta^S - \gamma(\frac{1}{r}-1))}]$. Further, when $\phi > \frac{\bar{v}}{2\left(\Delta^S - \gamma\left(\frac{1}{r}-1\right)\right)}$,

$$\delta_1^*(\phi) = t_n^\Delta - \delta\left(\phi, \Delta^S\right) - (M-1)s/\bar{v},$$

which apparently increases with ϕ.

Case-2: If $\phi^0 > \frac{\bar{v}}{2\Delta^S}$, the analysis for $\phi \in [\phi^0, \frac{\bar{v}}{2(\Delta^S - \gamma(\frac{1}{r}-1))}]$ is the same as that for $\phi \in [\frac{\bar{v}}{2\Delta^S}, \frac{\bar{v}}{2(\Delta^S - \gamma(\frac{1}{r}-1))}]$ in the previous case, and the analysis for $\phi > \frac{\bar{v}}{2(\Delta^S - \gamma(\frac{1}{r}-1))}$ is the same as the previous case, too.

In both cases, if $\beta_1 \le 0$, then $\delta_1^*(\phi)$ increases all the way with ϕ, which is consistent with the initial conjecture. However, if $\beta_1 > 0$, then $\delta_1^*(\phi)$ first increases with ϕ during $[0, \phi^0]$, then decreases with ϕ during $[\phi^0, \phi^E]$, and finally increases with ϕ during $[\phi^E, \infty)$, which is illustrated in Fig. 14(a). This is contradictory to the initial conjecture, such that when $\phi \in [\phi^0, \phi^E]$, further decreasing ϕ (or smoothing the trajectories) actually helps reduce the length of the queue. This finding suggests that traffic smoothing does not always worsen queue spillback. Instead, it may help alleviate queuing if the smoothing is done appropriately.

To illustrate this analysis result, we show some examples in the following presentation. The default parameters are set as $L = 1000$ m, $N = 100$, $\bar{v} = 16$ m/s, $s = 7$ m, $\tau = 1.5$ s, $r = 0.5$, $\Delta^S = 10$ m. With this setting, there is only one platoon in this traffic stream (i.e., $M = N$). Further, we obtain $\beta_1 = 162.9\,\text{m}^{0.5}$, $\phi^0 = 0.59$ m/s^2, $\phi^E = 0.97$ m/s^2, $\delta_1^*(\phi^0) = 49.2$ s and $\delta_1^*(\phi^E) = 11.0$ s. The complete $\delta_1^*(\phi)$ to ϕ curve for this default instance is shown in Fig. 14(b), which is consistent with the previous conclusion that $\beta_1 > 0$ indicates that $\delta_1^*(\phi)$ decreases on $[\phi^0, \phi^E]$. This also indicates that the end of queue is at 787.4 m for $\phi = \phi^0$ and at 176.7 m for $\phi = \phi^E$. The trajectories for these two cases are shown in Fig. 15.

Next, we vary the N value and investigate its impact on the queue length. As illustrated in Fig. 16(a), β_1 value increases linearly with N when there is only one platoon (or $M = N$). This can be also seen from the definition of β_1 in Eq. (48). Fig. 16(b) plots the values of $\delta_1^*(\phi^0)$ and $\delta_1^*(\phi^E)$ as β_1 increases (e.g., as a result of the increase of N). We see that $\delta_1^*(\phi^0)$ remains the same across different N value for the following reasons. Note that based on Eq. (47) and the definition, ϕ^0 shall remain the same at 0.59 regardless of N or β_1. Further, when $\phi = \phi_0$, no shock wave propagates backwards and the effect of traffic smoothing stays in a local area regardless of the N value. When $\beta_1 > 0$, ϕ^E shall split from ϕ^0, and as a result we see that $\delta_1^*(\phi^E)$ decreases as β_1 increases from 0. Note that $\left(\delta_1^*(\phi^0) - \delta_1^*(\phi^E)\right) \cdot \bar{v}$ implies the queue distance that traffic smoothing can reduce by lowering ϕ from ϕ^E to ϕ^0, which increases with β_1 (and thus N). Interestingly, this result indicates that even without modification of macroscopic traffic characteristics, backward propagation of stopping shock waves could be hampered by proper traffic smoothing, which only adjusts vehicle trajectories in a local area independent of the number of incoming vehicles (as long as the saturation rate remains the same). Further, as the number of incoming vehicles in a platoon increases, there actually could be more potential to reduce the queue length by traffic smoothing.

