Transportation Research Part B 95 (2017) 421–441

Contents lists available at ScienceDirect

Transportation Research Part B

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

Parsimonious shooting heuristic for trajectory design of connected automated traffic part II: Computational issues and optimization

Jiaqi Ma[a], Xiaopeng Li[b,*], Fang Zhou[a], Jia Hu[c], B. Brian Park[d]

[a] *Transportation Solutions and Technology Applications Division, Leidos, Inc., Reston, VA 20190, United States*
[b] *Department of Civil and Environmental Engineering, University of South Florida, FL 33620, United States*
[c] *Turner Fairbank Highway Research Center, Federal Highway Administration, VA 22101, United States*
[d] *Department of Civil and Environmental Engineering, University of Virginia, VA 22904, United States*

ARTICLE INFO

Article history:
Available online 14 July 2016

Keywords:
Connected vehicles
Automated vehicles
Traffic smoothing
Trajectory optimization
Traffic signal
Shooting heuristic
Customized numerical-gradient heuristic
Expedited objective evaluation

ABSTRACT

Advanced connected and automated vehicle technologies enable us to modify driving behavior and control vehicle trajectories, which have been greatly constrained by human limits in existing manually-driven highway traffic. In order to maximize benefits from these technologies on highway traffic management, vehicle trajectories need to be not only controlled at the individual level but also coordinated collectively for a stream of traffic. As one of the pioneering attempts to highway traffic trajectory control, Part I of this study (Zhou et al., 2016) proposed a parsimonious shooting heuristic (SH) algorithm for constructing feasible trajectories for a stream of vehicles considering realistic constraints including vehicle kinematic limits, traffic arrival patterns, car-following safety, and signal operations. Based on the algorithmic and theoretical developments in the preceding paper, this paper proposes a holistic optimization framework for identifying a stream of vehicle trajectories that yield the optimum traffic performance measures on mobility, environment and safety. The computational complexity and mobility optimality of SH is theoretically analyzed, and verifies superior computational performance and high solution quality of SH. A numerical sub-gradient-based algorithm with SH as a subroutine (NG-SH) is proposed to simultaneously optimize travel time, a surrogate safety measure, and fuel consumption for a stream of vehicles on a signalized highway section. Numerical examples are conducted to illustrate computational and theoretical findings. They show that vehicle trajectories generated from NG-SH significantly outperform the benchmark case with all human drivers at all measures for all experimental scenarios. This study reveals a great potential of transformative trajectory optimization approaches in transportation engineering applications. It lays a solid foundation for developing holistic cooperative control strategies on a general transportation network with emerging technologies.

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

http://dx.doi.org/10.1016/j.trb.2016.06.010
0191-2615/© 2016 Elsevier Ltd. All rights reserved.

1. Introduction

Highway traffic is frequently forced to decelerate and accelerate (or stop and go), particularly along signalized arterials due to alternating green and red lights. Vehicles engaged in repeated stop-and-go movements are exposed to high crash risks (Hoffmann and Mortimer, 1994), extra driver discomfort (Beard and Griffin, 2013), and excessive fuel consumption and emissions (Li et al., 2014). Further, when vehicles slow down or stop, traffic throughput decreases and capacity drops (Cassidy and Bertini, 1999), causing excessive travel delay.

In recent years, increasing research efforts have been on vehicle smoothing using advanced vehicle-based technologies on both freeways and arterials. See Part I of this study (Zhou et al., 2016) for a thorough review on this topic. One approach to smooth vehicle trajectories is variable speed limits (VSL) to dynamically regulate traffic speed based on real-time information of traffic states. On freeways, VSL is applied to mitigate sudden brakes that may trigger stop-and-go traffic, and this strategy is also referred to as speed harmonization (see Lu and Shladover (2014); Ma et al. (2016) for reviews on this topic). On signalized arterials, VSL is applied to guide a vehicle to arrive at the intersection during a green phase to avoid a full stop (e.g., Sanchez et al. (2006); Trayford et al. (1984a; 1984b); Wu et al. (2015)). These VSL-based approaches are relatively easy to implement on the existing infrastructure and have been widely deployed around the world (Robinson, 2000). However, VSL can only regulate traffic in an aggregated manner without much control of each individual vehicle's acceleration profile, and its performance is largely dependent on driver compliance (Brinckerhoff, 2010).

The other traffic smoothing approach, the focus of this study, is based on advanced connected and automated vehicle (CAV) technologies that enable precise control of individual vehicle trajectories (e.g., Ahn et al. (2013); Wang et al. (2014a); 2014b)). In addition to freeway speed harmonization, CAV brings opportunities for new paradigms of arterial intersection traffic control. With precise trajectory control, vehicles can either adjust their driving according to existing traffic signal timing plans to smoothly pass the intersection during green phases (e.g., Kamalanathsharma et al. (2013)), or coordinate with each other such that they can cross the intersection without an explicit traffic light like a school of fish (e.g., Dresner and Stone (2008); Lee and Park (2012)).

However, most existing studies mainly address individual trajectory control instead of coordinating a stream of vehicles that interact with each other. Most control methods developed so far either seek algorithm efficiency by ignoring detailed acceleration tuning (e.g., allowing speed jumps) or rely on complex algorithms that may impede real-time applications. Further, the integration of traffic signal timing and vehicle trajectory control is yet to be investigated. To address these research gaps, Part I of this study (Zhou et al., 2016) proposed a parsimonious Shooting Heuristic (SH) algorithm that can efficiently smooth trajectories of a stream of vehicles approaching a signalized intersection by controling detailed acceleration profiles. The SH algorithm represent each infinite-dimensional vehicle trajectory with a few segments of analytical quadratic curves, and therefore efficiently construct a large number of vehicle trajectories subject to physical limits, car-following safety and traffic signal timing. Further, this algorithm uses only a few parameters (i.e., acceleration levels) to control overall smoothness of the stream of vehicle trajectories, and thus it is suitable for real-time traffic smoothing optimization. The Part I paper describes this SH algorithm and reveals its elegant theoretical properties. To complete the entire study, this Part II paper proposes to embed SH into an efficient optimization framework that aims at finding the optimum vehicle trajectories on a signalized highway segment in order to minimize a number of traffic performance measures simultaneously, including travel time, fuel consumption, and safety risks. This paper also discusses properties of SH on optimality and computational complexity. Overall, this paper makes contributions to the literature from the following three perspectives: (1) simplifying the complex multi-trajectory optimization problem and consider only a few control variables, dealing with a small number of analytical trajectory segments; (2) theoretically showing the appealing properties of the SH algorithm on computational complexity and optimality; and (3) conducting numerical experiments to verify efficiency of this proposed optimization framework and draw managerial insights. Note that in this paper, we assume all vehicles are connected automated vehicles that are subject to centralized control. Although this assumption is somewhat utopian and cannot exactly match the reality in the near future when highway traffic is composed of both automated and manual vehicles, the results from this utopian scenario provide insights into the best CAV technologies can achieve and set an ultimate goal for using CAV technologies to improve existing traffic systems.

This paper is organized as follows. Section 2 states the studied CAV trajectory optimization problem, including problem settings, constraints and system objectives. Section 3 briefly reviews the SH algorithm developed in the Part I paper (Zhou et al., 2016), and analyzes its computational complexity and optimality properties. Efficient methods of evaluating the system objectives are also discussed. Section 4 demonstrates the proposed optimization framework, and tests its solution efficiency and related properties with numerical examples. Section 5 concludes this paper.

2. Problem statement

2.1. Trajectory construction problem

This section briefly reviews the problem of constructing a stream of feasible vehicle trajectories on a signalized one-lane highway segment as illustrated in Fig. 1. The details are presented in the Part I paper.

As illustrated in Fig. 1, we consider a highway segment of length L. The set of locations of this segment is $[0, L]$. There is a traffic signal at Location L. The signal timing is fixed with a green phase time of G and a red phase time of R. The signal

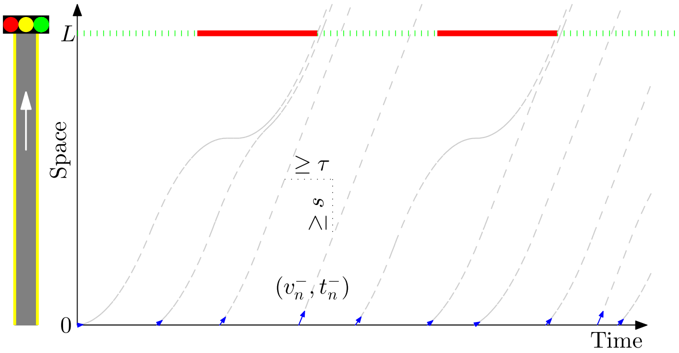


Fig. 1. Illustration of trajectory construction (from Zhou et al. (2016)).

starts with a green phase at time 0. The cycle length $C := G + R$. We define function

$$G(t) := \begin{cases} t, & \text{if } \bmod (t, C \in [0, R)); \\ \lceil t/C \rceil \cdot C, & \text{otherwise,} \end{cases}, \forall t \in (-\infty, +\infty),$$

which essentially finds the next r_m, the start of next green time. Note that $G(t)$ is an increasing function. Further, $G(t) = t$ indicates time t is in a green phase, or otherwise $G(t) > t$. Note that the assumption of fixed signal timing is only for simplicity of presentation and can be relaxed to any predictable signal phasing scenarios. Under ubiquitous communication between vehicles and infrastructure, we assume that a central controller can always access signal phasing information of the next planning horizon for vehicle trajectory design in advance. We leave the investigation of combined optimization of trajectory control and signal phasing for future studies.

A set of consecutive vehicles $\mathcal{N} := \{1, 2, \ldots, N\}$ travel from location 0 to L. The dynamics of these vehicles follow a number of constraints specified with the notion of trajectories. A vehicle *trajectory* is denoted with a second-order semi-differentiable function $p(t), \forall t \in (-\infty, \infty)$[1] such that its first order differential (or *velocity*) $\dot{p}(t)$ is absolutely continuous, and its second-order right-differential $\ddot{p}(t)$ (or *acceleration*) is Riemann integrable. We denote the set of all trajectories as $\bar{\mathcal{T}}$. For notation convenience, we denote the whole trajectory of function $p(t)$ with p, and the section of p over time interval $[t^-, t^+]$ by *trajectory section* $p(t^- : t^+), \forall -\infty \leq t^- \leq t^+ \leq \infty$. Let p_n denote the trajectory for vehicle $n, \forall n \in \mathcal{N}$, and we call $P := [p_n]_{n \in \mathcal{N}}$ a *trajectory vector* that contains all dynamic information of these vehicles. Trajectory vector P should satisfy the following constraints to be feasible.

Let v_n^- and t_n^- denote the speed and time when vehicle n enters location 0, $\forall n \in \mathcal{N}$, and we call pair $\left(v_n^-, t_n^-\right)$ the *entry boundary condition* of vehicle n. We assume that all vehicles' boundary conditions can be exactly predicted (e.g., with the advanced sensing technology in the future transportation infrastructure). Then each trajectory p_n shall satisfies the *entry boundary constraint* $p_n(t_n^-) = 0$ and $\dot{p}_n(t_n^-) = v_n^-$.

We assume all vehicles are identical. Every vehicle's acceleration is limited in $[\underline{a} < 0, \bar{a} > 0]$, and its speed range is $[0, \bar{v}]$. We say a trajectory section p is*kinetically feasible*if $\dot{p}(t) \in [0, \bar{v}]$ and $\ddot{p}(t) \in [\underline{a}, \bar{a}], \forall t \in (-\infty, \infty)$. We denote the set of all kinetically feasible trajectories as

$$\mathcal{T} := \left\{ p \in \bar{\mathcal{T}} | 0 \leq \dot{p}(t) \leq \bar{v}, \underline{a} \leq \ddot{p}(t) \leq \bar{a}, \forall t \in (-\infty, \infty) \right\}. \tag{1}$$

Therefore, every trajectory p_n shall fall in $\mathcal{T}$, which we call the *kinematic constraint.* Note that traditional car-following models may require that accelerations are dependent on variables such as distance gap and speed difference. The problem investigated in this paper, however, is different. We assume ubiquitous communication and existence of a central controller for trajectory construction for all vehicles. The central controller provides instructions on vehicle trajectories that can optimize the system performance, and these trajectories may not strictly follow a prescribed car-following model. For example, in general car-following models, acceleration is a non-increasing function of gap. This is mainly because of general car-following behavior of the rear vehicle to ensure safety. In our problem, since vehicles are controlled and coordinated by a central controller, as long as the safety constraint is met as discussed below, the rear vehicle can apply any acceleration as instructed by the algorithm. Therefore, we believe the bounded constrains for speeds and accelerations are sufficient for the purpose of this study.

To ensure safety in nominal conditions, a vehicle $n \in \mathcal{N} \backslash \{1\}$ should always be at least a minimum space separation s (which is usually the summation of the length of a vehicle and a safety buffer) behind its preceding vehicle $(n-1)$'s location a reaction time τ (or a communication delay) ago. We call the trajectory obtained by translating $p \in \mathcal{T}$ rightwards by τ and downwards by s the *shadow trajectory* of p. This shadow trajectory is denoted by adding a superscript 's', i.e., p^s, denoting the shadow trajectory of p. Then this *safety constraint* can be stated as $p_n(t) \leq p_{n-1}^S(t), \forall t \in (-\infty, \infty), \forall n \in \mathcal{N}$.

[1] While we are fully aware that a realistic trajectory is finite, we use time span $(-\infty, \infty)$ by extending two ends of a trajectory in a trivial way. This treatment facilitates the problem analysis without loss of generality.

Let $p^{-1}(l)$ denotes the first time when trajectory $p(t)$ arrives at location $l, \forall l \in (-\infty, \infty)$. Then $p^{-1}(L)$ denotes the time when this vehicle exits the highway segment, and it has to be during a green time, i.e., $G\big(p_n^{-1}(L)\big) = p_n^{-1}(L)$. We call this the *green exit time* constraint.

