Detecting Regularities of Traffic Signal Timing Using GPS Trajectories*

Juan YU†††, Peizhong LU††, Jianmin HAN†††‡, Nonmembers, and Jianfeng LU††(d), Member

SUMMARY  Traffic signal phase and timing (TSPaT) information is valuable for various applications, such as velocity advisory systems, navigation systems, collision warning systems, and so forth. In this paper, we focus on learning baseline timing cycle lengths for fixed-time traffic signals. The cycle length is the most important parameter among all timing parameters, such as green lengths. We formulate the cycle length learning problem as a period estimation problem using a sparse set of noisy observations, and propose the most frequent approximate greatest common divisor (MFAGCD) algorithms to solve the problem. The accuracy performance of our proposed algorithms is experimentally evaluated on both simulation data and the real taxi GPS trajectory data collected in Shanghai, China. Experimental results show that the MFAGCD algorithms have better sparsity and outliers tolerant capabilities than existing cycle length estimation algorithms.

key words: taxi GPS trajectory, traffic signals, cycle length estimation

1. Introduction

City-scale traffic signal phase and timing (TSPaT) information, including the cycle length and green lengths of phases within each cycle, is an indispensable input for various in-vehicle applications, such as velocity advisory system [1], [2], navigation systems [3], collision warning systems, and so forth. In the TSPaT information, the cycle length is the first important parameter to be estimated. For one thing, accurate estimation of the cycle length is a prerequisite for the estimation of other parameters, such as green lengths estimation in Kerper et al. [4] and Fayazi et al. [5]. For another, the accurate cycle length is also required for queue profiles estimation at intersections as introduced by Ramezani et al. [6], and queue profiles are important criteria for evaluating the performance of arterial road networks.

Several recent studies have been put forth on learning cycle lengths of traffic signals using trajectories of probe vehicles. Fayazi et al. [5] proposed a method to estimate the cycle length from a set of bus trajectories crossing the corresponding intersection. They modeled the cycle length estimation as an optimization problem to find the optimal cycle length minimizing the sum of squared normalized errors. Chuang et al. [7] proposed a two-step method to learn the cycle length from a set of green-start times extracted from high sampling rate GPS trajectories. However, their method requires high sampling rate trajectories and high penetration rate of probe vehicles. Protschky et al. [8] proposed a trial-and-error strategy to reconstruct the cycle length of an intersection using high sampling rate GPS trajectories. Their intuition was that a clear trajectory pattern could be observed when trajectories were folded with the correct cycle length, whereas no pattern could be detected when trajectories were folded with wrong cycle lengths.

In this paper, we aim to learn baseline timing cycle lengths of traffic lights using low sampling rate taxi GPS trajectories. For a specific signalized intersection, each passing taxi could contribute a small piece of information about states of corresponding traffic lights. We observe that the green-start event of a fixed-time traffic light could be captured by taxis approaching the intersection during red intervals, and the green-start event is a periodic event whose minimum period is the cycle length. Therefore, we could estimate baseline cycle lengths from a set of observations of the green-start event, i.e., green-start times. However, due to the low-sampling rate of GPS trajectories, positioning errors of GPS devices, and the low penetration rate of probe vehicles (taxis in this paper), observations about the green-start event are sparse, noisy, and even with outliers. Thus, baseline cycle lengths estimation using green-start times extracted from taxi GPS trajectories is a non-trivial task.

Our main contribution is that we model the cycle length estimation as the problem of period estimation from a sparse set of noisy observations, and propose a family of algorithms based on the idea of approximate greatest common divisor (AGCD), called the most-frequent approximate greatest common divisor (MFAGCD), to solve the problem. The family of MFAGCD algorithms is motivated by the fact that the period of a periodic event is very likely to be the greatest common divisor (GCD) of a set of sparse observations in the noise-free case.
2. Related Works

2.1 Traffic Signal Information Collection

Gowrishankar and Work[9] formalized the problem of estimating the control strategies of traffic controllers as an inverse optimal control (IOC) problem by observing the queue lengths at the intersection and the corresponding control actions. Gahrooei and Work[10] modeled the problem of inferring traffic signal phases at an intersection as a inference problem on a discrete-time hidden Markov model. Kerper et al.[4] designed an Traffic Light State Estimation (TLSE) method using shared high sampling rate velocity profiles from other drivers. Zhu et al.[11] proposed an system POVA to detect traffic lights status from taxis and buses probe data in Shanghai. Fayazi et al. [5] made use of a few days’ aggregated bus data in the city of San Francisco, CA, USA to estimate cycle times and the duration of reds for fixed-time traffic lights, and estimated the start of greens in real time by monitoring the movement of buses across intersections.