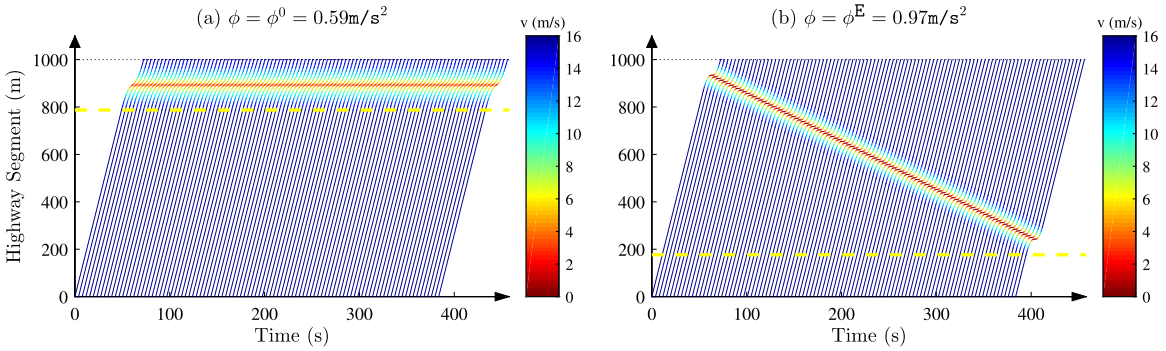

Fig. 15. Trajectories for ϕ^0 and ϕ^{E} at the default parameter values (where the dashed yellow line marks the end of the queue). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

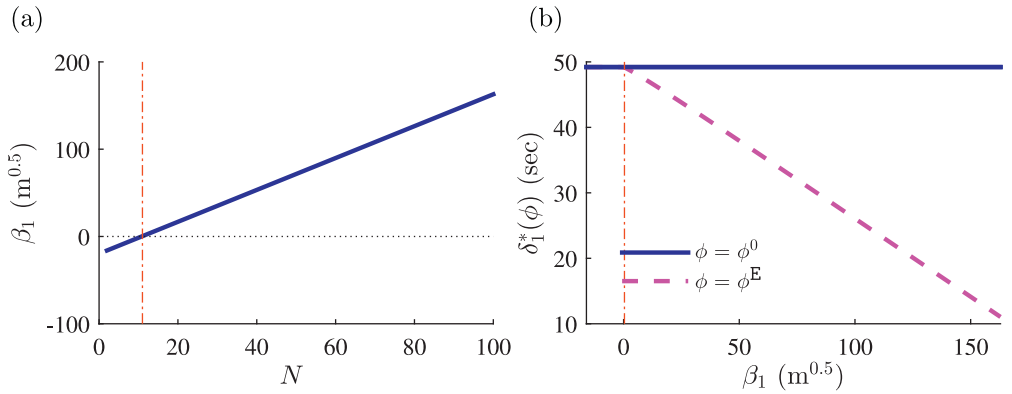

Fig. 16. Results for instances with varying N: (a) N vs. β_1; and (b) β_1 vs. $\delta_1^*(\phi^0)$ and $\delta_1^*(\phi^E)$.

