

# Soft Computing in Big Data Intelligent Transportation Systems

Chao Wang<sup>1,\*</sup>, Aili Wang<sup>2</sup>, Xi Li<sup>1</sup>, Xuehai Zhou<sup>1</sup> and Nadia Nedjah<sup>3</sup>

<sup>1</sup> School of Computer Science, University of Science and Technology of China  
Hefei, 230027, China

E-Mails: cswang@ustc.edu.cn, llxx,xhzhou@ustc.edu.cn

<sup>2</sup> School of Software Engineering, University of Science and Technology of China  
Suzhou, 215123, China

E-Mail: wangal@ustc.edu.cn

<sup>3</sup>Department of Electronics Engineering and Telecommunications  
Faculty of Engineering

State University of Rio de Janeiro, Brazil

E-Mail: nadia@eng.uerj.br

---

**Abstract:** The academic and industry have entered big data era in many computer software and embedded system related fields. Intelligent transportation system problem is one of the important areas in the real big data application scenarios. However, it is posing significant challenge to manage the traffic lights efficiently due to the accumulated dynamic car flow data scale. In this paper, we present NeverStop, which utilizes genetic algorithms and fuzzy control methods in big data intelligent transportation systems. NeverStop is constructed with sensors to control the traffic lights at intersection automatically. It utilizes fuzzy control method and genetic algorithm to adjust the waiting time for the traffic lights, consequently the average waiting time can be significantly reduced. A prototype system has been implemented at an EBox-II terminal device, running the fuzzy control and genetic algorithms. Experimental results on the prototype system demonstrate NeverStop can efficiently facilitate researchers to reduce the average waiting time for vehicles.

**Keywords:** Big data, intelligent transportation system, fuzzy control, genetic algorithm.

---

## 1. Introduction

Big data is now becoming common sense and the widely applied research area during the past decades [1]. With the pace of industrialization and urbanization, there are many research fields such as genome sequencing [2], intelligent transportation systems [3] [4], chip design [5]. In particular, due to the main traveling facility in the crowded city, the amount of the automobiles is increasing rapidly. Consequently the traffic problem gets more and more serious, which not only wastes resources and time, but also causes serious losses to the economy [6].

It has been proved that traffic problem causes 60 billion Euro losses in Germany each year. For example, Beijing, China, had a total of 4 million vehicles at the beginning of 2010 and increased another 800K during the past years. Congestion can lead to an increase in fuel consumption, air pollution, and difficulties in implementing plans for public transportation [7]. It is also reported that the average length of the waiting cars is 70 kilometers altogether in Germany. The annual financial cost of traffic congestion has swollen 97 billion and each day Germans burn approximately 300 million liters of fuel while idling in the heavy traffic. With the rapid developing of the economy and the explosive data scale of individual vehicles, China National Information Center announced that the car consumption was at an increasing speed of 13 percent and the total sale reached 5.80 million.

The intelligent transportation system problem is one of the important areas in the real big data application scenarios. A major problem causing the ineffective traffic scheduling is the explosive data scale of the car flow at the intersections. Traditional methodologies could not handle the tremendous data scale, therefore the automatic tuning problem is still pursuing by many researchers. To efficiently manage the traffic lights and further to reduce the average waiting time, in this paper we propose NeverStop, which is extended from our previous work [4] to manage both a single intersection and correlated area with multiple intersections. Unlike the conventional traffic management systems, NeverStop is able to detect the traffic flow information and then determines the intervals of traffic lights. Flexibility is one of the most primary critical principles during the design of NeverStop system. Besides determining the time of traffic lights dynamically, users can also customize their own schemes to fit special occasions and also can control the traffic lights instantly: for example, while encountering traffic accidents or bad weather. In particular, we claim following contributions of NeverStop system:

1. NeverStop obtains the information of the cars over passing the green traffic light and waiting for the red light using RFID devices. NeverStop sends the car flow data to remote central server that is deployed in EBOX II devices.
2. EBOX II device processes the data, and then figures out the delay-time of the traffic lights using the fuzzy control rule and genetic algorithms.
3. EBOX II collects the information of nearby intersections from the server, and adjusts the time. (If the NeverStop system is deployed in a single intersection, this step could be skipped).
4. NeverStop manipulates the intersection by switching the light or takes an adjustment according to information.

The remainder of this paper is decomposed as below. Section 2 summarizes the related study and the innovation concepts of the NeverStop system. In section 3 we present the NeverStop system overview and implementations. Section 4 describes the fuzzy control method and genetic algorithm. Section 5 illustrates the test plans and experimental results. Finally we conclude the paper in Section 6.

## **2. Related Work and Motivation**

The intelligent transportation system has been widely researched for decades [8]. In this section we review the related study, including the intelligent transportation system, fuzzy control, genetic algorithm and web services, respectively.

### *2.1 Intelligent Transportation System Review*

Intelligent transportation system has been a major research area during the past decade [9]. Now sensors have been an increasingly important method in the intelligent transportation system design [10]. An algorithm for the implementation of short-term prediction of traffic with real-time updating based on spectral analysis is described in [11]. The prediction is based on the characterization of the flow based on modal functions associated with a covariance matrix constructed from historical flow data. The number of these modal functions used for prediction depends on the local traffic characteristics. Although the method works well for the examples in this paper using the lower frequency modes, it can be adapted to include modes of higher frequency, as traffic conditions dictate.

Prominent short-term forecasting methods use different empirical and theoretical techniques. The empirical approaches (nonparametric and parametric) employ statistical methodology and/or heuristic methods for traffic flow forecasting. The nonparametric techniques include nonparametric regression [12] and artificial neural networks (ANNs) [13], [14], [15]. Considering two broad classes of models, namely, empirical and evolutionary, the distinction between examples of these and spectral-based forecasting can be seen. Reference [16] focuses on an algorithm for moving-object detection and tracking, given a sequence of distributed laser scan data of an intersection. The goal is to detect each moving object that enters the intersection; estimate state parameters such as size; and track its location, speed, and direction while it passes through the intersection.

## 2.2 Fuzzy Control Methodologies

There is a broad range of diverse technologies under the generic topic of intelligent transportation systems (ITS) that holds the answer to many of the transportation problems. However, fuzzy control methodologies have not been widely used. For example, reference [17] presents a novel interval type-2 fuzzy controller architecture proposed to resolve nonlinear control problems of vehicle active suspension systems. It integrates the Takagi-Sugeno (T-S) fuzzy model, interval type-2 fuzzy reasoning, the Wu-Mendel uncertainty bound method, and selected optimization algorithms together to construct the switching routes between generated linear model control surfaces. Reference [18] presents an adaptive cruise control (ACC) method. The main features of this kind of controller are the adaptation of the speed of the car to a predefined one and the keeping of a safe gap between the controlled car and the preceding vehicle on the road. Linda and Manic [19] proposed to evaluate the spatiotemporal risk based on the combination of online nearest neighbor and fuzzy inference.