2.2 Periodicity Analysis from Noisy and Incomplete Observations

Fogel and Gavish[12] firstly proposed a periodogram-based method to solve the unknown period. Casey and Sandler[13] proved that the true period is an approximate greatest common divisor of sampling records collected chronologically from a taxi, i.e., $t_{r} = R_{0} \rightarrow R_{1}, \ldots \rightarrow R_{n−1}$, where each record $R_{i} = \langle t_{i}, l_{i}, s_{i}, u_{i} \rangle$, $0 \leq i < n$, contains four attributes: (1) $t_{i}$ is the timestamp at which the record was sampled; (2) $l_{i} = \langle lat_{i}, lon_{i} \rangle$ is the snapshot GPS coordinates of the taxi; (3) $s_{i}$ is the snapshot speed of the taxi; (4) $u_{i}$ indicates whether there is passengers on-board or not.

To accurately capture the detailed driving behavior of the corresponding taxi and to eliminate the negative impact of stops caused by passengers’ on and off near intersections, each trajectory $tr$ should be preprocessed to satisfy the following conditions: (1) $t_{i+1} − t_{i} \leq \epsilon$, the time gaps of consecutive records should be less than $\epsilon$ for movement consistency, and we set $\epsilon = 90$ seconds; (2) $u_{i} = u_{i+1}$, records in a trajectory should be the same status; and (3) $n \geq 2$, the number of sample records in a trajectory should not be less than 2.

Road network. The road network of a city can be represented as a directed graph $G = \langle V, E \rangle$, where $V = \{v_{1}, \ldots, v_{n}\}$ is a set of nodes, and $E = \{e_{1}, \ldots, e_{m}\}$ is a set of directed edges. Each node $v = (lat_{v}, lon_{v})$ is a GPS point representing the center location of an intersection. Each edge $e = (v_{i}, v_{j})$ represents one direction of a road segment.

A two-way road segment is modeled as a pair of symmetric edges. For example, $e = (v_{i}, v_{j})$ and $e’ = (v_{j}, v_{i})$ represent two directions of a road segment, separately.

Driving direction. The driving direction of a vehicle while crossing an intersection $v \in G$ could be represented by its incoming and outgoing road segments, i.e., $e_{s} \rightarrow e_{y}$, where $e_{s} \in In(v)$ is the incoming road segment, and $e_{y} \in Out(v)$ is the outgoing road segment. For each driving direction, it can be characterized as one of the following types: through, right-turn, left-turn or u-turn, according to its turning direction. As right-turn and u-turn traffic movements are often free from traffic signal control, we only consider through and left-turn traffic movements in this paper.

Green-start time. A green-start time of a traffic light is the start time of a green interval assigned to it. At each signalized intersection, traffic flows are partitioned into groups, and are coordinated to access the intersection by group in an alternative manner. Each group of traffic flows is allowed to cross the intersection in its corresponding green intervals.

Cycle Length. The cycle length of a traffic light is the duration between two consecutive green-start times. As the green-start event of each phase or each traffic movement is a periodic event, the minimal period of this event is the cycle length which can be calculated from a set of green-start times.

3. Preliminaries

Trajectory. A taxi GPS trajectory consists of a sequence of sampling records collected chronologically from a taxi, i.e., $tr = R_{0} \rightarrow R_{1}, \ldots \rightarrow R_{n−1}$, where each record $R_{i} = \langle t_{i}, l_{i}, s_{i}, u_{i} \rangle$, $0 \leq i < n$, contains four attributes: (1) $t_{i}$ is the timestamp at which the record was sampled; (2) $l_{i} = \langle lat_{i}, lon_{i} \rangle$ is the snapshot GPS coordinates of the taxi; (3) $s_{i}$ is the snapshot speed of the taxi; (4) $u_{i}$ indicates whether there is passengers on-board or not.

To accurately capture the detailed driving behavior of the corresponding taxi and to eliminate the negative impact of stops caused by passengers’ on and off near intersections, each trajectory $tr$ should be preprocessed to satisfy the following conditions: (1) $t_{i+1} − t_{i} \leq \epsilon$, the time gaps of consecutive records should be less than $\epsilon$ for movement consistency, and we set $\epsilon = 90$ seconds; (2) $u_{i} = u_{i+1}$, records in a trajectory should be the same status; and (3) $n \geq 2$, the number of sample records in a trajectory should not be less than 2.

