IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received May 17, 2019, accepted June 22, 2019, date of publication June 26, 2019, date of current version July 12, 2019.

*Digital Object Identifier 10.1109/ACCESS.2019.2925134*

# Phase Timing Optimization for Smart Traffic Control Based on Fog Computing

**CHAOGANG TANG[1], SHIXIONG XIA[1], CHUNSHENG ZHU[2], AND XIANGLIN WEI[3]**
[1]School of Computer Science and Technology, China University of Mining and Technology, Xuzhou 221000, China
[2]Department of Electrical and Computer Engineering, The University of British Columbia, Vancouver, BC V6T 1Z4, Canada
[3]Nanjing Telecommunication Technology Research Institute, Nanjing 210007, China

Corresponding author: Shixiong Xia (xiasx@cumt.edu.cn)

This work was supported by the Fundamental Research Funds for the Central Universities under Grant 2019XKQYMS46.

**ABSTRACT** The explosive growth of motor vehicles in urban areas has heavily burdened the existing transportation systems. Consequently, there are recently emerging smart transportation paradigms that aim to ease urban transportation pressure. Smart traffic signal control, which is considered as one of the breakthrough technologies in smart transportation paradigm, has ushered significant academic and industry efforts for its considerable benefits. The state-of-the-art proposals usually rely on a centralized infrastructure with powerful computing abilities to deal with a large amount of different traffic data. However, the centralized processing approach is often hindered by a long and even unbearable response latency, restricting its wide deployment and applications in the real world. To overcome this latency-related issue and achieve near-optimal traffic signal control in nearly real time, we have proposed a non-centralized approach as well as a fog computing-based architecture, and thus traffic data can be handled in a smarter way. To be specific, the traffic data are processed right at where it was generated, i.e., at the edge rather than at a centralized facility. In this paper, the phase timing of a single intersection is to be handled real-time by a local fog node with a genetic optimization algorithm, and the task for the regional optimization is offloaded to the centralized cloud and executed. The simulation experiments are conducted on the simulator to evaluate the performance of our proposal, and the results confirm that the proposed architecture and algorithm have significant improvement on the average duration time compared to existing approaches.

**INDEX TERMS** Fog computing, smart transportation, genetic algorithm, signal control.

## I. INTRODUCTION

The explosive growth of motor vehicles in the wave of urbanization leads to a series of traffic pressures and issues in current cities. Significant academic and industry efforts have been spent on mitigating traffic flow and easing traffic pressures in the past few decades. Traffic signal control has benefitted a lot from the Information and Communication Technique (ICT), and quickly developed over the past few years. Typical traffic signal control has gone through several rounds of evolvements, from original fixed-time control systems to actuated or adaptive control systems, to handle the increasingly complicated and heterogeneous traffic data in real time. In fixed-time control systems, the sequence of phases is consistently fixed on one traffic signal cycle, while the timing in one cycle is pre-set based on statistical traffic flow data during different periods of one day. The principle behind this strategy is simple and easy to deploy without investing in new infrastructure. However, this kind of control system does not take into account the stochastic nature of traffic flow and unpredictability of vehicle arrival at intersections.

Loop detectors deployed near the intersections can detect and track traffic information such as the number of incoming vehicles. Thus the actuated control systems, to some extent, reflect the stochasticity of vehicles and improve the performance of traffic control, especially compared to fixed-time control systems. However, in the actuated systems, real-time data tracking traffic flow such as vehicle speed and acceleration can not be detected directly, although it plays an important role in optimizing Urban Traffic Control (UTC) and predicting the stochastic nature of traffic flow.

To further improve the performance of traffic control, adaptive traffic systems incorporate several technologies, such as loop detectors, camera detectors, infrared ray, radar,

The associate editor coordinating the review of this manuscript and approving it for publication was Antonella Longo.

to detect the real-time traffic information. These adaptive traffic control systems need real-time traffic data gathering and monitoring, and recent development of the integrated architecture such as Vehicular Ad-Hoc Networks (VANETs) and Internet of Things (IoT) techniques strive to make that possible.

Enhanced connection with sensing and collection capabilities increase the amount of real-time traffic flow data. Hence, there needs to be a way to optimize efficiently the phase timing based on the gathered traffic data. In addition, existing control strategies need to overcome the long decision and response latency from data processing. In fact, the traffic flow may have already changed greatly when the result of the optimization decision is returned to the traffic signal controller. As a result, the resulting optimal signal control strategy on traffic light scheduling cannot be suitable for the current traffic flow status quo. The challenge for centralized computation infrastructure lies in the data processing and the need for an instantaneous response for adaptive signal control systems.

To overcome this problem, the newly proposed fog computing paradigm, is a potential solution [1], [2]. Fog computing brings computation and storage resources to the edge of network, enabling highly computationally intensive applications to run while meeting strict delay requirements. The road situation (e.g, dry or wet, under construction, traffic accident), weather situation (e.g., sunny, rainy), vehicle states (location, speed, acceleration, etc.), and intersection information (e.g., the length of queue waiting at the intersection), can all be collected and processed by fog nodes in real time. Then the traffic signal controller can receive an instant response (e.g., extending the green time or starting new phase timing) to alleviate the traffic congestion and further ensure the driving safety.

We propose a fog computing based traffic signal control architecture. The collected traffic data will be mainly processed where the traffic data is generated, using efficient and fast algorithms to make real-time decisions.

The work presented in this paper has extended our previous work published in the conference [3]. The work in [3] has only introduced an architecture which is based on the fog computing to achieve real-time traffic control. It classifies the architecture into three layers and details the corresponding functionality of each layer. However, in our previous paper, neither mathematical formulation nor performance evaluation on this architecture was provided. So in contrasting fashion, in this paper, we not only provide the mathematical formulation on the phase timing optimization, but also adopt a genetic algorithm to solve this problem. The proposed phase timing strategy is also evaluated by an open-source simulator.

In specific, our main contributions are as follows:

1. We propose a fog computing based smart traffic signal control architecture, in which the traffic data are collected by various sensors, processed mainly at the edges (i.e., fog nodes) instead of remote cloud centers to reduce the latency as much as possible. The traffic data can be uploaded to the cloud center to ease the burden of fog nodes if necessary. Additionally, cloud computing mainly focuses on coordinating regional traffic control.
2. We propose a near real-time strategy to schedule the phase timing for signalized intersection based on the genetic algorithm.
3. We integrate the phase timing algorithm into existing traffic simulator and conduct a series of experiments to investigate the efficiency of our smart traffic control strategy. The simulation results show great potentials and prospects.

## II. RELATED WORKS

The majority of current literature is dedicated to achieve the real-time traffic signal control to mitigate the traffic flow. These works usually focus on the traffic signal configuration optimization, which includes the optimization of the phase sequences and the phase timing duration. They usually assume that the length of the traffic light cycle is unchangeable and the phase sequence in one light cycle is fixed. Most of existing works evaluate the performance of smart traffic control algorithms and strategies by means of microscopic traffic flow simulators (e.g., SUMO and PARAMICS ), and choose average travel time, average number of stops, and the waiting queue length at the intersection as their optimization objectives. But, to design an efficient traffic controlling strategy is complex because there is a number of issues that need to be addressed. For example, challenges and issues include the unpredictability of traffic flow, the fusion of traffic data with IoT (considering the heterogeneity of vehicles and V2X communication techniques).