## 2.3 Genetic Algorithms

Genetic Algorithms are becoming popular and effective in solving complex mathematical and scientific problems, including [20-22]. In the intelligent transportation systems, Genetic Algorithms have been utilized to achieve minimize driving [23] or waiting time [4]. The route guidance system [24], which provides driving advice based on traffic information about an origin and a destination, has become very popular along with the advancement of handheld devices and the global position system. Since the accuracy and efficiency of route guidance depend on the accuracy of the traffic conditions, the route guidance system needs to include more variables in calculation, such as real time traffic flows and allowable vehicle speeds. As variables considered by the route guidance system increase, the cost to compute multiplies. As handheld devices have limited resources, it is not feasible to use them to

compute the exact optimal solutions by some well-known algorithm, such as the Dijkstra's algorithm, which is usually used to find the shortest path with a map of reasonable numbers of vertices.

To solve this problem, [23] proposes to use the genetic algorithm to alleviate the rising computational cost. The authors use the genetic algorithm to find the shortest time in driving with diverse scenarios of real traffic conditions and varying vehicle speeds. The effectiveness of the genetic algorithm is clearly demonstrated when applied on a real map of modern city with very large vertex numbers. [25] provides a “real-time” traffic signal control strategy using genetic algorithms to provide nearoptimal traffic performance for intersections. Real-time traffic signal control is an integral part of the urban traffic control system and providing effective real-time traffic signal control for a large complex traffic network is an extremely challenging distributed control problem. [26] addresses the problem of selecting route to a given destination on an actual map under a static environment. The proposed solution uses a parallel genetic algorithm (PGA) implemented using High performance Cluster (HPC). [27] focuses on real-time adaptive signal optimization using genetic algorithms. The proposed adaptive signal system provides acyclic signal operation based on a rolling horizon real-time control approach. By applying random A\* algorithm, [28] clears out the biggest obstruction between the genetic algorithm and dynamic route guidance of how to get the initial generation of genetic algorithm. The developed models and algorithms are implemented with local electronic map and their computational performance is analyzed experimentally.

#### 2.4 Web based approaches

A comprehensive and presentable public transport information system is deemed invaluable for local residents and tourists all over the world. This is particularly necessary in view of the complex city structure and transportation system in Hong Kong. There are more than ten public transportation modes available, all with different operation schedules, fare structures, and routing characteristics. To assist commuters in making better use of public transport, the system needs to be not only user friendly and informative but intelligent enough to provide optimal route choices in terms of users' traveling behavior or preference as well.

The web-map public transport enquiry system, as described in [29], gives a good example of providing bilingual (English and Chinese) information on all public transports run in Hong Kong in the form of interactive maps and texts, as well as real-time derivation of optimal travelling routes for users in terms of multiple criteria, i.e., preferred mode, least changes, shortest travelling time, or lowest fare. Lu et al. [30], [31] developed a web-based visualization package, named CubeView, to aggregate data for identifying major traffic processing techniques.

#### 2.5 Innovation Concepts of NeverStop

Table 1 presents a summary of the state-of-the-art approaches. For each type, we list some typical references and analyzed the strengths and weaknesses, respectively. In spite of the state-of-the-art researches, there is no research that combines the fuzzy control methods, genetic algorithm and web services into a sound framework. NeverStop, proposed in this paper, absorbs the advantages of the above approaches. In particular, the innovation concepts and highlights are introduced as below:

**Table 1 Summary of the State-of-the-arts**

Type	References	Strengths	Weaknesses
Intelligent Transportation System	[13-15]	Motivated to shorten driving or waiting time	Neural networks could increase the complexity
Fuzzy Control	[17-19]	Solve nonlinear control problems	Not applicable to Intelligent Transportation System
Genetic Algorithms	[23-25]	Achieving shorten driving or waiting time	Complex for computing at servers
Web based approaches	[29-31]	Web inquiry interface	Not adaptive to local management

*1) RFID device*

We fix up a RFID tag with the information about the type and ID on the car. When the car passes the intersection, the information will be read by the RFID reader which has been placed on the roadside ahead of schedule. These information forms the car flow data, and then the data will be sent to EBox. The RFID device enables NeverStop system to be smart and flexible.

*2) Multiple selective modes*

NeverStop can run at a certain mode that is configured during initialization. When a sudden incident occurs, it can switch the mode at run time, marinating the traffic by the greatest extension. What's more, the administrator can configure their self-developed modes to satisfy special requirements.

*3) Web Service*

In order to ease the burden for programmers and the workload of each terminal device, NeverStop integrates the RPC (Remote Process Call) functionality at server side. The communication based on web service would package the data automatically, including the jam data, the input device command, and the data structures.

**3. System Overview and Implementations**

NeverStop has been designed to integrate into an intelligent traffic lights management system ensuring that there is minimal need for additional hardware, as is illustrated in Fig. 1. The system is comprised of the following components:

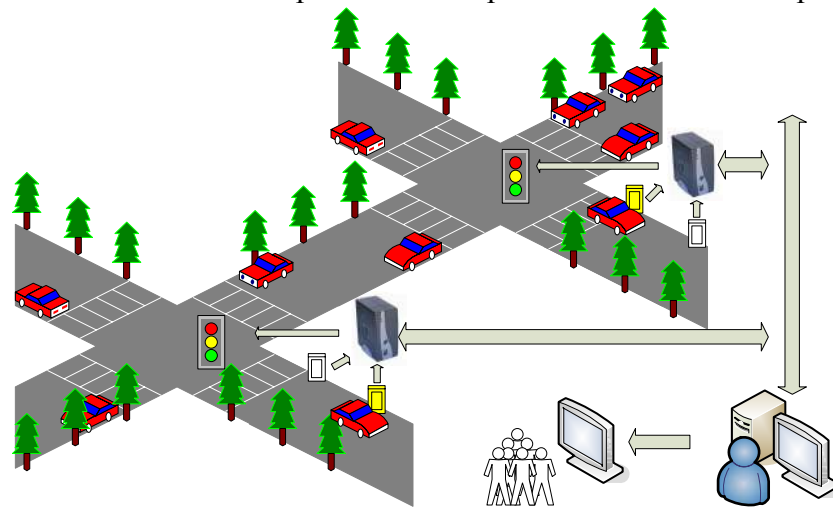
EBOX II: The hardware package housing the NeverStop System Controller and providing connectivity to the traffic lights, the RFID device and the Central Server. This unit runs a customized Windows CE OS image.

Light Control Unit: The controller to interface with the traffic lights and EBOX II. It may be consisted of one or several PCB(s).

RFID device: The device to send the car flow information to the EBOX II. It normally contains the following components: tags, readers, and antennas. We deploy the device both at the entrance and the exit of the lane.

Central Server: The server is in charge of making all the terminal EBox work in parallel and provides references and historical data. The central server primarily provides a means of guaranteeing

resilience in the accidental event of an unexpected catastrophic failure of NeverStop.



**Figure 1.** System Overview

Obtained with the above components, the system must adhere to the following performance requirements:

*(1) Frequency of RFID device*

The RF tag works on a certain frequency. Taking our system model into account, we choose ultra high frequency (UHF) or microwave tag for mobile object identification. Usually, as microwave RF tag, the effective distance for recognizing information can reach 3~5 meters.

*(2) Power Control*

Both RFID reader and the light control circuit need power supply modules. As the system is mainly installed in the city, the power supply is provided with chargers.

