Green Light Optimal Velocity Planning for Eco-driving: Computational Complexity and a Heuristic Algorithm*

Yang Zheng, Biao Xu, Shengbo Eben Li, Jianqiang Wang, and Keqiang Li

Abstract

In addition to traffic light signal, vehicles' motions can significantly affect the traffic throughput and fuel economy in a signalized road network. This paper considers a green light optimal velocity planning problem for eco-driving, where upcoming traffic signal information is utilized to find an optimal speed profile that avoids red lights and minimizes trip time and/or fuel consumption. We prove that this problem is NP-complete for the case where there are only binary velocity choices in each segment. The proof is based on a reduction from a known NP-complete problem, i.e., partition problem. It means that there are no polynomial-time algorithms to find a globally optimal solution for the general velocity planning problem. Consequently, we propose a genetic algorithm to obtain a sub-optimal solution, and we provide a detailed discussion on gene coding, population initialization, select operator and genetic operator. Numerical experiments demonstrate the effectiveness of our method, and also confirm that more improvements on fuel efficiency can be realized by taking multiple intersections into account simultaneously.

1 Introduction

During the past two decades, the increasing traffic demand has resulted in a heavy burden on the existing transportation systems, which sometimes leads to heavy traffic jams in some major cities [2]. For instance, the traffic congestion in US urban areas caused an estimated 3.1 billion gallons of extra fuel consumption and 6.9 billion additional hours on the road, resulting in a total cost of $160 billion in 2014 [3].

It is known that a significant amount of fuel consumption is spent by vehicles slowing down, idling behind, and accelerating away from signalized intersections [4, 5]. Reducing idling time and even avoiding red signals have the potential to greatly improve vehicles' fuel economy and reduce pollutant emissions [6]. Many efforts have been focused on designing advanced traffic signal control methods to optimize the signal timing, which help us reduce waiting time and eliminate unnecessary stops [7]. On the other hand, vehicles are major ingredients of the road transportation, the motions of which directly affect the traffic efficiency and fuel consumption during the signalized intersections. Some recent research has been exploring the concept of *eco-driving*, which provides real-time driving advices to further improve fuel economy [8, 9]. One of the important applications is to avoid red traffic signals by providing suitable velocity advisory, known as the green light optimal velocity planning problem [6, 10, 11, 12, 13, 14].

In principle, the green light optimal velocity planning is to utilize the upcoming traffic signal information to generate an optimal speed profile that avoids red lights and minimizes certain costs (*e.g.*, trip time and/or fuel consumption). This planning can be implemented as a smart-phone application that suggests suitable velocity to the driver [15], or it could be used as a reference velocity for the adaptive cruise control system [6]. This technique has recently received considerable attention due to its high potential to benefit the fuel efficiency [16]. For instance, Mandava *et al.* introduced a velocity planning algorithm for a single intersection, aiming to minimize the acceleration/decelereation rates [11]. A multi-segment speed planning was introduced for an artery with multiple signalized intersections in [13, 17]. Asadi and Vahidi developed a predictive cruise control in a traffic network with signalized intersections, where traffic light information was used to reduce idling time at stop lights [6]. A dynamic programming (DP) approach was proposed to optimize a vehicle's

*This is an extended technical report of our conference paper [1] in the 14th Intelligent Transportation Systems Asia Pacific Forum, Nanjing, China, 2015. The authors are with Department of Automotive Engineering, Tsinghua University, Beijing 100084, China.

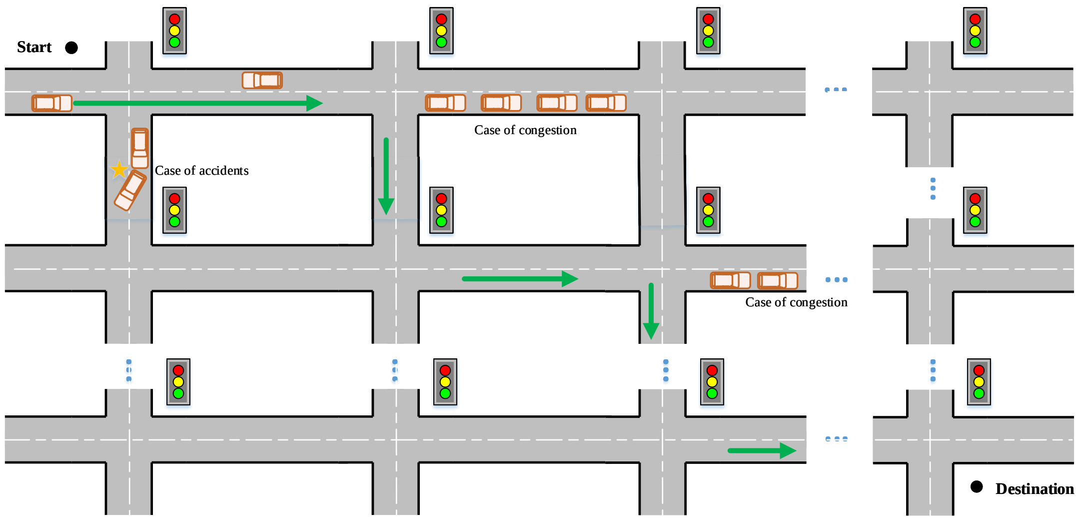


Figure 1: A trip route from the start point to the destination, indicated by the green arrows based on the traffic network information [23, 24].

trajectory that minimizes the fuel consumption level based on signal phasing and timing data [18]. A pruning and graph discretization-based approach was introduced to simplify the constrained optimization of a velocity planning problem in [12]. More recently, several realistic effects have been considered in the problem of green light optimal velocity planning, such as queue effects at intersections [19], probabilistic prediction of signal timing [10], driver's behavior adaptability [14] and impact on mixed traffic [20].

Due to the non-convexity introduced by the traffic signal timing, the majority of existing studies have focused on designing algorithms to find approximate optimal speed profiles using rule-based [6], heuristic-based [13] or DP-based [18] strategies. To the authors' best knowledge, there is no research that formally addresses the computational complexity of the green light optimal velocity planning problem. Also, due to the constraint on computation resource, some work only focuses on single intersection considering the nearest traffic light *sequentially*, *e.g.*, [6, 19, 14, 11, 20, 21]. However, as suggested in [13, 22], more improvement on fuel efficiency relies on the strategies taking multiple intersections into account *simultaneously* in a traffic network. In such cases, the computation of optimal velocity becomes quite challenging due to the disjointed feasible sets introduced by multiple available green phases at each intersection. The computation is also nontrivial if one tries to search the optimal velocity profile directly. For instance, if the maximum speed is 80 km/h and minimum speed is 30 km/h, and the speed scale for suggestion is 1 km/h, then the number of possible solutions is 40^{10} for a problem with ten intersections, where the solution space is too big to exhaustively search. Instead, approximated optimal solutions were sought based on heuristics and/or DP techniques; see *e.g.*, [13, 22, 18, 12] for details.