Academic research mostly focuses on designing adaptive and real-time algorithms to achieve the traffic signal control [4]–[8]. For example, an algorithm presented in [8] combines the genetic algorithm with machine learning algorithm to improve the performance of traffic control strategy. The optimization includes two phases. In phase one, it optimizes the phase timing duration in the traffic light cycle by genetic algorithm, and in phase two, it predicts the next phase timing duration by machine learning algorithms. However, this approach has serious time overheads and hence violates the principle of timeliness necessary for smart traffic control.

Krajzewicz *et al.* [9] present an agent based traffic light control algorithm to solve jams at intersections. They trigger the optimization depending on the queue length of different lanes. If the queue length exceeds the specified threshold, they increase the phase length one by one. The model is simple without mathematical formulation. It is not qualified enough to cope with increasingly complicated smart traffic scenarios. Guo *et al.* [5] present a scheme for traffic timing optimization under user equilibrium traffic. They model the optimization as a multi-dimensional search problem and simulate it in PARAMICS using a genetic algorithm.

Recently, fog computing has been introduced to smart transportation and smart cities [10]–[13]. Works [14]

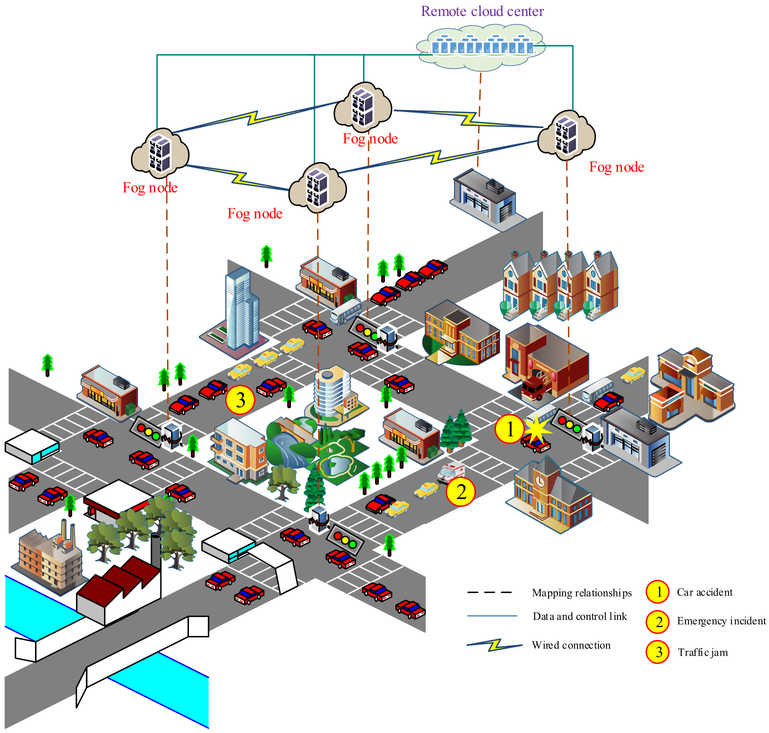



**FIGURE 1.** An application of fog computing in traffic signal control.

consider the road side unit (RSU) as a special fog node in vehicular fog computing. On one hand, RSU communicates with vehicles for information exchange and data propagation; On the other hand, it takes charge of data processing and algorithm execution for determining the optimal task scheduling plan.

Many works use merging V2X communication techniques to achieve a better performance of traffic control under assumption of certain market penetration. For instance, Priemer and Friedrich [15] propose to collect speed, acceleration and heading of vehicles based on V2I communication techniques, with the aim to improve the efficiency of the traffic control strategies at the intersections. Feng *et al.* [16] present a real-time and adaptive signal phase scheduling algorithm with V2I/V2V communication techniques to minimize the total delay minimization and the the queue length at intersections.

Moreover, Hou *et al.* in [12] proposed an architecture called vehicular fog computing (VFC), which takes full advantage of a collaborative multitude of end-user clients and nearby edge devices to perform the communication and computation tasks. Specifically, each moving or stopped vehicle can act as a fog node to collect traffic data and contribute to VFC. Works [10], [17] combine fog computing with VANETs, explore possibilities and further discuss the characteristics of fog computing and services in fog computing platform for VANETs. Challenges, open issues and future research directions are also put forth and discussed.

However, most of these works mainly focus on resource sharing among vehicles instead of traffic signal control. We also notice from the aforementioned works that the signal control strategies suffer from the running time of highly computational algorithms for phase timing optimization. The traffic control strategies fail to achieve expected performance because of long response latency. Furthermore, traffic control strategies that use fog computing in the intelligent transportation system focus on improving the road safety and traffic efficiency. Less attention has been paid, however, to the traffic light optimization with the aid of fog computing. Due to low response latency, location awareness and geographic distribution, fog computing can be considered as one of potential solutions to traffic phase timing optimization.

As far as we know, this is the first attempt to construct an architecture for traffic signal optimization based on fog computing while integrating a genetic algorithm based strategy into SUMO to investigate the performance of our fog computing based smart traffic control.

## III. APPLICATION SCENARIO AND SYSTEM ARCHITECTURE

In smart transportation, the wide deployment of various sensors enables the real-time monitoring and collection of massive amount of heterogeneous traffic data. Traffic data helps intelligent signal control systems design suitable signal controlling strategy and determine optimal phase timing plans. With the help of either infrastructure-based sensors (e.g., loop detectors, traffic monitoring cameras or radio) or vehicular networking technologies (e.g., V2V, V2I), fog nodes are in charge of traffic flow data retrieval.

A scenario is depicted in Fig.1 where fog computing drives decision making. Devices are equipped with computing facilities and located at the edge of the network, and they function as fog nodes in fog computing. For instance, RUS deployed for communicating with connected vehicles can be augmented to serve as fog node. In the near future, the smart traffic lights with computing powers can also be used as fog nodes.

In this scenario, near the traffic signal controller, we use special computer servers with appropriate computational abilities to act as fog nodes. This guarantees that the on-site real-time traffic data can be handled efficiently in real-time. In addition to applying fog computing to traffic phase timing optimization, two other common incidents are also exemplified in the scenario: one, car collision, and two, emergency events.

This scenario demonstrates several handling mechanisms which different incidents can trigger. For example, as the number of vehicles at an intersection increases, traffic congestion occurs. The traffic flow information is then collected by various sensors such as infrastructure-based sensors and vehicular networking technologies. Based on the traffic data, the fog node can cooperate with the traffic signal controller and optimize the phase timing to alleviate the traffic congestion. In addition, fog computing can be enhanced by adding a remote cloud, functioning as the cloud computing layer. Cloud computing has more powerful computing resources but at the expense of longer latency.

Besides traffic congestion, other incidents such as car collisions can also be recorded by nearby traffic-monitoring cameras or passing connected vehicles. They can forward this incident to nearby fog nodes in the case of an emergency and plan adjustments. First, the degree of collision-caused traffic is investigated, and based on the results of the analysis, the fog node intelligently decides whether or not to notify other passing vehicles of the incident with a much wider communication range compared to the equipped vehicles. In case of serious car accidents, fog nodes can further seek help directly from the traffic management center.

Lastly, one of the most common emergency incidents is to take patients to hospital by ambulance. How to arrive at hospital in the least amount of time is a life-and-death matter. In this scenario, the fog computing and cloud computing can cooperate together to guarantee that the ambulance arrives at hospital as quickly as possible. For instance, fog computing can determine temporary phase timing plan to ensure that the ambulance can pass the current intersection without waiting for traffic lights. After that, fog computing can restore regular phase timing. From a global viewpoint, the cloud computing can calculate the globally optimal driving route while taking into account green wave, the congestion level of different road segments, and other factors. The cooperation of fog computing and cloud computing can to a great extent improve the running efficiency of ambulances.