With these constraints, the trajectory construction problem is to identify a vector of trajectories for all vehicles in $\mathcal{N}$ in the following feasible trajectory vector set:

$$\mathcal{P} := \left\{ [p_n]_{n\in\mathcal{N}} \middle| \begin{array}{c} p_n \in \mathcal{T}, (p(t_n^-), \dot{p}(t_n^-)) = (0, v_n^-), G\big(p_n^{-1}(L)\big) = p_n^{-1}(L) \forall n \in \mathcal{N}, t \in (-\infty, \infty), \\ p_{n-1}(t-\tau) - p_n(t) \geq s, \forall n \in \mathcal{N}\backslash\{1\}, t \in (-\infty, \infty) \end{array} \right\}. \tag{2}$$

2.2. Trajectory optimization problem

Vehicle trajectories are arguably the most detailed traffic data among all that are available. Combined with certain vehicle specific data, vehicle trajectories provide complete information to determine most commonly used traffic performance measures, including travel time, fuel consumption, emissions, and safety. We denote the traffic performance for a given trajectory vector $P \in \mathcal{P}$ by $M(P)$, which could be either a single performance measure or a weighted summation of several measures. It can also be a vector if multiple measures are investigated simultaneously. Several candidate performance measures for composing $M(P)$ are described below.

When trajectory vector $P = [p_n]_{n\in\mathcal{N}}$ is given, the corresponding *average travel time* (per vehicle) can be simply formulated as:

$$T(P) := \frac{1}{N} \sum_{n\in\mathcal{N}} \left(t(p_n) := p_n^{-1}(L) - T_n^- \right), \tag{3}$$

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

The *system throughput* can be easily evaluated as:

$$R(P) := \frac{N}{p_N^{-1}(L) - t_1^-}, \tag{4}$$

There are a number of instantaneous fuel consumption and emission measures developed in the literature (e.g., CMEM (Barth et al., 2000), MOVES (Koupal et al., 2002), VT-Micro (Ahn, 1998; Ahn and Van Aerde, 2002), PHEM obtained from the quadratic time geography (Hausberger et al., 2003)). Such an instantaneous measure is usually a function of vehicle's location, velocity and acceleration at the measured time point, and thus we denote vehicle n's instantaneous measure at time t by function $e(p_n(t), \dot{p}_n(t), \ddot{p}_n(t))$. Then the *average fuel consumption* or *emissions* in general can be formulated as:

$$E(P) := \frac{1}{N} \sum_{n\in\mathcal{N}} \int_{t_n^-}^{p_n^{-1}(L)} e(p_n(t), \dot{p}_n(t), \ddot{p}_n(t)) dt, \tag{5}$$

Since the focus of this paper is development of a trajectory optimization framework rather than testing fuel-consumption models, for illustration purposes, we just use the VT-Micro model. This is reasonable because the VT-Micro model has been widely used in traffic engineering research as a simple yet relatively accurate measure of vehicle fuel consumption and emission levels. Then the e function is formulated in Eq. (6). Nonetheless, we do acknowledge that the VT-Micro model may have limitations for certain applications compared with other competing models. Should this be the case, we can just formulate the e function with other proper instantaneous fuel consumption functions, which shall not affect the optimization method proposed in this paper.

As shown in Eq. (6), due to the regression nature of the VT-Micro model, there are bounds for dependent variables (i.e., speeds and accelerations). If variable values exceed these bounds, it is recommended by Ahn and Van Aerde (2002) that "the boundary speed and acceleration levels are used instead to ensure realistic estimates." However, it is noted that truncation of speed and acceleration range of the model may cause underestimate of fuel consumption amounts and may lead to sub-optimal solutions to the final trajectory plan. Careful model choices should be considered in balancing computational efficiency and accuracy for extreme cases when acceleration rates may go beyond the bounds. Fortunately, the optimal results of most practical cases in the studied problem have smooth trajectories with relatively mild acceleration rates, which normally fall in the range of the VT-Micro model and shall not cause the boundary truncation issue.

$$e(\dot{p}_n(t), \ddot{p}_n(t)) = \exp\left\{ \sum_{i=0}^{3} \sum_{j=0}^{3} K_{ij}(\ddot{p}_n(t)) \left(\lfloor \dot{p}_n(t) \rfloor_0^{120 \text{ km/h}} \right)^i \left(\lfloor \ddot{p}_n(t) \rfloor_{-5 \text{ km/h/sec}}^{13 \text{ km/h/sec}} \right)^j \right\}. \tag{6}$$

where coefficient $K_{ij}(\ddot{p}_n(t))$ depends on the sign of $\ddot{p}_n(t)$, the vehicle type and the measure-of effectiveness (MOE) type (e.g., fuel consumption, emission of a certain pollutant). This study uses values of $\left[K_{ij}(\ddot{p}_n(t))\right]_{i,j=0,1,2,3}$ obtained from the test results with eight light duty vehicles collected by the Oak Ridge National Laboratory, as show in Table 2.[2] (Table 1)

[2] These data were extracted from the emission software developed by Dr. Hesham Rakha (http://filebox.vt.edu/users/hrakha/Software.htm).

Table 1
Coefficients for the MOE of fuel consumption (the units of fuel consumption, speed and acceleration are in liters/sec, kph, and kph/sec, respectively).

K_{ij}	$\ddot{p}_n(t)$ is positive				$\ddot{p}_n(t)$ is negative			
	$j=0$	$j=1$	$j=2$	$j=3$	$j=0$	$j=1$	$j=2$	$j=3$
$i=0$	−7.735	0.2295	−5.61E-3	9.773E-5	−7.73t	−0.01799	−4.27E-3	1.8829E-4
$i=1$	0.02799	0.0068	−7.722E-4	8.38E-6	0.02804	7.72E-3	8.375E-4	3.387E-5
$i=2$	−2.228E-4	−4.402E-5	7.90E-7	8.17E-7	−2.199E-4	−5.219E-5	−7.44E-6	2.77E-7
$i=3$	1.09E-6	4.80E-8	3.27E-8	−7.79E-9	1.08E-6	2.47E-7	4.87E-8	3.79E-10

Table 2
Default input parameter values.

Parameter	L(m)	G(s)	R(s)	N	$\underline{a}$ (m/s^2)	$\bar{a}$(m/s^2)	$\bar{v}$(m/s)	s(m)	τ(s)	f^s
Value	1000	25	25	50	−10	2	36	7	1	0.6

Traffic safety is often less intuitive to quantify with a deterministic and exact measure. Nonetheless, a number of surrogate models have been proposed as quantitative measures of risk or closeness to a potential collision. Such a surrogate measure can be fully quantified with vehicle trajectory information and can be easily integrated into the proposed optimization framework. For illustration purposes, this study uses a surrogate measure based on inverse time-to-collision (iTTC). At time t, the iTTC of a following vehicle $n \in \mathcal{N} \backslash \{1\}$ is $\frac{\dot{p}_n(t)-\dot{p}_{n-1}(t)}{p_{n-1}(t)-p_n(t)-g_{n-1}}$, where g_{n-1} is the vehicle length of vehicle $n-1$. The studied*safety surrogate measure* is formulated as an integral of the positive portion of iTTC over the travel time on the road segment for an average vehicle that follows others:

$$S(P) := \frac{1}{N} \sum_{n \in \mathcal{N} \backslash \{1\}} \int_{t_n^-}^{p_{n-1}^{-1}(L)} \left[\frac{\dot{p}_n(t) - \dot{p}_{n-1}(t)}{p_{n-1}(t) - p_n(t) - g_{n-1}} \right]^+ dt, \tag{7}$$

where operator $[\cdot]^+ := \max\{0, \cdot\}$. We use the above formulation for the following reasons. First, compared with other surrogate measures such as time-to-collision (TTC) (Hoffmann and Mortimer, 1994) and deceleration rate difference (DRD) (Park et al., 2011), iTTC, even with a negative value, is more positively related to collision risks Balas (2006). Therefore, the integral of iTTC is a reasonable indicator of a trajectory's overall collision risk. Secondly, iTTC can avoid certain singularities that may occur in other surrogate measures. For example, TTC goes to infinity as vehicles $n-1$ and n operates at the same speed. Thirdly, some other surrogate measures (e.g., TTC and DRD) are often used in conjunction with a binary threshold, which would be more computationally expensive to process in an integral (e.g., a Heaviside step function may have to be used and then the integral can be only evaluated numerically), while the integral of iTTC can be solved analytically using a proposed method in this paper.

System performance $M(P)$ can be formulated as one of, a weighted summation of, or a vector of measures (3)–(7). With this, the trajectory optimization problem is to identify a set of feasible trajectory vectors that optimizes (or minimizes) this system performance measure:

$$\min_{P \in \mathcal{P}} M(P) \tag{8}$$

Remark 1. While this study focuses on measures (3)–(7), the optimization framework can be easily extended to include other measures (e.g., driving comfort) as long as they can be characterized with trajectory vectors.

Note that controlled vehicle behavior may resemble some traits of human driving behavior such that the user acceptability and trust of automated vehicle technology can be improved among the driving public. This paper considers this though the use of kinetically feasible accelerations and driver behavior that are inspired by Newell's car-following model. In addition, the designed trajectory are smoother than Newell's simplified trajectory and this further improves driver comfort and acceptability. The user acceptability and trust of such technology itself is an interesting but complex issue and we left the comprehensive investigation for future studies.

3. SH-Based optimization framework

Problem (8) is an infinite dimension optimization problem with nonlinear objective and constraints. It is hard to solve it to an exact solution in a short time for real-time applications. To obviate the computational issue and draw insights of the problem structure, we propose a general optimization framework that builds upon the efficient shooting heuristic (SH) algorithm developed in the Part I paper (Zhou et al., 2016). Fig. 2 illustrates how this framework works. Basically, we first initialize a small set of control parameters and feed them to the SH algorithm to generate a feasible trajectory vector P. Then system performance $M(P)$ are evaluated by efficient methods based on the analytical properties of the SH solution. The evaluation result is fed to the central optimization algorithm. This algorithm checks whether P is already optimal. If

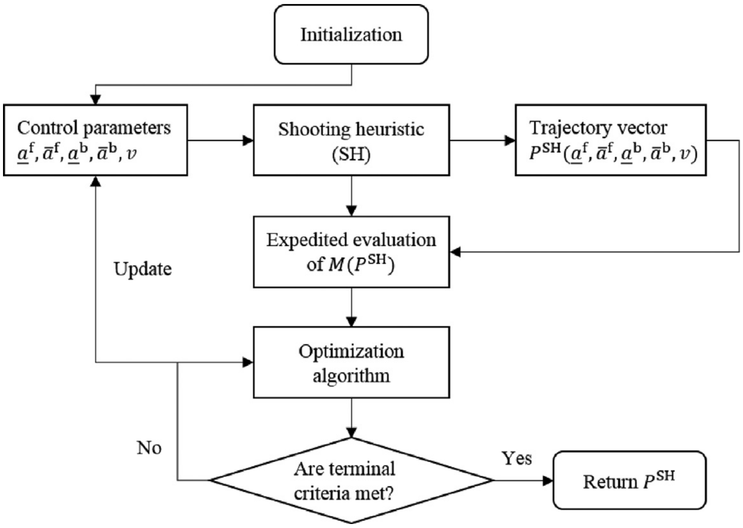

Initialization
Control parameters $\underline{a}^{\mathrm{f}}, \bar{a}^{\mathrm{f}}, \underline{a}^{\mathrm{b}}, \bar{a}^{\mathrm{b}}, v$
Shooting heuristic (SH)
Trajectory vector $P^{\mathrm{SH}}(\underline{a}^{\mathrm{f}}, \bar{a}^{\mathrm{f}}, \underline{a}^{\mathrm{b}}, \bar{a}^{\mathrm{b}}, v)$
Expedited evaluation of $M(P^{\mathrm{SH}})$
Update
Optimization algorithm
No
Are terminal criteria met?
Yes
Return P^{SH}

Fig. 2. Optimization framework.

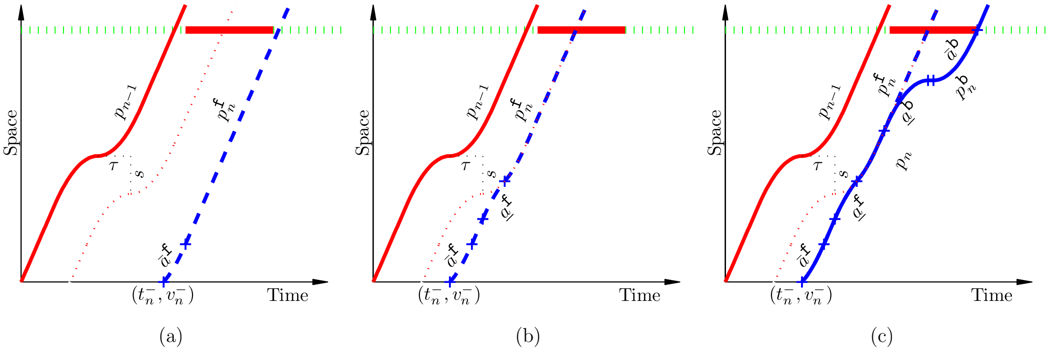


Fig. 3. (a) Forward shooting process without activating safety constraint; (b) forward shooting process with activating safety constraint; and (c) backward shooting process (adapted from Zhou et al. (2016)).

yes, it returns P as the optimal solution. Otherwise, it updates the control parameter values based on certain rules and goes to the next iteration. This section describes the components of this optimization framework and analyzes relevant properties. Section 3.1 revisits the SH algorithm and also investigates its analytical properties on computational complexity and optimality on travel time (3) and throughput (4). Section 3.2 proposes efficient methods to evaluate nonlinear integral measures of fuel consumption (5) and safety (7). Section 3.3.1 describes the optimization algorithm in detail.

3.1. Shooting heuristics

3.1.1. Algorithm revisit

This subsection briefly reviews the SH algorithm from Zhou et al. (2016), and interested readers can refer to it for detailed algorithmic descriptions and theoretical property analyses. To better serve the optimization framework, this paper slightly adapts the original SH algorithm by allowing the maximum cruising speed to be adjustable. The SH algorithm visits vehicles 1 through N sequentially and constructs a feasible trajectory as a piecewise quadratic function where every two consecutive *quadratic segments* are tangent for the current vehicle during each visit. This way, each constructed trajectory is differentiable everywhere and does not contain any speed jump. Fig. 3 illustrates the basic processes to construct a trajectory for each vehicle in the algorithm. Figs. 3(a) and (b) shows a forward shooting process, respectively in two cases, which constructs a kinematically feasible trajectory that satisfies the current vehicle's entry boundary condition and safety constraint against the preceding vehicle, if any. Fig. 3(c) shows a backward shooting process (BSP) that modifies the trajectory from FSP to the exit of the intersection during a green light.