In this work, we consider the problem of green light optimal velocity planning, and formally analyze its computational complexity. For the first time, we show that this problem is NP-complete for the case where there are only binary velocity choices in each segment between two intersections. The proof is based on a reduction from a known NP-complete problem, *i.e.*, partition problem. Therefore, there are no polynomial-time algorithms that are able to solve the general optimal velocity planning problem exactly unless P=NP. This conclusion provides a reasonable explanation for the fact that most previous work focused on heuristics and/or DP techniques instead of seeking a global solution. In this paper, to numerically obtain a sub-optimal solution, we introduce a heuristic algorithm, called genetic algorithm [25], with a detailed discussion of gene coding, population initialization, select operator and genetic operator. In contrast to the strategies considering the nearest traffic light one-by-one [6, 19, 14, 11, 20, 21], our algorithm directly takes multiple intersections into account. Numerical experiments are conducted to demonstrate effectiveness of our solution, which also confirm that more improvements on fuel efficiency can be realized when considering multiple intersections *simultaneously*.

The rest of this paper is organized as follows. Section 2 presents the problem statement of green light optimal optimal velocity planning. The computational complexity of a special case is discussed in Section 3, and we propose a genetic algorithm to solve the problem in Section 4. This is followed by numerical experiments in Section 5. Section 6 concludes the paper.

2 Problem Statement: Green Light Optimal Velocity Planning

In this section, we first present a general problem statement of green light optimal velocity planning, and then introduce some typical mathematical models used in the literature.

2.1 Green Light Optimal Velocity Planning Problem

Given a trip route from a start point to a final point in a traffic network (see Fig. 1 for example), our objective is to find an velocity profile that eliminates idling at red lights and minimizes certain costs (trip time and/or fuel consumption). This is referred to as the *green light optimal velocity planning* problem in our paper. Note that the route can be determined based on the traffic network condition, *e.g.*, using an in-vehicle route guidance system [23, 24].

To obtain a concise (yet accurate enough) mathematical model, the trip route is simplified as a straight road with a series of signalized intersections (see Fig. 2), which means we ignore the effects of possible left- or right-turn during the intersections. Also, we assume the vehicle has full knowledge of the traffic light timings obtained by infrastructure-to-vehicle (V2I) communication, and our analysis is carried out for a route with vehicles in a free traffic flow. As shown in Fig. 2, the route is assumed to have N intersections, with parameters defined as follows:

- A list of N intersections, denoted as $\mathbb{N} = \{1, 2, \ldots, N\}$;
- The distance between consecutive intersections, represented as $d_i, i = 2, \ldots, N$; and d_1 denotes the distance to the first upcoming intersection;
- The traffic light timing of each intersection, described by triples $\{T_i, G_i, g_{i1}\}$, where T_i is the traffic signal cycle length, G_i is the green phase length and g_{i1} is the start time of the first green phase in i-th traffic light;
- The initial speed at the beginning time, v_0.

For simplicity, the green light phase of the i-th traffic light is defined as

$$\bigcup_{j=1}^{\infty} [g_{ij}, r_{ij}) \, ,$$

where g_{ij} is the start of j-th green phase and r_{ij} is the start of j-th red phase. Note that the yellow phase is lumped into the red phase, *i.e.*,

$$\begin{cases} g_{ij} = g_{i1} + j \times T_i, \\ r_{ij} = g_{i1} + j \times T_i + G_i. \end{cases} \tag{1}$$

It is assumed that traffic timing of each intersection is fixed and the vehicle can obtain the traffic light timing of each intersection $\{T_i, G_i, g_{i1}\}$ through V2I communication, as used in [6, 10, 11, 12, 13].

From a mathematical viewpoint, the green light optimal velocity planning can be cast as a constrained optimization problem: trip time and/or fuel consumption is to be minimized by determining velocity profiles $V = \{v_1, v_2, \ldots, v_n\}$ for each segment, subject to the vehicle kinematics and time constraints for arrivals at a green phase, *i.e.*,

$$\begin{aligned} \min_{v_i} \quad & J = f(V) \\ \text{subject to} \quad & \text{1) Time constraint,} \\ & \text{2) Vehicle kinematics,} \\ & \text{3) Allowable bound for the speed,} \\ & \text{4) Speed choice space.} \end{aligned} \tag{2}$$

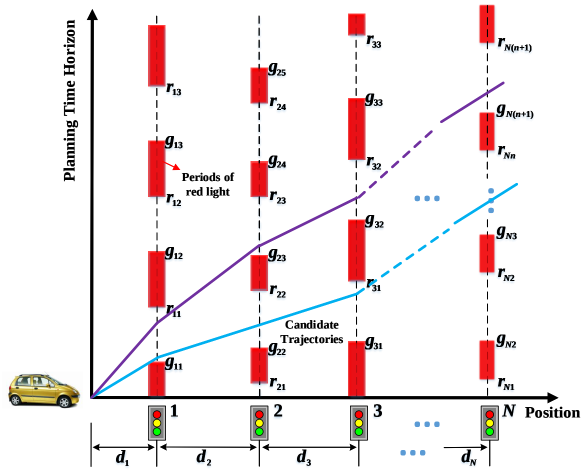


Figure 2: Schematic diagram of traffic lights distributed in a route. The green light optimal velocity planning is to find an optimal speed profile that eliminates idling at red signals using the upcoming traffic light information. d_i denotes the distance between two consecutive intersections; g_{ij}, r_{ij} represent the start of the j-th green phase, j-th red phase of the i-th traffic light, respectively.

where $f(V)$ denotes certain cost function, and the four types of constraints are defined as follows.

- *Time constraint*, which ensures the arriving time t_i at i-th intersection lie in the green phase,

$$t_i \in \bigcup_{j=1}^{\infty} [g_{ij}, r_{ij}) \,. \tag{3}$$

- *Vehicle kinematics*, which describes the vehicle's motion in each segment. This is highly related to the estimation of trip time and fuel consumption.
- *Allowable bound for the speed*, which is usually specified by traffic agency for each segment,

$$v_i \in [v_{i,\min}, v_{i,\max}] \,, i \in \mathbb{N}, \tag{4}$$

  where $v_{i,\min}, v_{i,\max}$ denote the minimal and maximal allowable velocity in the i-th segment, respectively.
- *Speed choice space*, which is required by designers. For instance, the velocity for suggestion can be continuous, or limited to integer numbers, or even restricted to binary values (*e.g.*, high speed or low speed).

$$v_i \in \mathbb{S}, i \in \mathbb{N}, \tag{5}$$

  where $\mathbb{S}$ denotes the speed choice space.