Road network. The road network of a city can be represented as a directed graph $G = \langle V, E \rangle$, where $V = \{v_{1}, \ldots, v_{n}\}$ is a set of nodes, and $E = \{e_{1}, \ldots, e_{m}\}$ is a set of directed edges. Each node $v = (lat_{v}, lon_{v})$ is a GPS point representing the center location of an intersection. Each edge $e = (v_{i}, v_{j})$ represents one direction of a road segment.

A two-way road segment is modeled as a pair of symmetric edges. For example, $e = (v_{i}, v_{j})$ and $e’ = (v_{j}, v_{i})$ represent two directions of a road segment, separately.

Driving direction. The driving direction of a vehicle while crossing an intersection $v \in G$ could be represented by its incoming and outgoing road segments, i.e., $e_{s} \rightarrow e_{y}$, where $e_{s} \in In(v)$ is the incoming road segment, and $e_{y} \in Out(v)$ is the outgoing road segment. For each driving direction, it can be characterized as one of the following types: through, right-turn, left-turn or u-turn, according to its turning direction. As right-turn and u-turn traffic movements are often free from traffic signal control, we only consider through and left-turn traffic movements in this paper.

Green-start time. A green-start time of a traffic light is the start time of a green interval assigned to it. At each signalized intersection, traffic flows are partitioned into groups, and are coordinated to access the intersection by group in an alternative manner. Each group of traffic flows is allowed to cross the intersection in its corresponding green intervals.

Cycle Length. The cycle length of a traffic light is the duration between two consecutive green-start times. As the green-start event of each phase or each traffic movement is a periodic event, the minimal period of this event is the cycle length which can be calculated from a set of green-start times.

4. The Problem and Motivation

Given a set of taxi GPS trajectories $D = \{tr_{j}\}_{j=0}^{N-1}$ and an intersection $v \in G$, our problem is to learn baseline timing cycle lengths of traffic lights at this intersection. It can be solved in two major stages. The first stage is to extract a set of green-start times from trajectories, which can be referenced from [17]. The second stage is to estimate baseline cycle lengths from extracted green-start times. This paper focuses on methods of the second stage.

For each signalized intersection, sets of green-start times are extracted by driving directions individually, and we only need to select the largest set of green-start times for the estimation. Because cycle lengths of different driving directions are always the same, and the more green-start times observed the better estimation accuracy can be achieved. In addition, we conduct the cycle length estimation within each hour of day individually considering that a fixed-time intersection often uses different time plans at different times of
Given an signalized intersection $v$, let $T_G^{[t_s, t_f]} = \{t_i\}_{i=0}^{w-1}$ be the set of green-start times at a given hour $[t_s, t_f]$, and we assume that green-start times in the set are sorted in chronological order. For the reasons that green-start times extracted from taxi GPS trajectories are: (1) incomplete, for the reason that taxis are only parts of traffic flows on the road network; (2) noisy, due to the low sampling rate of trajectories and GPS positioning errors; (3) with outliers, caused by misclassification of driving behaviors of taxis; (4) with duplicated observations, as one green-start time of a specific traffic lights might be captured by more than one taxis experienced with stop-and-go movements. Each green-start time $t_j \in T_G^{[t_s, t_f]}$ can be decomposed as

$$
\tilde{t}_j = t_s + o + c \times q_j + \epsilon_j, \quad (0 \leq j < n),
$$

where $t_s$ is the reference time, $o$ is the unknown common offset with regard to $t_s$, $0 \leq o < c$, $c$ is the fixed unknown cycle length as we assume that the cycle length did not change within the hour, $q_j$ is an unknown integer, $0 \leq q_j < (t_f - t_s)/c_{\min}$, $\epsilon_j$ is an unknown estimation error, and $\epsilon_j \in (-c/2, c/2)$. The $c_{\min}$ is the lower bound of the real cycle length which is often specified in official traffic signal design manual.

From the Eq. (1), we can observe that the cycle length estimation from $T_G^{[t_s, t_f]}$ is very similar with the period estimation problem considered by Fogel et al. [12], Casey et al. [13], Sidiroopoulos et al. [14], and Clarkson et al. [15]. However, our problem has the following characteristics compared with them. Firstly, the number of observations about the periodic event is limited, and is often less than 60 (one per minute). One reason is that multiple timing plans are often used at fixed-time signalized intersections, and each timing plan only last for several hours each day. The other reason is that the number of observations depends on the number of taxis which is hard to increase. Secondly, the precision requirement of cycle length estimation is not as high as that of the period estimation in the previously mentioned works, and estimates being accurate to a second is acceptable for our problem. Thus, the cycle length estimation problem prefers algorithms which could achieve high accuracy with less observations.