*(3) Active Network Connect*

The limitation relies on an active network between central servers and each terminal device. If connection is lost, NeverStop could lose some functions that depend on the network running environment. As in the case of multiple intersections, the micro adjustment of time will be unavailable. Therefore, it is preferred and desirable to deploy the NeverStop system on a local network other than on the internet.

### 3.1 Design Objectives

The design of the NeverStop system was guided by the following key design objectives:

*(1) Usability:* NeverStop is deployed at the intersections to manage the traffic. For convenience, the administrators do not need to be aware of the detailed implementation due to that the NeverStop is automatically configured. Therefore what is required is just “one-click” initialization. At the same time, administrator can supervise all the intersections at the server side in case that an accident happens.

*(2) Reliability:* As the system is designed to operate without user’s intervention, it is extremely important that the system runs reliably and has the ability to handle unexpected situations.

*(3) Extensibility:* The NeverStop system is designed for efficient expansion in big data era. To ensure maximum portability and versatility, it is imperative that the device configuration settings be remotely

obtained. This will enable the NeverStop system to grow into new markets easily.

### 3.2 System Description

The NeverStop system is composed of five components, three of which are hardware components (EBOX II, RFID device and Traffic Light Control Unit) and the others are software components (NeverStop System Controller Module and Central Server). Further more, the System Controller Module includes three subcomponent modules: Fuzzy Rule Module, Controller Module and Genetic Algorithm. The architecture module is presented in Figure 2.

The EBox is the central hardware component of the system. Within the EBox runs a Windows CE Image which is the operating system for the EBox. On top of this Image runs the NeverStop System Controller that determines when to get the traffic congestion state, when to get the delay time and when to change the phase. The NeverStop System Controller is connected through the EBox to the Traffic Light Control Unit via the serial port controller. The Traffic Light Control Unit is taking charge of the management of the traffic lights. The input module deals with the raw data from RFID and organizes them into useful information. Finally the Central Server operates the traffic lights remotely and is used to store the configurations for a group of neighborhood intersections. The NeverStop System Controller is connected to the Central Server via the EBox's RJ-45 Ethernet port for hardware connection.

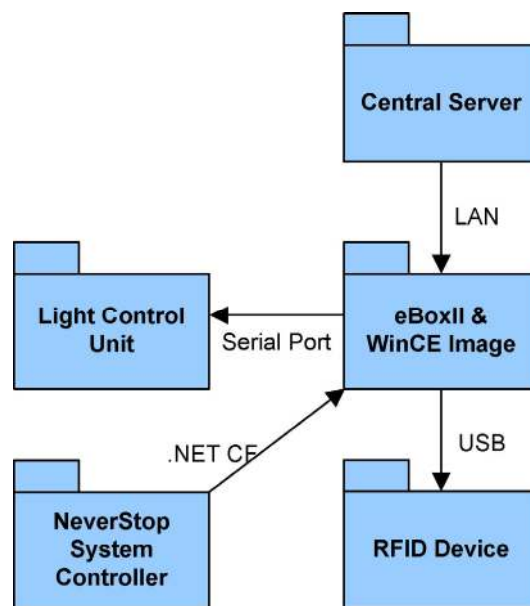


Figure 2. Architecture Modules

### 3.4 The Controller Module

In the controller module, we first illustrate the phase model, and then describe the single intersection model as well as the multiple intersection models, respectively. Finally, special modes are introduced to handle the unexpected situations.

#### (1) Phase Model

There are at least four phases at each intersection. When one phase is running, the other phases are disabled automatically. Please note the phase model is only for demonstration that it can be extended to

more phases for the real occasions. In order to maintain the ubiquitous and general, we use a four phase model in this paper, which are presented in Fig. 3.

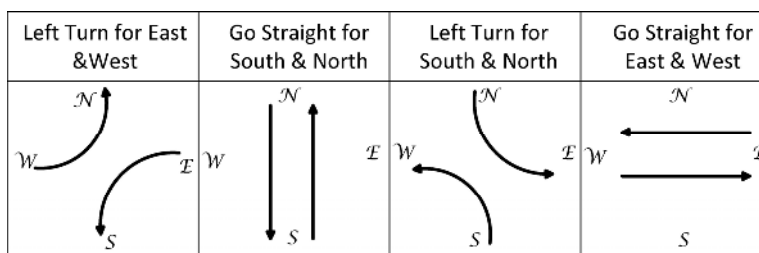


Fig. 3. Phase Model

According to the intersection status, there are several modes could be selected:

1. Single Intersection Model: Manage the traffic lights according to the traffic information of the local intersection.
2. Multiple Intersections Mode: Manage the traffic lights according to the local intersection and the traffic information of the neighborhood intersections.
3. Special Modes: The system has other configured modes. For example, when an ambulance or a fire truck comes, or even there is an accident, NeverStop should switch to corresponding mode to meet the individual requirement of accidental events.

(2) Single Intersection

As NeverStop can get the congestion from RFID device, the algorithm is used to manage the traffic of local intersection.

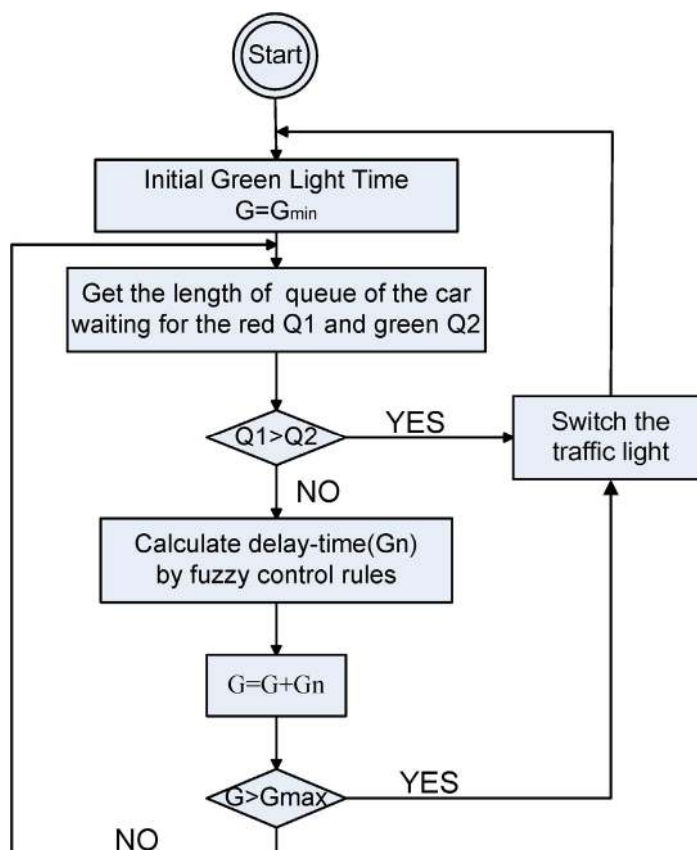


Fig. 4. Algorithm Step



When the traffic is smooth, each phase of traffic lights should be as short as possible, as yet no shorter than the minimum threshold in which the cars flow can go through the intersection exactly. When there is heavy traffic, the corresponding direction should take a longer time, yet not longer than a maximum threshold in which the drivers can wait. The control algorithm in single intersection is illustrated in Fig .4.