2.2 Modeling of Green Light Optimal Velocity Planning

The formulation (2) presents a highly abstract model for the velocity planning problem. Here, we present typical mathematical models of the involved cost function and constraints. According to different focuses reflected in the performance index J, the velocity planning problem (2) can be categorized into two cases: 1) trip-time oriented [6, 10]; and 2) fuel-consumption oriented [11, 18]:

1. *Trip-time oriented:* The objective is to find an optimal velocity profile such that the trip time is minimized without waiting at the red lights. Then, the performance index can be written as:

$$J_t = \sum_{i=1}^{N} h_i(d_i, v_i), \tag{6}$$

   where $h_i(d_i, v_i)$ denotes the time consumption in each segment i.

2. *Fuel-consumption oriented:* This aims to find a energy-saving speed profile, where the performance index is

$$J_f = \sum_{i=1}^{N} f_i(d_i, v_i), \tag{7}$$

where $f_i(d_i, v_i)$ represents the fuel consumption in the i-th segment.

Note that there exist a variety of fuel consumption models in the literature [26]. A typical one is the Virginia Tech Comprehensive Power-Based Fuel Consumption Model (VT-CPFM). In VT-CPFM, a polynomial of engine power is used to calculate fuel consumption rate, which is defined as

$$f_{\mathrm{c}} = \begin{cases} \alpha_0 + \alpha_1 E + \alpha_2 E^2, & \text{if } E > 0 \\ \alpha_0, & \text{if } E = 0 \end{cases} \tag{8}$$

where f_{c} is the fuel consumption rate (g/s), E denotes the engine power, and α_0, α_1, and α_2 are the coefficients of the fuel consumption model that need to be calibrated for a particular vehicle. Normally, the the engine power E can be computed as

$$E = v\left(m\dot{v} + \frac{1}{2}C_d A \rho_a v^2 + mgf\cos\theta + mg\sin\theta\right),$$

where m is the vehicle's mass, C_d is the drag coefficient, A is the vehicle frontal area, ρ_a is the air density, f is the rolling resistance coefficient of the tires, g is the gravitational acceleration, θ is the road slope, and v is the vehicle speed.

Meanwhile, the problem of green light velocity planning can also be grouped according to the assumptions for vehicle kinematics: 1) constant-speed type [13]; and 2) constant-acceleration type [22]:

1. *Constant-speed type:* It is assumed that the vehicle runs at a constant speed in each segment, ignoring the acceleration or deceleration process. This is the simplest case. Then, the arriving time t_i can be calculated as

$$t_i = \sum_{j=1}^{i} \frac{d_j}{v_j}, i \in \mathbb{N}. \tag{9}$$

2. *Constant-acceleration type:* It is one step improvement over the constant speed assumption, but also increases the complexity level. It accounts for the acceleration or deceleration process for velocity change form v_i running in the i-th segment to v_{i+1} running in the next segment. However, the acceleration or deceleration is assumed to be constant, *i.e.*,

$$v_{i+1} = v_i + a \times t_{a,i}, \tag{10}$$

where a represents the constant acceleration/deceleration value and $t_{a,i}$ is the time length for transition. In this case, the arriving time t_i is calculated as

$$t_i = \sum_{j=1}^{i} \left(\left| \frac{v_j - v_{j-1}}{a} \right| + \frac{d_j - l_j}{v_j} \right), \tag{11}$$

where

$$l_j = \left| \frac{v_j^2 - v_{j-1}^2}{a} \right|, j \in \mathbb{N}.$$

Therefore, the generic green light optimal velocity planning, formulated in (2), can be further categorized into four basic types: trip-time oriented and constant-speed type; trip-time oriented and constant-acceleration type; fuel-consumption oriented and constant-speed type; and fuel-consumption oriented and constant-acceleration type.

In the following, we first formally prove that green light optimal velocity planning under the assumption of trip-time oriented and constant-speed type is NP-complete when the speed suggestion is restricted to binary choice in each segment. Then, we develop a numerical heuristic algorithm to find an approximate solation to the generic green light optimal velocity planning problem (2).

Remark 1. We note that there also exist other variants of formulation, considering nonlinear vehicle dynamics [6, 21], probabilistic prediction of signal timing [10], driver's behavior adaptability [14], and queue effects at intersections [19]. These considerations are more in accordance to the realistic conditions, but they make the problem much more complicated as well. Consequently, many of them only focus on single intersection considering the nearest traffic light in a *sequential* fashion.

3 Computational Complexity Analysis

For the first step to theoretically investigate the complexity of green light optimal planning problem, we consider the case of trip-time oriented and constant-speed type. Then, the problem is concisely formulated as

$$\begin{aligned} \min_{v_i} \quad & \sum_{i=1}^{N} \frac{d_i}{v_i} \\ \text{subject to} \quad & \sum_{j=1}^{i} \frac{d_i}{v_i} \in \bigcup_{j=1}^{\infty} [g_{ij}, r_{ij}) \,, \\ & v_i \in [v_{i,\min}, v_{i,\max}] \,, \\ & v_i \in \mathbb{S}, \quad i \in \mathbb{N}. \end{aligned} \tag{12}$$

In this section, we discuss the computational complexity (12), where the speed suggestion is restricted to binary choice in each segment. We show that this special case is NP-complete by a reduction from the partition problem.

Remark 2. The formulation (12) has captured one key challenge in this problem, *i.e.*, the dynamic switching of traffic signal timing: $\bigcup_{j=1}^{\infty} [g_{ij}, r_{ij})$. This type of constraint naturally makes the feasible solution space disjoined and non-convex. Solving a non-convex optimization problem is in general computationally intensive. In this section, for the first time, we show that one special form of (12) actually belongs to NP-complete, which means there exist no polynomial-time algorithms to solve the general optimal velocity planning problem exactly unless P=NP.

3.1 Preliminaries on P/NP

We first introduce some definitions for complexity analysis for the sake of completeness; the interested reader can refer to [27] for more details.

- *Polynomial-time reduction:* Problem $\mathcal{A}$ is said to be polynomial-time reduced to problem $\mathcal{B}$ if there is a polynomial-time algorithm that transforms the inputs of $\mathcal{A}$ into those of $\mathcal{B}$, such that the transformed problem has the same output as the original problem.
- *Class P:* The general class of problems, where there exists an algorithm that can provide an answer in polynomial time, is called "class P" (polynomial) or "P".
- *Class NP:* The general class of problems, where an answer can be verified in polynomial time is called "class NP" (Non-deterministic polynomial) or "NP".
- *Class NP-complete:* A set of problems to each of which any other NP problem can be reduced in polynomial time, and whose solution can still be verified in polynomial time, is called NP-complete.

According to the definitions, we have P $\subseteq$ NP. The problems that belong to NP-complete cannot be solved in polynomial time unless P = NP. Many scientists believe that P $\neq$ NP [28], and up to now no one has been able to find a polynomial-time algorithm for any known NP-complete problem. To prove NP-completeness of a new problem, one common way is to reduce one known NP-complete problem to this new problem [27]. For solving NP-complete problems, practical methods are typically based on rule or heuristic algorithms, such as greedy algorithm and genetic algorithm [25, 27]. However, such methods can only provide approximated and sub-optimal solutions.