6. Conclusion

This paper investigates a trajectory smoothing problem for a general one-lane highway segment with pure CAVs and provides elegant theoretical insights and efficient algorithmic methods. Inspired by previous studies from Co-author Li's research team, this problem is simplified to one where each vehicle's trajectory is approximated with no more than five pieces of consecutive quadratic functions and all trajectories share identical acceleration and deceleration rates in the same platoon. This simplified problem is shown to have elegant theoretical properties in the objective shape and the feasible region. These properties lead to the development of an exact solution algorithm that efficiently solves the true optimum to this problem with only a series of analytical operations. The optimal solution can be intuitively interpreted as stretching all trajectories as smooth as the feasibility allows. Numerical examples reveal that the proposed analytical exact algorithm solves the problem much faster with the same or better solution quality compared with its numerical predecessor proposed earlier Ma et al. (2017). They also illustrate the applications of this algorithm to various CAV trajectory smoothing problems, e.g., on signalized segments and at non-stop intersections. Moreover, the numerical experiments indicate that the proposed algorithm is extendable to multi-lane highway traffic and also can handle stochastic vehicle arrival setting. Further, by constructing a homogeneous special case, we analyze how traffic smoothing affects propagation of the vehicle queue. We find that counter-intuitively, proper traffic smoothing may reduce the queue length or confine traffic slowdown within a local area without further propagation.

This fundamental study can be extended in a number of directions in the future. First, although we conjecture that the optimal solution to the simplified problem is likely near-optimum to the primary optimization problem, rigorous optimization models need to be built to quantify the optimality of the simplified solution. Further, while this study offers insights into a far-future scenario where all vehicle are CAVs, it is also worth investigating near-future scenarios with mixed traffic. The challenge in mixed traffic is that the trajectories of human driven vehicles may not be as regular and smooth as CAV trajectories, and they have more uncertainties and unpredictability. To overcome this challenge, well established car following models, either deterministic (Brackstone and McDonald, 1999) or stochastic (Tian et al., 2016), can be used in predicting human vehicle trajectories. And the new mixed traffic trajectory control method may need to populate CAV trajectories that suit the shapes of human vehicle trajectories (Yao et al., 2018) and to be adaptive to updating observations of human driven vehicles in real time (Ghiasi et al., 2017b). Next, while this study focuses on a basic roadway facility, it will be interesting

to investigate how to use it as a building block to investigate problems with more complex geometries. For example, it is worth investigating how to integrate longitudinal trajectory optimization with lateral vehicle control in multi-lane applications where lane changes may take place. Finally, some minor restrictions (e.g., identical speeds at the entrance and the exit of the highway segment) can be relaxed in future studies to suit more flexible problem settings.

Acknowledgment

This research is supported by the U.S. National Science Foundation through Grant CMMI#1453949.

Appendix A. Notation table

Variable	Description
a^-	deceleration magnitude variable
a^+	acceleration magnitude variable
$\underline{a}$	minimum deceleration
$\bar{a}$	maximum deceleration
A_p	coefficient of the general cost function
B_q	coefficient of the general cost function
C_r	coefficient of the general cost function
C	traffic signal cycle
$C_{\text{EA}}^{\text{VSP}}$	VSP objective with extreme accelerations
$C_{\text{EA}}^{\text{SA}}$	SA objective with extreme accelerations
$C_{\text{OT}}^{\text{VSP}}$	optimal VSP objective
$C_{\text{OT}}^{\text{SA}}$	optimal SA objective
$c(x_n)$	general cost function
Δ_n	travel delay for
$\Delta_{(n-1)n}$	potential time headway conflict
Δ^{S}	travel delay for vehicle 1
δ_{n1}^*	optimal duration of the first (quadratic) piece
$\hat{\delta}_{n1}$	maximum first-piece duration due to safety
$\bar{\delta}_{n1}$	maximum first-piece duration due to the depature time
δ_1^*	optimal first-piece duration across all vehicles
$\hat{\delta}_{n1}^{\text{lead}}$	first-piece duration function of the lead vehicle's states
$\hat{\delta}_{n1}^{\Delta}$	the difference from $\bar{\delta}_{n1}$ to $\hat{\delta}_{n1}$
δ_{ni}	first-piece duration variable
$e(\dot{x}_n(t), \ddot{x}_n(t))$	instantaneous cost
ϕ	auxiliary variable to indicate trajectory smoothness
$\underline{\phi}$	lower bound to ϕ
$\overline{\phi}$	upper bound to ϕ
ϕ^0	value of ϕ for a candidate peak of $\delta_1^*(\phi)$ in the homogeneous case.
ϕ^{E}	value of ϕ for a candidate valley of $\delta_1^*(\phi)$ in the homogeneous case
$\hat{\phi}$	given ϕ value
ϕ^+	lower bound to a stationary interval of ϕ
ϕ^-	upper bound to a stationary interval of ϕ
ϕ_n^{crit}	candidate bounds to stationary interval of ϕ
ϕ_n^{next}	next ϕ value where a lead vehicle changes to non-lead
ϕ^*	optimal ϕ value
$G_{n\phi^-}$	piecewise quadratic function of $\sqrt{\phi}$
G	green signal time
$\mathcal{G}$	function to move a time to the next green phase
ξ	coefficient of the VSP formula
ξ_n	random variable for arrival time generation
ξ_n'	random variable for departure time generation
h^{S}	minimum switching headway for the non-stop intersection
L	length of the investigated road segment
λ	auxiliary variable for the deceleration weight
λ^*	optimal λ value
$\mathbb{M}$	set of platoons
$\mathcal{M}$	one platoon include M vehicles
M	size of platoon $\mathcal{M}$
$\hat{\mathcal{M}}$	set of lead vehicles for given ϕ
$\hat{M}$	size of $\hat{\mathcal{M}}$
N	total number of CAVs