At initialization stage, the minimum time of green light is configured. When the delay ends, the algorithm compares the total delay time of current phase with the maximum time to decide whether NeverStop goes to the next phase or remain in current state. If the red queue is waiting longer than the green queue, then the traffic lights should be switched immediately. Otherwise, the delay time should be calculated by fuzzy control rules, and then an updated delay time is configured and compared to the max light time.

(3) Multiple Intersections

As the purpose of NeverStop is to reduce the average delay time for each vehicle, therefore we should not only consider about the local intersection, but also need to pay attention to the neighbor four intersections, so as to achieve the congestion in a small area.

The phase model for multiple intersections is the same with single intersections, which is presented in Figure 3. Meanwhile, the delay at each intersection is adjusted incrementally to coordinate with the adjacent neighbor intersections to minimize stops in the direction of dominant traffic flow. Fig. 5 illustrates the communication principle between the integrated components.

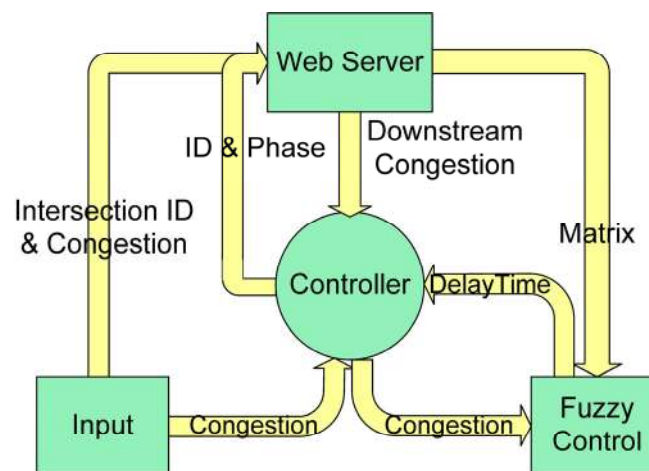


Fig. 5. Relations between the Modules

So in the case of multiple intersections, there are a few differences: After the delay time was calculated by the algorithm in single intersection model, NeverStop will not set up the delay time immediately. Instead, it asks for the information about the neighbor intersections from the central server. When the data was received by the EBox, NeverStop can get a micro adjustment according to the fuzzy control, and then get the accurate time that the current phase should delay. For example, if the neighbor intersection is in heavy jam, the delay time of current phase should be shorter. On the contrary, if the neighbor intersections have no congestion, the time of green light can be increased accordingly. Fig. 6 describes the relations between the classes.

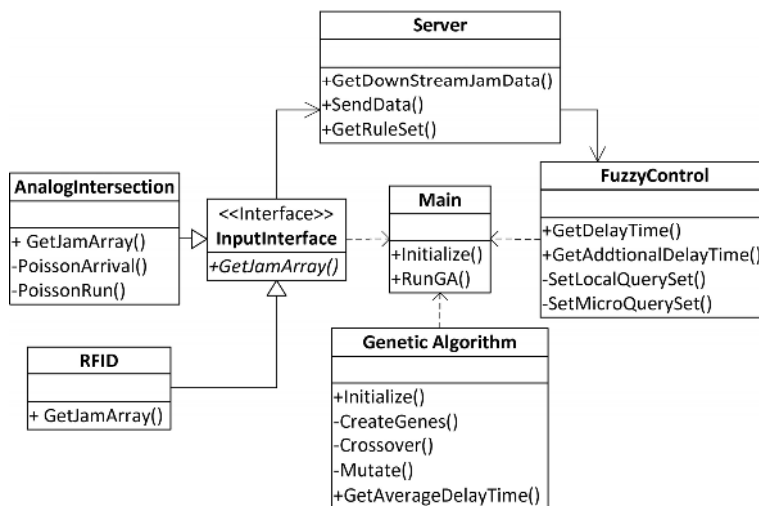


Fig. 6. Class overview

(4) Special Modes

The NeverStop system design should also cover the emergency like ambulance, fire truck, and even an accident. Some measures need to be taken to deal with these problems.

1. When an ambulance or fire truck comes through the intersection, all the other driveways should be paused. Due to that each car has an ID, and when the car passes the RFID device, the ID will be sent to the central server immediately. If the vehicle is recognized as an ambulance, EBox will offload a specific delay plan from server, and therefore emergent mode will be activated.

2. If a traffic accident happens, EBox will send an alarm signal to the server. Administrator should control the EBox from the server and inform this accident to the neighborhood intersections.

3.5 Input Module

1) RFID Device

Figure 7 illustrate the basic RFID system design based on RFID tags and the EBox devices. The components of the RFID System are composed of following components:

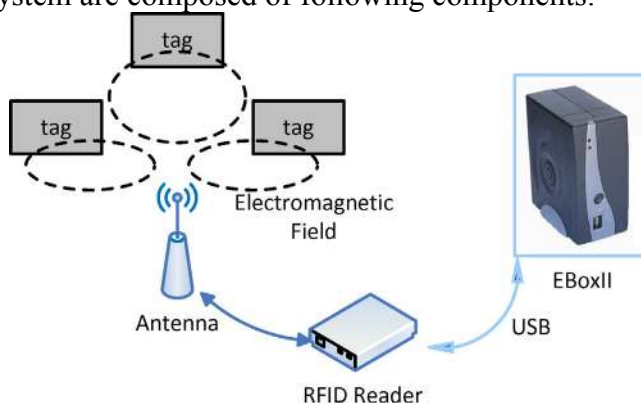


Figure 7. RFID System Design Specification

a) Tag

The electronic tag is made up of coupling elements and chips, and it stores the formatted information of the vehicles. Each tag has its unique coding mode to distinguish different vehicles. The

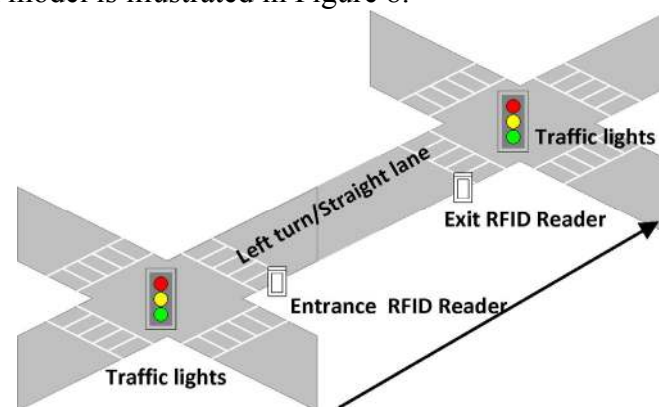
information solidified in the tag basically includes the registration number. In real usage, a tag should be attached on a vehicle.

b) Antenna

During the radio frequency communication, the antenna takes charge of transmitting the radio frequency signal between the tags and the RFID reader. The antenna must be small enough to attach on either the tag or the reader. Meanwhile it can transmit the max power throughout the tag. Due to the airborne interface, its polarization almost matches all query signals from the RFID reader. For the microwave RFID application, the antenna coupling mode is based on radar model.

c) RFID Reader

For an intersection with 4 directions, there are 3 lanes that lead the vehicles to turn left, go straight and turn right in each direction. Generally, in our four-phase intersection model, the lane for right turning is active always. That means any automobiles that tend to turn right on the lane will never barge up against the red light. As a consequence, our RFID Readers are fixed at both the left and the straight lanes, two on either one. The reader close to the traffic lights is used to record the vehicle flow out of the current lane, while the other one is used to record the vehicle flow into the current lane. According to the two record values, we can calculate the vehicle count that is passing to fuzzy control module. The intersection model is illustrated in Figure 8.