According to these descriptions, we can see that the proposed traffic signal control architecture in this paper includes three layers: cyber physical layer, fog computing layer and cloud computing layer. The cyber physical layer is to collect traffic flow data while fog computing layer and cloud computing optimize phase timing and determine the traffic signal controlling strategies. We will elaborate on these three layers in the next subsections.

### A. CYBER PHYSICAL LAYER

This layer is a densely distributed ecosystem, which consists of the infrastructure-based loop detectors and various video surveillance systems. Connected vehicles with mounted communication devices (e.g., On-Board Units, OBUs), and RSUs can communicate with each other for data collecting and forwarding.

Huge amounts of traffic data are constantly collected in this layer and then disseminated to the fog computing layer. However, the heterogeneity of traffic data from diverse sensors and actuators poses new issues concerning data extraction, storage and processing in the fog computing layer. Redundant information exists in the traffic information flow, since it can be captured by different sensors at the same time. Useless or redundant information should be identified and excluded as much as possible to conserve storage resources and improve traffic data handling. However, identical structure definitions for differently captured data (through connected vehicles or traffic monitoring cameras) may improve data extraction and processing, but produce more data redundancy and storage resources waste.

Another concern in this layer is the communication between different sensors and actuators in addition to 4G, WiFi, WLAN, ZigBee, Bluetooth, and other short-range communication techniques (DSRC) [18]. All of them can offer both V2V and V2I communications. Furthermore, DSRC (802.11p) enables fast data exchange among entities (vehicles and RUSs) at the same channels with no need for preparation of association and authentication, thus improving the flexibility of telecommunication service provisioning.

### B. FOG COMPUTING LAYER

This layer is composed of computing nodes with powerful computational and storage resources near the traffic signal controller. Fog nodes oversee data handling that includes data storage, analysis, and incident handling such as traffic signal optimization and emergency call services. The architecture for data and incident handling in the fog computing layer is depicted in Fig. 2.

The architecture is made up of several modules that serve different functions to optimize phase timing and determine traffic control strategies. Specifically, the model called Traffic Information Database from Fig. 2 stores the traffic data uploaded to the fog computing layer. The Orchestrator model is responsible for evaluating the traffic status and trigger different handling mechanisms. If traffic congestion occurs, a procedure for local optimization can be started. The modules surrounded by the dashed line box oversee local optimization. These models retrieve the traffic data from the Traffic Information Database and use appropriate algorithms to schedule phase sequence and optimize the phase timing. The resulting phase sequence and phase timing plans are

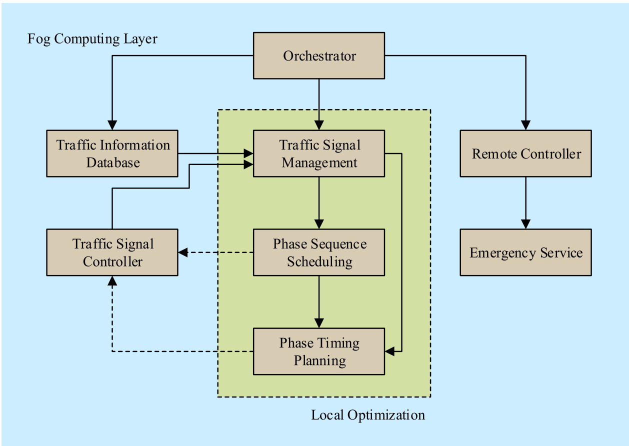



**FIGURE 2. Architecture of fog computing layer.**

forwarded to the model called Traffic Signal Controller model which is then responsible for adjusting the traffic lights, by extending the green time or resetting the phase timing duration in the next phase. In case of vehicle collisions or emergency incidents, the Orchestrator model will seek help directly from human services in the traffic management center.

The benefits of utilizing fog computing in the intelligent traffic signal control systems is to reduce the response delay significantly, compared to other existing controlling strategies. Thus, the traffic flow can be mitigated as much as possible.

### C. CLOUD COMPUTING LAYER

This layer is composed of multiple high-performance servers with more powerful computation and storage resources, compared to fog computing layer. The cloud computing layer differs from the fog computing layer in that the cloud computing layer serves multiple intersections while the fog computing layer serves only one intersection.

Based on the traffic data offloaded by fog computing layers, the cloud computing layer concentrates on regional traffic control coordination which involves multiple intersections. Comparatively, the fog node only optimizes the phase timing for the one intersection it serves. The fog computing fails to account for the upstream and downstream traffic flow before and after its intersections respectively. Therefore, the autonomous traffic signal control at fog nodes are not globally optimal in multiple-intersection scenarios. For example, the vehicles released at the current intersection may lead to traffic congestion at the immediate downstream intersection because the fog node responsible for the current intersection has no traffic flow data at the immediate downstream intersection. Ultimately, there is a need for both global and local optimization and cloud computing performs global optimization like planning the globally optimal phase timing for multiple intersections.

According to these descriptions, the proposed multi-layer signal control system takes into account both signal and multiple intersections. Some algorithms and strategies are subsequently adopted to achieve local and global optimization of the traffic signal control.

## IV. TRAFFIC CONTROL STRATEGY

Fig.1 shows a traffic jam to detail the fog computing based traffic signal control strategy. If a traffic jam occurs due to the inappropriate setting of phase timings at the intersection, optimization can be initiated by fog computing.

Some traffic data can be collected to aid the phase timing optimization and the periodic beacon messages from vehicles are sent to fog nodes when vehicles are approaching the intersection. Such beacon messages usually include the destination, speed, acceleration, timestamp. For clarification, Fig. 3 shows the corresponding interactions among several entities (i.e., vehicles, RSUs, fog nodes and cloud node).

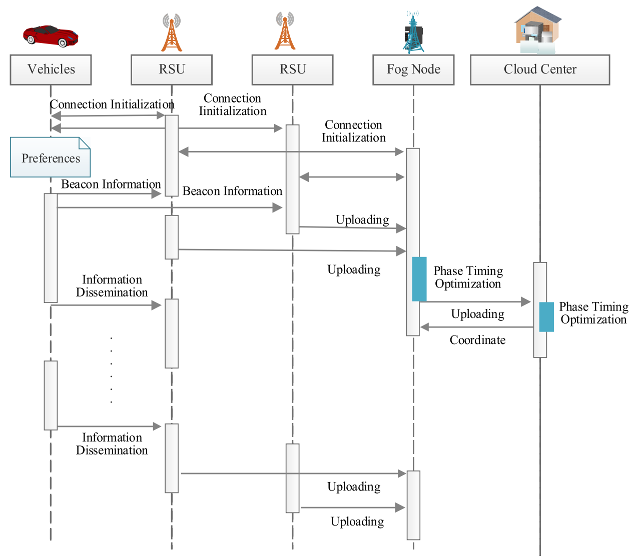



**FIGURE 3. Interactions among entities (vehicles, RSUs, fog nodes and cloud nodes).**

Through V2V/V2I communication technologies, RSUs can obtain queue length and the arrival rate of vehicles at the intersection. An initial communication link between RSUs and fog node is constructed and then RSUs send traffic messages to the fog node. The fog node starts the optimization module to optimize the traffic signal for single intersection and returns the results to the signal controller in real time. The controller subsequently extend the green time in current traffic light cycle or reschedule the phase timing for the next phases of the same traffic light cycle. On the other hand, cloud computing takes charge of traffic signal optimization for multiple signalized intersections, a more complicated process with extra factors such as the offset between two adjacent intersections.