(*continued on next page*)

Variable	Description
n	vehicle index
$\hat{n}_i$	the ith lead vehicle
s	safety distance gap
t_n^-	arrival time
t_n^+	departure time
τ	response time
t_n^Δ	travel time
t_{ni}	ending time of the ith piece of vehicle n's trajectory
ψ	coefficient of the VSP formula
ζ	coefficient of the VSP formula
r	saturation rate
R	red signal time
$\mathcal{R}^{\text{feas}}$	feasible region of ϕ
$\mathcal{R}_n^{\text{feas}}$	feasible region of ϕ associated with vehicle n
$\mathcal{R}_{nk}^{\text{feas}}$	feasible region of ϕ associated with vehicle n in the k^{th} stationary interval
$\bar{v}$	speed limit
$x_n(t)$	vehicle trajectory
$\dot{x}_n(t)$	velocity
$\ddot{x}_n(t)$	acceleration
$x_n^s(t)$	shadow trajectory
$\mathbf{x}$	vector of all trajectories

Appendix B. Proofs

Proposition 1

Proof. Let $\{\delta'_{n1}\}$ denote an feasible solution to $\{\delta_{n1}\}$. Then $\{\delta'_{n1}\}$ shall satisfy constraints (28) as follows:

$$0 \le \delta'_{n1} \le \begin{cases} \tilde{\delta}_{n1}(\phi), & \text{if } n = 1; \\ \min\left\{\tilde{\delta}_{n1}(\phi), \hat{\delta}_{n1}\left(\delta'_{(n-1)1}, \phi\right)\right\}, & \text{otherwise;} \end{cases}, \forall n \in \mathcal{M}.$$

Then first $\delta^*_{11}(\phi) = \tilde{\delta}_{11}(\phi) \ge \delta'_{11} \ge 0$ is apparently feasible to constraints (28) as well. Now we will use induction to show that $\delta^*_{n1}(\phi)$ is feasible for $n \in \mathcal{M}/\{1\}$. Assume $0 \le \delta'_{k1} \le \delta^*_{k1}(\phi)$ and $\delta^*_{k1}(\phi)$ satisfies constraints (28), which is obviously true for $k = 1$. Then for $n = k+1$, by definition (32), $\delta^*_{n1}(\phi) = \min\{\tilde{\delta}_{n1}(\phi), \hat{\delta}_{n1}(\delta^*_{(n-1)1}(\phi), \phi)\}$, which obviously satisfies constraints (28). Further, by definition $\hat{\delta}_{n1}\left(\delta_{(n-1)1}(\phi), \phi\right)$ apparently increases with the $\delta_{(n-1)1}(\phi)$ value, which indicates $\delta^*_{n1}(\phi) \ge \delta'_{n1} \ge 0$. This completes the induction proof. □