3.2 NP-complete result for binary speed choice

Here, we introduce the *partition problem*: given a set of N positive integers $\{p_1, p_2, \ldots, p_n\}$, decide whether there exists a subset $S \subseteq \{1, 2, \ldots, N\}$ such that

$$\sum_{i \in S} p_i = \sum_{i \notin S} p_i. \tag{13}$$

Lemma 1 ([29])**.** *Partition problem is NP-complete.*

Then, we state the main theorem of this paper.

Theorem 1. *The problem of green light optimal velocity planning* (12) *belongs to NP-complete, if the speed space $\mathbb{S}$ only have binary value,* i.e., $\mathbb{S} = \{v_{low}, v_{high}\}$*,* $v_{i,\min} \le v_{low}, v_{high} \le v_{i,\max}$*.*

Proof. The proof is based on a reduction of the partition problem to (12) with binary speed choices. Given any instance of the partition problem $\{p_1, p_2, \ldots, p_N\}$, we can construct an instance of (12) in the following way.

Step 1. Set p_i as the length of segment i, *i.e.*,

$$d_i = p_i, i \in \mathbb{N}.$$

Step 2. Make the green light phase of i-th intersection be sufficient large, where $i = 1, 2, \ldots, N-1$, such that the vehicle can pass these $N-1$ intersections using any speed choice.

Step 3. Make the traffic light cycle of the N-th intersection be sufficient large and only the following time point be green light phase

$$t_g = \frac{\sum_{i=1}^{N} d_i}{2} \times \left(\frac{1}{v_{\text{low}}} + \frac{1}{v_{\text{high}}}\right). \tag{14}$$

It is obvious that the construction in steps 1-3 can be finished in polynomial time. Then, it remains to show that the constructed instance of (12) is equivalent to the original partition problem, *i.e.*, any solution to (12) is a solution to the partition problem, and vice versa. Note that the optimal cost value of the constructed instance of (12) is t_g if it exists.

1) $\Rightarrow$: Suppose there is one suction to the constructed instance of (12). Then, there exist a subset $\hat{S} \subset \{1, 2, \ldots, N\}$, and the advisory speed for $\hat{S}$ is v_{low} and other segments are v_{high}, such that we have

$$\frac{\sum_{j \in \hat{S}} d_j}{v_{\text{low}}} + \frac{\sum_{j \notin \hat{S}} d_j}{v_{\text{high}}} = t_g, \tag{15}$$

where t_g is shown in (14). Note that, we have

$$\sum_{j \in \hat{S}} d_j + \sum_{j \notin \hat{S}} d_j = \sum_{i=1}^{N} d_i. \tag{16}$$

According to (14), (15) and (16), we have

$$\sum_{j \in \hat{S}} d_j = \sum_{j \notin \hat{S}} d_j. \tag{17}$$

This means a solution of the original partition problem is derived.

2) $\Leftarrow$: Suppose there is one solution to the partition problem, *i.e.*, there exist a subset $S \in \{1, 2, \ldots, N\}$ such that $\sum_{j \in S} d_j = \sum_{j \notin S} d_j$. Next, choose the velocity for the segments in S as v_{low} and the velocity for other segments as v_{high}. Then, we have

$$\frac{\sum_{j \in S} d_j}{v_{\text{low}}} + \frac{\sum_{j \notin S} d_j}{v_{\text{high}}} = \frac{\sum_{i=1}^{N} d_i}{2} \left(\frac{1}{v_{\text{low}}} + \frac{1}{v_{\text{high}}}\right) = t_g, \tag{18}$$

This indicates a solution to the constructed instance of (12) is obtained.

Therefore, we can claim that the partition problem can be polynomial-time reduced to the optimal velocity planning (12) with binary speed choices. According to Lemma 1, partition problem is NP-complete. Consequently, the problem (12) with binary speed choices belongs to NP-complete.

□

To the best of our knowledge, Theorem 1 is the first result in the literature that formally addresses the computational complexity of green light optimal velocity planning problem. Most of exiting studies only focus on the formulation of the mathematical model considering different factors, and then employ rule-based or DP-based algorithms to solve the problem [6, 10, 11, 12, 13, 14].

Remark 3. If the speed space $\mathbb{S}$ is in integer domain, the complexity of (12) remains an open question. Intuitively, it is much more difficult to solve (12) with more velocity choices. For instance, if the speed limit is set to $v_{i,\min} = 1\ m/s, v_{i,\max} = 30\ m/s$, there are 30 available choices for each segment. Then, the number of candidate velocity profiles to (12) is 30^N. However, the size of search space in Theorem 1 is 2^N since each segment only has two possible choices. If $\mathbb{S}$ is in continue domain, then the complexity of (12) is unknown as well. Note that a linear combination of two feasible solutions to (12) may be infeasible due to the constraint of traffic lights. This means the feasible region of (12) is not convex. It is in general nontrivial to solve non-convex problems.

4 Solution via a Heuristic Algorithm

In this section, we develop a numerical heuristic algorithm to solve the green light optimal velocity planning problem (2). Since we have proven that a special form of (12) is NP-complete, there are no effective algorithms to find a global optimal solution in polynomial time. Instead, we propose a genetic algorithm to obtain a sub-optimal solution. In particular, for problem (2), we consider a combination of trip time and fuel consumption as the cost function and take the constant acceleration into account. Then, (2) can be compactly written into

$$\begin{aligned} \min_{v_i} \quad & J_t + \rho J_f \\ \text{subject to} \quad & t_i \in \bigcup_{j=1}^{\infty} [g_{ij}, r_{ij}), \\ & v_i \in [v_{i,\min}, v_{i,\max}], \\ & v_i \in \mathbb{S}, \quad i \in \mathbb{N}, \end{aligned} \tag{19}$$

where $\mathbb{S}$ is in continuous space, t_i is defined in (11), and J_t denotes the trip time and J_f denotes the fuel consumption, and ρ is a cost balanced factor. In our case, J_t is calculated as

$$J_t = \sum_{i=1}^{N} \left(\left| \frac{v_j - v_{j-1}}{a} \right| + \frac{d_j - l_j}{v_j} \right),$$

and J_f is computed as

$$J_f = \sum_{i=1}^{N} \int_{t_{i-1}}^{t_i} f_{\text{c}} dt,$$

where f_{c} is the fuel consumption rate defined in (8). In this paper, we assume that (19) is feasible in the subsequent discussion; otherwise, stops at the red traffic light are unavoidable.

4.1 General framework

Genetic algorithm is a search technique that follows the evolution paradigm [30]. A population is first initialized and then genetic operators are applied to produce offsprings (corresponding to the exploration of the neighborhood), which converges to a nearly optimal solution over a large number of generations or iterations.

