Transportation Research Part B 119 (2019) 1–21

Contents lists available at ScienceDirect

Transportation Research Part B

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

Parsimonious trajectory design of connected automated traffic

Li Li[a], Xiaopeng Li[b,*]

[a] Department of Automation, BNRist, Tsinghua University, Beijing, 100084, China
[b] Department of Civil and Environmental Engineering, University of South Florida, FL 33620, USA

ARTICLE INFO

Article history:
Received 17 December 2017
Revised 9 October 2018
Accepted 12 November 2018
Available online 22 November 2018

Keywords:
Trajectory planning
Connected vehicles
Automated vehicles
Feasibility conditions
Parsimonious

ABSTRACT

One challenging problem about connected automated vehicles is to optimize vehicle trajectories considering realistic constraints (e.g. vehicle kinematic limits and collision avoidance) and objectives (e.g., travel time, fuel consumption). With respect to communication cost and implementation difficulty, parsimonious trajectory planning has attracted continuous interests. In this paper, we first analyze the feasibility conditions for a general continuous-time trajectory planning problem and then propose an analytical solution method for two important boundary trajectory problems. We further propose a discrete-time model with a more general objective function and a certain sparsity requirement that helps parsimonious planned trajectories. This sparsity requirement is implemented with a l_1 norm regulatory term appended to the objective function. Numerical examples are conducted on several representative applications and show that the proposed design strategy is effective.

1. Introduction

Connected and automated vehicles (CAV) hold a great potential to improve traffic safety and efficiency (Li and Wang, 2007) and have gained rapidly increasing interests during the last decade. The connected vehicle technology enables real-time information sharing between individual vehicles and infrastructures; and the automated vehicle technology enables vehicles to implement complicated and concise actions with the aid of advanced sensors and control actuators. The combination of connected vehicle and automated vehicle technologies further enables appropriate coordination of multiple vehicles to optimize both drivers/passengers' experience and the overall traffic performance (Lee and Park, 2012; Li et al., 2014c).

A number of pioneering studies have been conducted to realize the full potential of CAV based traffic operations by controlling CAV movements at particular facilities. For example, researchers studied how to utilize CAVs to improve mobility and safety at intersections (Li and Wang, 2006; Dresner and Stone, 2008; Li et al., 2014b; Li et al., 2014c; Guler et al., 2014; Ma et al., 2017; Zhou et al., 2017; Li and Zhou, 2017b; Wei et al., 2017; Zheng and Liu, 2017; Cassandras, 2018; Yu et al., 2018), to coordinate vehicles going through a merge ramp (Rios-Torres and Malikopoulos, 2017a,b; Letter and Elefteriadou, 2017), and to reduce fuel consumption and environmental impacts along highway segments (Ahn et al., 2013; Yang and Jin, 2014). These various applications are centered at a common problem of optimizing trajectories of a stream of vehicles to achieve a specific objective (e.g., reducing the overall fuel consumption of the studied vehicles or absorbing a jam to improve traffic efficiency).

* Corresponding author.
E-mail address: xiaopengli@usf.edu (X. Li).

https://doi.org/10.1016/j.trb.2018.11.006
0191-2615/

The trajectory planning problems studied by researchers in traffic flow field differ from the trajectory planning problems studied by researchers in vehicle control field. For vehicle control, we are usually interested in two-dimensional trajectory planning problems for individual vehicles, in which the longitudinal and lateral motion dynamics are considered. As summarized in (Li and Wang, 2007; Katrakazas et al., 2015; González et al., 2016; Liu et al., 2017), the major problem of these trajectory planning methods is that we must fully consider vehicle dynamics heavily affecting the geometry property of the candidate trajectories. The planning time horizon is really short (e.g. 10s –15 s) for vehicle control studies, since we do not need to consider the far future, because the driving environment may change quickly.

For traffic flow, we are usually interested in one-dimensional trajectory planning problems in which we consider the long-term expected movements of multiple vehicles. The planning time horizon is really long (e.g. 50s–100 s) for traffic flow studies. The dynamic models of vehicles are often simplified for two reasons. First, it is really hard to solve the resulting nonlinear optimization problems with many decision variables (note that we often). Second, it is widely assumed the modeling errors would not contribute too much for the loss of objective values, when a long planning time horizon is considered.

However, this central problem has not been adequately addressed in the literature. Due to the complex nature of multi-trajectory optimization problems, most existing studies either over-simplified trajectories (e.g., allowing infinite accelerations to have sudden speed jumps) or relied on complicated numerical procedures not suitable for real-time applications or ad-hoc heuristics without optimality assurance. There are two limitations of the traditional studies on this topic. First, numerical approaches alone cannot provide sufficient analytical knowledge and intuitive understanding of the problem and solution structures or fundamental insights, which may hinder discovering certain underlying managerial insights in real-world applications. Second, the obtained trajectories from these approaches may not be smooth and comfortable enough for individual vehicles to follow in practice, e.g., due to speed jumps or excessive acceleration and deceleration movements. While a few recent studies (Ma et al., 2017; Zhou et al., 2017) improved trajectory smoothness by approximating trajectories with only a simple piecewise quadratic function, the solution approaches are basically heuristics and cannot guarantee to find the exact optimal solution.

To address these two major limitations, this paper first formulates common constraints of the investigated continuous model for a general class of trajectory optimization problems. With the formulation, it analyzes theoretical feasibility conditions of the model on both an isolated individual vehicle and the entire vehicle stream considering car-following safety and communication limits. Based on the relevant theoretical properties, we discover analytical approaches to solve the exact optima to two important boundary trajectory problems. We show that any feasible trajectories for this general problem will be enclosed within an envelope formed by these two sets of boundary solutions. These theoretical results provide fundamental knowledge on the problem structure and solution properties.

Further, we propose a discrete-time counterpart model with a more general objective function and certain sparsity requirement that helps parsimonious planned trajectories. This sparsity requirement is implemented with a l_1 norm regulatory term appended to the objective function, which helps mitigate or eliminate speed differences between consecutive time points. This will improve the smoothness and parsimony of the obtained trajectories. With this problem formulation, we can vary the weight of the l_1 norm term to get different levels of parsimony of the obtained trajectory solution. Since this model is formulated as a mathematical program with all linear constraints, as long as the objective is linear or convex, it can be solved to the exact optimum with state-of-the-art linear programming and convex programming algorithms.

A set of numerical examples are conducted to illustrate the applications of this model to several relevant problems. For some traffic applications, e.g., jam absorption, the objective functions are linear so that the obtained optimization problems are linear programming problems that can be efficiently solved with off-the-shelf linear programming solvers. For some other traffic applications with nonlinear objective functions, e.g. eco-driving, we use the efficient Frank-Wolf algorithm, because it approximates a model even with a complex nonlinear objective into a series of simple linear programming sub-problems and thus has fast convergence rates and superior computational efficiency (Clarkson, 2010; Jaggi, 2013; Lacoste-Julien, 2016). The results show that the proposed design strategy is effective in solving all these problems and can yield relatively smooth trajectories suitable for real-world implementation.

The rest of this paper is organized as follows. *Section 2* presents the problem statement in a continuous form, analyzes theoretical properties and proposes analytical solutions to two special problems with boundary objectives. *Section 3* formulates this problem in a discrete-time model with a general objective function including a l_1-penalty term for trajectory parsimony. *Section 4* illustrates how to apply this model to three problems with different focuses. Finally, *Section 5* concludes the paper and discusses future research directions.

2. Continuous modeling and analysis

This section formulates a continuous modeling framework for the class of investigated trajectory optimization problems and analyzes its theoretical properties. *Section 2.1* states the basic notation and formulates the major constraints. *Section 2.2* analyzes the local feasibility and associated bounds for a single trajectory without considering safety constraints that couple consecutive trajectories. Based on the local feasibility results, *Section 2.3* investigates global feasibility for a stream of vehicles following one another incorporating the safety constraints. Through such global feasibility analysis, it also formulates exact analytical solutions to two special case problems with extreme objective values, which also serve as the upper and lower bounds to the global feasible region, respectively. Note that since the continuous model focus more on

feasibility analysis on the constraints with assist from simple objective functions that only highlight the bounds, we will instead leave analysis on the l_1-penalty term on more general objective functions to the next section on the discrete model. For the convenience of readers, we added a notation list in the Appendix.

2.1. The decision variables and the constraints

In this paper, we consider the following continuous-time trajectory planning problem within a control time horizon from time 0 to time T. We consider that a total of I identical vehicles, indexed by $i \in \mathcal{J} = \{1, 2, \cdots, I\}$, follow one another along a one-lane road. We set the indexes increasing upstream, and thus vehicle 1 is the lead vehicle and vehicle I is the last vehicle in this traffic stream. The decision variables of the investigated trajectory planning problem are positions for all vehicles at all time points, i.e., $\mathbf{s} := \{s^i(t)\}_{t\in[0,T],i\in\mathcal{J}}$, which fully characterize the dynamics of not only individual vehicles but also the collective traffic stream. These variables are subject to a number of constraints as formulated below.

1. The acceleration (or deceleration) rate of each vehicle at a time point, formulated as the second order derivative of the corresponding vehicle position, is bounded by mechanical limits as follows:

$$-d_{\max} \le \ddot{s}^i(t) \le a_{\max} \ \forall i, t \tag{1}$$

where $a_{\max}$ denotes the maximum acceleration rate and $-d_{\max}$ denotes the minimum acceleration rate (or the negation of the maximum deceleration rate). Here $a_{\max}$ and $d_{\max}$ should be both positive numbers.

2. The speed of each vehicle at a time point, formulated as the first-order derivative of the corresponding vehicle position, shall be no greater than the speed limit $v_{\max}$. Further, since we usually define that the vehicle cannot move backward, we also have the following monotonic condition:

$$0 \le \dot{s}^i(t) \le v_{\max} \ \forall i, t \tag{2}$$

3. We postulate that a vehicle i and the corresponding preceding vehicle $i-1$ has to be separated with a sufficient gap to avoid collisions. Also, the following vehicle needs to trail the preceding vehicle within a communication distance range to ensure they are connected. Thus we require their gap should be always no greater than a maximum limit to ensure reliable communication (Zhang et al., 2015; Abboud et al., 2016), as formulated below:

$$g_{\min} \le s^{i-1}(t) - s^i(t) - l \le g_{\max} \ \forall i \in \mathcal{J}\backslash\{1\}, t \tag{3}$$

where l denotes the length of a vehicle. $g_{\min}$ is the minimum distance gap to ensure safety and $g_{\max}$ is the maximum distance gap that allows reliable communication. Here, we assume that the influence of the communication lag between two consecutive vehicles is appropriately considered and modeled as a part of $g_{\min}$. Note that constraints (3) are the coupling constraints that link these multiple trajectories into on integrated problem. Without these constraints, investigated problem will reduce to an individual trajectory optimization problem, but the output trajectories may collide with each other, which is infeasible for realistic implementation.

4. We assume that the boundary conditions (the initial position and speed at time 0 and the final position at time T) are given. Previous studies (Li et al., 2018; Ma et al., 2017) claimed that similar boundary conditions with fixed locations and speeds are actually quite general and can be applied to trajectory optimization problems or their sub-problems on various infrastructure geometries, such as intersections, merge points, freeway bottleneck controls, etc. Nonetheless, different from the existing studies (Li et al., 2018), we do not constrain the departing speed, which allows for more general trajectory control schemes. These boundary condition constraints are formulated as:

$$s^i(0) = s^i_{\text{begin}}, \dot{s}^i(0) = v^i_{\text{begin}}, s^{i-}_{\text{final}} \le s^i(T) \le s^{i+}_{\text{final}}, \forall i, t, \tag{4}$$

which guarantees that the initial position and speed of vehicle i at time 0 are s^i_{begin} and v^i_{begin}, respectively, and the final position of vehicle i at time T has to be within $[s^{i-}_{\text{final}}, s^{i+}_{\text{final}}]$. Location bounds $s^{i-}_{\text{final}}, s^{i+}_{\text{final}}$ may be adjusted based on the needs of the specific problems. If no restriction is set, s^{i-}_{final} can be set as the minimum location (by decelerating with rate $-d_{\max}$ until stopping, i.e., $s^-_{s^i_{\text{begin}} v^i_{\text{begin}} 0}(T)$ according to the notation in the next subsection) and s^{i+}_{final} as the maximum location (by accelerating with rate $a_{\max}$ until reaching speed $v_{\max}$, and then maintaining maximum speed $v_{\max}$ if needed, i.e., see $s^+_{s^i_{\text{begin}} v^i_{\text{begin}} 0}(T)$ according to the notation in the next subsection) that vehicle i can reach kinematically at time T. Thus without loss of generality, this study postulates $s^{i-}_{\text{final}} \ge s^-_{s^i_{\text{begin}} v^i_{\text{begin}} 0}(T)$ and $s^{i+}_{\text{final}} \le s^+_{s^i_{\text{begin}} v^i_{\text{begin}} 0}(T)$.

Ineq. (1)-(4) compose the complete set of constraints of the investigated problem. Note that all the proposed constraints are linear constraints. In addition, for some applications, the lead vehicle's trajectory is given as

$$s^1(t) = s^1_t, \forall t, \tag{5}$$

where $\{s^1_t\}$ are the given locations of the lead vehicle at all time points. This is a set of optional constraints depending on the needs of actual applications.

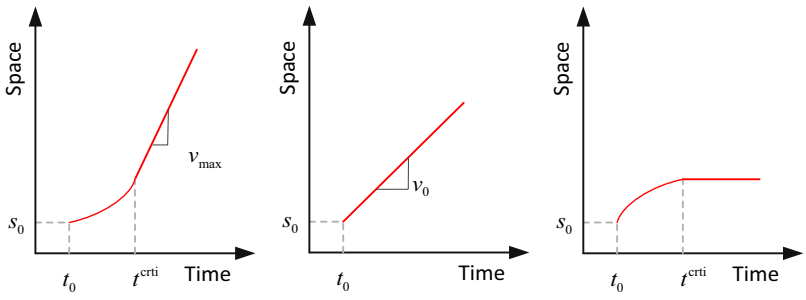


Fig. 1. An illustration of three kinds of locally feasible trajectories.

Remark 1. In the following feasibility analysis we only consider constraints (1)-(4). Whereas the analysis results can be easily adapted to the case with constraints (5) by adding an additional bound, and thus we omit the analysis for constraints (5).

2.2. Local feasibility analysis

This section investigates *local feasibility* for a generic vehicle i regarding physical constraints (1)-(2) and boundary conditions (4) (yet vehicle interaction constraints are ignored for now).