At each vehicle n, FSP first shoots two sequential quadratic segments, as illustrated in Fig. 3(a). The first segment starts off the entry boundary condition $\left(v_n^-, t_n^-\right)$ at location 0 (or its entry *state point* $\left(0, v_n^-, t_n^-\right)$ defined as a 3-tuple of location, speed and time) to reach a *target cruising speed* $v \in (0, \bar{v}]$ at a *forward acceleration*$\bar{a}^{\mathrm{f}} \in (0, \bar{a}]$. We can easily calculate that this segment takes $(v - v_n^-)/\bar{a}^{\mathrm{f}}$ time. The second segment is a straight ray at target cruise speed v all the way down. Note that the second segment exits location L at a time formulated as:

$$\hat{t}_n^+\left(v, \bar{a}^{\mathrm{f}}\right) := t_n^- + \begin{cases} \frac{-v_n^- - \sqrt{\left(v_n^-\right)^2 + 2\bar{a}^{\mathrm{f}}L}}{\bar{a}^{\mathrm{f}}}, & \text{if } L \le \frac{v^2 - \left(v_n^-\right)^2}{2\bar{a}^{\mathrm{f}}}; \\ \frac{L}{v} + \frac{\left(v - v_n^-\right)^2}{2\bar{a}^{\mathrm{f}}v}, & \text{otherwise.} \end{cases} \tag{9}$$

These two segments form a candidate shooting trajectory that will be tested against the safety constraint. As illustrated in Fig. 3(a), if this candidate shooting trajectory (the blue dashed curve) is always below the shadow trajectory of the preceding vehicle p_{n-1}^{S} (the red dotted curve), then the safety constraint is never violated and thus this candidate shooting trajectory is returned as the *forward shooting trajectory* for vehicle n. denoted by p_n^{f}. Note that this is always the case for the lead vehicle, because there is no vehicle ahead (or a dummy preceding vehicle can be set to location ∞), and therefore the safety constraint will never be violated. Otherwise, as illustrated in Fig. 3(b), if the candidate shooting trajectory overpasses p_{n-1}^{S}, then p_n^{f} is constructed by smoothly merging the candidate trajectory into p_{n-1}^{S} with a merging segment at a *forward deceleration* $\underline{a}^{\mathrm{f}} \in [\underline{a}, 0)$. If trajectory p_n^{f} from the forward shooting process is found to run into the red light, as illustrated in Fig. 3(c), a backward shooting process is activated to revise p_n^{f}. The backward shooting process first shifts the section of p_n^{f} above location L rightwards to the start of the next green phase. This shifted trajectory section forms the initial part of a *backward shooting trajectory* p_n^{b}. Then p_n^{b} continues to shoot backwards from this start point at a *backward acceleration* $\bar{a}^{\mathrm{b}} \in (0, \bar{a}]$ until getting close enough to p_n^{f} (which might require p_n^{b} to stop for some time if the separation between the backward shooting start point and p_n^{f} is long in relation to $\bar{a}^{\mathrm{b}}$). Then, p_n^{b} shoots backwards a merging segment at a *backward deceleration* $\underline{a}^{\mathrm{b}} \in [\underline{a}, 0)$ until getting tangent to p_n^{f}. Finally, merging p_n^{f} and p_n^{b} yields a feasible trajectory p_n for vehicle n . Such forward and backward shooting processes are executed from vehicle 1 through vehicle N consecutively, and then SH concludes with a feasible trajectory vector $[p_n]_{n \in \mathcal{N}} \in \mathcal{P}$. Note that SH only deals with a few number of quadratic segments, and each of them is analytically solvable. Further, only a few variables, including four acceleration rates $\left\{\bar{a}^{\mathrm{f}}, \underline{a}^{\mathrm{f}}, \bar{a}^{\mathrm{b}}, \underline{a}^{\mathrm{b}}\right\}$ and the target cruising speed v, are able to control the overall smoothness of all trajectories that determines all studied performance measures. Therefore, the proposed SH algorithm is efficient, parsimonious, simple to implement. It is appropriate to be included as a sub-routine in the presented optimization framework. We summarize these SH steps in the following pseudo code.

SH-1: Initialize acceleration parameters $\bar{a}^{\mathrm{f}}$, $\underline{a}^{\mathrm{f}}$, $\bar{a}^{\mathrm{b}}$ and $\underline{a}^{\mathrm{b}}$ and target cruising speed v, and set $n = 1$.

SH-2: Apply the FSP to obtain p_n^{f}.

SH-3: If $p_n^{\mathrm{f}-1}(L) = G\left(p_n^{\mathrm{f}-1}(L)\right)$, which indicates p_n^{f} exits location L at a green time, then set $p_n = p_n^{\mathrm{f}}$, go to the next step. Otherwise, the BSP is activated to revise p_n^{f} into p_n that exits location L at the beginning of the next green phase.

SH-4: If no feasible p_n^{f} is found from steps SH-2 and SH-3, the algorithm returns with no feasible solution. Otherwise, when $n < N$, set $n = n + 1$ and go back to SH-2. When $n = N$, a feasible trajectory vector $[p_n]_{n \in \mathcal{N}}$ is returned. We denote $[p_n]_{n \in \mathcal{N}}$ by functional $P^{\mathrm{SH}}\left(\underline{a}^{\mathrm{f}}, \bar{a}^{\mathrm{f}}, \underline{a}^{\mathrm{b}}, \bar{a}^{\mathrm{b}}, v\right)$ since this trajectory vector is determined by these five parameters.

3.1.2. Analytical properties

Note that every operation of the aformentioned SH algorithm can be analytically calculated, and thus this heuristic can be solved very efficiently. The theorem below shows that the upper bounds of the computational time and space needed are both $o(N^2)$, which is of low-order polynomial complexity and this verifies the efficiency of the SH algorithm.

Theorem 1. *The computational time complexity of the shooting heuristic SH1-4 is $o(N^2)$, and its computational space complexity is $o(N^2)$ as well.*

Proof. Since each quadratic segment is analytically solved, the computational time for processing a quadratic segment is bounded by a constant, so is the corresponding storage space. Then, to calculate the time and space complexities, we just need to count at most how many segments are processed in total. Let $\bar{k}_n$ denote the number of quadratical segments that p_n contains, $\forall n \in \mathcal{N}$. For vehicle 1, the FSP in SH-2 can at most generate 2 quadratic segments in p_1^{f} (see Fig. 3(a)). In case that the BSP in SH-3 is activated, it generates at maximum 4 more quadratic segments on top of those in p_1^{f} (see Fig. 3(c)). This indicates $\bar{k}_1 \le 6$. For trajectory $n \ge 2$, the FSP in SH-2 may copy quadratic segments from p_{n-1}^{S} in addition to the three segments prior to the merging point (see Fig. 3(b)), and thus p_n^{f} contains at most $\bar{k}_{n-1} + 3$ quadratic segments. Again, the BSP in SH-3, if activated, at most adds 4 more quadratic segments. Therefore, $\bar{k}_n \le \bar{k}_{n-1} + 7$. This indicates that $\bar{k}_n \le \bar{k}_{n-1} + 7 \le \ldots \le \bar{k}_1 + 7(n-1) < 7n$. Therefore, the total number of quadratic segments that the SH algorithm processes is $\sum_{n=1}^{N} \bar{k}_n < 7N(N+1)$. Thus both the computational time and the storage space are on the order of $o\left(\sum_{n=1}^{N} \bar{k}_n\right) = o\left(N^2\right)$. This completes the proof. □

Remark 2. For most realistic instances when traffic is not extremely congested, in every few vehicles, p_n^{f} needs to copy no more than a constant number of quadratic segments from p_{n-1}. This way, every $\bar{k}_n$ shall be bounded by a constant, and thus the time and space complexities of SH reduce to a linear function $O(N)$.

Next, we will investigate the optimality of the SH solution for the travel time measure (3) and throughput measure (4). We first identify a lower bound to travel time (3) in the following proposition.

Proposition 1. *Travel time (3) should have*

$$\min_{P\in\mathcal{P}} T(P) \geq \underline{T} := \frac{1}{N}\sum_{n\in\mathcal{N}}\left(\underline{t}_n^+ - t_n^-\right) \tag{10}$$

where $\underline{t}_n^+$ is the lower bound for the time when vehicle n exits location L as formulated below:

$$\underline{t}_{n10}^+ := \begin{cases} G\left(\hat{t}_n^+(\bar{v},\bar{a})\right), & \text{if } n=1; \\ G\left(\max\left\{\hat{t}_n^+(\bar{v},\bar{a}), \underline{t}_{n-1}^+ + s/\bar{v}+\tau\right\}\right) & \forall n=2,\ldots,N, \end{cases} \tag{11}$$

and exit time function $\hat{t}_n^+$ is defined in Equation (9).

Proof. We will use induction to prove this proposition. Let $[p_n]_{n\in\mathcal{N}}\in\mathcal{P}$ denote the optimal trajectory platoon that yields the minimum total travel time. The induction assumes that vehicle n's minimum exit time (at location L) $p_n^{-1}(L)$ is no less than $\underline{t}_n^+$ defined in (11). The quadratic time geography theory (QTG) proposed in Zhou et al. (2016) shows that no feasible trajectory in $\mathcal{T}$ through state point $\left(0, v_n^-, t_n^-\right)$ can pass location L before $\hat{t}_n^+(\bar{v},\bar{a})$. Therefore, when $n=1$, $p_n^{-1}(L)\geq\hat{t}_1^+(\bar{v},\bar{a})$. Further, the green exist time constraint indicates $p_n^{-1}(L)=G(p_n^{-1}(L))\geq G\left(\hat{t}_1^+(\bar{v},\bar{a})\right)=\underline{t}_1^+$. Thus, the induction assumption holds when $n=1$. Then we need to prove that, when this assumption holds for $n=k-1$, this assumption shall automatically hold for $n=k,\forall k\in\mathcal{N}\backslash\{1\}$. Again, QTG yields $p_k^{-1}(L)\geq\hat{t}_n^+(\bar{v},\bar{a})$. Further, due to the safety constraint, $p_k^{-1}(L)\geq p_{k-1}^{-1}(L+s)+\tau$. This yields $p_k^{-1}(L)\geq p_{k-1}^{-1}(L)+s/\bar{v}+\tau$ due to the vehicle kinematic constraint. Then the induction assumption indicates $p_k^{-1}(L)\geq\underline{t}_{k-1}^+ + s/\bar{v}+\tau$. With the previous QTG inequality, we obtain that $p_k^{-1}(L)\geq\max\left\{\hat{t}_n^+(\bar{v},\bar{a}),\underline{t}_{n-1}^+ + s/\bar{v}+\tau\right\}$. Further, the green exist time constraint indicates $p_k^{-1}(L)=G(p_k^{-1}(L))\geq G\left(\max\left\{\bar{p}_{0v_n^-t_n^-}^{-1}(L),\underline{t}_{n-1}^+ + s/\bar{v}+\tau\right\}\right)=\underline{t}_n^+$. This completes the proof. □

This minimum travel time bound further implies the maximum throughput bound as stated in the following corollary.

Corollary 1. *Throughput (4) should have*

$$\max_{P\in\mathcal{P}} R(P)\leq\bar{R} := \frac{N}{\underline{t}_N^+ - t_1^-} \tag{12}$$

With these results, the following theory shows that, under certain mild conditions, the SH solution can achieve travel time bound $\underline{T}$ and throughput bound $\bar{R}$.

Theorem 2. *When $L\geq\bar{v}^2/(2\bar{a})$, if $P=P^{\mathrm{SH}}\left(\underline{a}^{\mathrm{f}},\bar{a}^{\mathrm{f}},\underline{a}^{\mathrm{b}},\bar{a}^{\mathrm{b}},\bar{v}\right)$ (i.e., the SH solution with $\bar{a}^{\mathrm{f}}=\bar{a}^b=\bar{a}$, $\underline{a}^{\mathrm{f}}=\underline{a}^{\mathrm{b}}=\underline{a}$ and $v=\bar{v}$) is feasible, then $T(P)=\underline{T}$ and $R(P)=\bar{R}$. This also indicates that in this case $\min_{P\in\mathcal{P}}T(P)=\bar{T}$ and $\min_{P\in\mathcal{P}}R(P)=\bar{R}$.*

Proof. We again applies induction to prove it. The induction assumption is that $p_n^{-1}(L)=\underline{t}_n^+$. Let $[p_n]_{n\in\mathcal{N}}$ be the trajectories in $P^{\mathrm{SH}}\left(\underline{a}^{\mathrm{f}},\bar{a}^{\mathrm{f}},\underline{a}^{\mathrm{b}},\bar{a}^{\mathrm{b}},\bar{v}\right)$. Based on the definition of the SH, we see $p_1^{-1}(L)=\underline{t}_1^+$ apparently holds. Then, we assume $p_n^{-1}(L)=\underline{t}_n^+$ holds for $n=k-1$, and we will investigate whether it holds for $n=k,\forall k\in\mathcal{N}\backslash\{1\}$. Since$L\geq\bar{v}^2/(2\bar{a})$, $\dot{p}_{k-1}(t)=\bar{v},\forall t\geq p_{k-1}^{-1}(L)$. We obtain $p_{k-1}^{\mathrm{S}-1}(L)=p_{k-1}^{-1}(L)+s/\bar{v}+\tau$. Then, the induction assumption indicates that $p_{k-1}^{\mathrm{S}-1}(L)=\underline{t}_{k-1}^+ + s/\bar{v}+\tau$. If FSP trajectory p_k^{f} is not blocked by p_{k-1}^{S}, as illustrated by Fig. 3(a), we obtain $p_k^{\mathrm{f}-1}(L)=\hat{t}_k^+(\bar{v},\bar{a})$. Otherwise, p_k^{f} will merge into p_{k-1}^{S}before reaching location L, and thus $p_k^{\mathrm{f}-1}(L)=p_{k-1}^{\mathrm{S}-1}(L)$. In either case, $p_k^{\mathrm{f}-1}(L)=\max\left\{\hat{t}_k^+(\bar{v},\bar{a}),\underline{t}_{n-1}^+ + s/\bar{v}+\tau\right\}$. Then, the BSP will just result in $p_k^{-1}(L)=G(p_k^{\mathrm{f}-1}(L))=\underline{t}_n^+$. This proves the induction assumption for $n=k$, and the proof completes. □

Although SH can directly achieve optimality on travel time and throughput with all parameters set at their extreme values, optimizing other objectives such as fuel/emission measures (5) and safety measure (7) are not as intuitive. Therefore, a general optimization framework is still needed to optimize all these performance measures.