One advantage of the genetic algorithm with respect to other local search algorithms is that more variable space can be explored in each iteration, since a genetic algorithm can incorporate more strategies to generate

Algorithm 1 Genetic algorithm for the velocity planning

Input: *Problem statistics*: Intersection number N, green light phase $\bigcup_{j=1}^{\infty}[g_{ij}, r_{ij})$, speed bound $v_{i,\max}, v_{i,\min}$, distance d_i, constant acceleration a, initial speed v_0, cost balanced factor ρ
GA parameters: Maximum iterations $k_{\max}$, population size N_p, linear combination factor γ, probability of rule-based initiation p_i and probability of rule-based crossover p_c.
Output: Optimal speed profile: $V = [v_1, \dots, v_N]$

```
Initialize a population of N_p individuals using (20) (with probability 1 − p_i) and the rule-based initialization (with probability p_i).
Evaluate the fitness of the initialized population (indexed as 0) using (21)
for k = 1, ..., k_max do
    Selection: select parents from population k − 1 using the linear ranking strategy;
    Crossover: perform crossover on parents to create population k using (25) (with probability 1 − p_c) and rule-based crossover (with probability p_c);
    Mutation: perform mutation of population k;
    Fitness: evaluate the fitness of population k using (21);
    if Converged then
        break;
    end if
end for
Output the best individual in the population k as [v_1, ..., v_N].
```

new individuals both in the initial population phase and in the dynamic generation phase [31, 32]. The overall structure of our proposed genetic algorithm for (19) includes the following steps:

- *Coding:* The genes of the chromosomes (or individuals) describe the velocity profiles for a chosen route, and each chromosome denotes a possible solution to (19);
- *Initial population:* We randomly generate a portion of initial chromosomes, and also generate some feasible chromosomes considering the hard constraints in (19);
- *Fitness evaluation:* The fitness of each chromosome is evaluated by the trip time and fuel consumption and feasibility of the traffic light constraints (3);
- *Selection operator:* The selection operator is based on a linear ranking to choose the chromosomes for reproduction;
- *Offspring generation:* The new generation is obtained by a linear combination of selected chromosomes and gene mutation;
- *Stop criterion:* When the number of generations reach a pre-fixed maximum number or the solution does not change for a pre-fixed number of iterations, we stop the algorithm and output the best chromosome as a suboptimal solution.

4.2 Detailed design

Here, we present the detailed descriptions for the proposed genetic algorithm in each step.

4.2.1 Coding for solutions

To implement the genetic algorithm, we first need to design the chromosome representation for problem (19). Usually, the chromosome representation is not unique, but needs to adapt with the problem characteristic [25]. For our problem, we use the target velocity in each segment as the genes of a chromosome. The length of a chromosome is equal to the number of intersections, *i.e.*, a chromosome is represented by

$$V = \{v_1, v_2, \dots, v_N\}.$$

4.2.2 Initial population

Population initialization is critical in the design of genetic algorithm since it can affect the search space, convergence speed and also the final solution [33]. For our problem, we first randomly generate a portion of initial chromosomes that satisfy the bounds of allowable speeds and ignore the time constraint (3), *i.e.*,

$$v_i = v_{i,\min} + \tau(v_{i,\max} - v_{i,\min}), i \in \mathbb{N}, \tag{20}$$

where τ is a random variable between 0 and 1.

This randomly generating method is easy to implement and also allows us to search the entire space of possible solutions. However, one major disadvantage is that the initial chromosomes usually violate the time constraint (3). To guarantee some feasible initial candidates, we also use a rule-based strategy to generate some chromosomes (see the Supplementary Materials). In the implementation, a mix of these two methods are used to produce the initial population with *priori* probability p_i. For instance, $p_i = 0.2$ means 80% of the initial population is generated by the random method, and 20% is generated by the rule-based method.

4.2.3 Fitness evaluation

The fitness evaluation is to measure the quality of each chromosome, which is in the form of

$$\mathrm{Fit}(V) = h(V) + l(V), \tag{21}$$

where $h(V)$ reflects the trip time and fuel consumption of each chromosome, and $l(V)$ considers the hard time constraint (3). Therefore, given any chromosome $V = \{v_1, v_2, \ldots, v_N\}$, we should have

$$\begin{cases} l(V) = 0, & \text{if (3) is satisfied,} \\ l(V) > \max(h(V)), & \text{otherwise.} \end{cases} \tag{22}$$

Obviously, functions satisfying (22) are not unique. In our implementation, we use the following forms

$$l(V) = \sum_{i=1}^{N} n_i, \quad h(V) = 1 - e^{-J}, \tag{23}$$

where $J = J_t + \rho J_f$ is the combination of trip time and fuel consumption in (19), and n_i is the indicator of the feasibility of passing the i-th intersection, *i.e.*,

$$n_i = \begin{cases} 0, & \text{if } t_i \in \bigcup_{j=1}^{\infty} [g_{ij}, r_{ij}), \\ 1, & \text{otherwise.} \end{cases} \tag{24}$$

In each generation, all the chromosomes are evaluated and the best individual is recorded to the next generation.

4.2.4 selection operator

The selection phase is to choose the chromosomes for reproduction. In our approach, the strategy of *linear ranking* is used to select the chromosomes to be included in the mating pool, which is widely used in the literature.

Linear ranking: Chromosomes are sorted according to their fitness and a rank $r_i \in \{1, 2, \ldots, N_p\}$ is assigned to each individual, where N_p is the population size. The best individual, *i.e.*, the individual with lowest fitness value, gets rank N_p and the worst one gets rank 1. Then,

$$\frac{2r_i}{N_p(N_p+1)}, i = 1, \ldots, N_p,$$

is the probability of choosing the i-th individual in the ranking order.

Table 1: Problem statistics of the typical scenario

Parameters	Value
Number of intersections N	10
Distance $d_i(m)$	700, 580, 440, 760, 720, 520, 720, 660, 440, 600
Traffic light cycle length (s)	60, 55, 60, 65, 65, 65, 70, 70, 75, 70
Green phase length (s)	21, 23, 25, 21, 42, 42, 28, 42, 36, 46
Green phase start time (s)	37, 19, 46, 40, 6, 20, 39, 6, 16, 42
Initial velocity $v_0(m/s)$	10
Lower bound of velocity $v_{i,\min}(m/s)$	5.6
Upper bound of velocity $v_{i,\max}(m/s)$	22.2
Cost balance factor ρ	0.3
Constant acceleration $a(m/s^2)$	1.5

4.2.5 offspring generation

Once the chromosomes for reproduction have been selected, the next stage is to produce offsprings using two genetic operators: *crossover* and *mutation*. Crossover applies to a pair of selected chromosomes, aiming to obtain better result by exchanging information contained in the current individuals. Here, a linear combination is used to realize the crossover. Suppose the selected pair of chromosomes are $V_{1,\text{parent}}, V_{2,\text{parent}}$, two offspring are generated as

$$\begin{cases} V_{1,\text{child}} = \gamma V_{1,\text{parent}} + (1-\gamma) V_{2,\text{parent}}, \\ V_{2,\text{child}} = (1-\gamma) V_{1,\text{parent}} + \gamma V_{2,\text{parent}}, \end{cases} \tag{25}$$

where γ is a linear combination factor. Similar to the initialization step, the easily implementable linear combination (25) allows us to search all possible solutions between V_1 and V_2, but with the disadvantage of leading to possible infeasible candidates. In our implementation, we also use a rule-based strategy at *priori* probability p_c to generate offsprings that promises to improve feasibility (see the Supplementary Materials). Mutation applies to a single individual with the aim to introduce extra variability into the population to enhance the diversity. We first randomly select an individual form the population. Then, we read the genes of the individual from left to right, and generate a random probability value. If the value is less than the mutation probability, we generate a number from a normal distribution $N(v_i, 1)$ and assign it to the mutation position. We repeat this process until all the genes are read.