Definition 1. We say that s^i is *locally feasible* (or *locally infeasible*) if there exists a solution to constraints (1)-(2) and (4) associated with vehicle i (or there does not exist such a solution).

To facilitate the math notation, let $s_{s_0v_0t_0a_0}(t)$ denote the trajectory that starts at state (s_0,v_0,t_0) (this triple denotes location s_0, speed v_0, and time t_0) and initially accelerates with rate a_0. We only consider $a_0 \in [-d_{\max},a_{\max}]$ due to acceleration limits (1) and $v_0 \in [0, v_{\max}]$ due to speed constraints (2). As shown in Fig. 1, if $a_0 > 0$, after this trajectory accelerates to speed $v_{\max}$, it will cruise at this maximum speed. If $a_0 < 0$, after this trajectory decelerates to speed 0, it will remain being stopped for the rest of the time. With this, $s_{s_0v_0t_0a_0}(t)$ can be formulated as follows

$$\text{(a) } s_{s_0v_0t_0a_0}(t) = \begin{cases} s_0 + v_0(t-t_0) + 0.5a_0(t-t_0)^2, \forall t \in [t_0, t^{\text{crit}}], \\ s_{s_0v_0t_0a_0}(t^{\text{crit}}) + v_{\max}(t - t^{\text{crit}}), \forall t > t^{\text{crit}}, \end{cases} \text{ if } a_0 > 0;$$

$$\text{(b) } s_{s_0v_0t_0a_0}(t) = s_0 + v_0(t-t_0), \forall t \geq t_0, \text{ if } a_0 = 0;$$

$$\text{(c) } s_{s_0v_0t_0a_0}(t) = \begin{cases} s_0 + v_0(t-t_0) + 0.5a_0(t-t_0)^2, \forall t \in [t_0, t^{\text{crit}}], \\ s_{s_0v_0t_0a_0}(t^{\text{crit}}), \forall t > t^{\text{crit}}, \end{cases} \text{ if } a_0 < 0$$

where the critical time between the quadratic segment and the linear segment is defined as $t^{\text{crit}}(v_0, t_0, a_0) := \begin{cases} t_0 + \frac{v_{\max}-v_0}{a_0}, if\ a_0 > 0, \\ t_0 + \frac{v_0}{-a_0}, if\ a_0 < 0. \end{cases}$

We highlight two special trajectories with extreme accelerations, i.e., *upper bound trajectory* $s^+_{s_0v_0t_0}(t) := s_{s_0v_0t_0a_{\max}}(t)$ and *lower bound trajectory* $s^-_{s_0v_0t_0}(t) := s_{s_0v_0t_0(-d_{\max})}(t)$. These definitions allow us to state the necessary and sufficient conditions for local feasibility given below.

Proposition 1. *s^i is locally feasible if and only if*

$$\left[s^{i-}_{\text{final}}, s^{i+}_{\text{final}}\right] \cap \left[s^-_{s^i_{\text{begin}}v^i_{\text{begin}}0}(T), s^+_{s^i_{\text{begin}}v^i_{\text{begin}}0}(T)\right] \neq \emptyset. \tag{6}$$

Proof: First, it is easy to see that if $[s^{i-}_{\text{final}}, s^{i+}_{\text{final}}] \cap [s^-_{s^i_{\text{begin}}v^i_{\text{begin}}0}(T), s^+_{s^i_{\text{begin}}v^i_{\text{begin}}0}(T)] = \emptyset$, any trajectory that connects initial condition $(s^i_{\text{begin}}, v^i_{\text{begin}}, 0)$ and final condition $(s^i(T),\cdot,\ T)$ cannot satisfy both physical constraints (1)-(2). Thus no feasible trajectory exists in either of these conditions. This simply proves the necessity.

For the sufficiency, when condition (6) holds, let s^i_{final} be a value in $[s^{i-}_{\text{final}}, s^{i+}_{\text{final}}] \cap [s^-_{s^i_{\text{begin}}v^i_{\text{begin}}0}(T), s^+_{s^i_{\text{begin}}v^i_{\text{begin}}0}(T)]$. Then we can construct a trajectory as $s^i_\lambda(t) = \lambda s^+_{s^i_{\text{begin}}v^i_{\text{begin}}0}(t) + (1-\lambda)s^-_{s^i_{\text{begin}}v^i_{\text{begin}}0}(t)$ for some $\lambda \in [0, 1]$. Obviously $s^i_\lambda(t)$ satisfies physical constraints (1)-(2) for any $\lambda \in [0, 1]$. Further, we can see that $s^i_\lambda(T)$ continuously increases from $s^-_{s^i_{\text{begin}}v^i_{\text{begin}}0}(T)$ to $s^+_{s^i_{\text{begin}}v^i_{\text{begin}}0}(T)$ as λ increases from 0 to 1, and thus there exists some $\lambda \in [0, 1]$ such that $s^i_\lambda(T) = s^i_{\text{final}}$. With this λ value, trajectory s^i_λ shall satisfy boundary condition (4) as well. This proves the sufficiency. □

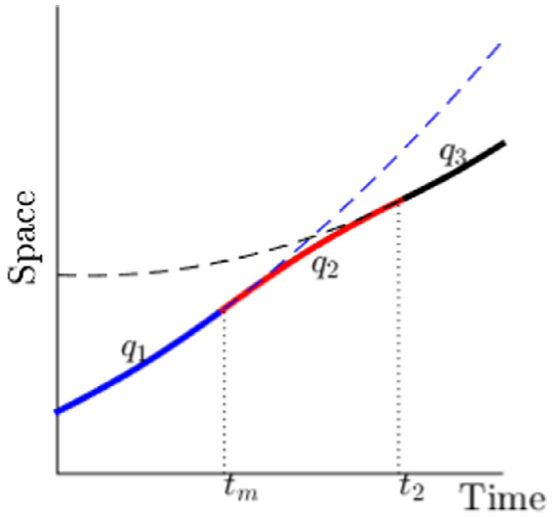


Fig. 2. An illustration of the merging operation.

When condition (6) holds, we will construct bounding trajectories for s^i. The trajectory that bounds all possible trajectories associated with vehicle i satisfying physical constraints (1)-(2) and boundary condition (4) from above (or from below) is referred as *the local upper* (*or lower*) *bound* for s^i and is denoted as s^{i+} (or s^{i-}). Conceptually, it is not difficult to imagine the shapes of these trajectories: s^{i+} first accelerates at $a_{\max}$ as long as possible (before reaching $v_{\max}$ or having to decelerate, whichever comes first), and finally decelerates at $-d_{\max}$ in the last moment just to ensure vehicle i stops at s^{i+}_{final} at time T; s^{i-} first decelerates at $-d_{\max}$ as long as possible (before a full stop or having to accelerate, whichever comes first, and finally accelerates at $a_{\max}$ during the right time window such that vehicle i passes location s^{i-}_{final} at time T.

To formulate these bounds, we first introduce the following terms.

Definition 2. We define *quadratic trajectory* q_{svta} as a quadratic function anchored at state (s, v, t) with an acceleration rate of a, as formulated:

$$q_{svta}(t') := s + v(t'-t) + 0.5a(t'-t)^2, \forall t'.$$

In this study, we only need to investigate segments of quadratic trajectories satisfying physical limits (1)-(2) in the investigated time horizon, and thus all quadratic trajectories mentioned in the following analysis should satisfy $v \in [0, v_{\max}]$, $a \in [-d_{\max}, a_{\max}]$, $t \in [0, T]$ unless stated otherwise.

Definition 3. We define the *merging operation* between two quadratic trajectories as follows. The first trajectory $q_{s_1v_1t_1a_1}$, short as q_1, is fixed at starting point (s_1,v_1,t_1) and shoots forward. The second trajectory $q_{q_{s_3v_3t_3a_3}(t_2)\dot{q}_{s_3v_3t_3a_3}(t_2)t_2a_2}$, short as q_2, goes backward with its initial point sliding along another quadratic function $q_{s_3v_3t_3a_3}$. As illustrated in Fig. 2, this merging operation is essential to solve a quadratic segment q_2 with acceleration a_2 that just gets tangent to both given trajectories q_1 and q_3. Thus in a sense q_2 is a segment that smoothly "merges" two intersected trajectories q_1 and q_3.

We only consider the case that q_2 is on the right side of q_1 (and thus obtain $t_2 \geq t_1$). Starting time t_2, which can be also denoted as function $t_2(s_1v_1t_1a_1,s_3v_3t_3a_3,a_2)$, can be analytically solved as follows for non-trivial cases.

$$t_2 = \begin{cases} -\frac{B}{2A} + \frac{\sqrt{B^2-4AC}}{2|A|}, \text{if } A > 0 \text{ and } a_1 - a_3 > 0; \\ -\frac{B}{2A} + \frac{\sqrt{B^2-4AC}}{2|A|}, \text{if } A > 0 \text{ and } a_1 - a_3 < 0; \\ -\frac{C}{B}, \text{if } A = 0 \end{cases}$$

where $A=(a_1-a_3)(a_2-a_3)$, $B=2(a_2-a_3)(v_1-v_3-a_1t_1+a_3t_3)$, $C=(v_1-v_3-a_1t_1+a_3t_3)^2-(2a_1-2a_2)(\frac{a_1t_1^2}{2}-\frac{a_3t_3^2}{2}+t_3v_3-v_1t_1+s_1-s_3)$.

Based on this solution, we can also analytically solve the tangent point between q_1 and q_2, denoted by t_m or function $t_m(s_1v_1t_1a_1,s_3v_3t_3a_3,a_2)$, as follows:

$$t_m = \frac{v_3 - v_1 + a_1t_1 - a_2t_2 + a_3(t_2 - t_3)}{a_1 - a_2}.$$

This merging operation can be further generalized to the case where q_1 is compound trajectory $s_{s_1v_1t_1a_1}$. This can be simply done by first set q_1 as quadratic trajectory $q_{s_1v_1t_1a_1}$ and solve the tangent point t_m. If $t_m \leq t^{\text{crit}}(v_1,t_1,a_1)$, this solves the

merging operation as well. Otherwise, we can just set q_1 as a linear trajectory $q_{s_{s_1 v_1 t_1 a_1}(t^{\text{crit}}) v_{\max} t^{\text{crit}} 0}$ and solve t_m again, which should yield the solution for compound trajectory $s_{s_1 v_1 t_1 a_1}$. Following this logic, we can generalize this merging operation for cases where both q_1 and q_3 are general trajectories each composed by a finite number of quadratic segments.

With the above analytical formulations for the merging operation, when condition (5) holds, we can analytically solve upper bound trajectory s^{i+} with Algorithm 1.

Algorithm 1. The analytical algorithm for the upper bound trajectory.

Initialization: We use superscript UB to associate certain terms with the upper bound trajectory. Set $q_1^{\text{UB}} = s_{s^i_{\text{begin}} v^i_{\text{begin}} 0 a_{\max}}$ and $q_2^{\text{UB}} = q_{s^{i+}_{\text{final}} v^i_{\text{final}} T(-d_{\max})}$ where v^i_{final} is a variable. Let $q(t_1\colon t_2)$ denote the segment of trajectory q between time t_1 and time t_2.

Step 1: Solve v^i_{final} such that q_1^{UB} and q_2^{UB} get tangent. If $v^i_{\text{final}} \in [0, v_{\max}]$, then s^{i+} is simply solved as $[q_1^{\text{UB}}(t_1 : t_m^{\text{UB}}), q_2^{\text{UB}}(t_m^{\text{UB}} : T)]$ where t_m^{UB} is the tangent time point between q_1^{UB} and q_2^{UB}. This solution is illustrated as the red curve in Fig. 3(a).

Step 2: Otherwise, fix q_1^{UB} as $s_{s^i_{\text{begin}} v^i_{\text{begin}} 0 a_{\max}}$, make q_3^{UB} as horizontal line $q_{s^{i+}_{\text{final}} 0 T 0}$, and set the merging acceleration as $-d_{\max}$. Conduct the merging operation between q_1^{UB} and q_3^{UB}, and solve starting point t_2^{UB}, tangent time t_m^{UB} and merging segment $q_2^{\text{UB}} := q_{s^{i+}_{\text{final}} 0 t_2(-d_{\max})}$. Then s^{i+} shall be $[q_1^{\text{UB}}(t_1 : t_m^{\text{UB}}), q_2^{\text{UB}}(t_m^{\text{UB}} : t_2^{\text{UB}}), q_3^{\text{UB}}(t_2^{\text{UB}} : T)]$. This solution is illustrated as the red curve in Fig. 3(b).

Similarly, s^{i-} can be analytically solved with Algorithm 2.

Algorithm 2. The analytical algorithm for the lower bound trajectory.

Initialization: We use superscript LB to associate certain terms with the lower bound trajectory. Set $q_1^{\text{LB}} = s_{s^i_{\text{begin}} v^i_{\text{begin}} 0(-d_{\max})}$ and $q_2^{\text{LB}} = q_{s^{i-}_{\text{final}} v^i_{\text{final}} T a_{\max}}$ where v^i_{final} is a variable.

Step 1: Solve v^i_{final} such that $q_1^{\text{LB}}$and q_2^{LB} get tangent. If $v^i_{\text{final}} \in [0, v_{\max}]$, then s^{i-} is simply solved as $[q_1^{\text{LB}}(t_1 : t_m^{\text{LB}}), q_2^{\text{LB}}(t_m^{\text{LB}} : T)]$ where t_m^{LB} is the tangent time point between q_1^{LB} and q_2^{LB}. This solution is illustrated as the blue curve in Fig. 3(a).

Step 2: Otherwise, fix q_1^{LB} as $s_{s^i_{\text{begin}} v^i_{\text{begin}} 0(-d_{\max})}$, make q_3^{LB} as angled line $q_{s^{i-}_{\text{final}} v_{\max}\ T 0}$, and set the merging acceleration as $a_{\max}$. Conduct the merging operation between q_1^{LB} and q_3^{LB}, and solve starting point t_2^{LB}, tangent time t_m^{LB} and merging segment $q_2^{\text{LB}} := q_{(s^{i-}_{\text{final}} - v_{\max}\ (T-t_2)) v_{\max}\ t_2 a_{\max}}$. Then, s^{i-} is $[q_1^{\text{LB}}(t_1 : t_m^{\text{LB}}), q_2^{\text{LB}}(t_m^{\text{LB}} : t_2^{\text{LB}}), q_3^{\text{LB}}(t_2^{\text{LB}} : T)]$. This solution is illustrated as the blue curve in Fig. 3(b).