Lemma 1

Proof. Based on Eq. (23), if $\phi \le \frac{\bar{v}}{2\Delta_n}$,

$$F_n^{\text{A}}(\phi, \lambda, p) = \sqrt{2\bar{v}\Delta_n}\left(\lambda^{1-p} + (1-\lambda)^{1-p}\right)\phi^{p-0.5},$$

which apparently increases with $\phi > 0$ when $p \ge 1$. Otherwise, if $\phi > \frac{\bar{v}}{2\Delta_n}$,

$$F_n^{\text{A}}(\phi, \lambda, p) = \bar{v}\left(\lambda^{1-p} + (1-\lambda)^{1-p}\right)\phi^{p-1},$$

which again increases with $\phi > 0$ when $p \ge 1$. This completes the proof. □

Lemma 2

Proof. Eq. (23) apparently shows that $F_n^{\text{A}}(\phi, \lambda, p)$ is symmetric with respect to $\lambda = 0.5$. Further,

$$\frac{dF_n^{\text{A}}(\phi, \lambda, p)}{d\lambda} = \min\left(\sqrt{2\bar{v}\Delta_n\phi}, \bar{v}\right) \cdot \phi^{p-1}(q-1)\left((1-\lambda)^{-p} - \lambda^{-p}\right),$$

which apparently is no greater than 0 when $\lambda \in (0, 0.5]$ and no less than 0 when $\lambda \in [0.5, 0)$. This completes the proof. □

Lemma 3

Proof. This lemma trivially holds when $q = 1$ since $F_n^V(\phi, 1)$ equals constant L. Then we only investigate the cases for $q = 2, 3, 4$. Based on formulation (24), if $\phi \le \frac{\bar{v}}{2\Delta_n}$,

$$\frac{dF_n^V(\phi, q)}{d\phi} = -\frac{\bar{v}^{q+1}}{(q+1)\phi^2}\sum_{i=3}^{q+1}\binom{q+1}{i}(i/2-1)\left(-\sqrt{\frac{2\Delta_n\phi}{\bar{v}}}\right)^i.$$

Since $\frac{2\Delta_n\phi}{\bar{v}} \le 1$ holds in this case, it is easy to verify that $\frac{dF_n^V(\phi,q)}{d\phi} \ge 0$ for $q = 2, 3, 4$.

Otherwise if $\phi > \frac{\bar{v}}{2\Delta_n}$,

$$\frac{dF_n^V(\phi, q)}{d\phi} = \left(\frac{1}{2} - \frac{1}{q+1}\right)\bar{v}^{q+1}\frac{1}{\phi^2}$$

which is greater than 0 for all $q > 1$. This completes the proof. □

Corollary 1

Proof. Based on the definition of $\delta_{n1}^*(\phi)$ with Eqs. (25), (27) and (32), we find that $\delta_{n1}^*(\phi)$, is continuous with ϕ. Then it is easy to see with Eq. (33) that $\delta_1^*(\phi)$ is continuous with ϕ as well. Further, we shall see that $\delta_1^*(\underline{\phi}) \le \min_{n\in\mathcal{M}}\tilde{\delta}_{n1}(\underline{\phi}) = 0$. Then if $\delta_1^*(\underline{\phi}) = 0$, then apparently, $\phi^* = \underline{\phi}$. Otherwise, $\delta_1^*(\underline{\phi}) < 0$. Then if $\delta_1^*(\phi) < 0, \forall\phi \in [\underline{\phi}, \bar{\phi}]$, ϕ^* does not exist. Otherwise, based on the intermediate value theorem, Eqs. (36) and (38) are equivalent. This completes the proof. □

Proposition 2

Proof. First as $\phi \to \infty$, $\hat{n}(n, \phi)$ has to be less than n, and thus $\delta_{n1}^{\Delta}(\phi) = t_n^+ - \left(n - \hat{n}(n,\phi)\right)\tau - t^+_{\hat{n}(n,\phi)} > 0, \forall n \in \mathcal{M}$, and thus $\hat{\mathcal{M}}(\phi) = \{1\}$. Then we only need to show as ϕ increases, if a lead vehicle n leaves $\hat{\mathcal{M}}(\phi)$, it should not come back to $\hat{\mathcal{M}}(\phi)$ and become a lead vehicle again. When vehicle n just leaves $\hat{\mathcal{M}}(\phi)$, then we have

$$\delta_{n1}^{\Delta}(\phi) = t_n^+ - \left(n - \hat{n}(n-1,\phi)\right)\tau - t^+_{\hat{n}(n-1,\phi)} + \sum_{n'=\hat{n}(n-1,\phi)+1}^{n}\left(\delta\left(\phi, \Delta_{n'(n'-1)}\right) - \delta(\phi, \Delta_{n'})\right) \ge 0, \forall n \in \mathcal{M}\backslash\{1\}.$$