We terminate the offspring generation phase if the number of chromosomes reaches the population size. The proposed algorithm ends when a maximal number of iterations is reached or the best individual does not change for a pre-fixed number, and we output the best individual as a sub-optimal solution. The major steps are summarized in Algorithm 1.

5 Numerical Experiments

This section presents a set of numerical experiments to validate the proposed method. We considered a typical scenario of a road with ten signalized intersections. The distance between two consecutive intersections was chosen from 400 m to 800 m randomly with step-length 20 m, and the length of each traffic light cycle was generated from 50 s to 80 s randomly with step-length 5 s. The problem statistics of a typical scenario is listed in Table 1, and the parameters of GA used in our simulations are given in Table 2.

We consider four different strategies in the proposed GA: 1) assuming only nearest traffic light information is available, *i.e.*, considering one intersection sequentially; 2) considering two intersections at a time; 3) considering five intersections at a time; 4) considering ten intersections simultaneously. For comparison, we

Table 2: Parameters in the genetic algorithm

Parameters	Value
Maximal number of iterations $k_{\max}$	1000
Population size N_p	500
Linear combination factor γ	0.33
Probability of rule-based initiation p_i	0.4
Probability of rule-based crossover p_c	0.5
Probability of crossover	0.95
Probability of mutation	0.1

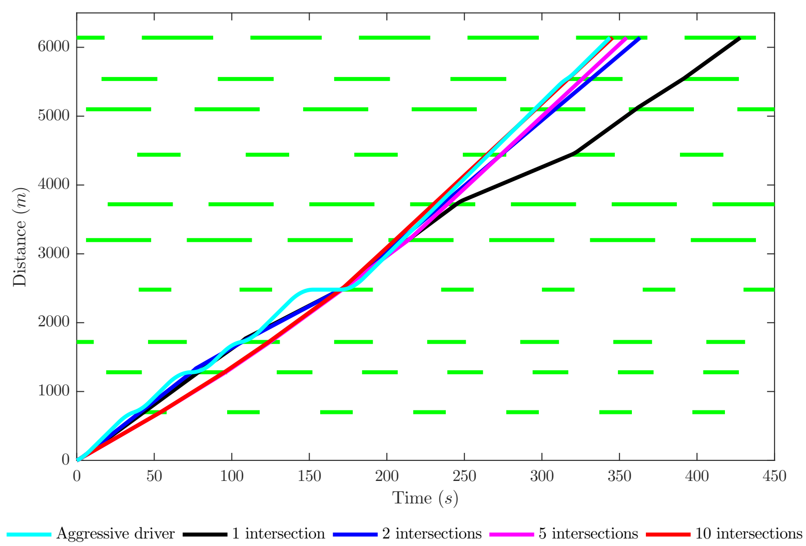
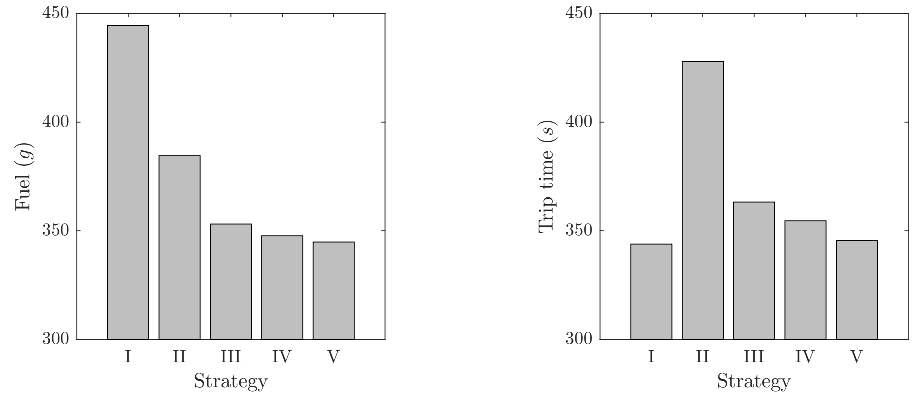


Figure 3: Vehicle trajectories for different strategies

also assume an aggressive driver's strategy in the simulations: the vehicle runs as the maximal velocity in each segment and slows down, idles behind, and accelerates away from signalized intersections if meeting the red lights.

Fig. 3 shows the trajectories when using these five strategies. It can be seen that the speed profiles from the proposed GA could avoid red lights while the aggressive driver strategy has to stop at the red lights. Also, if one only considers the nearest traffic light, the velocity profile might be greedy, which neglects the effects of the upcoming intersections. In the chosen scenario, the strategy of considering one intersection sequentially results in a speed profile that runs fast for the first three intersections and has to slow down for the forth intersection. This increases the fuel consumption. Also, for the seventh intersection, the vehicle has to slow down again since the previous velocity is not high enough. This inharmonious behaviour could be avoided if one considers more intersections at a time, as demonstrated by the trajectories provided by other strategies in Fig. 3. Meanwhile, as shown in Fig. 4, the greedy velocity profile also leads to a higher fuel consumption and higher trip time. In contrast, the strategy of considering ten intersection simultaneously provide the best improvement of fuel efficiency and reduction of trip time. Besides, Fig. 4(a) confirms that vehicles spend a significant amount of fuel consumption by slowing down, idling behind, and accelerating away from signalized intersections, which could be avoided using the upcoming traffic information, *e.g.*, solving a velocity planning problem.



(a) Fuel consumption

(b) Trip time

Figure 4: Comparison of different strategies: I: Aggressive driver; II: only considering 1 intersection; III: considering 2 intersections; IV: considering 5 intersections; V: considering 10 intersections simultaneously.

6 Conclusion

This paper studied the problem of green light optimal velocity planning problem. We have proven that this problem is NP-complete for the case where there are only binary velocity choices in each segment. Our proof is based a reduction from a known NP-complete problem. Then, a genetic algorithm has been introduced to obtain a sub-optimal solution, where a detailed discussion is provided on gene coding, population initialization, select operator and genetic operator. Numerical experiments demonstrated the effectiveness of our method, which also confirmed that more improvements on fuel efficiency could be realized by taking multiple intersections into account simultaneously. One future work is to consider the complexity analysis for the velocity planning problem in continuous domain, and another interesting direction of future work is to consider multiple vehicles, such as platoons [34], running through signalized intersections.