Note that both s^{i+} and s^{i-} are locally feasible with respect to vehicle i. Further, any vehicle i's locally feasible trajectory s^i must satisfy $s^{i-}(t) \le s^i(t) \le s^{i+}(t)$, $\forall t \in [0, T]$. Therefore, s^{i+} and s^{i-} are the tight lower and upper bounds, respectively, to the locally feasible region specified by constraints (1), (2) and (4) with respect to vehicle i.

2.3. Global feasibility analysis and analytical solutions to special cases

Next, we will specify the global feasible region considering vehicle interaction constraints (3) in addition to other constraints. To investigate this, we examine two special problems. The first problem solves the *most aggressive trajectory* formulated as follows:

$$\max_{s^i(t)} \sum_{i=1}^{I} \int_{t=0}^{T} s^i(t) dt. \tag{7}$$

subject to constraints (1)-(4). The second problem solves the *most conservative trajectory set*, formulated as follows:

$$\min_{s^i(t)} \sum_{i=1}^{I} \int_{t=0}^{T} s^i(t) dt. \tag{8}$$

subject to constraints (1)-(4).

If the lead vehicle is fixed, then constraints (5) will be added to the respective problem (7) or (8). Without loss of generality, we assume fixed lead trajectory $\{s^1(t)\}$ is composed of a finite number of quadratic segments (since any physically meaningful trajectory can be approximated with a finite piecewise quadratic function within any arbitrarily small error) complying with kinematic constraints (1) and (2). Here, aggressiveness can be interpreted as how fast a vehicle desires to pass a location. In the most aggressive trajectory set, every vehicle i will pass each location on the highway segment at the earliest time as kinematic constraints (1)-(2), safety and communication constraints (3) with respective to preceding vehicle i, and boundary constraints (4) (or (5)) allow. This tends to maximize traffic volume and minimize travel delay (e.g., every vehicle i wants to reach s^{i+}_{final} as soon as possible). While in the most conservative trajectory set, each vehicle on the contrary desires to pass each location at the latest time allowed by these constraints. This tends to minimize traffic volume and maximize travel delay (e.g., every vehicle i wants to stay back at s^{i-}_{final} by the end). Please also see Cassidy and Windover (1998) for the discussion of driving aggressiveness. Note that Problem (7) or (8) has a solution if and only if the investigated general problem is feasible.

We discover the following approach that can analytically solve Problem (7) and (8), when there exist feasible solutions to them. This approach can be also used to test whether the investigated problem is feasible. Before introducing this analytical solution approach, we define the following terms.

Definition 4. We define the *upper-bound merging operation* (UMO) for a series of trajectories, $\boldsymbol{s}$: $=\{s_1, s_2, \cdots, s_L\}$ as follows.

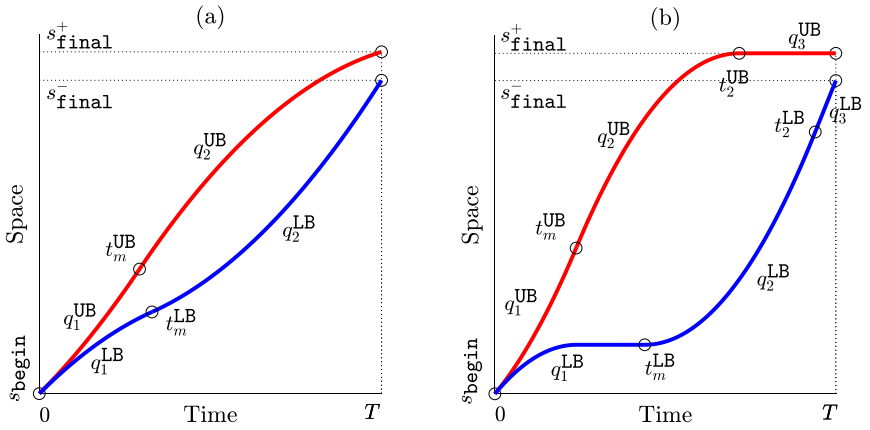


Fig. 3. Illustration of the upper (red) and lower (blue) bound trajectories: (a) for Step 1′s solution and (b) for Step 2′s solution.

UMO-0: Set s' as an empty trajectory (or an empty set). Set $t^- = 0$, $k = 1$;

UMO-1: Identify the trajectory from $\boldsymbol{s}$ with the minimum values in the neighborhood right side of t^- and denote this trajectory as s'_k, i.e. $s'_k(t) \le s_l(t),\ \forall l = 1, 2, \cdots, L, t \in [t^-, t^- + \delta], \exists \delta > 0$. Find $t^+ = \sup_{t \in (t^-,\ T]} s'_k(t) \le s_l(t),\ \forall l = 1, 2, \cdots, L$. If tie exists, select the k with the largest t^+. As shown in Fig. 3, append $s'_k(t^- : t^+)$ to s', i.e., $s' := [s', s'_k(t^- : t^+)]$, and set $t_k^- = t^-, t_k^+ = t^+$.

UMO-2: If $t^+ < T$, update $t^- = t^+$ and $k = k + 1$, and go to Step UMO-1. Otherwise, if $t^+ = T$, record $K = k$, and go to the next step.

UMO-3: Note that s' may not satisfy acceleration constraints (1). To revise it to be feasible, we repeat the merging operation between every pair of consecutive trajectory in the following. First, set $k = 1$ and $k' = k + 1$.

UMO-4: Conduct the merging operation with s'_k being q_1, $s'_{k'}$ being q_3 and $-d_{\max}$ being a_2. If the solved starting point $t_2 > t_{k'}^+$, set $k' = k' + 1$. If the solve merging point $t^m < t_k^-$, set $k = k - 1$. Repeat this step until finding $t_m \in [t_k^- : t_k^+]$ and $t_2 \in [t_k^- : t_k^+]$.

UMO-5: In s', replace kinky segment $[s_k(t_k^-, t_k^+), s_{k'}(t_{k'}^-, t_{k'}^+)]$ with smooth segment $[s_k(t_k^-, t_m), q_2(t_m, t_2), s_{k'}(t_2, t_{k'}^+)]$. If k' is less than K, set $k = k'$ and $k' = k + 1$, go to Step UMO-4. Otherwise, go to the next and final step.

UMO-6: Return the UMO trajectory of $\boldsymbol{s}$, denoted by $s^{\text{UMO}}(\boldsymbol{s})$: $= s'$.

Definition 5. Conversely, we also define the lower-bound merging operation (LMO) for a series of trajectories, $\boldsymbol{s}$: $= \{s_1, s_2, \cdots, s_L\}$ as follows. The major difference is that the maximum trajectory segments are selected and the merging operations use maximum acceleration $a_{\max}$

LMO-0: Set s' as an empty trajectory (or an empty set). Set $t^- = 0$, $k = 1$;

LMO-1: Identify the trajectory from $\boldsymbol{s}$ with the maximum values in the neighborhood right side of t^- and denote this trajectory as s'_k, i.e. $s'_k(t) \ge s_l(t),\ \forall l = 1, 2, \cdots, L, t \in [t^-, t^- + \delta], \exists \delta > 0$. Find $t^+ = \sup_{t \in (t^-,\ T]} s'_k(t) \ge s_l(t),\ \forall l = 1, 2, \cdots, L$. If tie exists, select the k with the largest t^+. Append $s'_k(t^- : t^+)$ to s', i.e., $s' := [s', s'_k(t^- : t^+)]$, and set $t_k^- = t^-, t_k^+ = t^+$.

LMO-2: If $t^+ < T$, update $t^- = t^+$ and $k = k + 1$, and go to Step UMO-1. Otherwise, if $t^+ = T$, record $K = k$, and go to the next step.

LMO-3: Note that s' may not satisfy acceleration constraints (1). To revise it to be feasible, we repeat the merging operation between every pair of consecutive trajectory in the following. First, set $k = 1$ and $k' = k + 1$.

LMO-4: Conduct the merging operation with s'_k being q_1, $s'_{k'}$ being q_3 and $a_{\max}$ being a_2. If the solved starting point $t_2 > t_{k'}^+$, set $k' = k' + 1$. If the solve merging point $t^m < t_k^-$, set $k = k - 1$. Repeat this step until finding $t_m \in [t_k^- : t_k^+]$ and $t_2 \in [t_k^- : t_k^+]$.

LMO-5: In s', replace kinky segment $[s_k(t_k^-, t_k^+), s_{k'}(t_{k'}^-, t_{k'}^+)]$ with smooth segment $[s_k(t_k^-, t_m), q_2(t_m, t_2), s_{k'}(t_2, t_{k'}^+)]$. If k' is less than K, set $k = k'$ and $k' = k + 1$, go to Step UMO-4. Otherwise, go to the next and final step.

LMO-6: Return the LMO trajectory of $\boldsymbol{s}$, denoted by $s^{\text{LMO}}(\boldsymbol{s})$: $= s'$.

Fig. 4(a) illustrates UMO for four trajectories $\boldsymbol{s} = \{s_1, s_2, s_3, s_4\}$, as illustrated by the dotted curves. The blue dashed curve marks lower envelope s' before being revised by the merging operations. We see that s' now is not smooth and has two kinks marked by the blue stars. s^{UMO} is the result after smoothing s' with the merging operation. We can also see that s^{UMO} becomes the highest trajectory satisfying kinematic constraints (1) and (2) that bounds all these four trajectories from below. Similarly, Fig. 4(b) illustrates LMO for the same four trajectories. Again we see that s' before the merging operation may have kinks violating the kinematic constraints. But after the merging operation, all these kinks are smoothed out in the resulting trajectory s^{LMO}, which apparently satisfies the kinematic constraints. Again, s^{UMO} becomes the lowest trajectory satisfying the kinematic constraints that bound all these four trajectories from above.

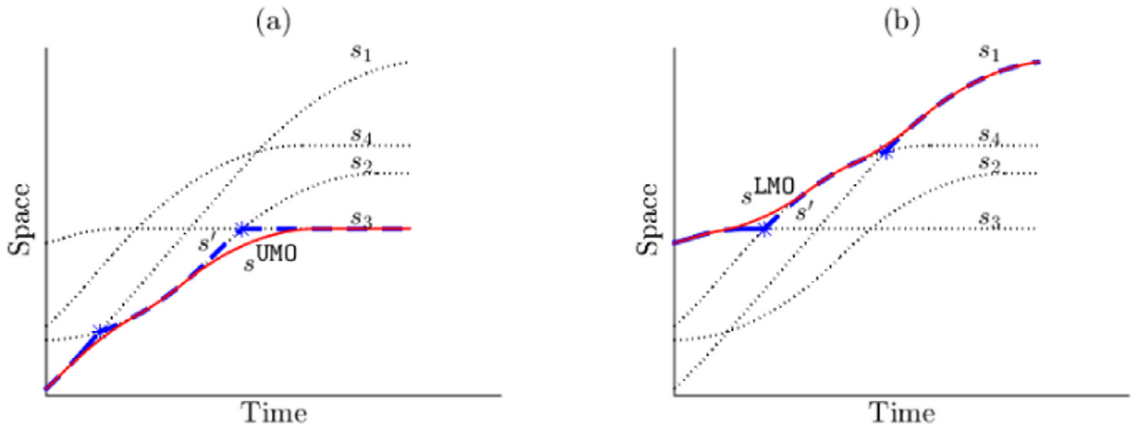


Fig. 4. An illustration of (a) UMO and (b) LMO.

Definition 6. We define the m^{th} *inner/outer shadow trajectory* of a given trajectory $q(t)$ as

$$q[m+](t) := q(t) - m(l + g_{\min}),$$
$$q[m-](t) := q(t) - m(l + g_{\max}),$$

where m is a (possibly negative) integer and superscript $+$ and $-$ indicate the inner and lower shadows, respectively. Note that $q[0+] = q[0-] = q$.

The above definition enables us to construct the upper and lower bound trajectories for problems (7) and (8) (and their variants with lead vehicle fixing constraints (5)) with the following definitions.

Definition 7. We define the upper/lower bound solution to a vehicle when the lead vehicle is not fixed (without constraints (5)) as follows:

$$\bar{s}^i := s^{\text{UMO}}\left(\left\{s^{j+}[(i-j)+]\right\}_{1\le j\le i}, \left\{s^{j+}[(i-j)-]\right\}_{i<j\le I}\right),$$
$$\underline{s}^i := s^{\text{LMO}}\left(\left\{s^{j-}[(i-j)-]\right\}_{1\le j\le i}, \left\{s^{j-}[(i-j)+]\right\}_{i<j\le I}\right).$$

Definition 8. We define the upper/lower bound solution to a vehicle when the lead vehicle is fixed (with constraints (5)) as follows:

$$\bar{s}^{i1} := s^{\text{UMO}}\left(s^1[(i-1)+], \left\{s^{j+}[(i-j)+]\right\}_{1<j\le i}, \left\{s^{j+}[(i-j)-]\right\}_{i<j\le I}\right),$$
$$\underline{s}^{i1} := s^{\text{LMO}}\left(s^1[(i-1)-], \left\{s^{j-}[(i-j)-]\right\}_{1<j\le i}, \left\{s^{j-}[(i-j)+]\right\}_{i<j\le I}\right).$$

With these definitions, we are ready to conduct the following theoretical analysis on the feasibility to the investigated problem.

Proposition 2. *Any feasible solution to constraints (1)-(4), denoted by $\{s^i\}$ if exists, shall satisfy $\underline{s}^i(t) \le s^i(t) \le \bar{s}^i(t), \forall 1 \le i \le I, t \in [0, T]$.*

Proof. We first investigate $s^i(t) \le \bar{s}^i(t)$. It is apparent that $s^i(t) \le \min_{1\le j\le i} s^{j+}[(i-j)+](t), \forall t$, due to the left inequality of constraint (3) and $s^i(t) \le \min_{i<j\le I} s^{j+}[(i-j)-](t), \forall t$ due to the right inequality of constraint (3).