According to Yu and Lu [18], the period estimation problem could be modeled as a general approximate greatest common divisor (AGCD) problem. The AGCD problem was firstly proposed by Howgrave-Graham et al. [19] in the filed of cryptanalysis of RSA, which is to recover the unknown approximate common divisor $c$ given any two large integers $x_1$ and $x_2$, where $x_1 = q_1 \times c + \epsilon_1$ and $x_2 = q_2 \times c + \epsilon_2$, $q_1$ and $q_2$ are unknown multiplicative factors, $\epsilon_1$ and $\epsilon_2$ are unknown error terms which are much smaller than $c$. It has been proved in [19] that the problem is solvable under the condition that $|\epsilon_i| < \sqrt{c}$, $(1 \leq i \leq 2)$. One of the most efficient methods for solving the problem is the GCD exhaustive search algorithm on the error space [20]. Specifically, the algorithm enumerates all possible integer error $\epsilon$ in the error space $[-\epsilon_{\max}, \epsilon_{\max}]$, computes the greatest common divisor (GCD) of $(x_1 - \epsilon)$ and $(x_2 - \epsilon)$, and finally returns the largest GCD as the estimate.

Motivated by the error space GCD exhaustive search algorithm [20], we propose a family of algorithms, named as the Most-Frequent Approximate Common Divisor (MFAGCD) algorithms, to solve the cycle length estimation problem. Our intuition is that when the error term of each time was not correctly removed, the resulting divisor would most probably be a small value which can be filtered out according to the prior knowledge about the range of the real cycle length; whereas when the error term of each time was correctly removed, the resulting divisor will be the cycle length or a multiple of the cycle length.

5. The MFAGCD Algorithms

The MFAGCD algorithms consist of two major steps. The first step is to compute candidate periods (or cycle lengths) and count their occurrence frequencies. The challenge faced in the first step is how to effectively eliminate the common offset while avoiding the impact of outliers. The main difference between our basic MFAGCD algorithm and the robust MFAGCD algorithm is that different strategies are chosen to calculate candidate periods in the first step. The second step is to find the final period estimate from these candidate periods according to their occurrence frequency distribution. Although observations considered in this paper are integers, the MFAGCD algorithms could also be easily extended to solve the problem considering float observations by scaling float observations to integers.

5.1 The Basic MFAGCD Algorithm

To estimate the cycle length $c$ at the hour $[t_s, t_f]$ from a given set of green-start times $T_G^{[t_s, t_f]} \subset T_G$, we need firstly eliminate the unknown common offset $o$. In the simplest way, the unknown common offset could be eliminated by subtracting the smallest element in the set of green-start times. Then, the cycle length estimation problem is to estimate the hidden approximate common divisor $c$ from $T$.

**Algorithm 1: The Basic MFAGCD Algorithm**

| Step | Description |
|------|-------------|
| 1    | $T = \{t_i - t_0\}_{i=1}^{n}, t_i \in T_G^{[t_s, t_f]}$; eliminate the common offset of each green start time in $T_G$ |
| 2    | $C \leftarrow \emptyset$; set the lower bound of the cycle time |
| 3    | for each $(t_i, t_j)$ where $t_i, t_j \in T$ and $t_i < t_j$ do |
| 4    | $C' \leftarrow \text{ApproxGCD}(t_i, t_j, \delta, c_{\min})$ |
| 5    | $C \leftarrow C \cup C'$ |
| 6    | count the frequency for each divisor in $C$ |
| 7    | $\hat{c} \leftarrow$ find the divisor with the highest frequency in $C$ |
| 8    | return $\hat{c}$ |
basic MFAGCD algorithm was firstly proposed in our previous paper [18]. For comparison, we list the basic MFAGCD in Algorithm 1.

The procedure ApproxGCD() is to calculate all possible greatest common divisors by exhaustively enumerate all possible errors. The function GCD() is to calculate the greatest common divisor of two given integers using the famous Euclidean algorithm [21, chapter 4.5.2, page:337]. The computational complexity of the basic MFAGCD algorithm is $O(n^2 \cdot \delta^2 \cdot \log c)$, where $n$ is the number of green-start times, $\delta$ is the error bound which can be learned from the green-start times, and $c$ is the cycle length.