3.2. Expedited objective evaluation

This study considers optimization objective (8) comprised of travel time (3), energy consumption (5) and safety (7). This system performance omits the throughput measure (4), because it contains redundant mobility information that travel time measures and it is easier to quantify the dollar value of travel time than throughput. For a given trajectory vector $P\in\mathcal{P}$

by using SH algorithm, it is easy to calculate a vehicle's travel time as the difference between vehicle's entry and exit times. However, fuel consumption (5) and safety (7) are complex nonlinear integrals. Although they can be evaluated by a numerical integral approach (Li et al., 2014), this approach needs to discretize a trajectory into a large number of cells. Therefore, it is computationally expensive and might not be suitable for real-time applications. Fortunately, one appealing feature of the SH solution is that the output vehicle trajectories are composed of a relatively small number of analytical quadratic segments. With this property, we propose the following expedited objective evaluation methods for each quadratic segment to reduce the computational burden. This is an added computational benefits of using the SH algorithm.

3.2.1. Fuel consumption

A closer observation of VT-Micro model (6) reveals that fuel consumption is determined by acceleration and speed values at each time. In the fuel consumption evaluation, we denote a generic quadratic segment by a three-element tuple (l^s, v, a), where l^s is the segment length, v is the start speed, and a is the constant acceleration rate. Note that all trajectory segments constructed by SH has constant accelerations and the change of state (e.g., from accelerating to the maximum speed to cruising at the maximum speed) will result in the end of one segment and start of another.

Therefore, the total fuel consumption along a trajectory segment (l^s, v, a) only depends on its start speed v, acceleration a, and time duration $\triangle t$, which is calculated using the value of l^s, v, and a for each quadratic segment. We call this*segment fuel consumption* and denote it by $E^s(l^s, v, a)$. Based on this observation, we propose a lookup table-based approximation method to estimate a quadratic segment's fuel consumption with a simple linear interpolation. We first construct a look-up table of fuel consumption with three dimensions, respectively for l^s, v, and a. Each dimension has a finite number of evenly distributed discrete values for the corresponding parameter within the corresponding range, i.e., $l^s \in [0, L]$, $v \in [0, \bar{v}]$, $a \in [\underline{a}, \bar{a}]$. Each entry of this table is segment fuel consumption value $E^s(l^s, v, a)$ pre-calculated with the numerical integral for the corresponding l^s, v and a values. Then, in each iteration of the online trajectory optimization, we traverse through all the quadratic segments in the SH output trajectory vector. For each quadratic segment (l^s, v, a), we find from the lookup table the closest two discrete values for l^s, v, and a, respectively, denoted by (l_0^s, l_1^s), (v_0, v_1), (a_0, a_1), respectively. Then, linear interpolation is applied to approximate $E^s(l^s, v, a)$ with pre-calculated lookup entries $E^S(l_0^s, v_0, a_0)$, $E^S(l_1^s, v_0, a_0)$, $E^S(l_0^s, v_1, a_0)$, $E^S(l_1^s, v_1, a_0)$, $E^S(l_0^s, v_0, a_1)$, $E^S(l_1^s, v_0, a_1)$, $E^S(l_0^s, v_1, a_1)$ and $E^S(l_1^s, v_1, a_1)$ in the following steps.

IP1: Calculate $l_d^s = (l^s - l_0^s)/(l_1^s - l_0^s) v_d := (v - v_0)/(v_1 - v_0)$, and $a_d := (a - a_0)/(a_1 - a_0)$, .

IP2: Interpolate along l^s at (v_0, a_0), (v_1, a_0), (v_0, a_1) and (v_1, a_1):

$$c_{00} := E^S(l_0^s, v_0, a_0) \cdot (1 - l_d^s) + E^S(l_1^s, v_0, a_0) \cdot l_d^s;$$

$$c_{10} = E^S(l_0^s, v_1, a_0) \cdot (1 - l_d^s) + E^S(l_1^s, v_1, a_0) \cdot l_d^s;$$

$$c_{01} = E^S(l_0^s, v_0, a_1) \cdot (1 - l_d^s) + E^S(l_1^s, v_0, a_1) \cdot l_d^s;$$

$$c_{11} = E^S(l_0^s, v_1, a_1) \cdot (1 - l_d^s) + E^S(l_1^s, v_1, a_1) \cdot l_d^s.$$

IP3: Then interpolate along v at a_0 and a_1:

$$c_0 = c_{00} \cdot (1 - v_d) + c_{10} \cdot v_d;$$

$$c_1 = c_{01} \cdot (1 - v_d) + c_{11} \cdot v_d.$$

IP4: Finally interpolate along a:

$$c = c_0 \cdot (1 - a_d) + c_1 \cdot a_d.$$

Then return c as the final estimated value for $E^s(l^s, v, a)$.

At the end, we add the fuel consumption values across all segments in the trajectory vector and divide the sum by N. This efficiently yields an good approximation of emission objective (5). Note that as we increase the density of the discrete values, the approximation error approaches to zero. Therefore, if we have enough offline computational capacity and pre-calculated $E^s(l^s, v, a)$ values for enough discrete values, this expedited fuel consumption evaluation approach is both efficient and accurate.

The lookup table approach is developed for accelerated fuel consumption evaluation during real-time applications and the table itself can always be generated offline. The case study results in Section 4 show that this approach is accurate for the VT-Micro model. Even if other more complicated (e.g., physics-based) fuel consumption models are used, this lookup table approach shall efficiently yield accurate MOE estimation. First, the lookup table needs to be estimated only once offline with much less computational resource constraints than real-time calculations, and thus the interpolation errors can be always reduced to an acceptable level with sufficient computation resources. Further, aggregating the measurements on a trajectory section rather than individually quantifying them at instantaneous time points helps minimize the overall error, because local errors likely cancel one another along the aggregation. While the approach is shown effective in the case studies in Section 4, it should be noted that the efficiency may be compromised when there are many look-up parameters. In this case, storing and extracting multi-dimensional tables may be not as efficient in real time. For specific on-line applications, further investigation is needed to evaluate the efficiency and the feasibility of this lookup table approach.

3.2.2. Safety

Safety objective (7) is comprised of the integrals of positive iTTC for all but the lead vehicles. Due to nonlinear operator $[\cdot]^+$, in general Eq. (7) can not be analytically solved. Fortunately, we find that this objective can be analytically solved for quadratic segments. Since SH trajectory vector, denoted by $P = [p_n]_{n\in\mathcal{N}}$, contains only analytical quadratic segments, we can expedite calculating safety objective (7) by analytically solving it for each quadratic segment. This section describes this expedited analytical method for the evaluation of safety objective (7).

For any vehicle $n \in \mathcal{N}$, since its trajectory p_n and the preceding trajectory p_{n-1} in SH solution P are both piecewise quadratic functions, we can find a finite number (denoted by K_n) of time points $t_{n1} := t_n^- < t_{n2} < \ldots < t_{nK_n} < t_{K_n+1} := p_{n-1}^{-1}(L)$. These time points are sorted in ascending order, and they cover the start and end of all trajectory segments of p_n and p_{n-1}. In the safety performance evaluation, we denote a quadratic segment by a five-element tuple (l, v, a, t', t''), where l is the segment start location, v is the start speed, a is the constant acceleration rate, t' is the start time and t'' is the end time. During each interval $[t_{nk}, t_{nk+1}]$, $\forall k = 1, 2\ldots, K_n$, we denote the quadratic segment of p_{n-1} by $\left(\hat{l}_{nk}, \hat{v}_{nk}, \hat{a}_{nk}, \hat{t}'_{nk}, \hat{t}''_{nk}\right)$ and the quadratic segment of p_n by $\left(l_{nk}, v_{nk}, a_{nk}, t'_{nk}, t''_{nk}\right)$. Then p_{n-1} and p_n can be formulated as follows, respectively:

$$p_{n-1}(t) = 0.5\hat{a}_{nk}(t - \hat{t}'_{nk})^2 + \hat{v}_{nk}(t - \hat{t}'_{nk}) + \hat{l}_{nk}, \forall t \in [t_{nk}, t_{nk+1}], k = 1, 2\ldots, K_n, n \in \mathcal{N}\backslash\{1\},$$

$$p_n(t) = 0.5a_{nk}(t - t'_{nk})^2 + v_{nk}(t - t'_{nk}) + l_{nk}, \forall t \in [t_{nk}, t_{nk+1}], k = 1, 2\ldots, K_n, n \in \mathcal{N}\backslash\{1\}.$$

With this, we obtain the difference between $p_{n-1}(t)$ and $p_n(t)$, which is also a quadratic function:

$$p_{n-1}(t) - p_n(t) = A_{nk}t^2 + B_{nk}t + C_{nk}, \dot{p}_n(t) - \dot{p}_{n-1}(t) = -2A_{nk}t - B_{nk}, \forall t \in [t_{nk}, t_{nk+1}], k = 1, 2\ldots, K_n, n \in \mathcal{N}\backslash\{1\}.$$

where $A_{nk} = 0.5(\hat{a}_{nk} - a_{nk})$, $B_{nk} = \hat{v}_{nk} - \hat{a}_{nk}\hat{t}'_{nk} - v_{nk} + a_{nk}t'_{nk}$, $C_{nk} = \hat{l}_{nk} - \hat{v}_{nk}\hat{t}'_{nk} + 0.5\hat{a}_{nk}\hat{t}'^{2}_{nk} - l_{nk} + v_{nk}t'_{nk} - 0.5a_{nk}t'^{2}_{nk}$. Then the iTTC in the integrand of safety objective (7) can be formulated as

$$\frac{\dot{p}_n(t) - \dot{p}_{n-1}(t)}{p_{n-1}(t) - p_n(t) - g_{n-1}} = \frac{-2A_{nk}t - B_{nk}}{A_{nk}t^2 + B_{nk}t + C_{nk} - g_{n-1}}, \forall t \in [t_{nk}, t_{nk+1}], k = 1, 2, \ldots, K_n, n \in \mathcal{N}\backslash\{1\}.$$

Due to the safety constraint, $p_{n-1}(t) - p_n(t) - g_{n-1}$ is always positive, and the sign of the this iTTC measure is determined by $\dot{p}_n(t) - \dot{p}_{n-1}(t)$. During an interval $[t_{nk}, t_{nk+1}]$, we only need to integrate the iTTC at time points when $\dot{p}_n(t) - \dot{p}_{n-1}(t)$ is positive in safety objective (7). Since $\dot{p}_n(t) - \dot{p}_{n-1}(t) = -2A_{nk}t - B_{nk}$ is linear, the set of such time points actually form a continuous time interval, i.e., $\left(\underline{t}_{nk}, \bar{t}_{nk}\right)$ where

$$\underline{t}_{nk} = \begin{cases} \min\{\max\{t_{nk}, -B_{nk}/2A_{nk}\}, t_{k+1}\}, & \text{if } A_{nk} < 0; \\ t_{k+1}, & \text{if } A_{nk} = 0 \text{ and } A_{nk} \geq 0; \\ t_k, & \text{otherwise,} \end{cases}$$

and

$$\bar{t}_{nk} = \begin{cases} \min\{\max\{t_k, -B_{nk}/2A_{nk}\}, t_{k+1}\}, & \text{if } A_{nk} > 0; \\ t_{k+1}, & \text{otherwise.} \end{cases}$$

With this, safety objective (7) for a SH solution can be simplified into the following analytical formulation

$$S(P) = \frac{1}{N}\sum_{n\in\mathcal{N}\backslash\{1\}}\sum_{k=1}^{K_n}\int_{\underline{t}_{nk}}^{\bar{t}_{nk}} \frac{-2A_{nk}t - B_{nk}}{A_{nk}t^2 + B_{nk}t + C_{nk} - g_{n-1}}dt = \frac{1}{N}\sum_{n\in\mathcal{N}\backslash\{1\}}\sum_{k=1}^{K_n}\ln\left(\frac{A_{nk}\underline{t}^2_{nk} + B_{nk}\underline{t}_{nk} + C_{nk} - g_{n-1}}{A_{nk}\bar{t}^2_{nk} + B_{nk}\bar{t}_{nk} + C_{nk} - g_{n-1}}\right),$$

which apparently can be very efficiently solved.

3.3. Optimization algorithms

The overall performance of a SH trajectory vector solution depends on values of the five parameters $(\bar{a}_1^{\mathrm{f}}, \underline{a}_1^{\mathrm{f}}, \bar{a}_1^{\mathrm{b}}, \underline{a}_1^{\mathrm{b}}, v_1)$, which need to be solved for with a certain optimization algorithm. As illustrated in Fig. 2, we aim to optimize a simplified optimization problem that can be formulated as

$$\max_{\bar{a}_1^{\mathrm{f}}, \underline{a}_1^{\mathrm{f}}, \bar{a}_1^{\mathrm{b}}, \underline{a}_1^{\mathrm{b}}, v_1} M\left(P^{\mathrm{SH}}(\bar{a}_1^{\mathrm{f}}, \underline{a}_1^{\mathrm{f}}, \bar{a}_1^{\mathrm{b}}, \underline{a}_1^{\mathrm{b}}, v_1)\right) \tag{13}$$

subject to

$$0 < \bar{a}^{\mathrm{f}}_1 \bar{a}^{\mathrm{b}}_1 \leq \bar{a},$$

$$\underline{a} \leq \underline{a}^{\mathrm{f}}_1 \underline{a}^{\mathrm{b}}_1 < 0,$$

$$0 \leq v_1 < \bar{v},$$

$$P^{\mathrm{SH}}(\bar{a}^{\mathrm{f}}_1, \underline{a}^{\mathrm{f}}_1, \bar{a}^{\mathrm{b}}_1, \underline{a}^{\mathrm{b}}_1, v_1) \in \mathcal{P}.$$

Optimization problem (13) is still nonlinear and cannot be explicitly formulated due to the SH algorithm. Nonetheless, since the problem only has 5 control parameters, a number of general heuristic algorithms (e.g., genetic algorithms, particle swarm) that only need to evaluate the objective for given variables without any knowledge of the problem structure can be potentially used to solve this problem. However, these general heuristics need to explore a relatively large solution space, and thus they may be too slow for real-time applications.

To ensure computational efficiency, Section (3.3.1) proposes a customized numerical-sub-gradient heuristic algorithm. This algorithm is applicable to the case when $M(P)$ in Eq. (8) is a single-valued function, i.e., a weighted summation of travel time, fuel consumption and safety measures. In fact, by varying the weights of the different objective components, this approach can also yield equivalent results as multi-objective optimization. Further, Section 3.3.2 also briefly describes an off-the-shelf multi-objective optimization algorithm that will be used in the numerical study to analyze the trade-offs between different objective components.