For notation convenience, denote $\bar{s}^{i+}(t) := \min\{\min_{1\le j\le i} s^{j+}[(i-j)+](t), \min_{i<j\le I} s^{j+}[(i-j)-](t)(t)\}$. Thus, it always holds that $s^i(t) \le \bar{s}^{i+}(t), \forall t$. Note that both $\bar{s}^i$ and $\bar{s}^{i+}$ have acceleration $-d_{\max}$ at the merging segments. However, due to the left inequality of acceleration constraint (1), it is impossible $s^i(t) > \bar{s}^i(t)$ at any merging segment. Therefore, $s^i(t) \le \bar{s}^i(t), \forall i, t$. We can use the same logic to prove $s^i(t) \ge \underline{s}^i(t), \forall i, t$. This proves this proposition. □

Corollary 1. *Any feasible solution to constraints (1)-(5), denoted by $\{s^i\}$ if exists, shall satisfy $\underline{s}^{i1}(t) \le s^i(t) \le \bar{s}^{i1}(t), \forall 1 < i \le I, t \in [0, T]$.*

Proposition 3. *There exists a feasible solution to constraints (1)-(4) if and only if condition (6) holds and*

$$\underline{s}^i(t) \le \bar{s}^i(t), \forall i, t. \tag{9}$$

If this condition holds, the optimal solution to Problem (7) is $\{\bar{s}^i\}_{\forall i}$ and that to Problem (8) is $\{\underline{s}^i\}_{\forall i}$.

Proof. Based on Proposition 1, if condition (6) does not hold, then there is no feasible solution at all. So we only need to investigate cases when condition (6) holds. In this case, to prove this proposition, Proposition 2 indicates that we only need to show that $\{\bar{s}^i\}_{\forall i}$ and $\{\underline{s}^i\}_{\forall i}$ are feasible solutions to constraints (1)-(4) if and only if $\underline{s}^i(t) \le \bar{s}^i(t), \forall i, t$.

The necessity is obvious, because if $\underline{s}^i(t) > \bar{s}^i(t)$ at a time, Proposition 2 indicates that no feasible solution exists. Then for the necessity, we first show that under these conditions, $\bar{s}^i$ exists and satisfies constraints (1), (2) and (4). Due to condition (6), we know all $\{s^{i+}\}$ and $\{s^{i-}\}$ exist and satisfy constraints (1) and (2), and thus so do their shadows. Denote trajectory sets $\bar{\boldsymbol{s}}^{i+} := \{s^{j+}[(i-j)+], \forall 1 \le j \le i\} \cup \{s^{j+}[(i-j)-],\ i < j \le I\}$ and $\underline{\boldsymbol{s}}^{i-} := \{s^{j-}[(i-j)-], \forall 1 \le j \le i\} \cup \{s^{j-}[(i-j)+],\ i < j \le I\}$. With slight abuse of notation, we let $\bar{\boldsymbol{s}}^{i+}(t)$ and $\underline{\boldsymbol{s}}^{i-}(t)$ denote the vector of trajectory values at time t. Further define $\bar{s}^{i+}(t) := \min \bar{\boldsymbol{s}}^{i+}(t)$ and $\underline{s}^{i-}(t) := \max \underline{\boldsymbol{s}}^{i-}(t)$.

Then, we know $\bar{s}^{i+}$ and $\underline{s}^{i-}$ satisfy constraints (2) everywhere and satisfy constraint (1) except at transitional points where they may not be differentiable. Nonetheless, after the UMO, $\bar{s}^i$ and $\underline{s}^i$ shall satisfy constraint (1) everywhere due to the smoothing effect of the merging segments. Meanwhile, constraint (2) shall remain hold, because each merge segment transits from one speed to another speed on $\bar{s}^{i+}$ or $\underline{s}^{i-}$, both of which are feasible to constraint (2).

Based on the merging operations generating $\bar{s}^i$ and $\underline{s}^i$, we know s^{i+} is an upper bound to $\bar{s}^i$, i.e., $\bar{s}^i(0) \le s^i_{\text{begin}}, \bar{s}^i(T) \le s_{\text{final}}{}^{i+}$, and s^{i-} is a lower bound to $\underline{s}^i$, i.e., $\underline{s}^i(0) \ge s^i_{\text{begin}}, \underline{s}^i(T) \ge s^{i-}_{\text{final}}$. This together with Condition (9) yields that $\bar{s}^i(0) = \underline{s}^i(0) = s^i_{\text{begin}}$ and $s_{\text{final}}{}^{i-} \le \bar{s}^i(T) \le \underline{s}^i(T) \le s^{i+}_{\text{final}}$.

Further, since $\bar{s}^i(0)$ starts from $s_{\text{begin}}{}^i$, then $\dot{\bar{s}}^i(0) \le v_{\text{begin}}{}^i$ since s^{i+} bounds $\bar{s}^i$ from above. Also, since $\underline{s}^i(0)$ starts from the same location s^i_{begin}, then $\dot{\underline{s}}^i(0) \ge v^i_{\text{begin}}$ since s^{i-} bounds $\underline{s}^i$ from below. This yields $\dot{\bar{s}}^i(0) \le v_{\text{begin}}{}^i \le \dot{\underline{s}}^i(0)$. On the other hand, due to condition (14) and $\bar{s}^i(0) = \underline{s}^i(0)$, we should have $\dot{\underline{s}}^i(0) \le v^i_{\text{begin}} \le \dot{\bar{s}}^i(0)$. This indicates $\dot{\underline{s}}^i(0) = \dot{\underline{s}}^i(0) = v^i_{\text{begin}}$. Thus $\bar{s}^i$ and $\underline{s}^i$ satisfy constraint (4).

Next, we will prove that these conditions ensure constraint (3) holds as well. We first investigate this for $\{\bar{s}^i\}_{\forall i}$. Note that $\bar{\boldsymbol{s}}^{(i-1)+} - l - g_{\min} = \{s^{j+}[(i-j)+], \forall 1 \le j \le i-1\} \cup \{s^{j+}[(i-j)-] + g_{\max} - g_{\min},\ i \le j \le I\}$ and $\bar{\boldsymbol{s}}^{i+} = \{s^{j+}[(i-j)+], \forall 1 \le j \le i-1\} \cup \{s^{j+}[(i-j)-],\ i \le j \le I\}$. Note that if shift trajectories in $\bar{\boldsymbol{s}}^{i+}$ indexed by $i \le j \le I$ upwards by positive offset $g_{\max} - g_{\min}$, we obtain $\bar{\boldsymbol{s}}^{(i-1)+} - l - g_{\min}$. Note that $\bar{s}^{i-1} - l - g_{\min}$ and $\bar{s}^i$ are actually the results by applying the UMO to trajectory sets $\bar{\boldsymbol{s}}^{(i-1)+} - l - g_{\min}$ and $\bar{\boldsymbol{s}}^{i+}$, respectively.

Since the offset of the composing trajectories are positive, is it obvious that $\bar{s}^{i-1}(t) - l - g_{\min} \ge \bar{s}^i(t), \forall t$, which proves the left inequality of constraint (3). Next, since $\bar{\boldsymbol{s}}^{(i-1)+} - l - g_{\max} = \{s^{j+}[(i-j)+] - (g_{\max} - g_{\min}), \forall 1 \le j \le i-1\} \cup \{s^{j+}[(i-j)-],\ i \le j \le I\}$, these composing trajectories are either the same as those in $\bar{\boldsymbol{s}}^{i+}$ or resulted from shifting down them. Similarly, the *UMO* shall preserve the bounding relationship, i.e., $\bar{s}^{i-1}(t) - l - g_{max} \le \bar{s}^i(t), \forall t$. This proves the right inequality of constraint (3).

The proof of constraints (3) for $\{\underline{s}^i\}_{\forall i}$ is exactly symmetric to that for $\{\bar{s}^i\}_{\forall i}$ and thus we omit it. This concludes the proof. □

Corollary 2. *There exists a feasible solution to constraints (1)-(5) if and only if condition (6)* holds and

$$\underline{s}^{i1}(t) \le \bar{s}^{i1}(t), \forall i, t.$$

Proposition 3. *and Corollary 2 are elegant analytical results that not only show the necessary and sufficient condition for the investigated problem to be feasible but also construct analytical solutions to the two extreme optimal solutions to Problems (7) and (8) (and their variants with fixing the lead vehicle's trajectory), respectively. It provides valuable insights into the structure of the general class of problems sharing the same constraints.*

Remark 2. Although the above analysis does not add a constraint on the range of jerk, the results for trajectory planning are still valid. Note that the investigated problem is trajectory planning that designs template trajectories for vehicles to follow in the next time horizon (e.g., on the order of minutes). However, the planned trajectories are different from real-time actual vehicle trajectories. The vehicles are actually controlled by a tracking controller to follow the designed trajectories within the error range allowed by $g_{\min}$. During the tracking control process, the jerk jumps in the template trajectories will be smoothed in the actual trajectories with slight location error admissible to $g_{\min}$ (since the planned trajectory locations have smooth shapes due to continuous speed everywhere). Thus, the jerk jumps in the template planned trajectories do not much affect the applicability of the above analysis results.

On the other hand, incorporating jerk range constraints in the above analysis is not difficult (though maybe quite tedious). Basically, all trajectory shapes (e.g., those in Fig. 1. 1–4) just need to be revised by adding a transitional cubic segment with the maximum/minimum jerk at each junction between two consecutive quadratic segments to ensure the jerks around the junction is still bounded. All the other analytical procedures and theoretical results will stay the same. This however will much complicate the math over the already complex formulations. Further, since the jerk jumps will be naturally smoothed out in tracking control as explained above, we just omit the detailed analysis on bounding jerks in this section.

While an analytical approach exists for these special cases, for more general objective functions, there may not exist an analytical solution approach. Instead, an efficient numerical solution approach is desirable for problems with more complex objectives. The following section will propose a numerical algorithm that can solve this problem with complex objective functions to the optimal solution in a very efficient manner.

3. Discrete modeling with general objectives

This section formulates this trajectory planning problem with a more general objective function. To better suit efficient numerical solution algorithms, we present the model formulation in the discrete-time format. *Section 3.1* presents the discretization of the models proposed in *Section 2*. *Section 3.2* proposes the model with a general objective function.

3.1. Decision variables and the constraints

In the discrete-time model, time horizon [0, T] is discretized into $n+1$ points, i.e., 0, δ, 2δ, ..., $n\delta$ where $\delta = T/n$. Note that when discretization interval δ is small enough, we can approximate any realistic vehicle trajectories (which should be Lipschitz continuous) within an arbitrarily small bound.

For the math convenience, we denote the entire set of time indexes as $\mathcal{J} \in \{0, 1, \cdots, n\}$. The trajectory of vehicle i is a series of $(n+1)$ space-time points $\{(s_j^i, j\delta)\}_{j\in\mathcal{J}}$, where s_j^i denotes the position of the front bumper of vehicle i at time $j\delta$, $\forall i \in \mathcal{J}, j \in \mathcal{J}$.

Corresponding to constraints (1)-(4), we have the following constraints

$$-d_{\max} \le \frac{\frac{s_{j+1}^i - s_j^i}{\delta} - \frac{s_j^i - s_{j-1}^i}{\delta}}{\delta} \le a_{\max}, \ \forall i, \ j = 1, \ \ldots, \ n-1 \tag{10}$$

$$0 \le \frac{s_j^i - s_{j-1}^i}{\delta} \le v_{\max}, \ \forall i, \ j = 2, \ \ldots, \ n., \tag{11}$$

$$g_{\min} \le s_j^{i-1} - s_j^i - l \le g_{\max}, \ \forall i, \ j \tag{12}$$

$$s_0^i = s_{\text{begin}}^i, \ \frac{s_1^i - s_0^i}{\delta} = v_{\text{begin}}^i, \ s_{\text{final}}^{i-} \le s_n^i \le s_{\text{final}}^{i+}, \ \forall i, \ j \tag{13}$$

Note that constraints (10)-(13) refer to discrete time points with interval δ. Yet as δ approaches to zero, these constraints are equivalent to constraints (1)-(4). Again, if needed, lead vehicle constraints (5) can be easily incorporated just by setting $\{s_j^1\}$ as parameters instead of variables. Since this adaptation is trivial from the mathematical programming point of view and we opt not to elaborate on constraints (5) in this remainder of this section.

Moreover, we can also numerically obtain the two important boundaries of the feasibility set by solving the discrete-time formulations of Problems (7) and (8) as

$$\max_{s_j^i} \sum_{i=1}^{I} \sum_{j=1}^{n} s_j^i \tag{14}$$

$$\min_{s_j^i} \sum_{i=1}^{I} \sum_{j=1}^{n} s_j^i \tag{15}$$

subject to Ineq. (10)-(13).

Problems (14) and (15) are linear programming problems that can be quickly solved (e.g., with the Simplex algorithm). This is important to connected vehicle applications which require real-time decisions and actions.

3.2. Model formulation with a general objective

In the above analysis, we mainly discuss the feasibility of the constraints and only investigated two special objectives. This section investigates how to solve this problem with more general objectives. Usually, for trajectory planning problems, the objective function J is defined as the summation of the polynomial functions of the position, velocity, acceleration and jerk of a vehicle along time (Li and Wang, 2003; Li and Wang, 2006; Li et al., 2014c; Ma et al., 2017; Zhou et al., 2017). If the objective function is convex, we can directly solve it and find the global optimal solution. If the objective function is non-convex, we can resort to the Frank–Wolf algorithm (Frank and Wolfe, 1956; Bertsekas, 1999; Nocedal and Wring, 2006; Jaggi, 2013) and find an acceptable local optimal solution after repeating the algorithm for enough times with randomly-selected initial solutions.

One major problem of these approaches is that the obtained trajectory for a vehicle might be very complicated, if the discretization interval is very small and the time range of interest is long. This often prevents us from implementing such trajectories via V2X communications in practice.

Instead, we often turn to find a trajectory that gives suboptimal performance but is much more parsimonious in shape. A natural idea is to find a trajectory with only a few regimes (Li and Wang, 2006; Li et al., 2014c; Ma et al., 2017).

Fortunately, we can directly achieve this goal by just adding a l_1 penalty into the objective function (Tibshirani, 1996; Efron et al., 2004; Hastie et al., 2009) as

$$\min_{s_j^i} J + \lambda \sum_{i=1}^{I} \sum_{j=1}^{n-1} \left| \left(s_j^i - s_{j-1}^i\right) - \left(s_{j+1}^i - s_j^i\right) \right| \tag{16}$$

where J is the original objective function and λ is a weighting coefficient controlling the parsimony of the trajectory.