As ϕ increases, when $\hat{n}(n-1,\phi)$ does not change, $\sum_{n'=\hat{n}(n-1,\phi)+1}^{n}\left(\delta\left(\phi, \Delta_{n'(n'-1)}\right) - \delta(\phi, \Delta_{n'})\right)$ shall increase and $\delta_{n1}^{\Delta}(\phi)$ shall remain non-negative. If $\hat{n}(n-1,\phi)$ changes, it has to decrease based on the formulation of $\delta^{\Delta}_{\hat{n}(n-1,\phi)1}(\phi)$. Thus $\delta_{n1}^{\Delta}(\phi)$ shall remain non-negative, too. This completes this proof. □

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.trb.2018.11.002.

References

Ahn, K., Rakha, H., Park, S., 2013. Ecodrive application: algorithmic development and preliminary testing. Transp. Res. Record J. Transp. Res. Board (2341) 1–11.

Brackstone, M., McDonald, M., 1999. Car-following: a historical review. Transp. Res. Part F Traffic Psychol. Behav. 2 (4), 181–196.

Day, C., Haseman, R., Premachandra, H., Brennan Jr, T., Wasson, J., Sturdevant, J., Bullock, D., 2010. Evaluation of arterial signal coordination: methodologies for visualizing high-resolution event data and measuring travel time. Transp. Res. Record J. Transp. Res. Board 2192 (1), 37–49.

De Nunzio, G., Canudas de Wit, C., Moulin, P., Di Domenico, D., 2013. Eco-driving in urban traffic networks using traffic signal information. In: Proceedings of the Fifty-Second Annual Conference on Decision and Control (CDC). IEEE, pp. 892–898.

Dresner, K., Stone, P., 2008. A multiagent approach to autonomous intersection management. J. Artif. Intell. Res. 31, 591–656.

Frey, H., Unal, A., Chen, J., Li, S., Xuan, C., 2002. Methodology for developing modal emission rates for EPA's multi-scale motor vehicle and equipment emission estimation systemPrepared by North Carolina state university for the office of transportation and air quality, us environmental protection agency. Ann Arbor, MI.

Ghiasi, A., Hussain, O., Qian, Z.S., Li, X., 2017a. A mixed traffic capacity analysis and lane management model for connected automated vehicles: a Markov chain method. Transp. Res. Part B Methodol. 106, 266–292.

Ghiasi, A., Ma, J., Zhou, F., Li, X., 2017b. Speed harmonization algorithm using connected autonomous vehicles. In: Proceedings of the Ninety-Sixth Annual Meeting on Transportation Research Board.

Guan, T., Frey, C.W., 2013. Predictive fuel efficiency optimization using traffic light timings and fuel consumption model. In: Proceedings of the Sixteenth International Conference on Intelligent Transportation Systems (ITSC). IEEE, pp. 1553–1558.

Guler, S.I., Menendez, M., Meier, L., 2014. Using connected vehicle technology to improve the efficiency of intersections. Transp. Res. Part C Emerg. Technol. 46, 121–131.

Hegyi, A., De Schutter, B., Hellendoorn, H., 2005. Model predictive control for optimal coordination of ramp metering and variable speed limits. Transp. Res. Part C Emerg. Technol. 13 (3), 185–209.