### Procedure ApproxGCD($t_1, t_2, \delta, c_{\min}$)

**Input:** $t_1$ and $t_2$ are integers, $\delta$: the error bound.

**Output:** $C$: a set of approximate greatest common divisors

1. foreach integer $\delta_1 \in [-E, E]$ do
2.     foreach integer $\delta_2 \in [-E, E]$ do
3.         $\hat{c} \leftarrow \text{GCD}(t_1 + \delta_1, t_2 + \delta_2);
4.         if $\hat{c} \geq c_{\min}$ then $C \leftarrow C \cup \{\hat{c}\}$
5. return $C$

5.2 The Robust MFAGCD Algorithm

In the basic algorithm, the first green-start time is utilized to eliminate the common offset. In the situation that outliers exist in the set of green-start times, the accuracy performance of the basic algorithm would be significantly deteriorate if the first minimum green-start time is happened to be an outlier. In addition, after the first elimination step, the error space could be enlarged and errors will not follow identical independent distribution. To further mitigate the negative impact of outliers on the common offset elimination step, we propose a robust MFAGCD algorithm. The algorithm is depicted in details in Algorithm 2.

In the robust MFAGCD algorithm, we enumerate all three elements and chose the smallest one among the three for eliminating the offset, instead of fixing the first element for eliminating the offset. For each three green-start times in the given set, we try to de-noise the selected times through enumerating all possible errors in the error space, and conduct the Euclidean algorithm over these de-noised times. In each three selected times, the smallest one is only used for eliminating the common offset. After enumerating all three combinations of green start times, we count the occurrence for each collected divisor. Finally, we return the divisor with the highest occurrence times as the estimate.

The computational complexity of the MFAGCD algorithm is $O(n^3 \cdot \delta^3 \cdot \log c)$, where $n$ is the number of green-start times, $\delta$ is the error threshold of green-start time estimation, $c$ is the cycle length, and $\log c$ is the computational cost the the famous Euclidean algorithm for computing the greatest common divisor. In the cycle length estimation problem, all the three parameters are small integers.

The value of cycle length $c$ often varies from 30 to 240, and $\delta$ is often much smaller than $c$. As we conducted the cycle length estimation hourly, the size of $n$ is also limited. Therefore, the increasing computational cost of the robust MFAGCD algorithm is acceptable in practice, and also the efficiency of algorithms is not the focus of this paper.

### Algorithm 2: The Robust MFAGCD Algorithm

**Input:** $TG_{v,t} = \{\delta_{j}\}_{j=0}^{N-1}$; a sequence of green start time estimations in chronological order, and $n > 3$; $\delta$: the estimation error bound; $c_{\min}$: the lower bound of the cycle time; $c_{\max}$: the upper bound of the cycle time.

**Output:** $\hat{c}$: the cycle length estimate

1. $C \leftarrow \emptyset$ ; // a set of candidate cycle lengths, initialized as empty
2. foreach $(t_i, t_j, t_k)$ where $t_i \in TG_v, t_j \in TG_v, t_k \in TG_v$, and $t_j < t_i < t_k$ do
3.    forall $\delta' \in [-\delta, \delta]$ do
4.         $t = t_i + \delta'$;
5.         $C' \leftarrow \text{ApproxGCD}(t_j - t, t_k - t, \delta, c_{\min})$;  
6.         $C \leftarrow C \cup C'$;
7.     count the frequency for each divisor in $C$;
8.     $\hat{c} \leftarrow$ find the divisor with the highest frequency in $C$;
9. return $\hat{c}$

6. Experiments

To evaluate the effectiveness of the proposed method, four algorithms are selected for the comparison using both simulation datasets and observational datasets extracted from real taxi GPS trajectories.

6.1 The Compared Algorithms

**Periodogram.** The periodogram-based algorithm [12] tries to solve the period estimation problem in frequency domain, and is to search for the right frequency resulting in the highest periodogram among the given frequency range.

**MSSNE.** The algorithm was proposed by Fayazi et al. [5], which treated the cycle length estimation problem as the optimization problem of minimizing the sum of squared normalized errors (MSSNE). Specifically, given a set of green start time estimates $TG_v = \{\delta_j\}_{j=0}^{N-1}$, and the range of feasible cycle length $[c_{\min}, c_{\max}]$, they defined the cycle length estimation problem as:

$$\hat{c} = \arg\max_{c \in [c_{\min}, c_{\max}]} \sum_{j=0}^{N-1} f(t_{j+1} - t_j, c),$$  \hspace{1cm} (2)