Indeed, the second term aims to reduce the number of non-zero acceleration/deceleration values along the time and thus controls the complexity of the trajectory. As shown in (Elad 2010; Bach et al., 2012), adding the l_1 norm (usually called the sparsity-inducing norm) penalty of the solution into the objective function helps obtaining a sparse solution (the elements of the solution vector are mostly zero) rather than a dense solution (the elements of the solution vector are mostly non-zero) for an optimization problem (Li, 2015). In theory, other penalty formulations (e.g., the l_2 norm) may also serve the purpose of smoothing trajectories. We just use the l_1 norm here since it fits the constructed algorithm the best. When λ is very large, the planning problem aims to find a trajectory of which the ac/deceleration rate is zero for most of time. When λ is very small, the planning problem aims to find a trajectory that gives the best performance, which however may yield an irregular shape. By varying λ we can find a good tradeoff between the optimality and the complexity of the obtained trajectory.

Introducing auxiliary variables $z_j^i \in \mathrm{R}^+$, Problem (16) can be rewritten as

$$\min_{s_j^i, z_j^i} J_{\text{sparse}} = J + \lambda \sum_{i=1}^{I} \sum_{j=1}^{n-1} z_j^i \tag{17}$$

$$\text{s.t. } \left| \left(s_j^i - s_{j-1}^i\right) - \left(s_{j+1}^i - s_j^i\right) \right| \le z_j^i. \tag{18}$$

Noticing that Problem (17) subject to Ineq.(10)-(13), (18) has a convex feasible region constrained by a series of linear constraints, we can solve it easily with off-the-shelf linear programming solvers if J is also a linear function. Further, if J is a convex function, the whole problem remains easy to solve with convex programming solvers (Boyd and Vandenberghe, 2004). If J is not convex, we may resort to some advanced algorithms. In the following sections, we will further discuss solution algorithms for several representative applications.

In addition, we can also define the smoothness and parsimoniousness as follows.

Definition 9. The smoothness of a discrete-time trajectory is characterized by the maximum acceleration rate associated with this trajectory. That is, for vehicle i, we have

$$\text{Smoothness}^i = \max_j \left| \left(s_j^i - s_{j-1}^i\right) - \left(s_{j+1}^i - s_j^i\right) \right|. \tag{19}$$

Definition 10. The parsimony of a discrete-time trajectory is characterized by the number of non-zero acceleration rate values associated with this trajectory. That is, for vehicle i, we have

$$\text{Parsimoniousness}^i = \sum_j \text{sgn}\left| \left(s_j^i - s_{j-1}^i\right) - \left(s_{j+1}^i - s_j^i\right) \right|. \tag{20}$$

4. Model applications

This section applies the proposed model to three representative problems with different focuses. The first problem on vehicle platoon control illustrates the model application to multiple vehicles without fixing the lead vehicle's trajectory. The second problem on eco-driving illustrates how the l_1 penalty term helps smooth and simplify the trajectory in car following. The third problem on jam-absorption driving illustrates how to smooth and simplify a trajectory jam across multiple vehicles when the lead vehicle trajectory is fixed and certain extra constraints are imposed. Finally, we discuss how to modify the proposed model to adapt other applications.

In all cases, we set $d_{\max} = a_{\max} = 2\text{m/s}$, $v_{\max} = 12\text{m/s}$, $g_{\min} = 15\text{m}$, $g_{\max} = 40\text{m}$.

4.1. Vehicle platoon control

This section illustrates the application of the proposed model to a vehicle platoon control case study, in which the trajectory of the lead vehicle is yet to determined (i.e., constraint (5) is not activated).

Suppose there is a four-vehicle platoon approaching a signalized intersection. We assume that the traffic light is red at the beginning of trajectory planning and will turn green after 15 s. So, we set the time range of this problem as $T = 15\text{s}$ and hope that the lead vehicle will right catch the green light at the stop-line when the green light just turns on. Thus we set $n = 15$ with time resolution $\delta = 1\text{s}$. We set the initial positions of vehicles as $s^1_{\text{begin}} = 69$ m, $s^2_{\text{begin}} = 40$ m, $s^3_{\text{begin}} = 18$ m, $s^4_{\text{begin}} = 0$ m. We set the stop-line at 180 m and the final gap between two consecutive vehicles are 20 m. The final position of the vehicles are set as $s^{1-}_{\text{final}} = s^{1+}_{\text{final}} = 180$ m, $s^{2-}_{\text{final}} = s^{2+}_{\text{final}} = 160$ m, $s^{3-}_{\text{final}} = s^{3+}_{\text{final}} = 140$ m, $s^{4-}_{\text{final}} = s^{4+}_{\text{final}} = 120$ m. The initial speeds of vehicles are set as $v^1_{\text{begin}} = 10\text{m/s}$, $v^2_{\text{begin}} = 12$ m/s, $v^3_{\text{begin}} = 12\text{m/s}$ and $v^4_{\text{begin}} = 11$ m/s.

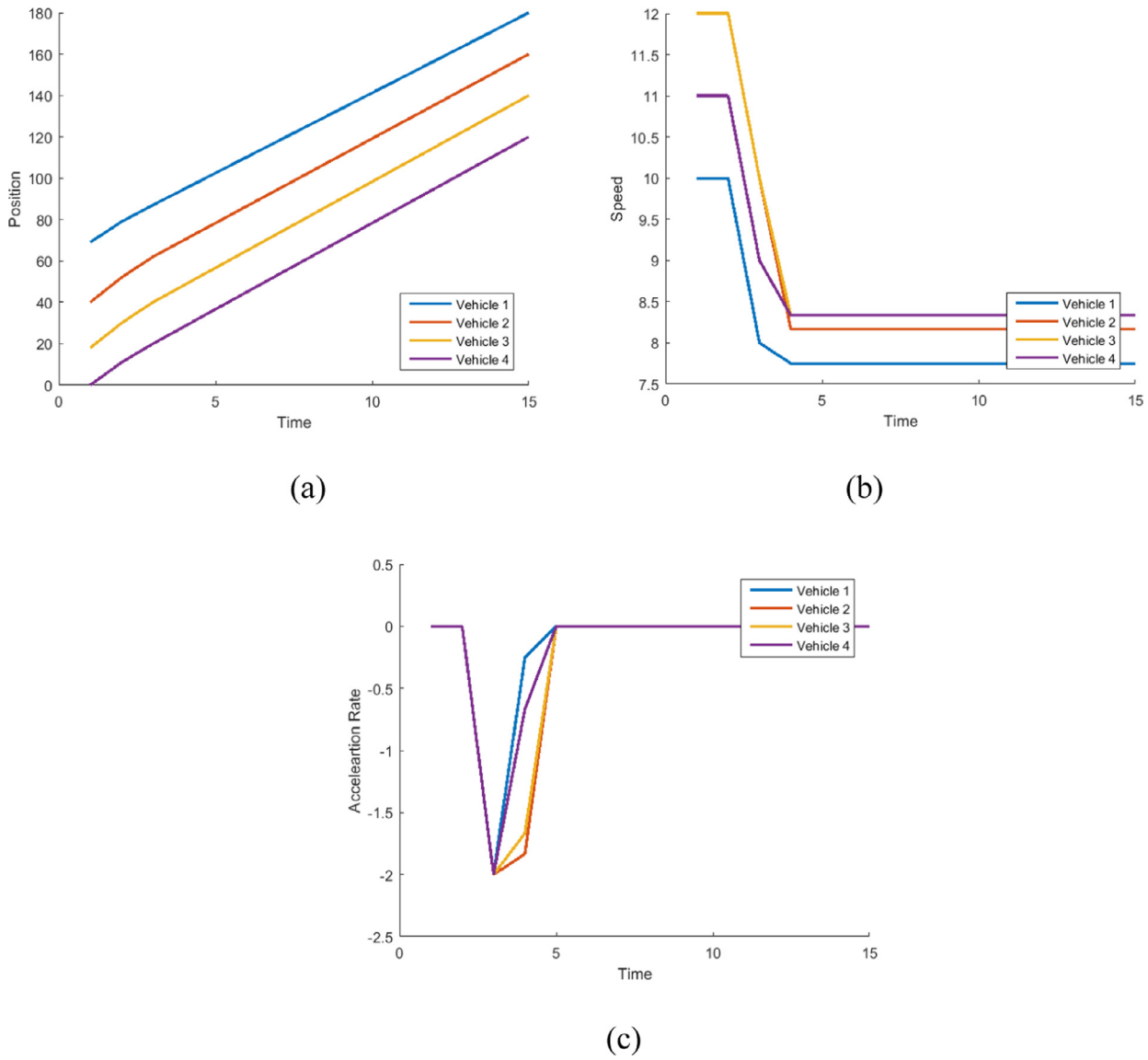


Fig. 5. (a) The time-space trajectories of the studied 4 vehicles; (b) the corresponding speed values of the studied 4 vehicles along the time; (c) the corresponding ac/deceleration values of the studied 4 vehicles along the time.

We set the objective as

$$\min_{s_j^i} \sum_{i=1}^{4} \sum_{j=1}^{n-1} \left| \left(s_j^i - s_{j-1}^i\right) - \left(s_{j+1}^i - s_j^i\right) \right| \tag{21}$$

which means that we seek the trajectories that have minimum acceleration rates from those compliant with the planned boundary conditions.

Fig. 5 shows the results. We can see that the proposed algorithm works when the trajectory of the lead vehicle is not given. The lead vehicle right catches the green light as required. Because the initial gaps and speeds of the following vehicles are different, these vehicles decelerate with varying deceleration rates accordingly to keep appropriate final gaps and avoid collisions.

4.2. Eco-Driving

This section illustrates the application of the proposed model to an eco-driving case study. In this application, the trajectory of the lead vehicle is given, and we only need to optimize the second vehicle's trajectory (thus $I = 2$). Eco-driving

is one hot topic in recent CAV research aiming reducing individual vehicle fuel consumption and improving driving comfort. Fuel consumption is often calculated as a function of the speeds and acceleration/decelerations approximated by the first- or second-order differences of positions (Demir et al., 2014). For example, the following fuel consumption function was proposed in (Ahn et al., 2002) and has been widely used

$$J = \sum_{j=2}^{n} F_j, \quad F_j = \begin{cases} \sum_{p=0}^{3}\sum_{q=0}^{3} W_{p,q} v_j^p a_j^q & \text{for } a_j \geq 0 \\ \sum_{p=0}^{3}\sum_{q=0}^{3} M_{p,q} v_j^p a_j^q & \text{for } a_j < 0 \end{cases} \tag{22}$$

$$v_j^i = \frac{s_j^i - s_{j-1}^i}{T/n}, \quad a_j^i = \frac{\frac{s_{j+1}^i - s_j^i}{T/n} - \frac{s_j^i - s_{j-1}^i}{T/n}}{T/n}, \tag{23}$$

where $W_{p,q}$, $M_{p,q}$ are preselected weighting coefficients.

The main challenge to solving this problem is that objective function J is non-convex. In this paper, we adopt the Frank-Wolf algorithm (Frank and Wolfe, 1956; Jaggi, 2013) to find a local minimal solution instead. The key idea of Frank-Wolf algorithm is to apply a special localized linear approximation for nonlinear optimization problems whose objective functions are not convex. Such a technique was widely used in transportation studies (Wong et al., 2002; Yin, 2008). The sketch of the specified algorithm is written as follows in Algorithm 3:

Algorithm 3. The Frank–Wolf algorithm.

Initialization: Solve the planning problem (22)-(23), (10)-(13), (18), and find a feasible solution (e.g. the most parsimonious solution that tracks the leading vehicle) $\{x_j^i|^{(0)}\}$, $j = 0, \ldots, n$, $i = 1, \ldots, I$,. Set iteration index $k = 0$.

Step 1. Solve the following direction-finding subproblem and get a solution $\{s_j^i|^{(k)}\}$

$$\min_{s_j^i} s^T \nabla J_{\text{sparse}}(\{x_j^i|^{(k)}\}) \quad s.t. s \in \Omega \tag{24}$$

where Ω is the feasibility set defined by Ineq. (10)-(13), (18). This subproblem is a linear programming problem and can be quickly solved with off-the-shelf linear programming solvers.

Step 2. Update the step size

$$\gamma \leftarrow \tfrac{2}{k+2} \tag{25}$$

Step 3. Update the solution and the iteration index

$$x_j^i|^{(k+1)} \leftarrow x_j^i|^{(k)} + \gamma\,(s_j^i|^{(k)} - x_j^i|^{(k)}), \ \forall j, \ k \leftarrow k+1 \tag{26}$$

and go to Step 1, if $\exists j$, $|\{x_j^i|^{(k+1)}\} - \{x_j^i|^{(k)}\}| > c$, where $c > 0$ is a pre-selected threshold. Otherwise, return the current trajectory $\{x_j^i|^{(k+1)}\}$.

In this paper, we resort to the Frank–Wolf algorithm for two reasons. First, the Frank–Wolf algorithm will converge to a local optimal solution soon and it had been shown to be useful in traffic engineering applications (Fukushima, 1984; Weintraub et al., 1985). Second, the Frank–Wolf algorithm was recently found to be an efficient algorithm for sparse optimization problems (Clarkson, 2010; Jaggi, 2013; Lacoste-Julien, 2016).

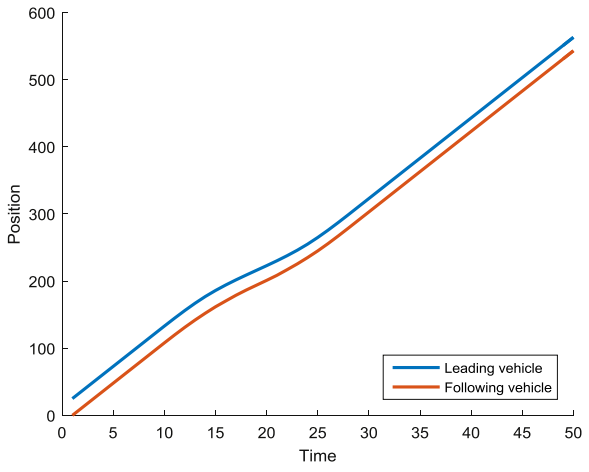
In the following numerical test, we set the initial speeds of the two studied vehicles as $v_{\text{begin}}^1 = v_{\text{begin}}^2 = 12$ m/s. Set $s_{\text{begin}}^1 = 25$ m, $s_{\text{final}}^{1+} = s_{\text{final}}^{1-} = 563$m. We also set $s_{\text{begin}}^2 = 0$m, and thus the initial gap is 25m. The final desired gap between these two consecutive vehicles is within [15, 20] m. In order words, we set $s_{\text{final}}^{2+} = s_{\text{final}}^{1+} - 15$, $s_{\text{final}}^{2-} = s_{\text{final}}^{1-} - 20$. The time range of this problem is $T = 50$ s. Thus we set $n = 50$ with resolution level $\delta = 1$ s.