3.3.1. Customized single objective algorithm

This section aims to solve the trajectory optimization problem when overall performance objective $M(P)$ is a weighted summation of $T(P)$ in Eq. (3), $E(P)$ in Eq. (5) and $S(P)$ in Eq. (7):

$$M(P) = w_T T(P) + w_E E(P) + w_S S(P). \tag{14}$$

Weights w_T and w_E can be simply obtained as the dollar value for unit travel time and unit fuel consumption, respectively. However, it is difficult to assign an exact value to w_S, since safety is a somehow a qualitative concept, and it is difficult to assign a monetary equivalent to iTTC. However, if there exists a reasonable w_S value whose corresponding optimal system performance is not very sensitive to small variations of its value, it will be a good candidate for Eq. (14). We propose the following method to identify such a w_S value. First, we determine a set of candidate w_S values (e.g., by observing the experimenting results), denoted by W, and we select a number of representative problem scenarios (e.g., arriving volumes), denoted by I. At any scenario $i \in I$, we denote the optimal SH solution that solves problem (13) with $w_S = w$ by $P^*_i(w)$. Further we also denote the objective value (14) with $w_S = w$ as $M(P, w)$. Then we say that a weight w is robust at scenario $i \in I$ if its optimal trajectory vector solution $P^*_i(w)$ yields an objective value (i.e., $M\big(P^*_i(w), w'\big)$) not far from the optimal objective (i.e., $M\big(P^*_i(w'), w'\big)$) even if the actual w_S value, denoted by w', is different from w. Since, we don not know the actual w_S value, we will just select a $w^* \in W$ that is the most robust across all scenarios given w_S has an equal probability to be any value in W. This can be formulated into a problem of minimizing the average relative difference across all these scenarios, as follows.

$$w^* = \arg\min_{w \in W} \frac{1}{|I| \cdot |W|} \sum_{i \in I} \sum_{w' \in W} \frac{M\big(P^*_i(w), w'\big) - M\big(P^*_i(w'), w'\big)}{M\big(P^*_i(w), w'\big)}. \tag{15}$$

With a single-valued objective $M(P)$ formulated, we then propose a numerical-sub-gradient (NG) search algorithm to solve a near-optimum solution to optimization problem (13). Basically, this algorithm starts with a vector of control parameter feasible to SH. Then, it evaluates the numerical sub-gradient by slightly perturbing each dimension of the current control parameter set. Next, it searches along this sub-gradient direction trying to find a solution point that yields a significant reduction of objective (13). If such a point is found, the control parameter vector moves to this new point. Otherwise, the algorithm may be stuck at a local optimum, and the control parameter vector just moves to a nearby point, even though the objective value does not decrease. The purpose is to get out of any local trap. This numerical sub-gradient search process is repeated until certain terminal criteria are met. During this process, whenever the objective of a new point is evaluated (e.g., at the perturbed points for numerical sub-gradient calculation), the optimal solution is updated to this point if it yields an objective value smaller than all previous evaluations. Note that although the general idea of this algorithm is similar to classic numerical sub-gradient search algorithm for local optimization, it has certain capability of jumping out of a local optimum. It also has a good efficiency since every evaluation of the objective is used to update the optimal solution. The detailed steps of this customized NG algorithm is described in the following pseudo code.

NG1: Initialize integer counters η_1, η_2, η_3, η_4, linear search step size α, local perturbation δ, percentage tolerance ϵ%, and pick a feasible control parameter vector $\left[\bar{a}^{\mathrm{f}}_1, \underline{a}^{\mathrm{f}}_1, \bar{a}^{\mathrm{b}}_1, \underline{a}^{\mathrm{b}}_1, v_1\right]$ to start with. Set $P^* := P_1 := P^{\mathrm{SH}}(\bar{a}^{\mathrm{f}}_1, \underline{a}^{\mathrm{f}}_1, \bar{a}^{\mathrm{b}}_1, \underline{a}^{\mathrm{b}}_1, v_1)$ and evaluate $M^* := M_1 := M(P_1)$. Set iteration index $i = 1$. For the convenience of the presentation, we index the control parameters as $[c_{i1}, c_{i2}, c_{i3}, c_{i4}, c_{i5}] := \left[\bar{a}_i^{\mathrm{f}}, \underline{a}_i^{\mathrm{f}}, \bar{a}_i^{\mathrm{b}}, \underline{a}_i^{\mathrm{b}}, v_i\right]$, and we denote the control parameter vector by $\mathbf{c}_i := [c_{ik}]_{k=1,2,\ldots,5}$.

NG2: This step calculates the numerical sub-gradient at iteration i through the following sub-steps.

NG2-1: Set the initial perturbation $\delta' = \delta$ and dimension index $k = 1$.

NG2-2: We perturb thek^{th} dimension of $\mathbf{c}_i$ by δ' to generate $\mathbf{c}_{ik} := \left[c_{i1}, \ldots, c_{ik} + \delta', \ldots, c_{i5}\right]$. Run SH to check whether $P_{ik} := P^{\mathrm{SH}}(\mathbf{c}_{ik})$ is feasible. If not, set $\delta' = \delta'/2,\ k = 1$, and repeat this step. Otherwise, evaluate $M_{ik} := M(P_{ik})$. If $M_{ik} < M^*$, update the best solution $P^* = P_{ik}$ and $M^* = M_{ik}$. Set $k = k + 1$ and repeat this step if $k < 5$, or go to the next step if $k = 5$.

NG2-3: Calculate the numerical sub-gradient $\mathbf{g}_i = \left[(M_{ik} - M_i)/\delta'\right]_{k=1,\ldots,5}$.

NG3: This step identifies the best linear search step size at iteration i through the following sub-steps.

NG3-1: Set the initial search step $\alpha' = \alpha$.

NG3-2: Set $\mathbf{c}_{i+1} := \mathbf{c}_i + \alpha' \mathbf{g}_i$. Run SH to check the feasibility of $P_{i+1} := P^{\mathrm{SH}}(\mathbf{c}_{i+1})$. If P_{i+1} is not feasible, set $\alpha' = \alpha'/2$ and repeat this step. Otherwise, evaluate $M_{i+1} := M(P_{i+1})$. If this relative change is in $[0, \epsilon\%)$, go to Step NG-3-5. Otherwise, set temporary count $\eta' = 1$. If relative change $(M_i - M_{i+1})/M_i \geq \epsilon\%$, set $\eta' = 1$, and go to Step NG3-3. Otherwise, if this relative change is negative, go to NG3-4.

NG3-3: Update $\alpha' = \alpha' + \alpha$, and then set $\mathbf{c}_{i+1} := \mathbf{c}_i + \alpha' \mathbf{g}_i$. If $P^{\mathrm{SH}}(\mathbf{c}_{i+1})$ is feasible, set $P_{i+1} = P^{\mathrm{SH}}(\mathbf{c}_{i+1})$ and evaluate $M_{i+1} := M(P_{i+1})$. If relative change $(M_i - M_{i+1})/M_i \geq \epsilon\%$ and $\eta' < \eta_1$, set $\eta' = \eta' + 1$ and repeat this step. In all other cases, Step NG-3-5.

NG3-4: Update $\alpha' = \alpha'/2$, and then set $\mathbf{c}_{i+1} := \mathbf{c}_i + \alpha' \mathbf{g}_i$. If $P^{\mathrm{SH}}(\mathbf{c}_{i+1})$ is feasible, set $P_{i+1} = P^{\mathrm{SH}}(\mathbf{c}_{i+1})$ and evaluate $M_{i+1} := M(P_{i+1})$. If relative change $(M_i - M_{i+1})/M_i < 0$ and $\eta' < \eta_2$, set $\eta' = \eta' + 1$ and repeat this step. In all other cases, step out of the linear search and go to Step NG-3-5.

NG3-5: If $M_{i+1} < M^*$, update $P^* = P_{i+1}$ and $M^* = M_{i+1}$.

NG-4: Return trajectory vector $P^*$and the corresponding objective $M^*$when $i > \eta_3$ or M^* has not improved in the last η_4 consecutive iterations. Otherwise, set $i = i + 1$, go to Step NG-2.

3.3.2. Multi-objective optimization algorithm

To visualize how different components in objective $M(P)$ interact with each other, we will investigate the case when $M(P)$ contains multiple objectives as a vector of independent elements, i.e.,$[T(P), E(P), S(P)]$. In this case, we will apply the existing Non-dominated Sorting Genetic Algorithm-II (NSGA-II) to solve multi-objective optimization problem (13). This algorithm starts from an initial population of feasible solutions. Then it updates the initial population by (i) generating new solutions with crossover and mutation operations and (ii) eliminating inferior ones in the current population by a specific sorting mechanism. This can be repeated through iterations until satisfying solutions are found or certain terminal criteria are met. The crossover and mutation operations are standard steps in GA (Deb and Agrawal, 1994; Raghuwanshi and Kakde, 2004). The ranking mechanism is implemented at the following two levels. Based on the evaluation results, it first sorts these solutions based on their dominating relationships: We call solution $\mathbf{c}$ (or a control parameter vector) is dominated by another solution $\mathbf{c}'$ if $T(P^{\mathrm{SH}}(\mathbf{c})) \geq T(P^{\mathrm{SH}}(\mathbf{c}'))$, $E(P^{\mathrm{SH}}(\mathbf{c})) \geq E(P^{\mathrm{SH}}(\mathbf{c}'))$ and $S(P^{\mathrm{SH}}(\mathbf{c})) \geq S(P^{\mathrm{SH}}(\mathbf{c}'))$. In other words, if a solution $\mathbf{c}$ is dominated by another solution $\mathbf{c}'$, no objective value of the trajectory vector produced with $\mathbf{c}$ is better that of $\mathbf{c}'$, and we would rather prefer $\mathbf{c}'$ and discard $\mathbf{c}$. This first level of sorting basically ranks the solutions into a number of fronts, and each front contains all solutions that are not dominated by any other solution excluding those contained in the fronts of lower ranks. Next, NSGA-II sorts the solutions in the same front based on their their crowing distances, i.e., how far a solution is from its neighbors based on their objectives in the multi-dimensional hyper space. A solution with a larger crowding distance has a lower ranking. With these two levels of sorting, each solution in the current population has a unique ranking, and only a certain number of solutions with the lowest rankings will be kept in the next iteration. Readers can refer to Deb et al. (2002) for details of NSGA-II.

4. Numerical studies

This section illustrates the proposed optimization framework with the following synthetic numerical examples. The default values for the input parameters (defined in Section 2.1) are shown in Table 2. We may vary one or multiple parameter values to construct different problem instances. The numerical studies are conducted using Matlab on a PC with 2.9 GHz CPU, 8 GB RAM.

In Table 2, parameter $f^{\mathrm{s}} \in (0, C/G]$ is a traffic saturation rate that is used to generate the entry boundary conditions. Basically, we set $t_1^- = 0$ and generate the entry times as $t_n^- = t_{n-1}^- + (\tau + s/\bar{v})\left(1 + \xi_n\left(\frac{C}{f^{\mathrm{S}}G} - 1\right)\right), \forall n \in \mathcal{N}\backslash 1$ where ξ_n is an uniformly distributed random number over [0, 2], and ξ_n values are independent across different n values. Then we generate v_n^- randomly backwards from $n = N$ to $n = 1$. Each time, we assign a random value uniformly distributed over $[0, \bar{v}]$ to v_n^-. When $n < N$, if v_n^- leads to a violation to the safety constraint with acceleration $\bar{a}^{\mathrm{f}} = \bar{a}/2$ and deceleration $\underline{a}^{\mathrm{f}} = \underline{a}/10$, we will change it to the minimum value that does not violate it.

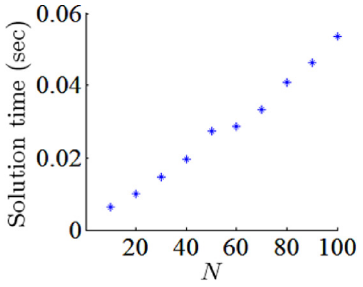


Fig. 4. Solution time v.s. N.

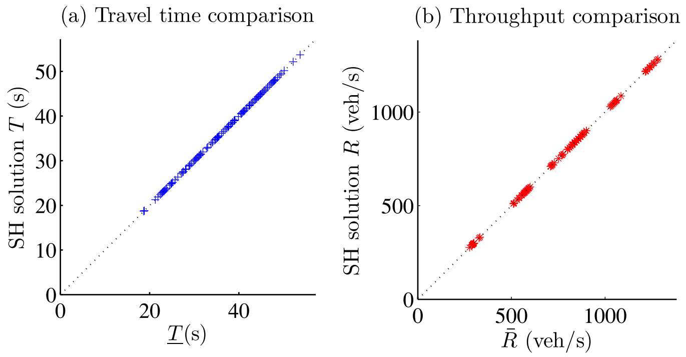


Fig. 5. Optimality comparison between SH solutions and theoretical bounds.

To evaluate the benefit of automated trajectory control, we construct a benchmark instance that represents the manually-driven traffic counterpart. We adapt Gipps' model (Gipps, 1981) as the manual-driving rule for every vehicle. Please refer to Part I of this study (Zhou et al., 2016) for this car-following model formulation for generating the benchmark trajectories.

4.1. Verification of theoretical properties

This subsection verifies the theoretical properties of SH discussed in Section 3.1.2. We first investigate its computational complexity. Fig. 4 plots the solution times for a series of numerical experiments as N varies from 10 to 100, where the SH control parameters are set to their extreme values, i.e., $\left(\underline{a}^{\mathrm{f}}, \bar{a}^{\mathrm{f}}, \underline{a}^{\mathrm{b}}, \bar{a}^{\mathrm{b}}, v\right) = (\underline{a}, \bar{a}, \underline{a}, \bar{a}, \bar{v})$. We see that the solution time increases almost only linearly as N increases. This increasing trend is even less than the worst-case quadratic time complexity bound predicted by Theorem 1, and it is consistent with the linear complexity discussed in Remark 2. Note that it only takes less than 0.06 second to solve the instance with $N = 100$. This indicates that the SH algorithm is efficient and suitable for real-time applications even if the number of vehicles is relatively large.