Iglesias, I., Isasi, L., Larburu, M., Martinez, V., Molinete, B., 2008. I2v communication driving assistance system: on-board traffic light assistant. In: Proceedings of the Sixty-Eighth Conference on Vehicular Technology Conference (VTC). IEEE, pp. 1–5.
Jiang, R., Hu, M.-B., Zhang, H., Gao, Z.-Y., Jia, B., Wu, Q.-S., 2015. On some experimental features of car-following behavior and how to model them. Transp. Res. Part B 80, 338–354.
Kamalanathsharma, R.K., Rakha, H., et al., 2013. Multi-stage dynamic programming algorithm for eco-speed control at traffic signalized intersections. In: Proceedings of the Sixteenth International IEEE Conference on Intelligent Transportation Systems (ITSC). IEEE, pp. 2094–2099.
Lee, J., Park, B., 2012. Development and evaluation of a cooperative vehicle intersection control algorithm under the connected vehicles environment. IEEE Trans. Intell. Transp. Syst. 13 (1), 81–90.
Letter, C., Elefteriadou, L., 2017. Efficient control of fully automated connected vehicles at freeway merge segments. Transp. Res. Part C Emerg. Technol. 80, 190–205.
Li, L., Wang, F.-Y., 2006. Cooperative driving at blind crossings using intervehicle communication. IEEE Trans. Veh. Technol. 55 (6), 1712–1724.
Li, X., Cui, J., An, S., Parsafard, M., 2014. Stop-and-go traffic analysis: theoretical properties, environmental impacts and oscillation mitigation. Transp. Res. Part B Methodol. 70, 319–339.
Li, X., Ouyang, Y., 2011. Characterization of traffic oscillation propagation under nonlinear car-following laws. Transp. Res. Part B 45 (9), 1346–1361.
Li, X., Peng, F., Ouyang, Y., 2010. Measurement and estimation of traffic oscillation properties. Transp. Res. Part B 44 (1), 1–14.
Li, X., Wang, X., Ouyang, Y., 2012. Prediction and field validation of traffic oscillation propagation under nonlinear car-following laws. Transp. Res. Part B 46 (3), 409–423.
Li, Z., Elefteriadou, L., Ranka, S., 2014b. Signal control optimization for automated vehicles at isolated signalized intersections. Transp. Res. Part C Emerg. Technol. 49, 1–18.
Lu, X.-Y., Shladover, S., 2014. Review of variable speed limits and advisories: theory, algorithms, and practice. Transp. Res. Record J. Transp. Res. Board (2423) 15–23.
Ma, J., Li, X., Zhou, F., Hu, J., Park, B.B., 2017. Parsimonious shooting heuristic for trajectory design of connected automated traffic part ii: computational issues and optimization. Transp. Res. Part B Methodol. 95, 421–441.
Ma, J., Li, X., Shladover, S., Lu, X.Y., Dailey, D., 2016. Freeway speed harmonization. IEEE Trans. Intell. Veh. 1 (1), 1–12.
Mandava, S., Boriboonsomsin, K., Barth, M., 2009. Arterial velocity planning based on traffic signal information under light traffic conditions. In: Proceedings of the Twelfth International IEEE Conference on Intelligent Transportation Systems (ITSC). IEEE, pp. 1–6.
Sanchez, M., Cano, J.-C., Kim, D., 2006. Predicting traffic lights to improve urban traffic fuel consumption. In: Proceedings of the Sixth International Conference on ITS Telecommunications.
Smith, C.C., McGehee, D.Y., Healey, A.J., 1978. The prediction of passenger riding comfort from acceleration data. J. Dyn. Syst. Meas. Control 100 (1), 34–41.
Spiliopoulou, A.D., Papamichail, I., Papageorgiou, M., 2009. Toll plaza merging traffic control for throughput maximization. J. Transp. Eng. 136 (1), 67–76.
Sun, W., Zheng, J., Liu, H.X., 2017. A capacity maximization scheme for intersection management with automated vehicles. Transp. Res. Procedia 23, 121–136.
Tian, J., Jiang, R., Jia, B., Gao, Z., Ma, S., 2016. Empirical analysis and simulation of the concave growth pattern of traffic oscillations. Transp. Res. Part B Methodol. 93, 338–354.