where the function $f(\cdot, \cdot)$ is defined as

$$f(x, y) = \left( \frac{x - y \cdot \lfloor \frac{x}{y} \rfloor + 0.5}{y/2} \right)^2,$$  \hspace{1cm} (3)

which is to calculate the squared normalized error. Then, they introduced the cycle length exhaustive search algorithm
to find the optimal estimate. In particular, the algorithm therein tries all feasible cycle lengths and selects the one minimizing the sum of squared normalized errors as the estimate. We figure out that the MSSNE algorithm [5] is a variant of the SLS2-ADJ algorithm [14]. Their major difference is that Fayazi et al. [5] used the squared normalized errors rather than the squared errors used by Sidiropoulos et al. [14], whereas this difference does not make much difference on their accuracy performance.

**iMSSNE.** The improved MSSNE (iMSSNE) algorithm is an improved version of the MSSNE algorithm, and its optimization function is defined as

$$\hat{c} = \arg \max_{c \in [c_{\min}, c_{\max}]} \sum_{0 \leq j < n-1} f(t_k - t_j, c). \quad (4)$$

Apart from periodogram-based algorithm [12], several algorithms have been proposed to estimate the period from noisy and sparse observations in signal processing area, including SLS2-based algorithms [14], [15], and LLS-based algorithms [15]. According to McKilliam et al. [22], the periodogram-based algorithm has the highest accuracy among them, so we only add the periodogram algorithm into our comparison for simplicity. For the MEA algorithm [13] and the algorithm proposed by Li et al. [16], as their estimation was less accurate than that of the basic MFAGCD algorithm [18], we do not compare with them again in this paper. In addition, the method proposed by Chuang et al. [7] requires dense green-start times for the cycle length estimation, which is not suitable for handling sparse observations, and therefore it is not suitable for the sparse situation considered in the paper.

6.2.2 Accuracy of Cycle Length Estimation

In order to get the ground truth about traffic signal settings at the three selected intersections, we recorded a few days videos about the traffic lights at the intersections, and extracted the real setting information from these videos. The cycle length estimation was conducted hourly from 07:00 am to 19:00 pm of each day from Dec. 28, 2014 to Jan. 10, 2015 (172 hours). For the compared algorithms, we assume that the cycle length ranges from 30 seconds to 240 seconds, which is reasonable for practical timing settings.

Figure 1 shows the accuracy results of the five algorithms using the real world taxi trajectory dataset of the three selected intersections. From the figure we can see that: (1) the rMFAGCD algorithm achieves higher estimation accuracy than the other algorithms in most cases; (2) the rMFAGCD algorithm performs more stable than the other algorithms as it can achieve high accuracy performance (> 80%) in all the three selected intersections. Thus, we can conclude that the proposed rMFAGCD algorithm has better error and outliers tolerant capability than that of the other
algorithms, as the extracted green-start times from taxi GPS trajectories are noisy, incomplete, with outliers and with duplicated observations.

In addition, the accuracies of the rMFAGCD algorithm for intersection 2 and 3 are higher than that for intersection 1. Because both intersection 2 and 3 are larger than intersection 1, and therefore more taxis crossed the two intersections, resulting in more GPS trajectories.

6.3 Comparison with Simulation Datasets

To further evaluate the robustness of the proposed algorithm to outliers, we design a generator to simulate the generation of sparse and noisy observation datasets. It can generate different types of simulation datasets via setting the four parameters: the cycle length $c$, the total number of cycles $n$ within the whole observation process, the observed cycles $m$, and the standard deviation $\sigma$ of the Gaussian noise distribution. In our experiments, we fix the cycle length $c = 90$ and the total number of cycles $n = 40$. In order to make the size of simulation data be similar with that of the real data, we let the number of observations $m$ vary from 4 to 20. Let $\sigma$ varies from 1 to 6 in our simulation. For each set of parameters, we run the estimation for 1000 times and then average the results.

Figure 2 demonstrates the results of these algorithms on simulated datasets without outliers. From Fig. 2 we can see that the rMFAGCD algorithm has similar accuracy with the iMSSNE algorithm. The accuracy of the rMFAGCD algorithm is more stable than the other algorithms with the

**Fig. 2** Accuracy comparison using simulation datasets without outliers.

**Fig. 3** Accuracy comparison using simulation datasets with outliers.
variations of noise threshold $\delta$ and number of observations $m$. The periodogram algorithm shows low accuracy when the noise threshold $\delta$ is small, and the MSSNE algorithm performs bad when the noise threshold grows large.