Next, we will examine the relevant optimality properties. We construct a number of instances that meet the condition of Theorem 2 by varying N from 50 to 100, f^s from 0.2 to 1.8, and L from 100 m to 1000 m. We run each instance with the control parameters again set to their extreme values, i.e., $\left(\underline{a}^{\mathrm{f}}, \bar{a}^{\mathrm{f}}, \underline{a}^{\mathrm{b}}, \bar{a}^{\mathrm{b}}, v\right) = (\underline{a}, \bar{a}, \underline{a}, \bar{a}, \bar{v})$ and obtain the output trajectory vector $P := P^{\mathrm{SH}}(\underline{a}, \bar{a}, \underline{a}, \bar{a}, \bar{v})$. Then, we measure travel time $T(P)$ with Eq. (3) and throughput with Eq. (4) for each instance, and we use Eqs. (3) and (12) to calculate the corresponding theoretical bounds. Fig. 5 compares the SH solution measures with their theoretical bounds for all instances. We see that for every instance, the SH solutions are exactly identical to their theoretical bounds. This verifies Theorem 3.1.2.

4.2. Fuel consumption evaluation accuracy

This subsection tests the accuracy of using the lookup table approach proposed in Section 3.2 to approximate the integral fuel consumption measure in objective (5). The lookup table created in this study has 2953 cells in total, with 19 levels of speed v (between 0 and 36 m/h levels with an interval of 2 m/s), 21 levels of length l^s (between 0 and 2000 m with an interval of 100 m), and 13 levels of acceleration a (between −10 and 2 m/s^2 with an interval of 1 m/s^2). We generate three sets of vehicle trajectories with $N = 50$ by varying parameters in Table 2. Specifically, these parameter sets are: 1) $\bar{a} = \bar{a}^{\mathrm{f}} = \bar{a}^{\mathrm{b}} = 2$ m/s^2, $\underline{a} = \underline{a}^{\mathrm{f}} = \underline{a}^{\mathrm{b}} = -10$ m/s^2 and $v = 30$ m/s; 2) $\bar{a} = \bar{a}^{\mathrm{f}} = \bar{a}^{\mathrm{b}} = 1$ m/s^2, $\underline{a} = \underline{a}^{\mathrm{f}} = \underline{a}^{\mathrm{b}} = -5$ m/s^2 and $v = 30$ m/s; and 3) $\bar{a} = \bar{a}^{\mathrm{f}} = \bar{a}^{\mathrm{b}} = 1$ m/s^2, $\underline{a} = \underline{a}^{\mathrm{f}} = \underline{a}^{\mathrm{b}} = -2$ m/s^2 and $v = 30$ m/s. The fuel consumption amount for each of the 150 trajectories in the three test sets is calculated with both the lookup table approach and the original integral

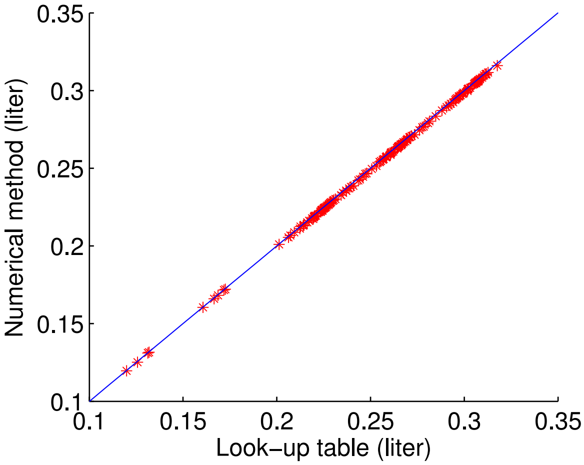


Fig. 6. Accuracy test of lookup table method for energy evaluation.

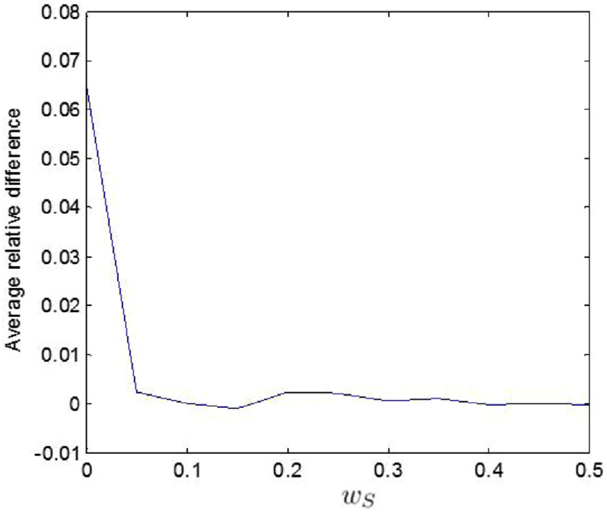
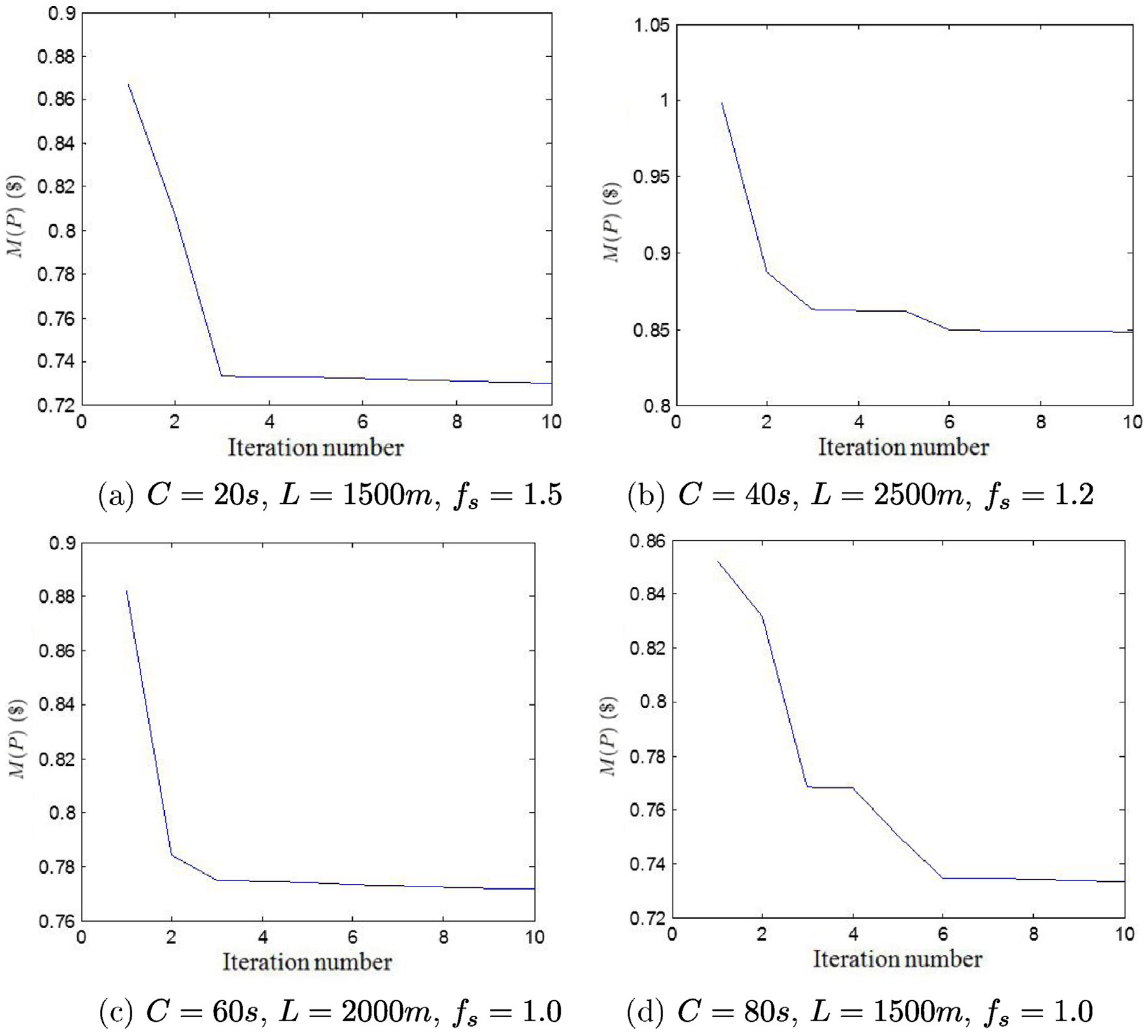


Fig. 7. Determination of safety weight w_S.

(with a numerical integral method). The comparison results are plotted in Fig. 6. It is shown that the fuel consumption amounts generated by both approaches are very close. We find their average error is 0.43%, and the maximum error is 0.98%. Obviously, increasing the parameter levels (or decreasing the corresponding intervals) will further reduce the approximation error. Considering that the errors of the state-of-the-art fuel consumption models are in general much larger than 0.98%, we deem that this approximation accuracy of is acceptable and sufficient. For efficiency, the fuel consumption results in the subsequent results are all calculated with the lookup table approach.

4.3. Single objective optimization

This section demonstrates the proposed single objective algorithm NG1-4. Weights w_T and w_E can be determined by the equivalent monetary values of travel time and fuel consumption, and we set $w_T = \$20/\text{h}$ and $w_E = \$1/\text{l}$. Also, Eq. (15) is used to find a robust safety weight w_S for the single objective optimization problem (13) under a set of problem scenarios I with 64 factorial combinations of varying parameters: four levels for congestion (saturation rate f_S=0.6, 0.9, 1.2, and 1.5), four levels for segment length L (1000, 1500, 2000, 2500 m), and four levels for signal cycle lengths C (20, 40, 60 and 80 s. We set $G = R = C/2$ in the following experiments). We also limit $w_S \in [0, 0.6]$ with a incremental interval of 0.05. This range limits safety term $w_S S(P)$ from being out of proportion to the other two terms $w_T T(P)$ and $w_E E(P)$ in objective (14). The results of average relative difference (15) across all w_S values are plotted in Fig. 7. We notice that when w_S< 0.1, the average relative differences are much larger than other cases. When $w_S \geq 0.1$, the average relative differences maintain at similar small levels, implying that any $w_S \in [0.1, 0.6]$ can be a robust safety weight for single objective optimization problem (13). Therefore we set $w_S = 0.1$ in this study.



(a) $C = 20s,\ L = 1500m,\ f_s = 1.5$ (b) $C = 40s,\ L = 2500m,\ f_s = 1.2$

(c) $C = 60s,\ L = 2000m,\ f_s = 1.0$ (d) $C = 80s,\ L = 1500m,\ f_s = 1.0$

Fig. 8. Algorithm efficiency with four sets of system parameters.

With all weights determined, we are ready to test optimization algorithm NG1-4. The acceleration parameters $\bar{a}^{\mathrm{f}}_1, \bar{a}^{\mathrm{b}}_1$ is bounded by $[0, \bar{a} = 2]$ m/s^2, deceleration $\underline{a}^{\mathrm{f}}_1, \underline{a}^{\mathrm{b}}_1$ by $[\underline{a} = -10, 0]$ m/s^2 and cruising speed v_1 by $[9, \bar{v} = 36]$ m/s ([20, 80] mph). To expedite the convergence and improve the final solution quality, for each instance, the NG algorithm starts at three selected initial solutions in parallel. These three initial solutions are 1) $\bar{a}^{\mathrm{f}} = \bar{a}^{\mathrm{b}} = 1$ m/s^2, $\underline{a}^{\mathrm{f}} = \underline{a}^{\mathrm{b}} = -2$ m/s^2, $v = 30$ m/s; 2) $\bar{a}^{\mathrm{f}} = \bar{a}^{\mathrm{b}} = 1$ m/s^2, $\underline{a}^{\mathrm{f}} = \underline{a}^{\mathrm{b}} = -5$ m/s^2, $v = 30$ m/s and 3) $\bar{a}^{\mathrm{f}} = \bar{a}^{\mathrm{b}} = 1.5$ m/s^2, $\underline{a}^{\mathrm{f}} = \underline{a}^{\mathrm{b}} = -7$ m/s^2, $v = 30$ m/s. The best solution among these three threads will be selected as the near-optimum final solution for this instance.

Fig. 8 shows the convergence performance of the NG algorithm on four examples with different f_s, L, and C values. In all four examples, we see that system objective $M(P)$ drops significantly within the first 6 to 8 iterations and then stays relatively stable afterwards. Based on this observation, the algorithm iteration number can be limited to a small value (e.g., 10) to suffice the requirement in the real-time applications while maintaining the solution quality. Table 3 summarizes the NG algorithm's solution efficiency and quality with the iteration limit of 10. We see that the solutions times of the NG algorithm for all eight test instances are below 13 s and the average is averagelyIt is easy to understand since fuel consumption rates are higher when vehicle speeds are higher on the basis of the VT-Micro model as shown in Eq. (6). less 10 s. Note that this time is one order of magnitude lower than the planning period for all 50 vehicles (around 100 s). This efficiency can be further improved if high-performance computers with multiple CPUs are used. This proves that the NG algorithm is fast enough for real-time applications. Table 3 also shows the NG algorithm significantly improves the objective value from the initial solutions, with an average of 14.16% improvement. We also compare the NG algorithm's performance with that of the traditional GA algorithm. GA takes over 10 min to solve a single instance with a population of 50 and a preset interation number of 300, and apparently it is not suitable for real-time applications. We see that at the objectives from

Table 3
Comparison of the optimization results with NG and GA. Notes: M^I is the best objective value from the three initial solutions; M^* and M^{GA} are the optimal objective values obtained with the NG algorithm and the GA algorithm, respectively; T^{NG} is the solution time from the NG algorithm.