We set the lead vehicle's ac/deceleration rate vary from 0 to -1m/s^2 during [10, 15)s, from -1 m/s^2 to 1 m/s^2 during [15, 25] s, and from 1 m/s^2 to 0 /s^2 during [15, 16) s. The initial feasible solution is set as the solution that we get after solving the planning problem (22)-(23), (10)-(13), (18).

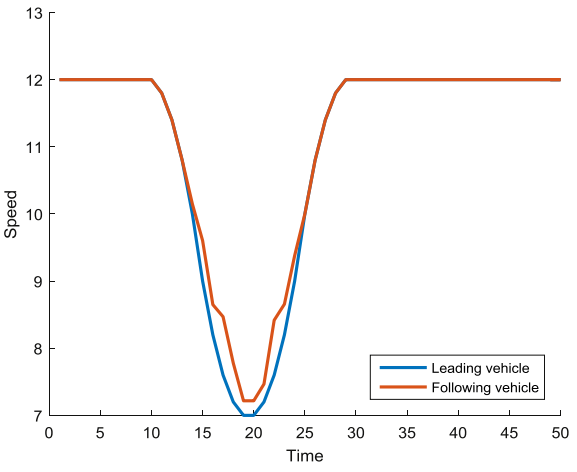
According to (Ahn et al., 2002), we set the objective function as

$$F_j = \begin{cases} 0.01 \cdot v_j + 0.023 \cdot a_j + 0.01 \cdot a_j^2 & \text{for } a_j \geq 0 \\ 0.006 \cdot v_j + 0.01 \cdot a_j^2 & \text{for } a_j < 0 \end{cases}. \tag{27}$$

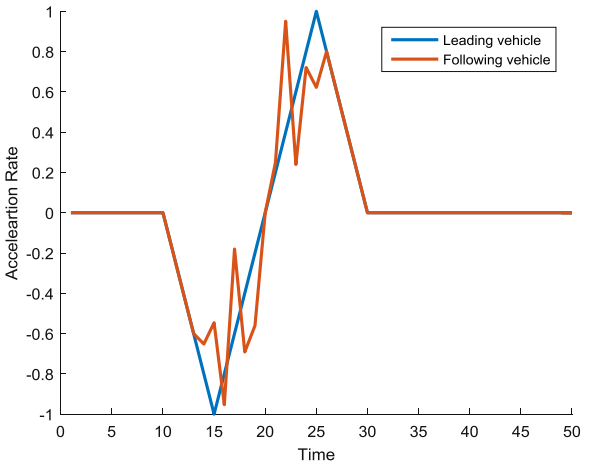
Fig. 6 shows the results with $\lambda = 0$ that converge after 6 iterations of the Modified Frank-Wolf algorithm. This solution contains 18 nonzero acceleration values and the corresponding objective value is 4.82. Fig. 7 shows the results with $\lambda = 1$ that converge after only 4 iterations. This solution contains 16 nonzero acceleration values and the corresponding objective value is 4.55. When we set $\lambda = 10$ or even larger, we again get the most parsimonious solution that we can get after solving the planning problem (22)-(23), (10)-(13), (18) after only 2 iterations. As shown in Fig. 8, this solution contains 14 nonzero acceleration values and the corresponding objective value is 4.67. Therefore, by varying the value of sparsity penalty coefficient λ, we can control the parsimoniousness of the trajectory. The larger the sparsity penalty coefficient λ, the simpler the solution is, despite that the optimality of the objective value might be slightly compromised. Considering that fuel consumption functions usually have at least a few percent errors, from an engineering practice point of view, losing a little optimality in this numerical solution does not really matter.



(a)



(b)



(c)

Fig. 6. (a) The time-space trajectories of the studied two vehicles; (b) the corresponding speed values; (c) the corresponding ac/deceleration values; where $\lambda = 0$.

4.3. Jam-Absorption driving

Traffic jams frequently occur on highways and cause significant economic losses as well as environment pollutions. As a result, the origin and evolution mechanism of traffic jams received consistent research interests (Daganzo, 1999; Helbing et al., 2009; Kerner, 2009; Li et al., 2010). Various strategies had been proposed to avoid or mitigate jams. Especially, several researchers became to show interests on absorbing a traffic jam via adaptive driving without making new jams (Tao et al., 2005; Nishi et al., 2013; Taniguchi et al., 2015; He et al., 2017). As a result, researchers have proposed various driving strategies and studied the conditions that compression and expansion waves caused by these driving strategies will meet and cancel out each other.

The basic idea of the jam absorb driving strategy is to guide the absorbing vehicles to vary their speeds appropriately so as to reduce the time of being captured by a jam. Based on the above parsimonious trajectory planning framework, we can formulate the corresponding planning problem as

$$\min_{s_j^i} \sum_i \sum_{j=1}^{n-1} \left| \left(s_j^i - s_{j-1}^i\right) - \left(s_{j+1}^i - s_j^i\right) \right| \tag{28}$$

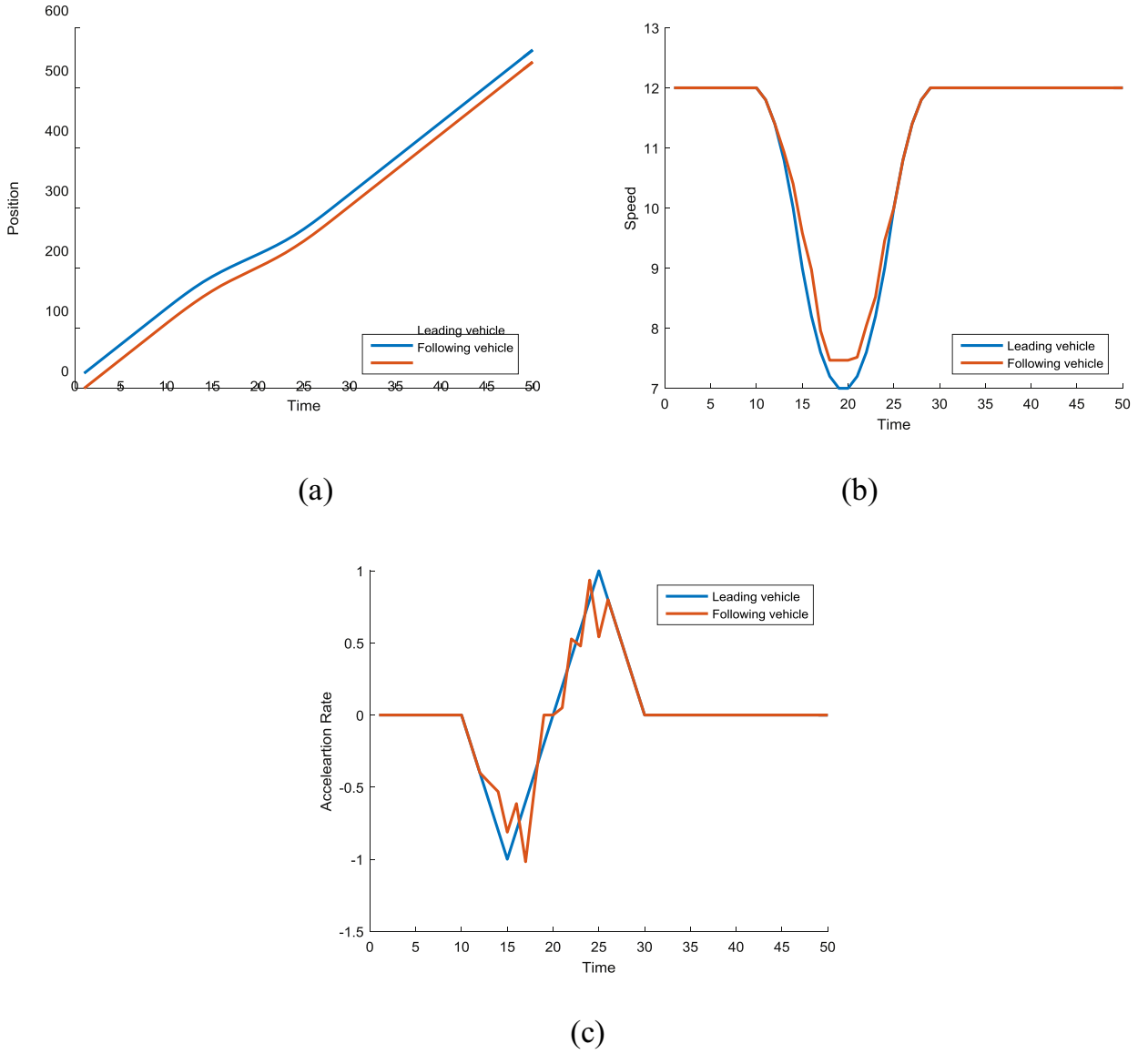


Fig. 7. (a) The time-space trajectories of the studied two vehicles; (b) the corresponding speed values; (c) the corresponding ac/deceleration values; where $\lambda = 1$.

subject to the constraints Ineq. (9)-(12) and also the following string stability constraints

$$s_j^{i-1} - s_j^i \leq s_{j-1}^{i-1} - s_{j-1}^i, \ \forall i = 2, \ \ldots, \ I, \ j = 1, \ \ldots, \ n \tag{29}$$

Objective (28) requires taking as simple as possible actions to absorb the jam, since it actually requires to minimize the l_1-norm of the nonzero acceleration values along the time.

Constraint (29) indicates that the following vehicle will vary its speed accordingly in advance to track the lead vehicle. As shown in (Nishi et al., 2013; Taniguchi et al., 2015; He et al., 2017), the initial gap between two consecutive vehicles is usually larger than their final gap to make this condition hold.

The planning problem (28), (10)-(13), (18), (29) is convex and easy to solve. When its objective is a l_1-norm function, we could apply the Alternating Direction Method of Multipliers (ADMM) algorithm to solve it quickly. Interested readers may check (Boyd and Vandenberghe, 2004; Parikh and Boyd, 2013) and also *Appendix A* of our previous paper (Li et al., 2015) for detailed explanations.

We can solve this problem by sequentially increasing the number of the encountered vehicles. Based on our previous analysis in *Section 2.2*, there should exist a feasible solution, if only constraints (10)-(13) are considered with $s_n^i = s_{\text{final}}^i$ correctly given. So, if the planning problem (28), (10)-(13), (18) and (29) is infeasible with respect to constraints related to a

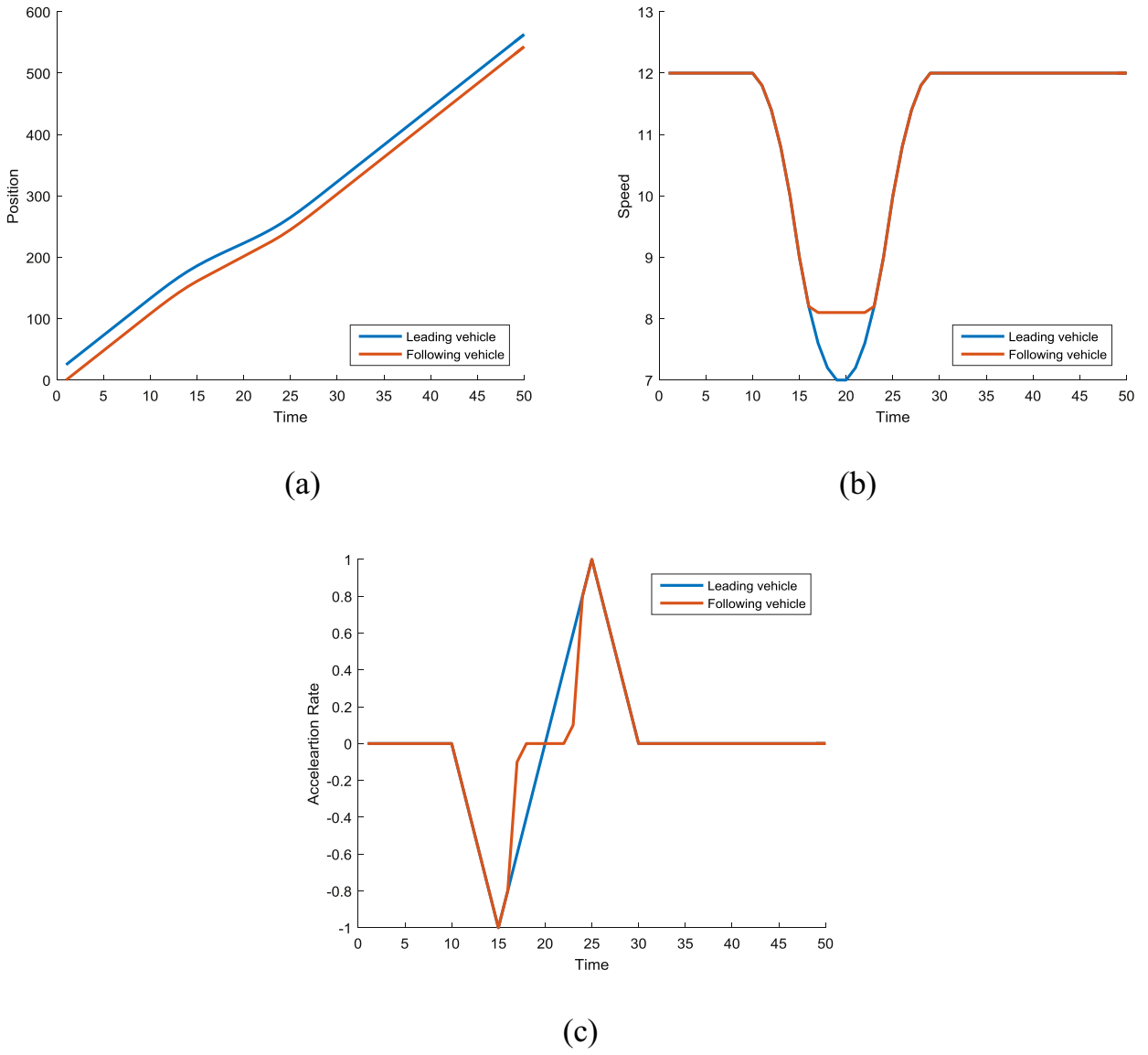


Fig. 8. (a) The time-space trajectories of the studied two vehicles; (b) the corresponding speed values; (c) the corresponding ac/deceleration values; where $\lambda = 10$.

vehicle, say i^*, the only possible situation is Ineq.(29) associated with i^*does not hold. This actually indicates that vehicle i^* cannot reach the desired final gap, mainly because of the maximum speed limit. In other words, the jam has already been fully absorbed by the first $(i^* - 1)$ vehicles.