**Figure 8.** Intersection Model

When a RFID reader sends a request order, the order will be modulated to the RF working frequency and transmitted by the antenna. The RF tag responds the signal irradiated on its surface through the reflected echo signals. Thereafter, the RFID reader demodulates the echo signal and parses the solidified information. Then the useful vehicle flow data will be picked up to wait for a further disposal. On the other hand, the untouched information collection from tag must be returned to the application system, which forms the API (Application Program Interface) to the programmers. In our system, the reader connects with EBox II by USB connectors. The detailed communication structure is presented in Figure 9.

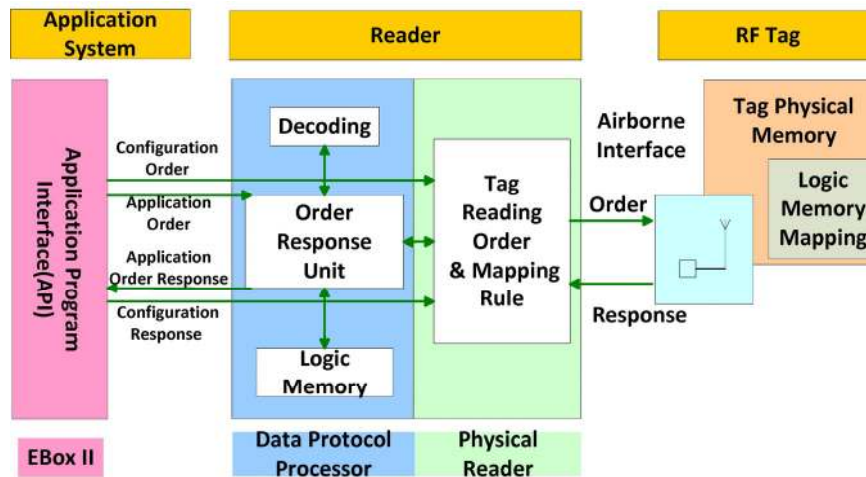


Figure 9. RFID Communication Structure

2) Poisson Arrival Simulation

In order to evaluate the status of real applications, we used Poisson arriving model to evaluate the feasibility and validity of the NeverStop system. Poisson arrival model has the following benefits in the initial stages of an integrated project:

First, it is not necessary to deploy the RFID readers in each lane; also the designers do not need to stick the tags on the cars. It saves plenty of time to design the system and integrate the modules.

Second, the simulation data is practically similar with the realistic data based on the mathematic model of producing vehicles.

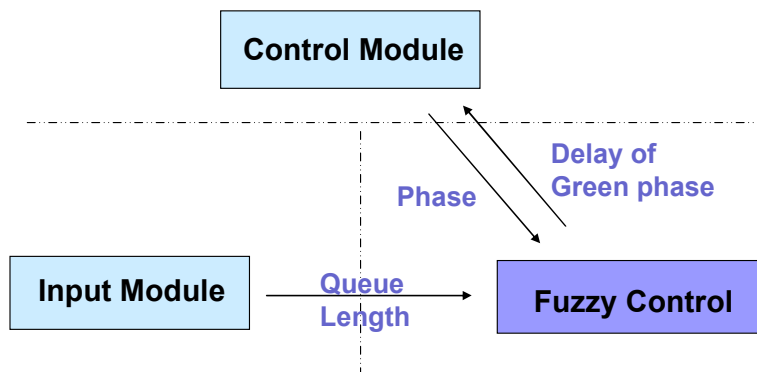
Third, we can modify the data by hands to test the stability and robustness of our algorithm. That offers us much help to improve and adjust our work ceaselessly, which has a significant advantages to the further phase with RFID devices.

4 Fuzzy Control and Genetic Algorithms

4.1 Fuzzy Rules Module

The conventional traffic management system always uses optimal design methodology, which first sets up a mathematic model, figures it out and then calculates a solution of the lights-distribution. However the above method is inadequate to cope with complex time-varying traffic situations. We are facing such a situation with various types of vehicles and high random, therefore the traffic system is challenging to be described by accurate mathematical models.

Fuzzy control method is adequate for qualitative modeling of complex systems. In order to avoid complex computation, we use the fuzzy control rules to operate the process. As a consequence, the burden of the system is alleviated and the real time property is boosted. Besides, the fuzzy control methods can be easily distributed to distinct intersections with different traffic conditions. Fuzzy control is used to provide the critical information delay time to system controller module. The processing flow is illustrated in Figure 10.

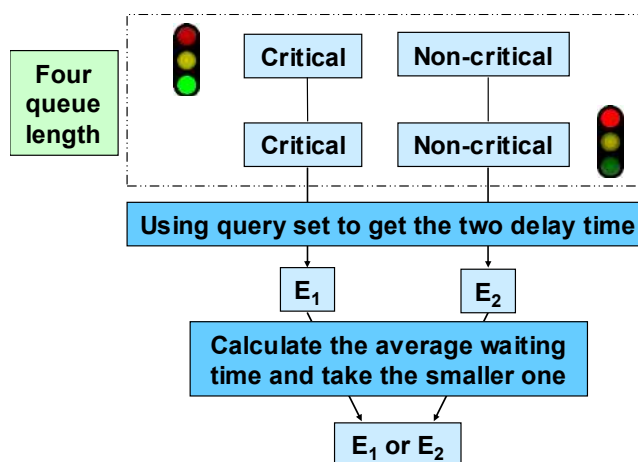


**Figure 10.** Processing Flow Chart of Fuzzy Control

In the flow chart of fuzzy control, the input is the accurate value of car flow data  $Q$ , and it can be received by the RFID device. Then we derive a fuzzy value  $Q_0$  of  $Q$ , and this step is called fuzzy operation. Then we get the delay time of current green phase using fuzzy control rules, but only a fuzzy value  $G_0$ . The last step we get the accurate value of the delay time  $G$  using fuzzy estimate.

Three data sets are required to complete the above process: the set of traffic volumes, the set of fuzzy control rules and the set of delay time, these three sets are corresponding to the above three steps. The set of traffic volumes helps to fuzzy the accurate car flow data into fuzzy value, for example the value '1' is transformed into 'very few' while '10' is mapped to 'many'. The set of fuzzy control rules represent the human reasoning process and there are two inputs and one output. If the congestion state of current phase is 'very few' and the congestion state of next phase is 'many' we will get an output 'short'. The set of delay time is used to transform the fuzzy value of delay time into accurate value, for example 'short' refers to 4 seconds approximately.

For every two inputs we need to calculate the delay time as above process, and in order to avoid repeated computation we calculate the delay time for all possible inputs when system is initialized. Thereafter the derived results form another set - the query set. While running, we can get the delay time only by using the formed query set. The coordination of fuzzy control is presented in Figure 11.



**Figure 11.** Coordination of Fuzzy Control

Fuzzy control has been used in the traffic schedule field for a long time. In this paper we take novel coordination fuzzy control and use genetic algorithm to evolve the proposed fuzzy control rules. By

using genetic algorithm we can update the fuzzy control query set while the NeverStop system is running and it does great help to improve the system performance.