## V. PROBLEM FORMULATION

To illustrate the application of fog computing to smart traffic control, we formulate the traffic control problem and further present the optimization objective in this section. Due to early stage of development of V2X techniques, not all vehicles are mounted with the OBUs but with promising growth in the foreseeable future. Therefore, we assume that the vehicles in this paper are OBU-mounted with full market penetration, which enables the direct communications among several entities as shown in Fig.3. Several important metrics for traffic optimization can be obtained via these communication techniques, such as the arrival speed, the queue length, and the distance between any vehicle in the queue and the intersection. To formulate the traffic optimization problem, we list key notations in Table 1 to be used through the paper.

**TABLE 1.** Notations for smart traffic control.

| Symbol | Description |
|---|---|
| $C$ | The length of traffic signal cycle |
| $P$ | Set of traffic phases |
| $K$ | The number of cycles |
| $L$ | Number of lanes in the same direction for one green phase |
| $S$ | Number of directions for one green phase |
| $\delta$ | The threshold triggering the optimization |
| $Veh_i^{pk}$ | The $i^{th}$ vehicle in the waiting queue for phase p during cycle $k$ |
| $AR_{pk}$ | The arrival rate of vehicles in phase $p$ during cycle $k$ |
| $G_{pk}^{min}$ | Minimal green time limitation for phase $p$ during cycle $k$ |
| $G_{pk}^{max}$ | Maximal green time limitation for phase $p$ during cycle $k$ |
| $G_{pk}$ | Green time duration for phase $p$ during cycle $k$ |
| $A$ | Acceleration when waiting vehicles start to pass the intersection |
| $V_{max}$ | Maximal speed limitation when vehicles are accelerated |
| $Dis(Veh_i^{pk})$ | The distance between $veh_i^{pk}$ and the intersection |
| $Num_{pk}$ | The initial number of queuing vehicles for phase $p$ during cycle $k$ |

Traffic phases usually refer to different signal states of traffic light at the intersection, which allow different and partially conflicting traffic flows to pass the intersection without vehicle collision. We adopt a traditional 4-arm intersection with four phases in the paper. Fig. 4 shows an example of 4-arm intersection with four phases. For example, in phase 1, only two traffic flows (e.g., east to south and west to north) are permitted while the traffic flow from other directions must wait at the intersection.

Before going further into the problem formulation of smart traffic control, we make some assumptions as follows.

First, we assume that the length of traffic light cycle is fixed. Thus, the optimization on traffic control is to adjust the phase timing residing in each single traffic light cycle.

Second, we assume that each vehicle in the waiting queue starts with the same acceleration when traffic light turns green. This assumption neglects the performance differences found in individual experience and driving habits and would otherwise complicate evaluating traffic control of the signalized intersections.

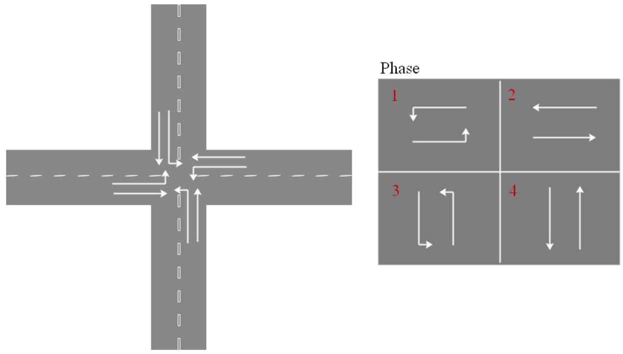



**FIGURE 4.** An example of 4-arm intersection with four traffic phases.

Third, the faster the vehicles pass the intersection, the more number of vehicles can pass the intersection within the designated green light time. However, faster speeds correlate to higher chances of car accidents, especially considering the pedestrian movement as well. Accordingly, we designate maximal vehicle acceleration and speed in this paper. Lastly, we assume that V2X communication techniques can record the distance between each vehicle in queue and the intersection.

In our smart traffic signal control scenario based on fog computing, we use the queue length to denote the performance evaluation metric of signalized intersection control. Given the number of traffic light cycles $K$, we optimize the phase timing within each single cycle, to minimize the queue length of vehicles waiting at the intersection. Then, a threshold $\delta$ is introduced to trigger the optimization process. When the number of vehicles waiting exceeds the threshold, the fog node starts to optimize the phase timing, and the new phase timing plan can be retrieved after the calculation in the fog computing layer. Then, results are sent to the traffic signal controller, which then reschedules the phase timing in the current traffic light cycle. By reducing or extending the corresponding phase timing, traffic jam can be alleviated at the intersection.

Based on the descriptions above, the objective function can be defined as below:

$$Minimize(Q): \sum_{k=1}^{K}\sum_{p=1}^{|P|}\sum_{s=1}^{S}\sum_{l=1}^{L} \max\{0, Num_{pk} + AR_{pk} \cdot G_{pk} - \arg\max_{i}\{Dis(Veh_i^{pk}) | Dis(Veh_i^{pk}) \leq D_0\}\}$$

$$\text{s.t. } D_0 = \frac{V_{max}^2}{2A} + V_{max} \cdot \max\{0, G_{pk} - \frac{V_{max}}{A}\} \tag{1}$$

$$\sum_{p=1}^{|P|} G_{pk} = C \tag{2}$$

$$G_{pk}^{min} \leq G_{pk} \leq G_{pk}^{max} \tag{3}$$

$$Num_{pk} \leq \delta \tag{4}$$

$$1 \leq p \leq |P| \tag{5}$$

$$1 \leq k \leq K \tag{6}$$

Constraint (1) calculates the maximal driving distance of a vehicle from stationary without exceeding $V_{max}$ in the given green time based on the principles of physics. Constraint (2) ensures that the length of traffic light cycle is unchangeable regardless of the adjustments to phase timing in the cycle. Usually, the initial phase timings can be planed based on the historical traffic data. That data is obtained by various sensors and V2X communication techniques. Constraint (3) reschedules phase timing so that it falls into a valid range specified by $G_{pk}^{min}$ and $G_{pk}^{max}$. Finally, constraint (4) states that the optimization process begins to work only if the initial queue length $Num_{pk}$ exceeds the threshold $\delta$.

## VI. PHASE TIMING OPTIMIZATION

In this section, a genetic algorithm is adopted to solve this optimization problem. We denote the genetic algorithm based phase timing rescheduling approach by GAPTR. In GAPTR, we encode the green time of four different phases shown in Fig.4 into a chromosome. Each gene in the chromosome is represented by binary variable 0 or 1. Fig. 5 shows an encoding example that corresponds to the four phases at an intersection. Each chromosome represents an individual attempt to schedule the phase timing.

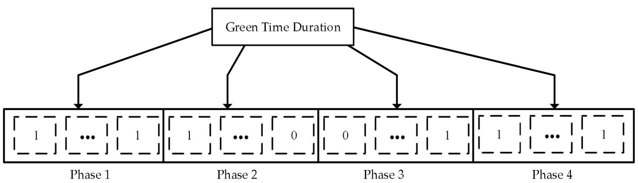



**FIGURE 5. An example of chromosome encoding.**

The fitness function is used to evaluate best functioning individuals. Better individuals tend to be reserved with higher possibilities than that of bad ones. In GAPTR, we aim to optimize the queue length of waiting vehicles as shown in the problem $Q$. Therefore, we choose the objective function as the fitness function directly in this paper.

For the selection, crossover and mutation operator, we adopt the conventional ways in genetic algorithms. For example, a roulette-wheel approach is used to do the selection operation and multiple point crossover and mutation are adopted in the crossover and mutation operation, respectively. Fig. 6 shows an example of multiple point crossover operation.