In the following numerical test, we consider a typical jam absorption scenario. The trajectory of the lead vehicle is given and the trajectories of the following vehicles need to be determined.

We set the initial speeds of all vehicles as 12 m/s, $v^1_{\text{begin}} = v^2_{\text{begin}} = \ldots = 12\text{m/s}$. We set the initial gap between every two consecutive vehicles as 25m. More precisely, we set $s^1_{\text{begin}} = 150$ m, $s^2_{\text{begin}} = 125$ m, $s^3_{\text{begin}} = 100$ m, $s^4_{\text{begin}} = 75$ m, $s^5_{\text{begin}} = 50$ m, $s^6_{\text{begin}} = 25$ and $s^7_{\text{begin}} = 0$ m. The final desired gap between these two consecutive vehicles is 20m. More precisely, we set $s^{1-}_{\text{final}} = s^{1+}_{\text{final}} = 708$ m, $s^{2-}_{\text{final}} = s^{2+}_{\text{final}} = 688$ m, $s^{3-}_{\text{final}} = s^{3+}_{\text{final}} = 668$ m, $s^{4-}_{\text{final}} = s^{4+}_{\text{final}} = 648$ m, $s^{5-}_{\text{final}} = s^{5+}_{\text{final}} = 628$ m, $s^{6-}_{\text{final}} = s^{6+}_{\text{final}} = 608$ and $s^{7-}_{\text{final}} = s^{7+}_{\text{final}} = 588$ m. The time range of this problem is $T = 50$ s. Thus we set $n = 50$ with resolution level $\delta = 1$ s.

To simulate the initial jam, we set the lead vehicle to decelerate with constant rate -1.2m/s^2 for 5 s during [10, 15) s and accelerate with constant rate 1.2s/s^2 for 5 s during [15, 16) s.

Fig. 9 shows the results, where the trajectories of the following vehicles are calculated by solving the planning problem (28), (10)-(13), WITHOUT constraint (29). We can see that the jam can be absorbed by the first 6 vehicles, but NOT in an

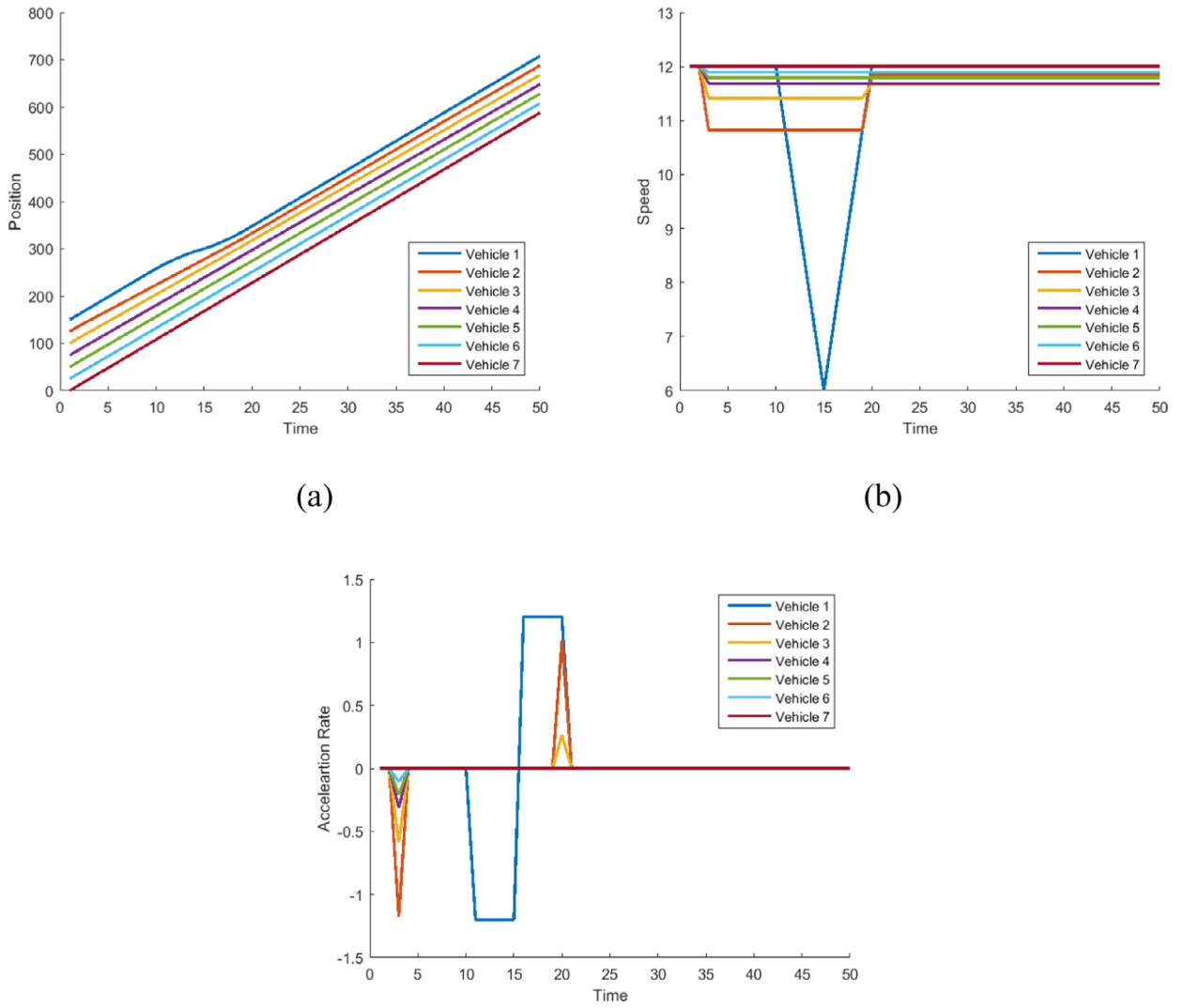


Fig. 9. (a) The time-space trajectories of the studied 7 vehicles; (b) the corresponding speed values of the studied 7 vehicles along the time; (c) the corresponding ac/deceleration values of the studied 7 vehicles along the time. Here, constraint (29) is NOT considered.

asymptotical manner. The gap between Vehicle 1 and Vehicle 2 decreases at the beginning but unexpectedly increases a little after 20 sec. This is mainly because we cannot accurately know in advance the required trajectory planning time range to implement jam absorption. So, if constraint (29) is not considered, the variation of the gap between two consecutive vehicles may not always be monotonous.

Fig. 10 shows the results, where the trajectories of the following vehicles are sequentially calculated by solving the planning problem (28), (10)-(13), and (29). We can see that the jam can be absorbed by the first 6 vehicles in an asymptotical manner.

4.4. Discussions

The above model could be further modified to adapt other applications.

First, we can add constraints on the jerk (the time derivative of the ac/deceleration rate) to each trajectory. In some applications, we need to limit the jerk to make riding more comfortable or consider mechanical constraints of ac/deceleration rates. To this end, we can explicitly add constraints on the jerk as

$$-jerk_{\min} \leq \frac{s^i_{j+2} - 3s^i_{j+1} + 3s^i_j - s^i_{j-1}}{\delta^3} \leq jerk_{\max}, \forall i, j = 1, \cdots, n-2. \tag{30}$$

Note that the above constraints remain linear and thus adding them to the problem does not much increase difficulty in solving the problem numerically.

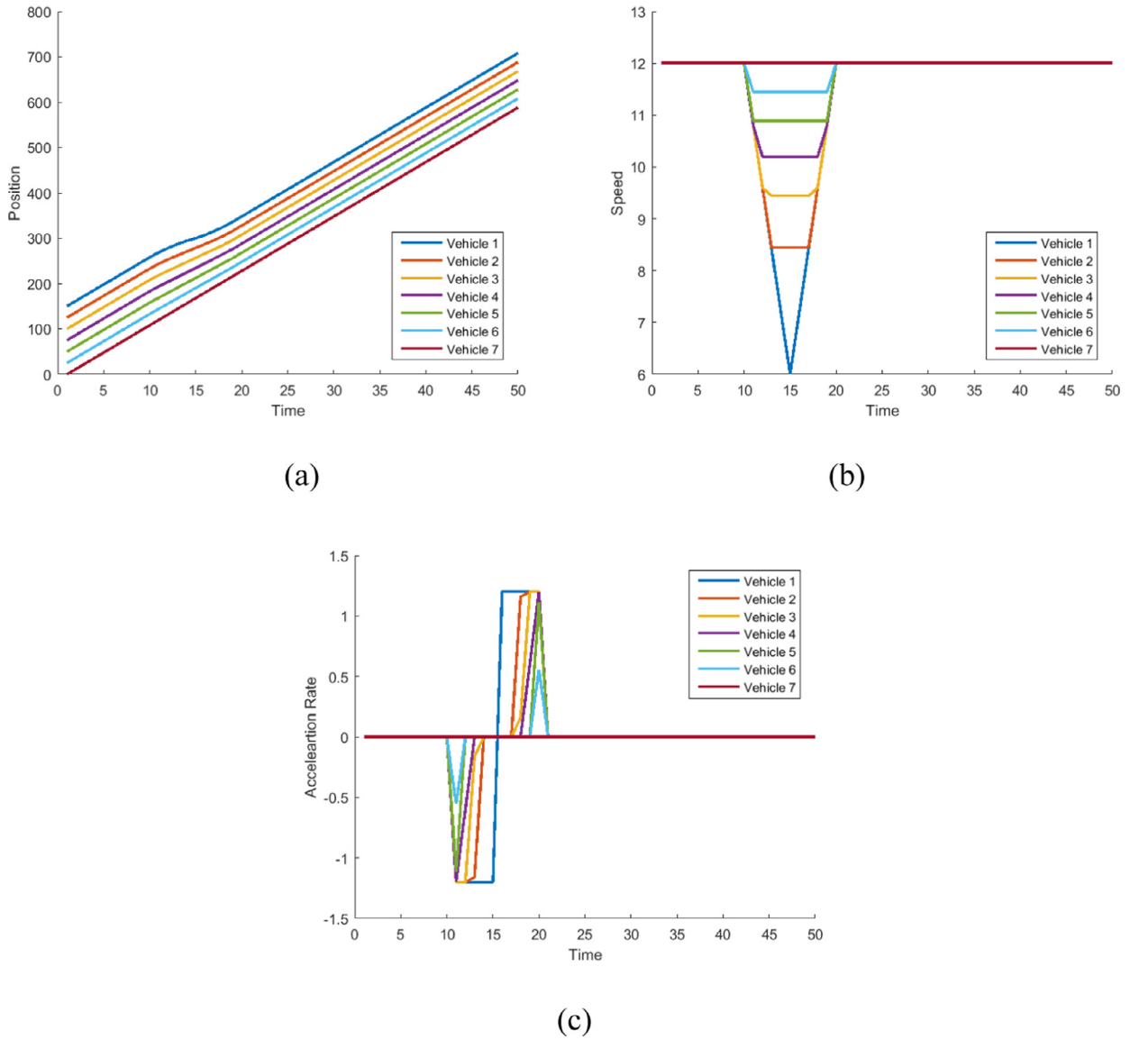


Fig. 10. (a) The time-space trajectories of the studied 7 vehicles; (b) the corresponding speed values of the studied 7 vehicles along the time; (c) the corresponding ac/deceleration values of the studied 7 vehicles along the time. Here, constraint (29) is considered.

Second, we do not consider too detailed vehicle kinematic characteristics (e.g., the dependence of the acceleration limit with the speed, or the jerk limits) in this paper; otherwise, we will reach a complex nonlinear equality constrained optimization problem which is really hard to solve. A similar example could be found in our recent car-following model calibration paper (Li et al., 2016a), in which we optimize a problem with a series of nonlinear equality constraints. Currently, there is no convenient method to solve such a problem online. For trajectory planning applications, we have to make a tradeoff between model accuracy and application feasibility.

Further, since the problem is a trajectory smoothing problem, the optimal solution usually yields a trajectory with already small acceleration much lower than the smallest acceleration limit at the highest possible speed. So, we can just set $a_{\max}$ smaller than the minimum acceleration limit at the highest possible speed, which shall not much impact the solution optimality. Actually, this assumption is just for trajectory planning step. In the trajectory following control step, the AV control will naturally smooth out acceleration jumps, if any, since the acceleration levels are not too high.

Third, we can easily adapt the proposed model to heterogeneous ac/deceleration rates and speed limits by assigning the following constraints for each vehicle individually.

Continuous-time case

$$-d^i_{\max} \le \ddot{s}^i(t) \le a^i_{\max} \ \forall i, t \tag{31}$$

$$0 \le s^i(t) \le v^i_{\max} \ \forall i, t \tag{32}$$

Discrete-time case

$$g^i_{\min} \le s^{i-1}_j - s^i_j - l \le g^i_{\max}, \ \forall i, \ j \tag{33}$$

$$-d^i_{\max} \le \frac{\frac{s^i_{j+1}-s^i_j}{\delta} - \frac{s^i_j - s^i_{j-1}}{\delta}}{\delta} \le a^i_{\max}, \ \forall i, \ j = 1, \ \ldots, \ n-1 \tag{34}$$

The above adaptation does not change the model structure and thus all the theoretical analysis and numerical algorithms shall directly apply. The reason that the original model formulation does not include this adaptation is that this study intends to focus on the aspects that compose the major contributions, including theoretical analyses on the feasibility and special bounding solutions, efficient numerical modeling and solution approaches, and their applications to various real-world problems.

Fourth, we neglect various uncertainties such as actuation lag, communication delay, and imperfect acceleration implementation, in the above discussion. As summarized in our recent paper Meng et al., 2018), there is a convenient way to consider various uncertainties. Assuming all these uncertainties can be reflected as the uncertainty on vehicle position measurements, we get a lumped parameter model. To avoid collisions caused by such uncertainty, we simply increase the minimum distance gap $g_{\min}$ appeared in ((3) or (12) to ensure safety. Some related testing results can be found in (Meng et al., 2018). Besides, almost all existing trajectory planning methods summarized in surveys (Li and Wang, 2007; Katrakazas et al., 2015; González et al., 2016) assumed that the errors are bounded. If the vehicles detect significant accumulation errors, we will re-start the trajectory planning based on the new measurements to solve this problem.