Conventional fuzzy control methods determine congestion state only use the 'critical flow' of current phase and the next phase. We calculate the average waiting time of the cars and then determine which one to use - the critical one or the non-critical one.

#### 4.2 Genetic Algorithms

NeverStop system calculates the time by the query set, which is figured out by the set of traffic volumes, the set of delay time and the set of fuzzy control rules. The set of fuzzy control rules has a significant impact on the result, therefore we should find out appropriate fuzzy control rules to satisfy our needs and improve the effect of the rules. With the help of genetic algorithm and using average waiting time of the car as critical criterion, the rules will perform acceptable after 100 generations. In particular, the genetic algorithm is illustrated in following steps:

1) Generate the first generation and encapsulate it in an array list. One of these genes is specified, including all the membership values, while others are generated at random and may only consist of a few values.

2) Calculate the query set from the set of traffic volumes, the set of delay time and the set of fuzzy control rules.

3) Set up an Intersection Model, and maintain this model from a Poisson arrival speed and a leaving speed. Run the model for certain duration for several times. While running, the query set is used to manage the traffic. Calculate the average waiting time in the end.

4) Evolution between generations. The evolution procedure between generations is composed of two key operations: crossover and mutation.

*a) Crossover operator:* Select parent genes by the crossover probability. Generate two child genes from the parents. Considering the population num, we used uniform crossover operations. In the prototype demonstration, we set the probability of crossover to 80%. All the crossover operators are generated by random, coding in binary template.

*b) Mutation operator:* Rather than the crossover operator, we also select genes by the mutation probability. Then mutate the genes at random point, the probability of mutation of each representative chromosome is configured to 20%. Within each chromosome, every genome has a mutation probability of 5%.

*c) Cost function:* We use the Average Waiting Time (AWT) of each vehicle as the cost function. After each generation is generated, a new population is achieved. Thus, the AWT for each chromosome will be calculated separately. Then all the chromosomes will be sorted in order and prepared for the next generation.

5) Calculate the average waiting time as step 3. Sort all the genes by the fitness, select the genes whose average waiting time is shorter and go to the next generation.

## 5 Experiment Results and Analysis

We have built a simulation framework of the NeverStop system. The simulation architecture is composed of two key modules: 1) the NeverStop controller application that is implemented in EBoxII

embedded devices, with the .NET Compact Framework, and 2) the Poisson Arrival simulation model carving the detailed situation. Both Modules are implemented in C#, therefore the simulation could be easily extended and ported to other research platforms on various embedded devices. The simulated traffic flow results are provided with XML data format in SOAP web services.

5.1 Mathematical Analysis of Poisson Arrival Simulation

The Poisson probability density function is as following (fixing sampling time  $t^*$ ):

$$P_r(n) = \frac{\lambda^n}{n!} e^{-\lambda}, n > 0 \tag{1}$$

Assume  $n$  is the vehicle number,  $\lambda$  is a parameter that figures the number of vehicles arriving in sampling time  $t^*$ . Let  $\lambda = \alpha \cdot t^*$ . Here  $\alpha$  is the average arriving speed (veh/s) that depends on different road section and different time. We can work out a concrete value by a field survey. Taking the Shayang Road in Chongqing City of China as an example, it sets 15s as a sampling time and records 70 sets of data. Each dataset includes numbers of vehicles that go through the simulation section in sampling time and frequencies of these numbers that occurs in total 70 measures. According the data, we can work out the exception of arriving rate  $\lambda$ . Putting the  $\lambda$  value into the Poisson probability density function, we get a theoretical curve. Comparing the curve with the practical one, the two curves almost reach a superposition within the allowance of error. The Poisson density and distribution function is illustrated in Figure 12.

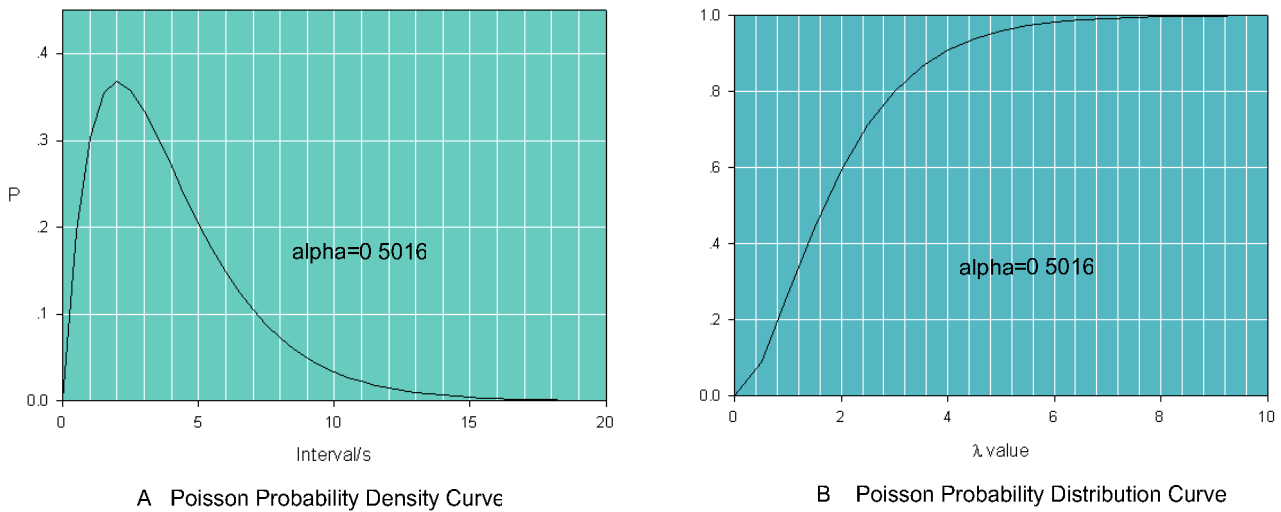


Figure 12. Poisson density and distribution function.

When  $n=1$  and keeping  $t^*$  as a variable,  $P_1(t^*)$  denotes a probability density function of interval time between any two successive cars.

$$P_1(t^*) = \alpha t^* e^{-\alpha t^*}, t^* > 0 \tag{2}$$

or

$$P_1(\lambda) = \lambda e^{-\lambda}, \lambda > 0, \lambda = \alpha \cdot t^* \tag{3}$$

Integrating the function above, yields the probability distribution function:

$$F(\lambda) = (-\lambda - 1)e^{-\lambda} + 1, \lambda > 0 \tag{4}$$

In general,  $F(\lambda)$  is monotonic increasing and positive definite. In particular, the poisson simulation algorithm is described in Algorithm 1:

ALGORITHM 1 POISSON SIMULATION ALGORITHM

```

Input: Car flow set T
Output: interval S,  $s \in S$  for each  $t \in T$ 

1 Initialize i = 1
2 for i < T.size do
3     p = random (0.0,1.0) // Get a random number p between 0.0 and 1.0
4     Let p = F( $\lambda$ ) derive  $\lambda$  //calculate by using Newton iteration method
5      $s_i = \lambda / \alpha$  // Figure the interval of ith and (i+1)th car
6     i++
7 endfor
    
```

According to the algorithm we could fully simulate the car flow by recording the interval time of two successive cars at the intersection. For example, when the light turns red, there are only arriving cars in the lane, and no cars leave. When the light turns green, besides the coming jam data, there is also car flow out of the intersection. In this occasion, we can simulate the congestion by selecting a reasonable leaving speed.