C(sec)	L(m)	f^s	T^{NG} (sec)	M^I($)	M^*($)	M^{GA}($)	$\left(M^I - M^*\right)/M^I$	$\left(M^* - M^{GA}\right)/M^*$
60	1500	0.9	12.56	0.736	0.667	0.662	9.43%	0.66%
60	1500	1.5	10.74	0.772	0.662	0.689	14.32%	−4.14%
60	2500	0.9	5.48	1.080	0.862	0.944	20.21%	−9.63%
60	2500	1.5	8.79	1.217	0.954	1.024	21.56%	−7.25%
80	1500	0.9	9.40	0.857	0.748	0.705	12.63%	5.80%
80	1500	1.5	7.57	1.004	0.843	0.741	16.06%	12.14%
80	2500	0.9	7.15	0.990	0.844	0.906	14.71%	−7.33%
80	2500	1.5	3.82	0.959	0.917	0.863	4.37%	5.89%
Average							14.16%	−0.48%

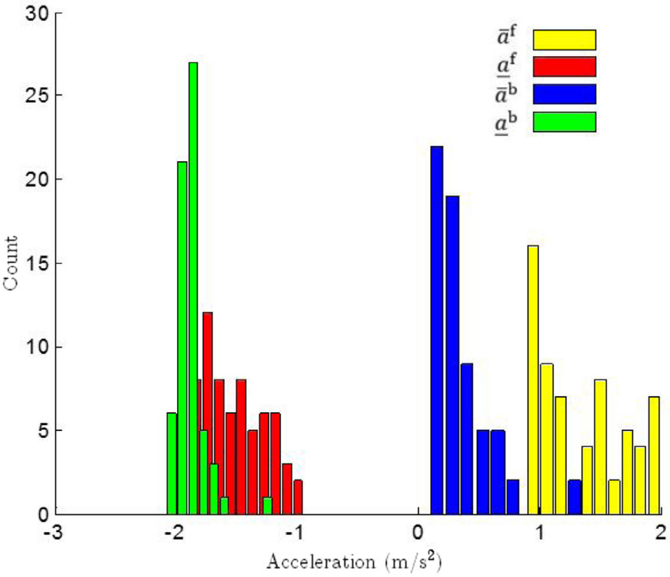


Fig. 9. Distribution of four acceleration values.

these two algorithms are not statistically different. This indicates that on average, the NG algorithm can obtain solutions as good as those from the traditional GA algorithm , and the NG solution optimality is relatively robust.

Fig. 9 illustrates the distribution of the optimal values of four acceleration variables $\bar{a}^{\mathrm{f}}, \underline{a}^{\mathrm{f}}, \bar{a}^{\mathrm{b}}, \underline{a}^{\mathrm{b}}$ across all 64 test instances. We observe that the optimal values for each acceleration variable are clustered within a relatively narrow range. The practical implication of this observation is that with less stringent requirements of absolute mathematical optimality or limited computational resources in engineering practices, the acceleration values may be directly selected from these ranges (e.g., based on traffic states and infrastructure features). This would obviate the need for iterative optimization and thus further improve real-time implementation efficiency. Nonetheless, a closer look does reveal visible dispersion of the optimal values for each acceleration within its range. For example, while the most frequent optimal $\bar{a}^{\mathrm{f}}$ value is around 1 m/s^2, there are still more than 20 instances in which the optimal $\bar{a}^{\mathrm{f}}$ values are between 1.5 and 2 m/s^2. In this regard, when computational resources are abundant, iterative optimization shall further improve the solution optimality and bring more savings.

It is also found that the optimal v value is quite consBy integrating human driver car-following and lane-changing behavior, the proposed algorithm can be also used to investigate mixed traffic with different CAV market penetration rates. istent across different scenarios, with a mean of 28 m/s and standard deviation of 2.65 m/s. This is probably because larger v values may cause larger speed oscillation, thereby more fuel consumption and larger iTTC. Smaller v values may cause relatively slow vehicles movements and increase the total travel time. In real world applications, the v values can be actually fixed to a certain value (e.g., 28 m/s) to further improve the algorithm efficiency without much affecting the optimality results.

Fig. 10 compares the theoretically lower-bound travel time values $\underline{T}$ (Eq. (1)) with the travel time values T in the optimized results across all It is easy to understand since fuel consumption rates are higher when vehicle speeds are higher on the basis of the VT-Micro model as shown in Eq. (6).64 instances. It shows that the optimal travel time T values, despite

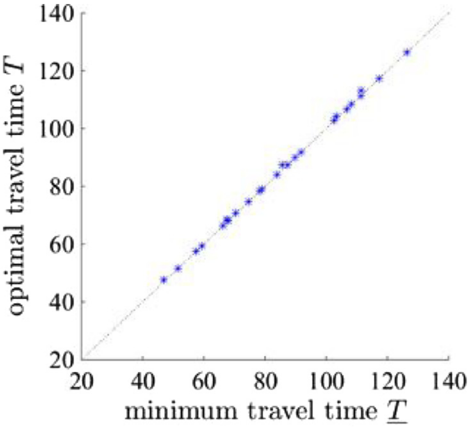


Fig. 10. Comparison between optimal travel time and mimimum travel time.

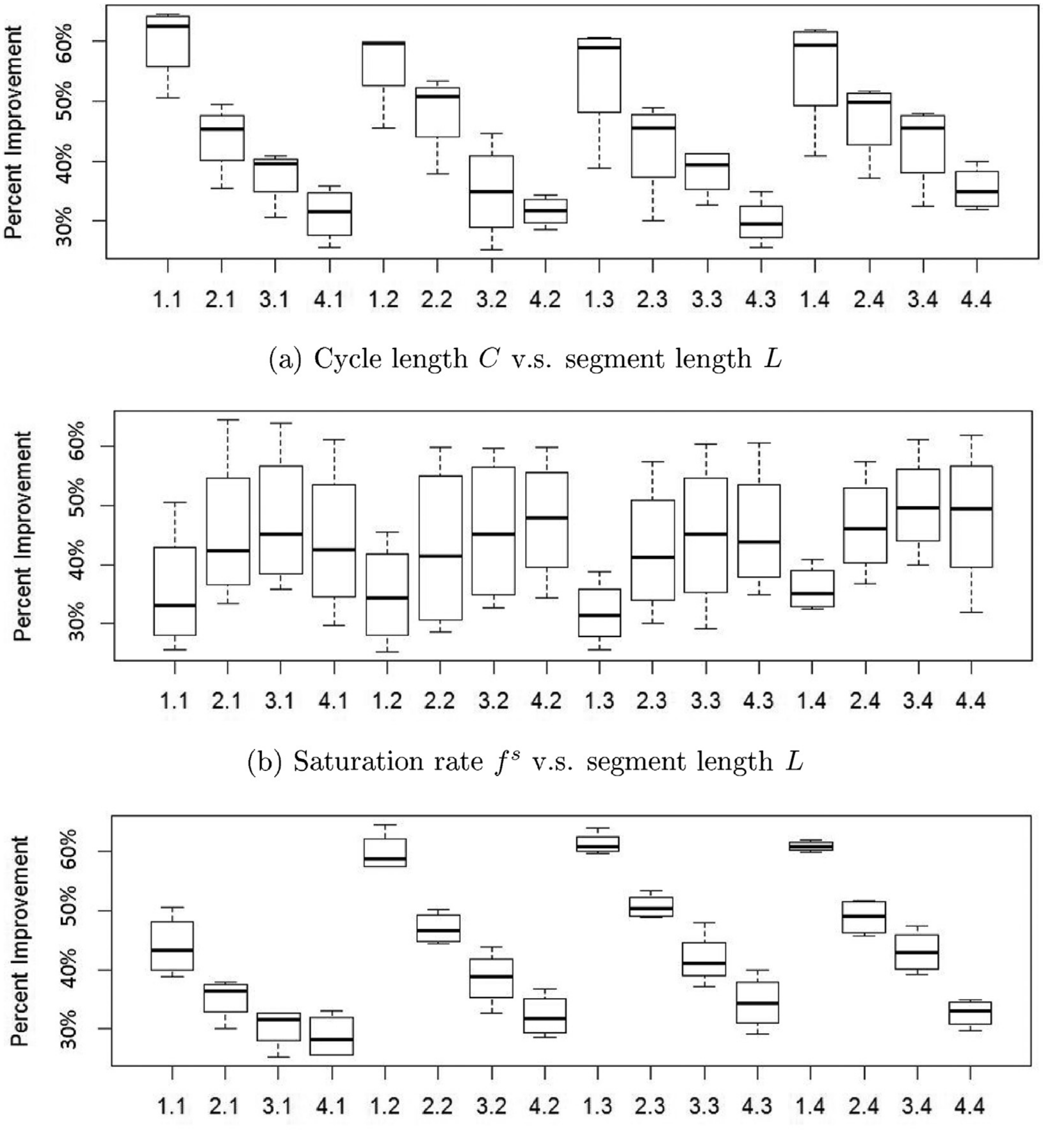
Table 4
Comparison of the CAV traffic performance with the benchmark manual traffic (by applying IDM) performance. Notes: $M^B$is the benchmark total system performance; $T^B$is benchmark system travel time; $E^B$is benchmark system fuel consumption; $S^B$is benchmark system safety performance; $M^*$is the optimal total system performance; $T^*$is optimal system travel time; $E^*$is optimal system fuel consumption; $S^*$is optimal system safety performance; ΔM is percent improvement of total system performance; ΔT is percent improvement of system travel time; ΔE is percent improvement of system fuel consumption; ΔS is percent improvement of system safety performance.

C(s)	L(m)	f^s	M^B($)	T^B(hr)	E^B(liter)	S^B	M^*($)	T^*(hr)	E^*(liter)	S^*	ΔM	ΔT	ΔE	ΔS
60	1500	0.9	0.991	0.021	0.315	2.488	0.667	0.017	0.226	0.932	32.68%	18.52%	28.28%	62.53%
60	1500	1.5	1.195	0.026	0.330	3.414	0.662	0.018	0.218	0.786	44.64%	30.26%	34.01%	76.97%
60	2500	0.9	1.533	0.035	0.458	3.676	0.862	0.026	0.264	0.859	43.79%	27.67%	42.31%	76.64%
60	2500	1.5	1.812	0.042	0.467	4.968	0.954	0.030	0.261	0.871	47.34%	28.57%	44.09%	82.47%
80	1500	0.9	1.049	0.022	0.325	2.892	0.748	0.019	0.245	1.243	28.66%	12.85%	24.57%	57.01%
80	1500	1.5	1.284	0.028	0.341	3.848	0.843	0.023	0.218	1.559	34.33%	15.88%	36.13%	59.47%
80	2500	0.9	1.334	0.033	0.423	2.519	0.844	0.026	0.244	0.709	36.74%	19.69%	45.53%	71.86%
80	2500	1.5	1.347	0.035	0.402	2.358	0.917	0.026	0.289	1.122	31.93%	27.28%	42.38%	52.44%
Average											37.51%	22.59%	28.09%	67.43%

conflicting with the other two objectives (fuel consumption E and safety S), are still very close to their corresponding bounds $\underline{T}$. This indicates that the trajectories constructed by the SH-based optimization framework always yield near-optimum travel time objective values (and thus near-optimum throughput values, too). This implies that we may simply fix each vehicle's exit time according to its minimum travel time $\underline{T}$, and focus trajectory optimization on only fuel consumption and safety objectives.

Table 4 shows the improvements of the optimal NG solutions over the benchmark manual driving. We see that the total system performance can improve by 37.51% on average through CAV control, with travel time improved by 22.59%, fuel consumption by 28.09%, and safety by 67.43%. Note that these results only factor in the effect of trajectory smoothing. If further enhancements from other advantageous CAV features (e.g., reduced headway) are considered, the benefits can be much more than those presented in this table. This demonstrates the CAV technologies, if properly utilized, can significantly improve traffic performance in all measures.

Fig. 11 presents the statistical results (i.e., the box plots) on the sensitivity of the performance improvement (CAV traffic v.s. manual traffic) to L, C, and f^s values with the optimization results from all 64 scenarios I. The x-axis indicates the combination of two different system parameters among L, C, and f^s. For example, the "3.1" in Fig. 11(a) means the scenario of segment length level "3" ($L = 1500$ m) and cycle length level "1" ($C = 20$ s). Figs. 11(a) and (c) show that the performance improvement increases significantly as the cycle length goes down. This is probably because smaller cycle lengths may result in more stop-and-go patterns at the intersection in the manual traffic, and the optimized trajectories have more room to improve by smoothing vehicle trajectories and mitigating the stop-and-go patterns. Based on Figs. 11(b) and (c), we find that the percent improvements at the medium saturation levels (f_s= 0.9 or 1.2) are larger than those at the low saturation level ($f_s = 0.6$) and high saturation level ($f_s = 1.5$). This is probably because when the saturation rate is low, there are potentially less stop-and-go traffic and thus the improvement is smaller; and when the saturation rate is high, there are less room for CAV to manipulate the trajectories and make further improvement. Fig. 11(a) also shows that there is no clear trend of the effect of segment length L on the percent improvement. This means the proposed algorithm can be potentially applied to a large range of roadway segments with similar effects. Additionally, we notice that, under different segment lengths L (Fig. 11(a)) and saturation rates f^s (Fig. 11(a)), there are similar decreasing patterns of system performance as cycle length L increases. This implies no obvious interaction effects between C and either of L or f^s.



(a) Cycle length C v.s. segment length L

(b) Saturation rate f^s v.s. segment length L

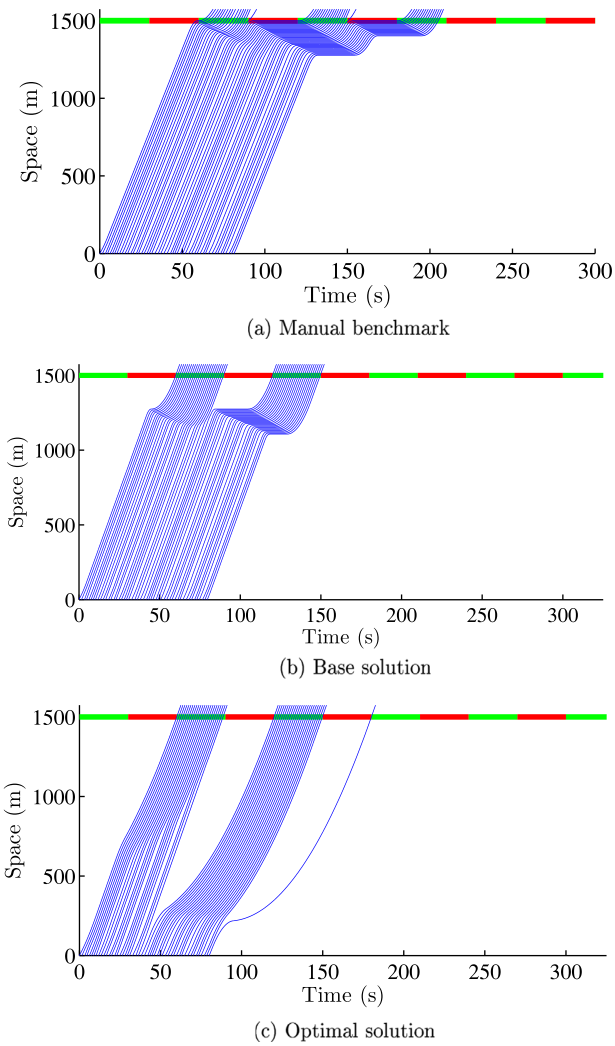
(c) Cycle length C v.s. saturation rate f^s

Fig. 11. Sensitivity analysis of optimization results. Notes: four levels of cycle length C : “1” - 20 sec, “2” - 40 sec, “3” - 60 sec, “4” - 80 sec; four levels of segment length L: “1” - 1000 m, “2” - 1500 m, “3” - 2000 m, “4” - 2500 m; and four levels of saturation rate f^s: “1” - 0.6, “2” - 0.9, “3” - 1.2, “4” - 1.5.