Figure 3 plots the estimation accuracy of these algorithms on simulated datasets with outliers. In Fig.3, there is evidence that the accuracy of the rMFAGCD algorithm is better than that of the other algorithms in most situations, especially the MSSNE algorithm. For the MFAGCD and rMFAGCD algorithms, negative effects caused by outliers could be mitigated by increasing the number of observations, and the MFAGCD algorithm requires more extra number of observations than that of the rMFAGCD algorithm. The MSSNE algorithm is very sensitive to outliers, and even in the situation that the number of observations grows. Thus, the rMFAGCD algorithm has better outliers-tolerant capability than the other algorithms.

7. Conclusions and Future Works

In this paper we formulate the baseline cycle lengths discovery for traffic signals as the problem of period estimation from sparse and noisy observations, and design a robust MFAGCD algorithm to solve the problem. The proposed algorithm has better accuracy than that of other algorithms for the reason that it does not rely on the requirement that errors in observational datasets are normally distributed as other algorithms do. Although the computational complexity of the robust MFAGCD algorithm is higher than the basic MFAGCD algorithm and other algorithms, the accuracy performance of the robust MFAGCD to noise and outliers is the steady and reliable than the other compared algorithms. In addition, the extra computational cost of the robust MFAGCD algorithm is acceptable in practice. Because the number of observations used to estimate the cycle lengths is often very small ($5 \sim 60$).

In the future, we plan to learn green lengths for each timing plan based on the cycle length estimate, and to further determine the shift times of timing plans at each day for fixed-time signalized intersections that use multiple timing plans. This work is supported by National Natural Science Foundation of China under Grant 61402418, 61503342, 61672468, 61602418, MOE (Ministry of Education in China) Project of Humanity and Social Science under Grant 12YJCZH142, 15YJCZH125, Zhejiang Provincial Natural Science Foundation of China under Grant LY15F020013, and Social development project of Zhejiang provincial public technology research under Grant 2017C33054.

Acknowledgments

This work is supported by National Natural Science Foundation of China under Grant 61402418, 61503342, 61672468, 61602418, MOE (Ministry of Education in China) Project of Humanity and Social Science under Grant 12YJCZH142, 15YJCZH125, Zhejiang Provincial Natural Science Foundation of China under Grant LY15F020013, and Social development project of Zhejiang provincial public technology research under Grant 2017C33054.