5.2 Results on Poisson Arrival Prediction and Practical Curve

We used component level testing (Black box approach) to ensure that each of the individual components was robust and performing optimally. NeverStop was tested on extreme cases and was subjected to conditions faced that might occur in actual use.

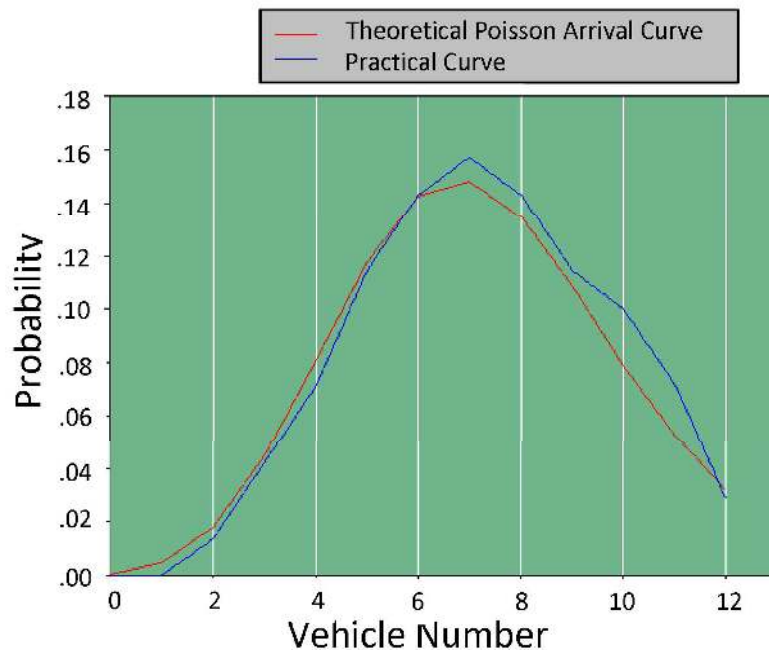


Figure 13 Vehicle Num-P Curve

The running time of Genetic Algorithm depends on the population number, the generation number,



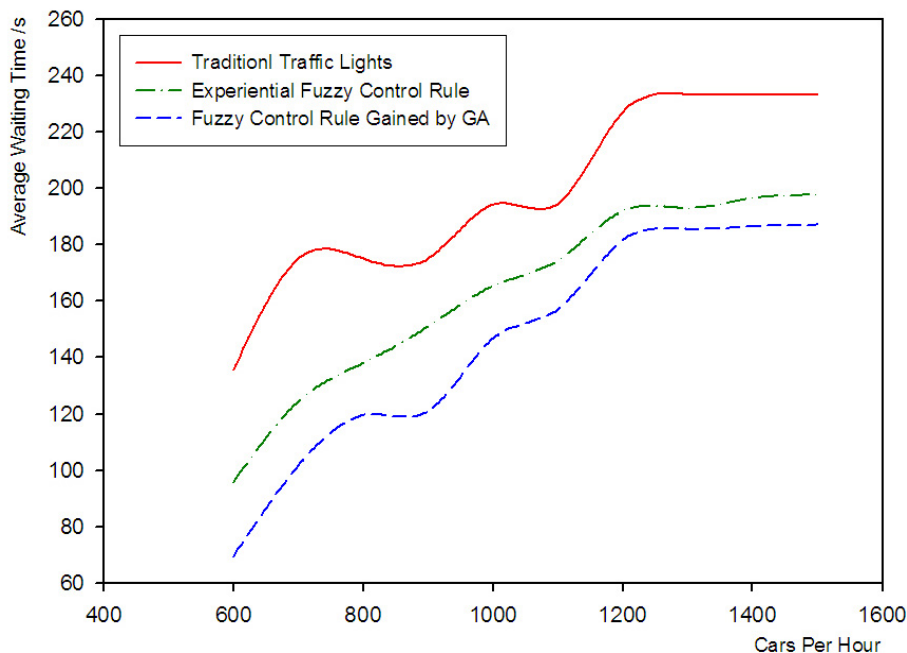
the running time and the frequency of the intersection model. Figure 13 presents the experimental results of the Poisson arrival simulation. From this Figure we can derive that the simulated Poisson procedure is very similar to the situation in the real world.

**Table 2** Genetic Algorithm Running Time

Population	Generation	Running Time	Running Frequency	Computing Time
100	50	600s	10	203s
200	50	600s	10	390s
100	100	600s	10	388s
100	50	1000s	10	334s

To prove the effectiveness of our algorithm, we compute average waiting time as our critical criterion. Our algorithm is superior to the conventional traffic lights, also has an advantage of the human experience. Table 2 illustrates the genetic algorithm running time. The computing time increases in line with the population, generation and running time respectively. When computing time increase to 390s at 200 population, 50 generation, 600s running time and 10 frequency. The lowest computing time occurs at the 100 populations and 50 generation.

The experimental results of the NeverStop system are shown in Figure 14. The results are achieved under the simulation that the cars coming at Poisson arrival model: the left-turn proportion is 20 percent, 80 percent moving straight.



**Figure 14** Experimental Results of the Fuzzy Controller with GA

5.2 Result of Algorithm Test

Figure 14 illustrates the experimental results, which includes the average waiting time for 1) traditional traffic lights system; 2) the traffic is controlled only by fuzzy control rules (as is described in Section 4.1); and 3) the traffic is controlled by both fuzzy control rules (in Section 4.1) and genetic

algorithm (in Section 4.2).

The X-axis represents the car arriving rate per hour, while the Y-axis is the average waiting time. Among the three different methods, fuzzy control rules gained by genetic algorithm have the least average waiting time, as is illustrated by the blue slashed line. As the arriving speed of cars increases, the average waiting time becomes longer simultaneously. Our genetic algorithm has better performance than the other two methods. When the arriving speed is 600, 900, and 1200 cars per hour, the efficiency of genetic algorithm is higher than conventional traffic lights by 48.72%, 31.42%, and 20%. Experimental results on prototyping system demonstrate the effectiveness of the NeverStop.

## 6 Conclusions and Future Work

In this paper, we have proposed an intelligent transportation system with RFID sensors NeverStop, which could control the traffic lights at intersection automatically. It utilizes the state-of-the-art fuzzy control method and genetic algorithms to leverage the waiting time for the traffic lights automatically. Experimental results on the prototype system in EBox-II demonstrate NeverStop can reduce the average waiting time significantly.

There are numerous future directions worth pursuing. First, improved genetic algorithms will be important to efficiently get a tuning plan. Second, we also plan to study about the arrival relationship with other mathematical models, exploring potential effect on the scheduling plans. Finally, as our model only considers the near-perfect readings from the sensors, it would be our future work to incorporate uncertainty into the approach.