Tielert, T., Killat, M., Hartenstein, H., Luz, R., Hausberger, S., Benz, T., 2010. The impact of traffic-light-to-vehicle communication on fuel consumption and emissions. In: Proceedings of the Internet of Things (IOT). IEEE, pp. 1–8.
Trayford, R., Doughty, B., van der Touw, J., 1984a. Fuel economy investigation of dynamic advisory speeds from an experiment in arterial traffic. Transp. Res. Part A General 18 (5), 415–419.
Trayford, R., Doughty, B., Wooldridge, M., 1984b. Fuel saving and other benefits of dynamic advisory speeds on a multi-lane arterial road. Transp. Res. Part A General 18 (5), 421–429.
Treiber, M., Kesting, A., 2013. Traffic flow dynamics. Traffic Flow Dynamics: Data, Models and Simulation ISBN 978-3-642-32459-8. Springer-Verlag Berlin Heidelberg 1.
Van Arem, B., van Driel, C., Visser, R., 2006. The impact of cooperative adaptive cruise control on traffic-flow characteristics. IEEE Trans. Intell. Transp. Syst. 7 (4), 429–436.
Von Stryk, O., Bulirsch, R., 1992. Direct and indirect methods for trajectory optimization. Ann. Oper. Res. 37 (1), 357–373.
Wang, M., Daamen, W., Hoogendoorn, S.P., van Arem, B., 2014a. Rolling horizon control framework for driver assistance systems. part I: mathematical formulation and non-cooperative systems. Transp. Res. Part C Emerg. Technol. 40, 271–289.
Wang, M., Daamen, W., Hoogendoorn, S.P., van Arem, B., 2014b. Rolling horizon control framework for driver assistance systems. part II: cooperative sensing and cooperative control. Transp. Res. Part C Emerg. Technol. 40, 290–311.
Wang, M., Daamen, W., Hoogendoorn, S.P., van Arem, B., 2016. Connected variable speed limits control and car-following control with vehicle-infrastructure communication to resolve stop-and-go waves. J. Intell. Transp. Syst. 20 (6), 1–14.
Wei, Y., Liu, J., Li, P., Zhou, X., 2016. Longitude trajectory optimization for autonomous vehicles: an approach based on simplified car-following model. In: Proceedings of the Ninety-Fifth Annual Meeting on Transportation Research Board.
Wu, G., Boriboonsomsin, K., Zhang, W.-B., Li, M., Barth, M., 2010. Energy and emission benefit comparison of stationary and in-vehicle advanced driving alert systems. Transp. Res. Record J. Transp. Res. Board 16 (5), 98–106.
Wu, X., He, X., Yu, G., Harmandayan, A., Wang, Y., 2015. Energy-optimal speed control for electric vehicles on signalized arterials. IEEE Trans. Intell. Transp. Syst. PP (99), 1–11. doi:10.1109/TITS.2015.2422778.
Xu, Z., Wang, Y., Wang, G., Li, X., Hao, R., An, Y., Zhao, X., 2018. Trajectory optimization for a connected automated traffic stream: comparison between exact model and fast heuristics. Working paper, Chang'an University & University of South Florida.
Yang, H., Jin, W.-L., 2014. A control theoretic formulation of green driving strategies based on inter-vehicle communications. Transp. Res. Part C Emerg. Technol. 41, 48–60.
Yang, K., Guler, S.I., Menendez, M., 2016. Isolated intersection control for various levels of vehicle technology: conventional, connected, and automated vehicles. Transp. Res. Part C Emerg. Technol. 72, 109–129.
Yao, H., Cui, J., Li, X., Wang, Y., An, S., 2018. A trajectory smoothing method at signalized intersection based on individualized variable speed limits with location optimization. Transp. Res. Part D Transp. Environ. 62, 456–473.
Zhou, F., Li, X., Ma, J., 2017. Parsimonious shooting heuristic for trajectory design of connected automated traffic part i: theoretical analysis with generalized time geography. Transp. Res. Part B Methodol. 95, 394–420.
Zohdy, I.H., Rakha, H.A., 2016. Intersection management via vehicle connectivity: the intersection cooperative adaptive cruise control system concept. J. Intell. Transp. Syst. 20 (1), 17–32.