References

[1] B. Asadi and A. Vahidi, “Predictive Cruise Control: Utilizing Upcoming Traffic Signal Information for Improving Fuel Economy and Reducing Trip Time,” IEEE Trans. Control Syst. Technol., vol.19, no.3, pp.707–714, May 2011.
[2] M. Kerper, C. Wewetzer, and M. Mauve, “Analyzing vehicle traces to find and exploit correlated traffic lights for efficient driving,” 2012 IEEE Intelligent Vehicles Symposium (IV), pp.310–315, June 2012.
[3] Y. Zhao, Y. Zhang, T. Yu, T. Liu, X. Wang, X. Tian, and X. Liu, “CityDrive: A map-generating and speed-optimizing driving system,” 2014 Proc. IEEE INFOCOM, pp.1986–1994, April 2014.
[4] M. Kerper, C. Wewetzer, A. Sasse, and M. Mauve, “Learning Traffic Light Phase Schedules from Velocity Profiles in the Cloud,” 2012 5th International Conference on New Technologies, Mobility and Security (NTMS), pp.1–5, May 2012.
[5] S.A. Fuyazi, A. Vahidi, G. Mahler, and A. Winckler, “Traffic Signal Phase and Timing Estimation From Low-Frequency Transit Bus Data,” IEEE Trans. Intell. Transp. Syst., vol.16, no.1, pp.19–28, Feb. 2015.
[6] M. Ramezani and N. Geroliminis, “Queue Profile Estimation in Congested Urban Networks with Probe Data,” Computer-Aided Civil and Infrastructure Engineering, vol.30, no.6, pp.414–432, June 2015.
[7] Y.-T. Chuang, C.-W. Yi, Y.-C. Tseng, C.-S. Nian, and C.-H. Ching, “Discovering Phase Timing Information of Traffic Light Systems by Stop-Go Shockwaves,” IEEE Trans. on Mobile Comput., vol.14, no.1, pp.58–71, Jan. 2015.
[8] V. Protschky, C. Ruhlhammer, and S. Feit, “Learning Traffic Light Parameters with Floating Car Data,” 2015 IEEE 18th International Conference on Intelligent Transportation Systems (ITSC), pp.2438–2443, Sept. 2015.
[9] S. Gowrishankar and D.B. Work, “Estimating traffic control strategies with inverse optimal control,” 2013 16th International IEEE Conference on Intelligent Transportation Systems - (ITSC), pp.2304–2309, Oct. 2013.
[10] M.R. Gahrooei and D.B. Work, “Inferring Traffic Signal Phases From Turning Movement Counters Using Hidden Markov Models,” IEEE Trans. Intell. Transp. Syst., vol.16, no.1, pp.91–101, Feb. 2015.
[11] Y. Zhu, X. Liu, M. Li, and Q. Zhang, “POVA: Traffic Light Sensing with Probe Vehicles,” IEEE Trans. Parallel Distrib. Syst., vol.24, no.7, pp.1390–1400, July 2013.
[12] E. Fogel and M. Gavish, “Parameter estimation of quasi-periodic sequences,” Proc. 1988 International Conference on Acoustics, Speech, and Signal Processing, ICASSP ’88, pp.2348–2351 vol.4, April 1988.
[13] S.D. Casey and B.M. Sadler, “Modifications of the euclidean algorithm for isolating periodicities from a sparse set of noisy measurements,” IEEE Trans. Signal Process., vol.44, no.1, pp.58–71, Jan. 2015.
[14] I.V.L. Clarkson, “Approximate maximum-likelihood period estimation from sparse, noisy timing data,” IEEE Trans. Signal Process., vol.56, no.5, pp.1779–1787, May 2008.
[15] Z. Li, J. Wang, and J. Han, “Mining periodicity for sparse and incomplete event data,” Proc. 18th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD ’12, Beijing, China, Aug. 2012.
[16] J. Yu and P. Lu, “Learning traffic signal phase and timing information from low-sampling rate taxi GPS trajectories,” Knowledge-Based Systems, vol.110, pp.275–292, Oct. 2016.
[17] J. Yu and P. Lu, “AGCD: A robust periodicity analysis method based on approximate greatest common divisor,” Frontiers of Information Technology & Electronic Engineering, vol.16, no.6, pp.466–473, 2015.
[18] N. Howgrave-Graham, “Approximate integer common divisors,” Revised Papers from the International Conference on Cryptography and Lattices, CaLC ‘01, London, UK, UK, vol.2146, pp.51–66,
Juan Yu received the M.S. degree in Computer Science from Zhejiang Normal University in 2010, and received the PhD degree in Computer Science from Fudan University in 2017. She joined the Smart City Research Center, Hangzhou Dianzi University, Hangzhou, China. Her research interests include data mining and data privacy preservation.

Peizhong Lu received the M.S. degree in mathematics from the University of Information Engineering, Zhengzhou, China, in 1987, and the Ph.D. degree in mathematics from the Chinese Academy of Sciences, Beijing, China, in 1998. He is a professor in computer science, Fudan University, Shanghai, China. His research interests include error-correcting codes and information security.

Jianmin Han received the M.S. degree in Computer Science from DaQing Petroleum Institute in 1992, and the Ph.D. degree in Computer Science from East China University of Science and Technology in 2009. He is a professor in the School of Mathematics-Physical and Information Engineering, Zhejiang Normal University. His research interests include data privacy and game theory.

Jianfeng Lu received his B.S. degree Computer Science from Wuhan University of Science and Technology in 2005, and the PhD degree in Computer Science from Huazhong University of Science and Technology in 2010. He is an associate professor in the School of Mathematics-Physical and Information Engineering, Zhejiang Normal University. His research interests include access control and game theory.

---

[20] Y. Chen and P.Q. Nguyen, “Faster Algorithms for Approximate Common Divisors: Breaking Fully-Homomorphic-Encryption Challenges over the Integers,” Advances in Cryptology – EUROCRYPT 2012, Lecture Notes in Computer Science, vol.7237, pp.502–519, April 2012.

[21] D.E. Knuth, The Art of Computer Programming, Volume 2: Seminumerical Algorithms, 3rd ed., Addison Wesley Longman, 1998.

[22] R.G. McKilliam, I.V.L. Clarkson, and B.G. Quinn, “Fast Sparse Period Estimation,” IEEE Signal Process. Lett., vol.22, no.1, pp.62–66, Jan. 2015.

[23] J. Yuan, Y. Zheng, C. Zhang, X. Xie, and G.-Z. Sun, “An Interactive-Voting Based Map Matching Algorithm,” Proc. 11th International Conference on Mobile Data Management, Washington, DC, USA, pp.43–52, 2010.

Springer-Verlag, 2001.