It should be pointed out that no existing studies aimed to solve all the problem in the stage of trajectory planning. The output of our algorithm is a reference trajectories. We will apply a tracking controller to suppress the deviation between this reference trajectory and the empirical trajectory. Moreover, every autonomous vehicles will monitor the gap between the leading vehicle and itself to stop for avoiding crash caused by inaccurate planned trajectory and re-start the trajectory planning based on the new measurements.

Fifth, our trajectory planning method can also be used to guarantee the stability of traffic streams, although this paper is for trajectory planning rather than tracing control.

As shown in our previous studies (Li and Ouyang, 2011; Li et al., 2012; Li et al., 2014a), the string instability of vehicle platoons are caused by inappropriate car-following behaviors of human drivers. There are generally two ways to suppress traffic oscillation and maintain the string stability of vehicle platoons. In the first way, we design special feedback controllers to decrease traffic oscillations; see discussions in (Ward and Wilson, 2011; Sterna et al., 2018). In the second way, we make special feed-forward trajectory planning to absorb traffic oscillations.

Actually, we have already proposed a sufficient condition to guarantee the string stable trajectory planning as (29). It is easy to prove its sufficiency according to the rigorous definitions of string stability in (Stüdli et al., 2017).

5. Conclusions

This paper investigates a general multi-trajectory planning problem on a highway with the CAV technology. Noting that communication cost and implementation difficulty naturally lead to parsimonious trajectory requirements, we propose a parsimonious trajectory design method. Both continuous-time type vehicle trajectory planning problem and discrete-time type vehicle trajectory planning problem are discussed here. The continuous-time model yields theoretical insights into the feasibility conditions of the trajectory solution, which leads to the analytical solutions to the two special boundary trajectory problems. For the discrete-time model, we appended a l_1 norm term with a customized weight in the objective to regulate trajectory smoothness and parsimoniousness. The numerical studies on three representative applications reveal the efficiency of our method. They also shed insights into how the l_1 norm term help regulate the smoothness and parsimoniousness of individual vehicle trajectories and a platoon of vehicle trajectories.

This study can be extended in several directions. It will be interesting to integrate this model with other high-level control models (e.g., signal timing and ramp metering) to explore CAV based controls for a highway corridor or even a transportation network. It will be of near-future significance if mixed traffic including human-driven vehicles is considered. Further, we are building a new testing ground for intelligent vehicles (Li et al., 2016b) and are manufacturing several testing vehicles. When the hardware is ready, field experiments can be conducted to validate and calibrate the theoretical models.

Acknowledgements

This work was supported in part by National Natural Science Foundation of China (Grant No. 91520301), National Key R&D Program in China (Grant No. 2016YFB0100906) and the US National Science Foundation through Grant #1558887.

Appendix: Notation

Symbols	Meaning
T	The length of time horizon
n	The number of discretization points
δ	The discretization interval
$s^i(t)$	The continuous-time position of the front bumper of vehicle i
$s_{s_0 v_0 t_0 a_0}(t)$	The trajectory that starts at state (s_0,v_0,t_0) (this triple denotes location s_0, speed v_0, and time t_0) and initially accelerates with rate a_0.
s^i_{j+1}	The discrete-time position of the front bumper of vehicle i at time $j\delta$
$a_{\max}$	The maximum acceleration rate
$d_{\max}$	The maximum deceleration rate
$v_{\max}$	The maximum allowable speed
$g_{\min}$	The minimum allowable gap between the front bumper of the following vehicle and the rear bumper of the leading vehicle
$g_{\max}$	The maximum allowable gap between the front bumper of the following vehicle and the rear bumper of the leading vehicle, if we consider two vehicles are in car following status
v^i_{begin}	The initial speed of vehicle i
s^{i-}_{final}	The minimum allowable position of vehicle i
s^{i+}_{final}	The maximum allowable position of vehicle i

References

Abboud, K., Omar, H.A., Zhuang, W., 2016. Interworking of DSRC and cellular network technologies for V2X communications: A survey. IEEE Trans. Vehicular Technol. 65 (12), 9457–9470.

Ahn, K., Rakha, H., Park, S., 2013. Ecodrive application: Algorithmic development and preliminary testing. Transp. Res. Rec. 2341, 1–11.

Ahn, K., Rakha, H., Trani, A., Van Aerde, M., 2002. Estimating vehicle fuel consumption and emissions based on instantaneous speed and acceleration levels. ASCE J. Transp. Eng. 128 (2), 182–190.

Bach, F., Jenatton, R., Mairal, J., Obozinski, G., 2012. Convex Optimization with Sparsity-Inducing Norms. In: Sra, S., Nowozin, S., Wright, S.J. (Eds.), Optimization For Machine Learning. MIT Press.

Bertsekas, D., 1999. Nonlinear Programming. Athena Scientific.

Boyd, S., Vandenberghe, L., 2004. Convex Optimization. Cambridge University Press, Cambridge, UK.

Cassandras, C.G., 2018. Automating mobility in smart cities. Ann. Rev. Control 44, 1–8.

Cassidy, M.J., Windover, J.R., 1998. Driver memory: Motorist selection and retention of individualized headways in highway traffic. Transp. Res. Part A 32 (2), 129–137.

Clarkson, K.L., 2010. Coresets, sparse greedy approximation, and the Frank-Wolfe algorithm. ACM Trans. Algor. 6 (4).

Daganzo, C.F., 1999. Remarks on traffic flow modeling and its applications. In: Traffic and Mobility. Springer, Berlin, Heidelberg, pp. 105–115.

Demir, E., Bektas, T., Laporte, G., 2014. A review of recent research on green road freight transportation. Eur. J. Oper. Res. 237, 775–793.

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

Efron, B., Hastie, T., Johnstone, I., Tibshirani, R., 2004. Least angle regression. Ann. Stat. 32 (2), 407–499.

Elad, M., 2010. Sparse and Redundant Representations. Springer.

Frank, M., Wolfe, P., 1956. An algorithm for quadratic programming. Naval Res. Logistics Q. 3 (1-2), 95–110.

Fukushima, M., 1984. A modified Frank-Wolfe algorithm for solving the traffic assignment problem. Transp. Res. Part B 18 (2), 169–177.

González, D., Pérez, J., Milanés, V., Nashashibi, F., 2016. A review of motion planning techniques for automated vehicles. IEEE Trans. Intell. Transp. Syst. 17 (4), 1135–1145.

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

Hastie, T., Tibshirani, R., Friedman, J., 2009. The Elements of Statistical Learning: Data Mining. Inference, and Prediction, 2nd edition Springer-Verlag, New York, NY, USA.

He, Z., Zheng, L., Song, L., Zhu, N., 2017. A jam-absorption driving strategy for mitigating traffic oscillations. IEEE Trans. Intell. Transp. Syst. 18 (4), 802–813.

Helbing, D., Treiber, M., Kesting, A., Schönhof, M., 2009. Theoretical vs. empirical classification and prediction of congested traffic states. Eur. Phys. J. B 69, 583–598.

Jaggi, M., 2013. Revisiting Frank-Wolfe: Projection-Free sparse convex optimization. J. Mach. Learn. Res. 28 (1), 427–435.

Katrakazas, C., Quddus, M., Chen, W.H., Deka, L., 2015. Real-time motion planning methods for autonomous on-road driving: State-of-the-art and future research directions. Transp. Res. Part C 60, 416–442.

Kerner, B.S., 2009. Introduction to Modern Traffic Flow Theory and Control: The Long Road to Three-Phase Traffic Theory. Springer-Verlag, Berlin.

Lacoste-Julien, S., 2016. Convergence rate of Frank-Wolfe for non-convex objectives. CoRR abs/1607.00345.

Lee, J., Park, B., 2012. Development and evaluation of a cooperative vehicle intersection control algorithm under the connected vehicles environment. IEEE Trans. Intelligent 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 80, 190–205.

Li, L., 2015. Selected Applications of Convex Optimization. Springer, New York, NY, USA.

Li, L., Chen, X., Zhang, L., 2016a. A global optimization algorithm for trajectory data based car-following model calibration. Transp. Res. Part C 68, 311–332.

Li, X., Cui, J., An, S., Parsafard, M., 2014a. Stop-and-go traffic analysis: theoretical properties, environmental impacts and oscillation mitigation. Transp. Res. Part B 70, 319–339.

Li, Z., Elefteriadou, L., Ranka, S., 2014b. Signal control optimization for automated vehicles at isolated signalized intersections. Transp. Res. Part C 49, 1–18.

Li, L., Huang, W.-L., Liu, Y., Zheng, N.-N., Wang, F.-Y., 2016b. Intelligence testing for autonomous vehicles: A new approach. IEEE Trans. Intell. Vehicles 1 (2), 158–166.

Li, X., Ouyang, Y., 2011. Characterization of traffic oscillation propagation under nonlinear car-following laws. Transp. Res. Part B 45 (9), 1346–1361.

Li, L., Su, X., Wang, Y., Lin, Y., Li, Z., Li, Y., 2015. Robust causality dependence mining in big data network and its application to traffic flow prediction. Transp. Res. Part C 58, 292–307.

Li, L., Wang, F.-Y., 2003. An integrated design framework for driver/passenger-oriented trajectory planning. In: Proceedings of IEEE Conference on Intelligent Transportation Systems, pp. 1764–1769.

Li, L., Wang, F.-Y., 2006. Cooperative driving at blind crossings using intervehicle communication. IEEE Trans. Vehicular Technol. 55 (6), 1712–1724.

Li, L., Wang, F.-Y., 2007. Advanced Motion Control and Sensing for Intelligent Vehicles. Springer, New York, NY, USA.

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, L., Wen, D., Yao, D., 2014c. A survey of traffic control with vehicular communications. IEEE Trans. Intell. Transp. Syst. 15 (1), 425–432.
Li, X., Peng, F., Ouyang, Y., 2010. Measurement and estimation of traffic oscillation properties. Transp. Res. Part B 1 (44), 1–14.
Li, X., Ghiasi, A., Qu, X., Xu, Z., 2018. A piecewise trajectory optimization model for connected automated vehicles: Exact optimization algorithm and queue propagation analysis. Transp. Res. Part B 118, 429–456.
Li, P., Zhou, X., 2017b. Recasting and optimizing intersection automation as a connected-and-automated-vehicle (CAV) scheduling problem: A sequential branch-and-bound search approach in phase-time-traffic hypernetwork. Transp. Res. Part B 105, 479–506.
Liu, W., Li, Z., Li, L., Wang, F.-Y., 2017. Parking like human: A direct trajectory planning solution. IEEE Trans. Intell. Transp. Syst. 18 (12), 3388–3397.
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 95, 421–441.
Meng, Y., Li, L., Wang, F.-Y., Li, K., Li, Z., 2018. Analysis of cooperative driving strategies for non-signalized intersections. IEEE Trans. Vehicular Technol. 67 (4), 2900–2911.
Nishi, R., Tomoeda, A., Shimura, K., Nishinari, K., 2013. Theory of jam-absorption driving. Transp. Res. Part B 50, 116–129.
Nocedal, J., Wring, S.J., 2006. Numerical Optimization, 2nd edition Springer-Verlag, New York, NY, USA.
Parikh, N., Boyd, S., 2013. Proximal Algorithms. Found. Trends Optim. 1 (3), 123–231.
Rios-Torres, J., Malikopoulos, A.A., 2017a. Automated and cooperative vehicle merging at highway on-ramps. IEEE Trans. Intell. Transp. Syst. 18 (4), 780–789.
Rios-Torres, J., Malikopoulos, A.A., 2017b. A survey on the coordination of connected and automated vehicles at intersections and merging at highway on-ramps. IEEE Trans. Intelligent Transp. Syst. 18 (5), 1066–1077.
Sterna, R.E., Cui, S., Monache, M.L.D., Bhadani, R., Bunting, M., Churchill, M., Hamilton, N., Haulcy, R., Pohlmann, H., Wu, F., Piccolih, B., Seibold, B., Sprinkle, J., Work, D.B., 2018. Dissipation of stop-and-go waves via control of autonomous vehicles: Field experiments. Transp. Res. Part C 89, 205–221.
Stüdli, S., Seron, M.M., Middleton, R.H., 2017. From vehicular platoons to general networked systems: String stability and related concepts. Ann. Rev. Control 44, 157–172.
Taniguchi, Y., Nishi, R., Ezaki, T., Nishinari, K., 2015. Jam-absorption driving with a car-following model. Physica A 433, 304–315.
Tao, R., Wei, H., Wang, Y., Sisiopiku, V.P., 2005. Modeling speed disturbance absorption following current state-control action-expected state chains: Integrated car-following and lane-changing scenarios. Transp. Res. Record 1934, 83–93.
Tibshirani, R., 1996. Regression shrinkage and selection via the lasso. J. R. Stat. Soc. 58 (1), 267–288.
Ward, J.A., Wilson, R.E., 2011. Criteria for convective versus absolute string instability in car-following models. Proc. R. Soc. A 467, 2185–2208.
Wei, Y., Avci, C., Liu, J., Belezamo, B., Aydın, N., Li, P., Zhou, X., 2017. Dynamic programming-based multi-vehicle longitudinal trajectory optimization with simplified car following models. Transp. Res. Part B 106, 102–129.
Weintraub, A., Ortiz, C., González, J., 1985. Accelerating convergence of the Frank-Wolfe algorithm. Transp. Res. Part B 19 (2), 113–122.
Wong, S.C., Wong, W.T., Leung, C.M., Tong, C.O., 2002. Group-based optimization of a time-dependent TRANSYT traffic model for area traffic control. Transp. Res. Part B 36 (4), 291–312.
Yang, H., Jin, W.-L., 2014. A control theoretic formulation of green driving strategies based on inter-vehicle communications. Transp. Res. Part C: Emerging Technol. 41, 48–60.
Yin, Y., 2008. Robust optimal traffic signal timing. Transp. Res. Part B 42 (10), 911–924.
Yu, C., Feng, Y., Liu, H.X., Ma, W., Yang, X., 2018. Integrated optimization of traffic signals and vehicle trajectories at isolated urban intersections. Transp. Res. Part B 112, 89–112.
Zheng, J., Liu, H.X., 2017. Estimating traffic volumes for signalized intersections using connected vehicle data. Transportation Research Part C 79, 347–362.
Zheng, K., Zheng, Q., Yang, H., Zhao, L., Hou, L., Chatzimisios, P., 2015. Reliable and efficient autonomous driving: The need for heterogeneous vehicular networks. IEEE Commun. Mag. 53 (12), 72–79.
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 95, 394–420.