Finally, to give an impression of the trajectory patterns before and after the optimization, Fig. 12 plots the time-space vehicle trajectories in the manual benchmark, the CAV base solution (obtained by SH algorithm using extreme control parameters (2, −10, 2, −10. 36)), and the CAV optimal solution from the NG algorithm (with control parameters (0.98, −1.51, 0.33, −1.88, 29.1)) for the instance with $C = 60$ s, $L = 1500$ m, and $f_s = 1.5$. We see that both the base solution and the optimal solution have much better throughput rates compared with the manual benchmark due to the high intersection passing speed. The optimal solution can further smooth the vehicle trajectories and eliminate full stops to gain benefits from fuel saving and safety risk reduction. This qualitatively explains two key benefits from CAV control that we investigate: retaining traffic throughput and mitigating stop-and-go traffic.

4.4. Multi-objective optimization

Fig. 13 plots the 3-D Pareto front of the multi-objective optimization problem with default parameters in Table 2. This solution is obtained by running the NSGA-II algorithm with 500 populations with 100 generations. With an irregular Pareto front surface as shown in Fig. 13(a), we further project the solutions to all two-dimensional surfaces to derive more insights.



(a) Manual benchmark

(b) Base solution

(c) Optimal solution

Fig. 12. Comparisons of trajectories before and after optimization.

A few observations are obtained for this case study. In Fig. 13(b), the travel time and the fuel consumption are generally negatively correlated to each other. This is because a slightly longer travel time gives more flexibility for smoothing trajectories and reducing stop-and-go patterns as opposed to the minimum travel time that is however associated with extreme accelerations (as Theorem 2 indicates). The same trend is observed between the travel time and the safety performance, as shown in Fig. 13(c). This is intuitive since slower traffic can be usually safer than faster traffic. Also, based on Eq. (7), smaller speeds are more likely to yield smaller speed differences between two consecutive vehicles and thus smaller iTTC values. Fig. 13(d) implies the fuel consumption and the safety performance are positively correlated, and this is also intuitive since both of them are dependent on instant speeds and accelerations. As a conclusion, the travel time is a conflicting objective against both fuel consumption and safety performance. This observation further emphasizes the need for leveraging the trade-off between three objectives. This trade-off can be also tuned by varying w_T, w_E and w_S values in the single objective optimization discussed in Section 3.3.1.

5. Conclusion

Advanced connected and automated vehicle (CAV) technologies offer new opportunities for highway traffic smoothing by optimizing vehicle trajectories. Although controlling an individual or isolated object trajectory is a commonly-encountered problem in certain fields, optimizing trajectories of a stream of highway vehicles that interact with each other has been seldom studied in transportation engineering. By adapting a parsimonious shooting heuristic (SH) proposed in Part I of this study (Zhou et al., 2016) as a building block, this paper proposes numerical sub-gradient-based optimization framework with SH as a subroutine (NG-SH) that can simultaneously optimize a range of performance measures for a platoon of vehicles

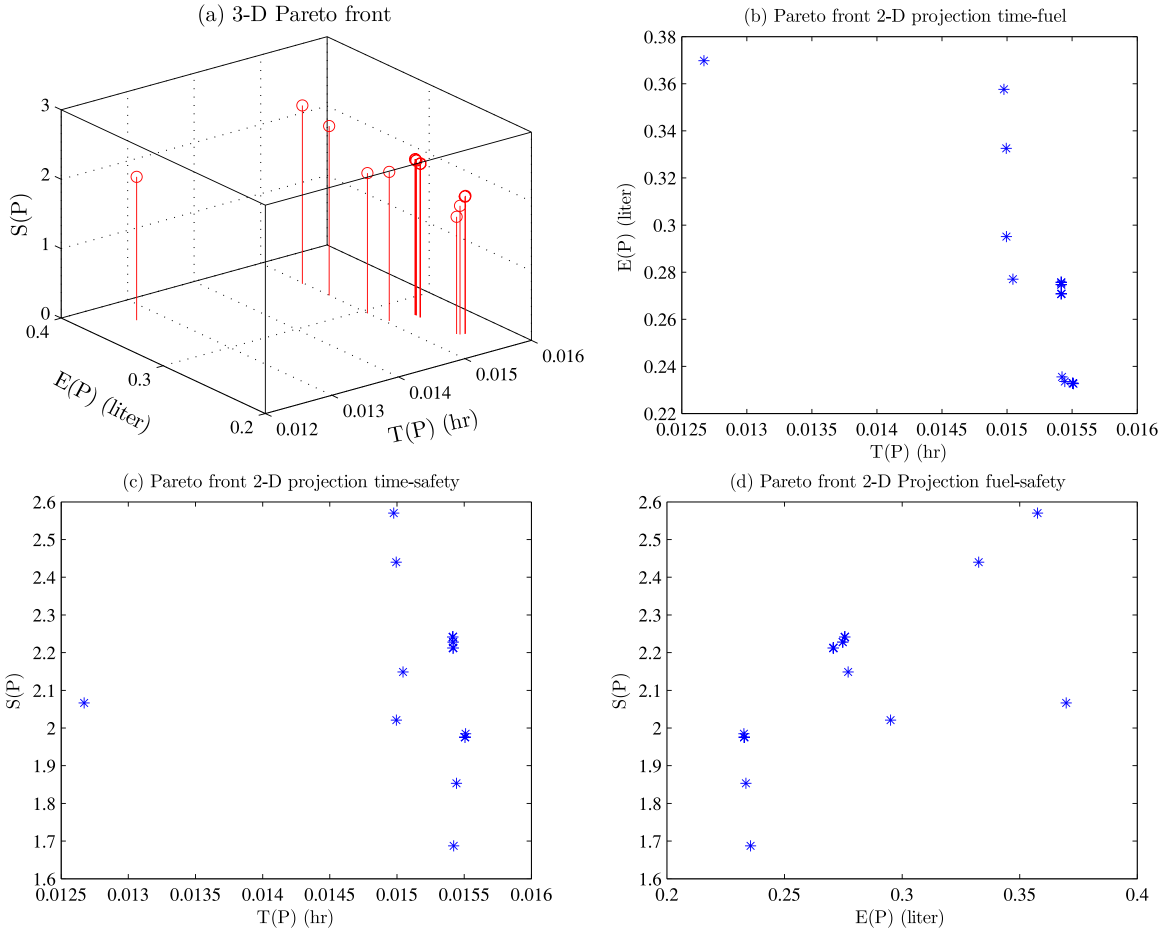


Fig. 13. 3-D Pareto fronts.

on a signalized highway section. It is found that SH possesses appealing theoretical properties on computation complexity and solution optimality. This enables the NG optimization to efficiently yield near-optimum solutions. Numerical examples are conducted to illustrate these computational and theoretical findings. They showed that, for experimental scenarios investigated in this study, vehicle trajectories produced from the proposed optimization method significantly outperforms the benchmark case (with all human drivers) at all measures, only by trajectory smoothing. With the default parameters, the optimized CAV trajectories outperform the counterpart manual trajectories by 43.8% on average across all measures, including travel time reduction by 36.7%, fuel consumption saving by 39.7% and safety risk mitigation by 67.1%. This optimization approach can be easily adapted to a multi-objective optimization problem, with which we analyzed the trade-off between different objectives. We find the travel time objective conflicts with the fuel consumption and the safety performance, while the fuel consumption and the safety objectives are consistent. Overall, this optimization framework enables the original infinite-dimensional trajectory optimization with intimidating complexity to be solved very efficiently, and thus is suitable for real-time applications when relevant technologies are in place in the near future.

This study reveals a great potential of transformative trajectory optimization in transportation engineering applications. It lays a solid foundation for devising holistic cooperative control strategies on a general transportation network with emerging technologies. The study can be further extended in multiple areas, such as calibration with field experiments, more complex geometries and heterogeneous vehicles. In particular, an interesting extension of this research is to consider mixed traffic that includes both CAV and human-driven vehicles. For instance, in mixed traffic, we can use SH to control the trajectories for the subset of compliant CAV vehicles and then we estimate the trajectories of the remaining human-driven vehicles with classic car following laws such as Newell's model. In this case, the human-driven vehicle trajectories are affected or indirectly controlled by their nearby CAV vehicles. It is an interesting research extension to examine the potential of improving mixed traffic performance by controlling the subset of CAV vehicles. Also, these single-lane single-segment results can be used as building blocks to investigate more complex geometries, e.g., multi-segment corridors, intersections, multi-lane highways, and even a complete network. By integrating human driver car-following and lane-changing behavior, the proposed algorithm can be also used to investigate mixed traffic with different CAV market penetration rates. The current optimization framework is essentially based on a constructive heuristic algorithm and may not yield exact optimal solutions. It is worth exploring exact algorithms that can solve exact optima or provide optimality bounds. Further, when detailed trajectory data along signalized segments are available, it is desirable to investigate the benchmark manual driving case and make relevant comparisons with calibrated car-following models. Last, in real-time applications, separate control algorithms, e.g., feedback control, need to be developed to let the vehicle follow designed trajectories. The efficiency of such control algorithms will be critical to reducing the gap between actual benefits of vehicle control and the upper bound benefits from designed trajectories.

Acknowledgments

This research is supported in part by the U.S. National Science Foundation through Grants CMMI CAREER #1558887, CMMI #1558889 and CMMI #1541130, and by the U.S. Department of Transportation Federal Highway Administration through Grant DTFH61-12-D-00020. The work presented in this paper remains the sole responsibility of the authors.

References

Ahn, K., 1998. Microscopic Fuel Consumption and Emission Modeling. Virginia Polytechnic Institute and State University, the US Master's thesis.
Ahn, K., Rakha, H., Park, S., 2013. Ecodrive application: Algorithmic development and preliminary testing. Transp. Res. Record (2341) 1–11.
Ahn, K., R. H. T. A., Van Aerde, M., 2002. Estimating vehicle fuel consumption and emissions based on instantaneous speed and acceleration levels. J. Transp. Eng. 128 (2), 182–190.
Balas, V.E., B. M. M., 2006. Driver assisting by inverse time to collision. World Automation Congress (WAC). wAC.
Barth, M., An, F., Younglove, T., Levine, C., Scora, G., Ross, M., Wenzel, T., 2000. Development of a comprehensive modal emissions model. National Cooperative Highway Res. Prog. Transp. Res. Board National Acad..
Beard, G.F., Griffin, M.J., 2013. Discomfort during lateral acceleration: Influence of seat cushion and backrest. Appl. Ergonomics 44 (4), 588–594.
Brinckerhoff, P., 2010. Synthesis of active traffic management experiences in europe and the united states. Federal Highway Admin..
Cassidy, M., Bertini, R., 1999. Some traffic features at freeway bottlenecks. Transp. Res. Part B 33 (1), 25–42.
Deb, K., Amrit Pratap, S.A., Meyarivan, T., 2002. A fast elitist multi-objective genetic algorithm: Nsga-ii.. IEEE Trans. Evol. Comput. 6 (2), 182–197.
Dresner, K., Stone, P., 2008. A multiagent approach to autonomous intersection management. J. Artif. Intell. Res. 31, 591–656.
Gipps, P.G., 1981. A behavioural car-following model for computer simulation. Transp. Res. Part B 15 (2), 105–111.
Hausberger, S., Rodler, J., Sturm, P., Rexeis, M., 2003. Emission factors for heavy-duty vehicles and validation by tunnel measurements. Atmos. Env. 37 (37), 5237–5245.
Hoffmann, E.R., Mortimer, R.G., 1994. Drivers' estimates of time to collision. Accident Anal. Prevention 26 (4), 511–520.
Kamalanathsharma, R.K., Rakha, H., et al., 2013. Multi-stage dynamic programming algorithm for eco-speed control at traffic signalized intersections. In: Intelligent Transportation Systems-(ITSC), 2013 16th International IEEE Conference on. IEEE, pp. 2094–2099.
Koupal, J., Michaels, H., Cumberworth, M., Bailey, C., Brzezinski, D., 2002. Epa's plan for moves: a comprehensive mobile source emissions model. In: Proceedings of the 12th CRC On-Road Vehicle Emissions Workshop, San Diego, CA.
Lee, J., Park, B., 2012. Development and evaluation of a cooperative vehicle intersection control algorithm under the connected vehicles environment. Intell. Transp. Syst. IEEE Trans. 13 (1), 81–90.
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 70, 319–339.
Lu, X.-Y., Shladover, S., 2014. Review of variable speed limits and advisories: Theory, algorithms, and practice. Transp. Res. Record (2423) 15–23.
Ma, J., Xiaopeng, L., Steve Shladover, X.-Y.L., Dailey, D., 2016. Freeway speed harmonization. IEEE Trans. Intell. Vehic. 1 (1), 1–12.
Park, B., Chen, Y., Hourdos, J., 2011. Opportunities for preventing rear-end crashes: findings from the analysis of actual freeway crash data. J. Transp. Safe. Security 3 (2), 95–107.
Robinson, M.D., 2000. Examples of variable speed limit applications. EasyWay Annual Forum, Lisbon, Portugal.
Sanchez, M., Cano, J.-C., Kim, D., 2006. Predicting traffic lights to improve urban traffic fuel consumption. In: 2006 6th International Conference on ITS Telecommunications.
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 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 18 (5), 421–429.
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 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 40, 290–311.
Wu, X., He, X., Yu, G., Harmandayan, A., Wang, Y., 2015. Energy-optimal speed control for electric vehicles on signalized arterials. Intell. Transp. Syst. IEEE Trans. PP (99), 1–11. doi:10.1109/TITS.2015.2422778.
Zhou, F., Li, X., Ma, J., 2016. Parsimonious shooting heuristic for trajectory control of connected automated traffic part I: Theoretical analysis with generalized time geography. Transp. Res. Part B.