Acknowledgment

The authors would like to thank Prof. Wenxun Xing at Tsinghua University for his helpful comments on P/NP issues.

Appendix

6.1 Rule-based initialization

To improve the probability of feasible individuals in the initialization phase, we employ a rule-based strategy at probability p_i, as detailed in Procedure 1. The key idea of the proposed rule-based strategy is to find a feasible trip time window for each intersection; see (26). In Procedure 1, it is assumed that each segment d_i is long enough for the vehicle's acceleration/deceleration process, *i.e.*,

$$d_i \geq \max\left\{\frac{v_{i,\max}^2 - v_{i-1}^2}{2|a|}, \frac{v_{i-1}^2 - v_{i,\min}^2}{2|a|}\right\}$$

Procedure 1 Rule-based initialization

Input: Intersection Number N, Green light phase $\bigcup_{j=1}^{\infty}[g_{ij}, r_{ij})$, speed bound $v_{i,\max}, v_{i,\min}$, distance d_i, constant acceleration value a, and initial speed v_0

Output: Speed profile: $V = [v_1, \ldots, v_N]$

1: Initialize $t_0 = 0$, $v_0 = v_0$.

2: **for** $i = 1, \ldots, N$ **do**

3: Calculate the bounds of arrival time $t_{\min}, t_{\max}$

$$t_{\min} = t_{i-1} + \left|\frac{v_{i,\max} - v_{i-1}}{a}\right| + \frac{2|a|d_i - (v_{i,\max}^2 - v_{i-1}^2)}{2|a|v_{i,\max}},$$
$$t_{\max} = t_{i-1} + \left|\frac{v_{i-1} - v_{i,\min}}{a}\right| + \frac{2|a|d_i - (v_{i-1}^2 - v_{i,\min}^2)}{2|a|v_{i,\min}}.$$

4: Calculate the feasible travel time window:

$$G = [t_{\min}, t_{\max}] \bigcap \left(\bigcup_{j=1}^{\infty}[g_{ij}, r_{ij})\right) \tag{26}$$

5: Choose the arrival time randomly $t_i \in G$.

6: Calculate the trip time during the i-th segment

$$t_{\text{seg}} = t_i - t_{i-1}.$$

7: Compute the velocity v_i as

$$v_i = \begin{cases} v_{i-1} - |a|t_{\text{seg}} + \\ \qquad \sqrt{-2v_{i-1}|a|t_{\text{seg}} + a^2 t_{\text{seg}}^2 + 2|a|d_i}, \ \text{if } \dfrac{d_i}{v_{i-1}} < t_{\text{seg}} \\ v_{i-1} + |a|t_{\text{seg}} - \\ \qquad \sqrt{2v_{i-1}|a|t_{\text{seg}} + a^2 t_{\text{seg}}^2 - 2|a|d_i}, \quad \text{otherwise} \end{cases} \tag{27}$$

8: **end for**

6.2 Rule-based crossover

As mentioned in Remark 1, the linear combination (25) for crossover is able to to search all possible solutions between two candidates, but it may also lead to infeasible individuals due to the non-convexity of the feasible domain. In our implementation, we use a rule-based crossover at probability p_c defined in Procedure 2. Similar to the rule-based initialization, the key idea is to find feasible trip time window that is defined by both the selected parents and green light phase; see (28).

Procedure 2 Rule-based crossover

Input: Intersection Number N, Green light phase $\bigcup_{j=1}^{\infty}[g_{ij}, r_{ij})$, speed bound $v_{i,\max}, v_{i,\min}$, distance d_i, constant acceleration value a, initial speed v_0, and two selected chromosomes $V_1 = [v_{1,1}, v_{1,2}, \ldots, v_{1,N}], V_2 = [v_{2,1}, v_{2,2}, \ldots, v_{2,N}]$

Output: Two offsprings: $V_3 = [v_{3,1}, \ldots, v_{3,N}]$, $V_4 = [v_{4,1}, \ldots, v_{4,N}]$

1: Initialize $t_{3,0} = 0, t_{4,0} = 0,\ v_{3,0} = v_0, v_{4,0} = v_0$.

2: **for** $i = 1, \ldots, N$ **do**

3: Calculate the bounds of arrival time $t_{\min,k}, t_{\max,k} (k = \{3, 4\})$

$$\begin{aligned} t_{\min,k} &= t_{k,i-1} + \left|\frac{v_{i,\max} - v_{k,i-1}}{a}\right| \\ &\quad + \frac{2|a|d_i - (v_{i,\max}^2 - v_{k,i-1}^2)}{2|a|v_{i,\max}}, \\ t_{\max,k} &= t_{k,i-1} + \left|\frac{v_{k,i-1} - v_{i,\min}}{a}\right| \\ &\quad + \frac{2|a|d_i - (v_{k,i-1}^2 - v_{i,\min}^2)}{2|a|v_{i,\min}}. \end{aligned}$$

4: Calculate the trip time $T_k, k = \{1, 2\}$ of the parents, and find the time interval T.

$$\begin{aligned} T_k &= \sum_{j=1}^{i}\left(\left|\frac{v_{k,j} - v_{k,j-1}}{a}\right| + \frac{2|a|d_j - |v_{k,j}^2 - v_{k,j-1}^2|}{2|a|v_{k,j}}\right), \\ T &= [\min(T_1, T_2), \max(T_1, T_2)]. \end{aligned}$$

5: Calculate the feasible travel time window $(k = \{3, 4\})$:

$$G_k = [t_{\min,k}, t_{\max,k}] \bigcap \left(\bigcup_{j=1}^{\infty}[g_{ij}, r_{ij})\right) \bigcap T. \tag{28}$$

6: Choose the arrival time randomly $t_{k,i} \in G_k, k = \{3, 4\}$.

7: Calculate the trip time during the i-th segment

$$t_{k,\text{seg}} = t_{k,i} - t_{k,i-1}, k = 3, 4$$

8: Compute the velocity $v_{k,i}, k = \{3, 4\}$ using an equation similar to (27).

9: **end for**

References

[1] Yang Zheng, Shengbo Li Eben, Biao Xu, Keqiang Li, and Jianqiang Wang. Complexity analysis of green light optimal velocity problem: an np-complete result for binary speed choices. In *Proceedings of the 14th Intelligent Transportation System Asia Pacific Forum*, 2015.

[2] Junping Zhang, Fei-Yue Wang, Kunfeng Wang, Wei-Hua Lin, Xin Xu, and Cheng Chen. Data-driven intelligent transportation systems: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 12(4):1624–1639, 2011.

[3] David Schrank, Bill Eisele, Tim Lomax, and Jim Bak. 2015 urban mobility scorecard. *Texas A&M Transp. Inst., College Station, TX, USA*, 2015.