In Fig. 6, the crossover operation seems to exchange gene segments in phases, which, however, is not necessary in practice. We can leverage a random binary vector of the same length as the chromosome to conduct the crossover operation.The value of element in the vector denotes which parent the gene of the corresponding position in the offspring comes from. For example, if the value is 1 in the current position of vector, the gene value in the corresponding position of offspring comes from parent 1, and otherwise, parent 2. Similar operation can be applied to mutation operation with random vector. The only difference is that the value of element in the vector denotes whether the gene of the corresponding position in the offspring mutates or not.

## VII. SIMULATION AND RESULTS ANALYSIS

To evaluate the performance of the smart traffic control proposed in this paper, we have conducted extensive experiments in this section. Specifically, we first introduce the basic experimental setups and the simulation platform called Simulation of Urban Mobility (SUMO). Then we show the experimental results, followed by the analysis.

### A. EXPERIMENTAL INITIALIZATION

SUMO is an open source, microscopic, space-continuous and time-discrete traffic flow simulation platform [19]. It generates the road networks, vehicles and driving routes in the experiments. To integrate GAPTR into SUMO and realize on-line interaction and controlling of the traffic lights and the traffic flow, we use an external application called Traffic Control Interface (TraCI) [20]. It acts as a client to access the simulation artifacts and gives instructions to change the simulation process on demand via a socket connection. Fig.7 shows the evaluation process which integrates GAPTR into SUMO and introduces TraCI as a client to instruct the simulation.

Using the inductive loops and ICT techniques, SUMO monitors and records the traffic flow information such as the queue length of vehicles waiting at the intersection to simulate the traffic flows at the signalized intersection. As a client, TraCI can retrieve the simulation information after it establishes a TCP connection with SUMO as shown in Fig.7. Next, TraCI is responsible for deciding whether the optimization process is triggered. In other words, when the queue length exceeds the threshold, the optimization process is activated. The related traffic data, such as the queue length, vehicle acceleration, and arrival rate, are sent to the fog node which oversees traffic signal optimization through GAPTR. Due to its powerful computation abilities and low latency, the resulting phase timing plan can be returned nearly in real time. Then, the resulting scheduling plan is used to change the traffic signal light settings in SUMO via TraCI. Table 2 lists the parameters involved in the evaluation, in which the value ranges as well as the default values are given.

The fitness function as defined in minimization problem Q is a performance indicator which represents the efficiency of signalized intersection regulation. There are also other metrics which can act as performance indicators to evaluate the efficiency of intersection regulation. For example, intuitively, the fewer number of vehicles remains in queue after traffic light cycles, the less travel time for a vehicle. We use the average travel time (ATT) as the metric to evaluate GAPTR in this paper. The travel time denotes the time taken for a vehicle to travel from the original to the destination. And thus ATT is the mean travel time of all vehicles involved in the simulation, an important metric to evaluate the overall performance of the road network.

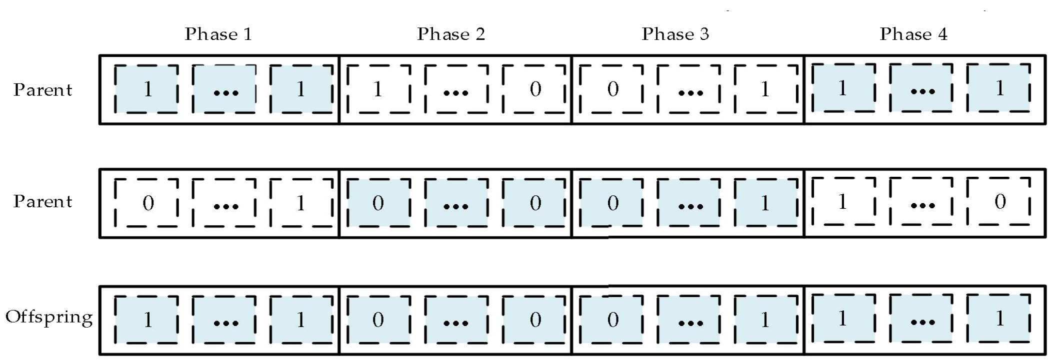



**FIGURE 6.** An example of crossover operation.

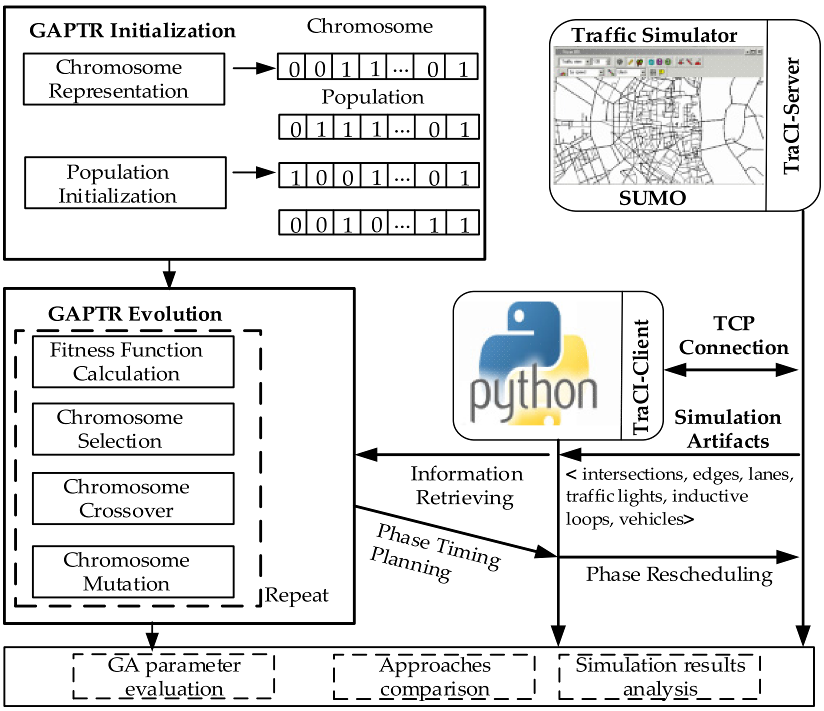



**FIGURE 7.** The evaluation process using SUMO and TraCI.

We choose an intersection with four arms as our smart traffic control scenario. To simplify the simulation process, we set the number of lanes in the same direction for one green phase to one lane and the number of directions for one green phase to two directions. 658 vehicles with random routes are generated in the road network. The default value of green phase is set to 30 seconds.

Two different kinds of calculation for GAPTR are proposed, which leads to two different strategies for smart traffic control, denoted as *Sig_GAPTR* and *Both_GAPTR*. *Sig_GAPTR* accounts for the number of vehicles waiting in queue at only one side, when deciding whether or not to trigger the optimization process. Usually, *Sig_GAPTR* chooses the side with the larger number of waiting vehicles. Conversely, *Both_GAPTR* considers the number of vehicles waiting in queue at both sides. Fig. 8 shows an example to illustrate this situation by denoting the state of traffic light at the intersection. In the current state, vehicles on lines 1 and 3 should be waiting for traffic lights while vehicles on line 0 and 2 are permitted to pass the intersection. Once the number of vehicles from either line 1 or line 3 exceeds the threshold, *Sig_GAPTR* starts the optimization process. *Both_GAPTR* starts the optimization process, only if the number of vehicles from both sides exceeds the threshold. We will compare the performance of the two strategies in the next simulation. In addition, we will introduce two additional approaches to compare with the two versions of GAPTR strategies. We denote the two approaches by *Baseline* approach and *Greedy* approach, respectively.

The *Baseline* approach does not reschedule the phase timing once the phase duration is determined. In this setting, the green phase in the corresponding phases is set to the

**TABLE 2.** Parameter involved in experiments.

| Name | Value | Default Value |
|---|---|---|
| Population size | [100, 1000] | 500 |
| Crossover probability | [0.2, 0.8] | 0.4 |
| Mutation probability | [0.01, 0.1] | 0.02 |
| Chromosome length | [10, 40] | 24 |
| $C$ | [90, 120] | 95 |
| $K$ | [100, 200] | 150 |
| $S$ | [1, 2] | 2 |
| $L$ | [1, 4] | 1 |
| $\acute{\varepsilon}$ | [10, 35] | 20 |
| $G_{pk}^{min}$ | [10, 20] | 10 |
| $G_{pk}^{max}$ | [35, 45] | 40 |
| $V_{max}$ | [15, 30] | 25 |

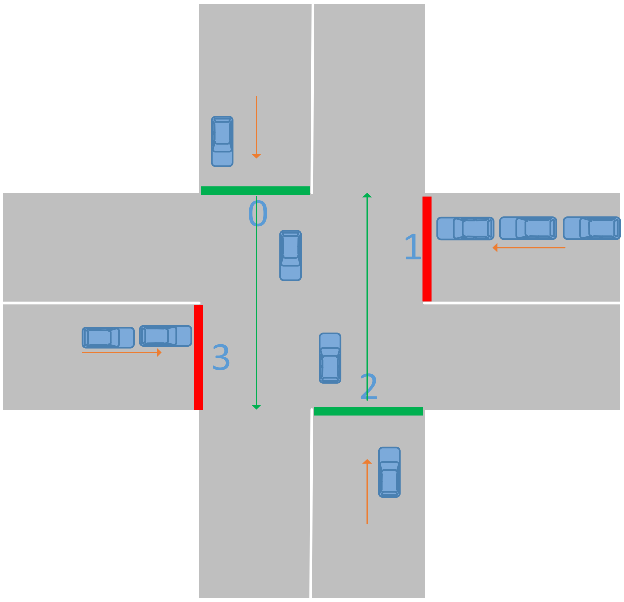



**FIGURE 8.** An example to trigger optimization in *Sig_GAPTR* and *Both_GAPTR*.

default values. Similar to GAPTR, the *Greedy* approach can reschedule the phase timing. When the number of vehicles from either side exceeds the threshold, *Greedy* approach starts the optimization process extending or reducing the green time duration. The duration of the green time makes sure that all the vehicles in queue pass the intersection. Considering the incoming vehicles from time to time, *Greedy* approach schedules the current phase to the next one when the time gap between two consecutive vehicles detected by the loop detectors exceeds the specified time which is set to three seconds in the simulation.

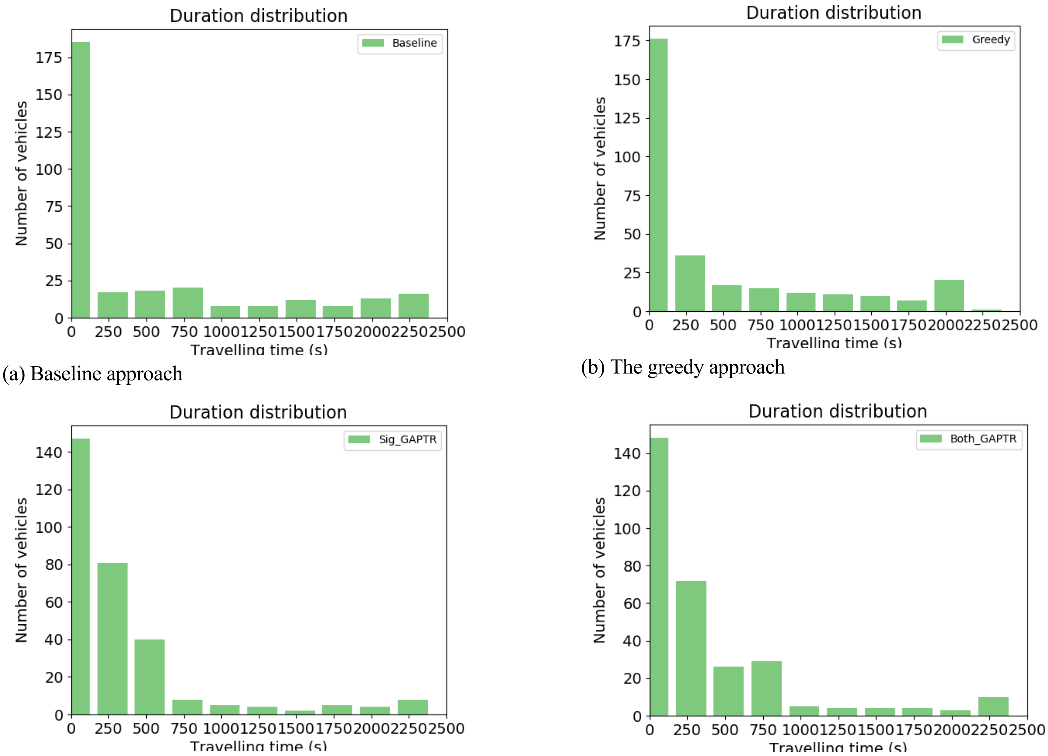
### B. SIMULATION RESULTS AND ANALYSIS

We conducted the first set of experiments to investigate the effectiveness of *Sig_GAPTR* and *Both_GAPTR*, compared to the *Baseline* and the *Greedy* approach. To visualize the outputs from the simulation, we use a tool named *plot_tripinfo_distributions.py* provided by SUMO to read the generated *tripinfo-files* and plot the selected attribute. Fig. 9 shows the results where the x-coordinate represents the travelling time of all vehicles across the simulation and the y-coordinate represents the corresponding number of vehicles under different travelling times. In the experimental settings, we randomly generate 500 vehicles and corresponding routes for the four approaches and set the simulation time to 5000 seconds.

From the results, we observe that there are more vehicles with travelling times less than 250 seconds in the *Baseline* and *Greedy* approaches compared to the *Sig_GAPTR* and *Both_GAPTR*: 175 vehicles travel the whole route in less than 250 seconds for the *Baseline* and *Greedy* approaches, versus 140 vehicles travel the whole route in less than 250 seconds for *Sig_GAPTR* and *Both_GAPTR*. However, the *Sig_GAPTR* and *Both_GAPTR* strive to achieve better performance compared to the *Baseline* and *Greedy* approaches by reducing the travelling time of all vehicles from a global perspective. What that means is that the number of vehicles with travelling time less than 1000 seconds is significantly greater than the *Baseline* and *Greedy* approaches. To sum up, the ATT for each strategy is 673, 592, 395 and 432, respectively. Compared to the *Baseline*, *Sig_GAPTR* and *Both_GAPTR* reduce the ATT by 41% and 36%, respectively. Compared to *Greedy*, *Sig_GAPTR* and *Both_GAPTR* reduce the ATT by 33% and 27%, respectively. This makes sense since *Sig_GAPTR* and *Both_GAPTR* find the optimal phase time in each traffic light cycle. Although the process is time consuming, the phase time scheduling can almost be achieved in real time using fog nodes.