[4] Shengbo Eben Li, Shaobing Xu, Xiaoyu Huang, Bo Cheng, and Huei Peng. Eco-departure of connected vehicles with v2x communication at signalized intersections. *IEEE Transactions on Vehicular Technology*, 64(12):5439–5449, 2015.

[5] Aleksandar Stevanovic, Jelka Stevanovic, Kai Zhang, and Stuart Batterman. Optimizing traffic control to reduce fuel consumption and vehicular emissions: Integrated approach with vissim, cmem, and visgaost. *Transportation Research Record: Journal of the transportation research board*, (2128):105–113, 2009.

[6] Behrang Asadi and Ardalan Vahidi. Predictive cruise control: Utilizing upcoming traffic signal information for improving fuel economy and reducing trip time. *IEEE transactions on control systems technology*, 19(3):707–714, 2011.

[7] Pitu Mirchandani and Larry Head. A real-time traffic signal control system: architecture, algorithms, and analysis. *Transportation Research Part C: Emerging Technologies*, 9(6):415–432, 2001.

[8] Kanok Boriboonsomsin, Alexander Vu, and Matthew Barth. Eco-driving: Pilot evaluation of driving behavior changes among us drivers. *University of California Transportation Center*, 2010.

[9] Y Saboohi and H Farzaneh. Model for developing an eco-driving strategy of a passenger vehicle based on the least fuel consumption. *Applied Energy*, 86(10):1925–1932, 2009.

[10] Grant Mahler and Ardalan Vahidi. Reducing idling at red lights based on probabilistic prediction of traffic signal timings. In *American Control Conference (ACC), 2012*, pages 6557–6562. IEEE, 2012.

[11] Sindhura Mandava, Kanok Boriboonsomsin, and Matthew Barth. Arterial velocity planning based on traffic signal information under light traffic conditions. In *Intelligent Transportation Systems, 12th International IEEE Conference on*, pages 1–6. IEEE, 2009.

[12] Giovanni De Nunzio, Carlos Canudas Wit, Philippe Moulin, and Domenico Di Domenico. Eco-driving in urban traffic networks using traffic signals information. *International Journal of Robust and Nonlinear Control*, 2015.

[13] Marcin Seredynski, Wojciech Mazurczyk, and Djamel Khadraoui. Multi-segment green light optimal speed advisory. In *Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW), 2013 IEEE 27th International*, pages 459–465. IEEE, 2013.

[14] Xuehai Xiang, Kun Zhou, Wei-Bin Zhang, Wenhu Qin, and Qingzhou Mao. A closed-loop speed advisory model with driver's behavior adaptability for eco-driving. *IEEE Transactions on Intelligent Transportation Systems*, 16(6):3313–3324, 2015.

[15] Emmanouil Koukoumidis, Margaret Martonosi, and Li-Shiuan Peh. Leveraging smartphone cameras for collaborative road advisories. *IEEE Transactions on mobile computing*, 11(5):707–723, 2012.

[16] Konstantinos Katsaros, Ralf Kernchen, Mehrdad Dianati, and David Rieck. Performance study of a green light optimized speed advisory (glosa) application using an integrated cooperative its simulation platform. In *Wireless Communications and Mobile Computing Conference (IWCMC), 2011 7th International*, pages 918–923. IEEE, 2011.

[17] Biao Xu, Fang Zhang, Jianqiang Wang, and Keqiang Li. B&b algorithm-based green light optimal speed advisory applied to contiguous intersections. In *CICTP 2015*, pages 363–375. 2015.

[18] Raj Kishore Kamalanathsharma and Hesham A Rakha. Leveraging connected vehicle technology and telematics to enhance vehicle fuel efficiency in the vicinity of signalized intersections. *Journal of Intelligent Transportation Systems*, 20(1):33–44, 2016.

[19] Hao Yang, Hesham Rakha, and Mani Venkat Ala. Eco-cooperative adaptive cruise control at signalized intersections considering queue effects. *IEEE Transactions on Intelligent Transportation Systems*, 2016.

[20] Nianfeng Wan, Ardalan Vahidi, and Andre Luckow. Optimal speed advisory for connected vehicles in arterial roads and the impact on mixed traffic. *Transportation Research Part C: Emerging Technologies*, 69:548–563, 2016.

[21] Baisravan HomChaudhuri, Ardalan Vahidi, and Pierluigi Pisu. Fast model predictive control-based fuel efficient control strategy for a group of connected vehicles in urban road conditions. *IEEE Transactions on Control Systems Technology*, 2016.

[22] Yugong Luo, Shan Li, Shuwei Zhang, Zhaobo Qin, and Keqiang Li. Green light optimal speed advisory for hybrid electric vehicles. *Mechanical Systems and Signal Processing*, 87:30–44, 2017.

[23] Jun Hu, Ioannis Kaparias, and Michael GH Bell. Spatial econometrics models for congestion prediction with in-vehicle route guidance. *IET Intelligent Transport Systems*, 3(2):159–167, 2009.

[24] Yu Marco Nie, Xing Wu, John F Dillenburg, and Peter C Nelson. Reliable route guidance: A case study from chicago. *Transportation Research Part A: Policy and Practice*, 46(2):403–419, 2012.

[25] Darrell Whitley. A genetic algorithm tutorial. *Statistics and computing*, 4(2):65–85, 1994.

[26] Shengbo Eben Li, Kun Deng, Yang Zheng, and Huei Peng. Effect of pulse-and-glide strategy on traffic flow for a platoon of mixed automated and manually driven vehicles. *Computer-Aided Civil and Infrastructure Engineering*, 30(11):892–905, 2015.

[27] Christos H Papadimitriou. *Computational complexity*. John Wiley and Sons Ltd., 2003.

[28] Lance Fortnow. The status of the P versus NP problem. *Communications of the ACM*, 52(9):78–86, 2009.

[29] Michael R Garey and David S Johnson. *Computers and intractability*, volume 29. wh freeman New York, 2002.

[30] John H Holland. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press, 1992.

[31] F Pezzella, G Morganti, and G Ciaschetti. A genetic algorithm for the flexible job-shop scheduling problem. *Computers & Operations Research*, 35(10):3202–3212, 2008.

[32] Xinxin Du, Kyaw Ko Ko Htet, and Kok Kiong Tan. Development of a genetic-algorithm-based nonlinear model predictive control scheme on velocity and steering of autonomous vehicles. *IEEE Transactions on Industrial Electronics*, 63(11):6970–6977, 2016.

[33] Shahryar Rahnamayan, Hamid R Tizhoosh, and Magdy MA Salama. A novel population initialization method for accelerating evolutionary algorithms. *Computers & Mathematics with Applications*, 53(10):1605–1614, 2007.

[34] Yang Zheng, Shengbo Eben Li, Jianqiang Wang, Dongpu Cao, and Keqiang Li. Stability and scalability of homogeneous vehicular platoon: Study on the influence of information flow topologies. *IEEE Transactions on Intelligent Transportation Systems*, 17(1):14–26, 2016.