The second set of experiments is conducted to evaluate the efficiency of the four different strategies. Fig. 10 shows the results. The x-coordinate represents the number of steps, denoted by $N$, which is used to generate the vehicles in a random way. Since there are twelve lanes for one intersection. Each value of $N$ corresponds to at most $12*N$ vehicles. In doing this, we add the randomness and unpredictability to the simulation. For the result, the ATT is used as the evaluation metric in the simulation. We can observe that the *Baseline* approach has the largest values of ATT while *Sig_GAPTR* has the smallest values of ATT in most cases. The GAPRT strategies ( *Sig_GAPTR* and *Both_GAPTR*) have the better performance compared to *Baseline* and *Greedy* strategies. *Greedy* and *Baseline* had the largest ATT values of 1000 and 1200, respectively. In each case, both the vehicles and corresponding routes are generated randomly and hence some extreme situations can be generated which lead to this result. For instance, if the number of vehicles in one direction is tremendously large, *Baseline* traffic control strategy with fixed phase time may handle the situation efficiently.

Table 3 denotes the performance comparison compared to *Baseline* and *Greedy* approach. For example, when the number of steps is 1000, the ATT for *Sig_GAPTR* and *Both_GAPTR* is 565 and 659, respectively. Compared with




(a) Baseline approach
(b) The greedy approach
(c) Sig_GAPTR
(d) Both_GAPTR

**FIGURE 9.** **The number of vehicles distribution according to travelling time with four different approaches.**

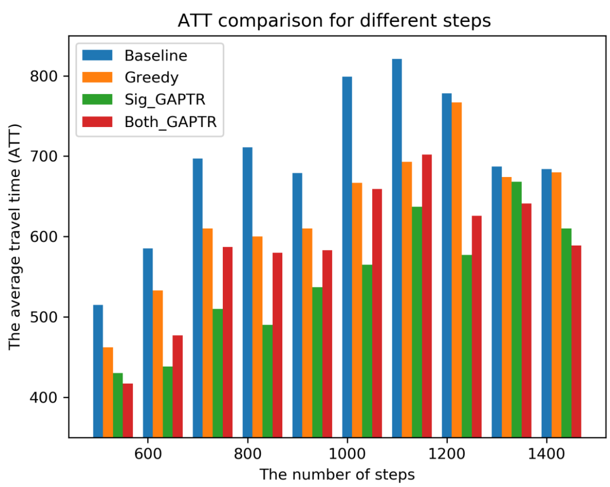



**FIGURE 10.** **The average travel time comparison under number of steps.**

*Baseline* and *Greedy*, *Sig_GAPTR* reduces the ATTs by 29.3% and 15.2%, respectively. Similarly, *Both_GAPTR* reduces the ATTs by 17.5% and 1.2%, respectively. From the table, we can observe that in general *Sig_GAPTR* has relatively better performance than *Both_GAPTR*. Both *Sig_GAPTR* and *Both_GAPTR* have better performance than *Baseline* and *Greedy*. When the number of steps is 1100, *Both_GAPTR* increases the ATT by 1.3% compared to Greedy approach. The reason is similar to the case in Fig. 10. The random generation of vehicles and routes can occasionally result in an extreme situation with this result.

Genetic algorithm searches the best individual over the huge population space by generating new population iteratively. But they brings large time overheads and hence violates the principle of timeliness necessary for smart traffic control. Accordingly, fog computing is introduced to reduce the response latency. In addition to the large population size, the number of generations also has great affects on the time overheads. Generally, the probability to find the best solution with regards to the fitness value is becoming increasingly greater as the number of generations increases. However, if an overall optimization is achieved, it makes no sense to continue increasing the number of generations.

To investigate the relationships between best individuals and the number of generations, we have conducted another set of experiments, shown in Fig. 11. The x-coordinate represents the number of generations and y-coordinate the corresponding fitness values. The optimization process of smart traffic control is triggered by the number of waiting vehicles in their own lanes. Four cases, which can trigger the optimization process, are generated randomly, denoted by C1, C2, C3 and C4, respectively. Each case in the simulation is actually a vector in which each element denotes the number of waiting vehicles in its own lane.

**TABLE 3. Performance comparison compared to baseline and greedy.**

| Number of steps | | 500 | 600 | 700 | 800 | 900 | 1000 | 1100 | 1200 | 1300 | 1400 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| ATT(*Sig_GAPTR*\|*Both_GAPTR*) | | 430\|417 | 438\|477 | 510\|587 | 490\|580 | 537\|583 | 565\|659 | 637\|702 | 577\|626 | 668\|641 | 610\|589 |
| *Sig_GAPTR* | *Baseline* | 16.5% | 25.1% | 26.8% | 31% | 20.9% | 29.3% | 22.4% | 25.8% | 2.7% | 10.9% |
| | *Greedy* | 6.9% | 17.8% | 16.3% | 18.3% | 11.9% | 15.2% | 8.8% | 24.8% | 8.9% | 10.3% |
| *Both_GAPTR* | *Baseline* | 19.1% | 18.4% | 15.7% | 18.5% | 14.4% | 17.5% | 14.6% | 19.6% | 6.7% | 13.9% |
| | *Greedy* | 9.7% | 10.5% | 3.8% | 3.3% | 4.4% | 1.2% | -1.3% | 18% | 4.9% | 13.6% |

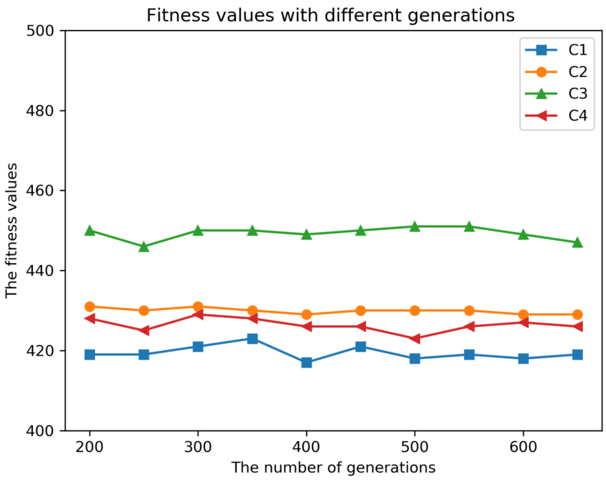



**FIGURE 11. Fitness values with different number of generations.**

We observe that the best individual in regards to fitness values can be found within five hundred generations in most cases. We use the roulette-wheel approach to select the offspring to generate the new population. The individuals, that violate the constraints in current population, are filtered out in the next generation. Thus, there is a greater probability that better individuals that satisfy the constraints are generated in the next generation. For instance, the best individual for C3 are found at about 250 generations, while the best individual for C1 is found at about 400 generations. Totally speaking, in our case, it suffices to set the number of generations within five hundred generations.

## VIII. CONCLUSIONS

With the help of advanced ICTs (such as V2X techniques), huge amount of heterogeneous traffic data can be gathered. To process these information and realize traffic signal control usually requires the employment of efficient control strategies and algorithms. However, existing traffic signal control strategies have serious response-time overheads. To achieve a smart traffic light control, we have proposed a fog computing based traffic signal control strategy in this paper, as well as a traffic signal control architecture. In the architecture, the phase timing task for a single intersection can be handled by a local fog node in real time, and regional optimization task will be left for the centralized cloud. Testing our proposition, we integrate our GA based traffic signal control approach into SUMO via TraCI. The experimental results have shown that the control strategies are more efficient than *Baseline* and *Greedy* strategies.

To realize a fog-upgraded traffic light control, effective transportation specified software entities running in fog nodes are critical. The software entities include application software as well as a few middleware that could support dedicated software design with several APIs or functions. Therefore, in the future works, we are planning designing and realizing software entities which can support flexible update and upgrade to adapt to new control policies.

## REFERENCES

[1] C. Perera, Y. Qin, J. C. Estrella, S. Reiff-Marganiec, and A. V. Vasilakos, "Fog computing for sustainable smart cities: A survey," *ACM Comput. Surv.*, vol. 50, no. 3, p. 32, Oct. 2017.

[2] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the Internet of Things," in *Proc. MCC Workshop Mobile Cloud Comput.*, Helsinki, Finland, Aug. 2012, pp. 13–16.

[3] C. Tang, X. Wei, and J. Liu, "Application of sensor-cloud systems: Smart traffic control," in *Security, Privacy, and Anonymity in Computation, Communication, and Storage* (Lecture Notes in Computer Science). Melbourne, NSW, Australia: Springer, 2018, pp. 192–201.

[4] M. R. J. Sattari, H. Malakooti, A. Jalooli, and R. M. Noor, "A dynamic vehicular traffic control using ant colony and traffic light optimization," in *Proc. ICSS*, Wroclaw, Poland, 2014, pp. 57–66.

[5] J. Guo, Y. Kong, W. Huang, J. Cao, Y. Wei, and Z. Li, "A model and genetic algorithm for area-wide intersection signal optimization under user equilibrium traffic," *Math. Comput. Simul.*, vol. 155, pp. 92–104, Jan. 2019.

[6] E. Eriskin, S. Karahancer, S. Terzi, and M. Saltan, "Optimization of traffic signal timing at oversaturated intersections using elimination pairing system," *Procedia Eng.*, vol. 187, pp. 295–300, May 2017.

[7] Z. Zhou and M. Cai, "Intersection signal control multi-objective optimization based on genetic algorithm," *J. Traffic Transp. Eng.*, vol. 1, no. 2, pp. 153–158, 2014.

[8] B. Zhao, C. Zhang, and L. Zhang, "Real-time traffic light scheduling algorithm based on genetic algorithm and machine learning," in *Proc. Int. Conf. Internet Vehicles*, Chengdu, China, Dec. 2015, pp. 385–398.

[9] D. Krajzewicz, E. Brockfeld, J. Mikat, J. Ringel, C. Rössel, W. Tuchscheerer, R. Wösler, and P. Wagner, "Simulation of modern traffic lights control systems using the open source traffic simulation SUMO," in *Proc. Ind. Simul. Conf.*, Jun. 2005, pp. 1–4.

[10] C. Huang, R. Lu, and K.-K. R. Choo, "Vehicular fog computing: Architecture, use case, and security and forensic challenges," *IEEE Commun. Mag.*, vol. 55, no. 11, pp. 105–111, Nov. 2017.

[11] J. Feng, Z. Liu, C. Wu, and Y. Ji, "AVE: Autonomous vehicular edge computing framework with ACO-based scheduling," *IEEE Trans. Veh. Technol.*, vol. 66, no. 12, pp. 10660–10675, Dec. 2017.

[12] X. Hou, Y. Li, M. Chen, D. Wu, D. Jin, and S. Chen, "Vehicular fog computing: A viewpoint of vehicles as the infrastructures," *IEEE Trans. Veh. Technol.*, vol. 65, no. 6, pp. 3860–3873, Jun. 2016.

[13] C. Zhu, L. Shu, V. C. M. Leung, S. Guo, Y. Zhang, and L. T. Yang, "Secure multimedia big data in trust-assisted sensor-cloud for smart city," *IEEE Commun. Mag.*, vol. 55, no. 12, pp. 24–30, Dec. 2017.

[14] C. Tang, X. Wei, C. Zhu, W. Chen, and J. J. P. C. Rodrigues, "Towards smart parking based on fog computing," *IEEE Access*, vol. 6, pp. 70172–70185, 2018.

[15] C. Priemer and B. Friedrich, "A decentralized adaptive traffic signal control using V2I communication data," in *Proc. Int. IEEE Conf. Intell. Transp. Syst.*, St. Louis, MO, USA, Oct. 2009, pp. 1–6.

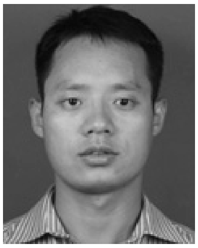
[16] Y. Feng, K. L. Head, S. Khoshmagham, and M. Zamanipour, "A real-time adaptive signal control in a connected vehicle environment," *Transp. Res. C, Emerg. Technol.*, vol. 55, pp. 460–473, Jun. 2015.

[17] K. Kang, C. Wang, and T. Luo, "Fog computing for vehicular ad-hoc networks: Paradigms, scenarios, and issues," *J. China Univ. Posts Telecommun.*, vol. 23, no. 2, pp. 56–65, Apr. 2016.

[18] A.-M. Cailean, B. Cagneau, L. Chassagne, V. Popa, and M. Dimian, "A survey on the usage of DSRC and VLC in communication-based vehicle safety applications," in *Proc. 21st IEEE Symp. Commun. Veh. Technol. Benelux*, Delft, The Netherlands, Nov. 2014, pp. 69–74.

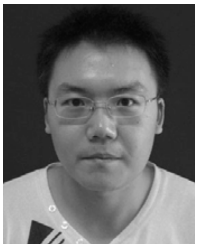
[19] D. Krajzewicz, G. Hertkorn, C. Rössel, and P. Wagner, " SUMO (Simulation of Urban MObility)—An open-source traffic simulation," in *Proc. 4th Middle East Symp. Simulation Modelling*, 2002, pp. 183–187.

[20] A. Wegener, M. Piórkowski, M. Raya, H. Hellbrück, S. Fischer, and J.-P. Hubaux, "TraCI: An interface for coupling road traffic and network simulators," in *Proc. 11th Commun. Netw. Simulation Symp.*, Ottawa, ON, Canada, Apr. 2008, pp. 155–163.

**CHAOGANG TANG** received the B.S. degree from the Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2007, and the Ph.D. degree from the School of Information Science and Technology, University of Science and Technology of China, Hefei, China, and the Ph.D. degree from the Department of Computer Science, City University of Hong Kong, in 2012, through the joint Ph.D. Program. He is currently an Assistant Professor with the China University of Mining and Technology. His research interests include mobile cloud computing, fog computing, the Internet of Things, big data, and WSN.

**SHIXIONG XIA** is currently a Professor with the China University of Mining and Technology, Xuzhou, China. His research interests include pattern recognition, intelligent information processing, and information security.



**CHUNSHENG ZHU** received the B.E. degree in network engineering from the Dalian University of Technology, China, in 2010, the M.Sc. degree in computer science from St. Francis Xavier University, Canada, in 2012, and the Ph.D. degree in electrical and computer engineering from The University of British Columbia, Canada, in 2016, where he is currently a Postdoctoral Research Fellow of the Department of Electrical and Computer Engineering. He has authored over 120 papers or accepted by refereed international journals, such as the IEEE Transactions on Industrial Electronics, the IEEE Transactions on Computers, the IEEE Transactions on Information Forensics and Security, the IEEE Transactions on Emerging Topics in Computing, the IEEE Systems Journal, the *IEEE Communication Magazine*, and IEEE Access, and conferences, such as the IEEE GLOBECOM and the IEEE ICC. His current research interests mainly include wireless sensor networks, cloud computing, the Internet of Things, social networks, and security.

**XIANGLIN WEI** received the bachelor's degree from the Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2007, and the Ph.D. degree from the University of Science and Technology, Nanjing, in 2012. He is currently a Researcher with the Nanjing Telecommunication Technology Research Institute, Nanjing. His research interests include cloud computing, peer-to-peer networks, network anomaly detection, network measurement, and distributed system design and optimization. He has served as an editorial member of many international journals and a TPC member of several international conferences. He has also organized a few special issues for many reputable journals.