## Acknowledgements

This work was supported by the National Science Foundation of China under grants (No. 61379040, No. 61272131, No. 61202053), Jiangsu Provincial Natural Science Foundation (No. SBK201240198), Natural Science Foundation of Fujian Province of China (No.2014J05075), Open Project of State Key Laboratory of Computer Architecture, Institute of Computing Technology, Chinese Academy of Sciences (No. CARCH201407), and the Strategic Priority Research Program of CAS (No. XDA06010403). The authors deeply appreciate many reviewers for their insightful comments and suggestions. We would like to thank Dr. Bo Chen, Dr. Fuxin Ren and Dr. Wenyuan Yin for their valuable contribution on the engineering and discussions.

## References and Notes

- [1].Doug Howe, Maria Costanzo, et al., *Big data: The future of biocuration*. Nature, 2008. **455**(7209): p. 47-50.
- [2].Chao Wang, Xi Li, et al., *Heterogeneous Cloud Framework for Big Data Genome Sequencing*. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 2014. **12**(1): p. 166-178.
- [3].Jelena Fiosina, Maxims Fiosins, et al., *Big Data Processing and Mining for Next Generation Intelligent Transportation Systems*. Jurnal Teknologi, 2013. **63**(3).
- [4].Chao Wang, Xi Li, et al. *An Intelligent Transportation System using RFID based Sensors*. in *The 15th IEEE International Conference on High Performance Computing and Communications (HPCC 2013)*. 2013.

- [5].Chao Wang, Xi Li, et al. *SODA: Software Defined FPGA based Accelerators for Big Data*. in *Design, Automation & Test in Europe*. 2015.
- [6].Jelena Fiosina, Maksims Fiosins, et al. *Mining the Traffic Cloud: Data Analysis and Optimization Strategies for Cloud-Based Cooperative Mobility Management*. in *Int. Sym. on Management Int. Systems. Adv. in Int. Syst. and Comp.* 2013: p. 25-32.
- [7].J. Shawe-Taylor, T. de Bie, et al., *Data mining, data fusion and information management*. IEE Proceedings - Intelligent Transport Systems, 2006. **153**(3): p. 221-229.
- [8].Junping Zhang, Fei-Yue Wang, et al., *Data-Driven Intelligent Transportation Systems: A Survey*. Intelligent Transportation Systems, IEEE Transactions on, 2011. **12**(4): p. 1624-1639.
- [9].Chin-Feng Lai, Yueh-Min Huang, et al., *Adaptive Body Posture Analysis Using Collaborative Multi-Sensors for Elderly Falling Detection*. IEEE Intelligent Systems, 2010. **25**(2): p. 20-30.
- [10].Ousmane Diallo, Joel J. P. C. Rodrigues, et al., *Real-time Data Management on Wireless Sensor Network: a Survey*. Journal of Network and Computer Applications, 2012. **35**(3): p. 1013-1021.
- [11].T. T. Tchakian, B. Basu, et al., *Real-Time Traffic Flow Forecasting Using Spectral Analysis*. Intelligent Transportation Systems, IEEE Transactions on, 2012. **13**(2): p. 519-526.
- [12].G. A. Davis and N. L. Nihan, *Nonparametric regression and shortterm freeway traffic forecasting*. J. Transp. Eng, 1991. **117**(2): p. 178-188.
- [13].P. C. Vythoulkas. *Alternative approaches to short-term traffic forecasting for use in driver information systems*. in *in Proc. 12th Int. Symp. Traffic Flow Theory Transp.* 1993: p. 485-506.
- [14].D.-H. Lee W. Zheng, and Q. Shi., *Short-term freeway traffic flow prediction: Bayesian combined neural network approach*. J. Transp. Eng., 2006. **132**(2): p. 114-121.
- [15].X. Jiang and H. Adeli, *Dynamic wavelet neural network model for traffic flow forecasting*. J. Transp. Eng., 2005. **131**(10): p. 771-779.
- [16].Huijing Zhao, Jie Sha, et al., *Detection and Tracking of Moving Objects at Intersections Using a Network of Laser Scanners*. Intelligent Transportation Systems, IEEE Transactions on, 2012. **13**(2): p. 655-670.
- [17].Jiangtao Cao, Ping Li, et al., *An Interval Fuzzy Controller for Vehicle Active Suspension Systems*. Intelligent Transportation Systems, IEEE Transactions on, 2010. **11**(4): p. 885-895.
- [18].J. E. Naranjo, C. Gonzalez, et al., *Adaptive fuzzy control for inter-vehicle gap keeping*. Intelligent Transportation Systems, IEEE Transactions on, 2003. **4**(3): p. 132-142.
- [19].Ondrej Linda and Milos Manic, *Online Spatio-Temporal Risk Assessment for Intelligent Transportation Systems*. Intelligent Transportation Systems, IEEE Transactions on, 2011. **12**(1): p. 194-200.
- [20].Palesi Maurizio and Givargis Tony, *Multi-objective design space exploration using genetic algorithms*, in *Proceedings of the tenth international symposium on Hardware/software codesign*. 2002, ACM: Estes Park, Colorado.
- [21].L.M. Mourelle, R.E. Ferreira, et al., *Migration Selection of Strategies for Parallel Genetic Algorithms Implementation on NoCs*. International Journal of Electronics, 2010. **97**(10): p. 1227-1240.
- [22].Yang Li and Li Guo. *An Efficient Network Anomaly Detection Scheme Based on TCM-KNN Algorithm and Data Reduction Mechanism*. in *Information Assurance and Security Workshop, 2007. IAW '07. IEEE SMC*. 2007: p. 221-227.
- [23].Lin Chu-Hsing, Yu Jui-Ling, et al. *Genetic Algorithm for Shortest Driving Time in Intelligent Transportation Systems*. in *Multimedia and Ubiquitous Engineering, 2008. MUE 2008. International Conference on*. 2008: p. 402-406.
- [24].Hitoshi Kanoh and Nobuaki Nakamura. *Route Guidance with Unspecified Staging Posts Using Genetic Algorithm for Car Navigation Systems*. in *in Proc. Intelligent Transportation Systems*. 2000: p. 119-124.
- [25].Leena Singh, Sudhanshu Tripathi, et al., *Time Optimization for Traffic Signal Control Using Genetic Algorithm*. International Journal of Recent Trends in Engineering, 2009. **2**(2): p. 4-6.

- [26].J. Arunadevi, A. Johnsanjeevkumar, et al. *Intelligent Transport Route Planning Using Parallel Genetic Algorithms and MPI In High Performance Computing Cluster*. in *Proceedings of the 15th International Conference on Advanced Computing and Communications*. 2007: IEEE Computer Society: p. 578-583.
- [27].J. Lee, B. Abdulhai, et al., *Real-Time Optimization for Adaptive Traffic Signal Control using Genetic Algorithms*. *Journal of Intelligent Transportation Systems*, 2005. **9**(3): p. 111-122.
- [28].Liang Zou, Jianmin Xu, et al., *Application of Genetic Algorithm in Dynamic Route Guidance System* *Journal of Transportation Systems Engineering and Information Technology*, 2007. **7**(3): p. 45-48.
- [29].Pun-Cheng L. S. C., *An Interactive Web-Based Public Transport Enquiry System With Real-Time Optimal Route Computation*. *Intelligent Transportation Systems, IEEE Transactions on*, 2012. **13**(2): p. 983-988.
- [30].S. Shekhar, C. T. Lu, et al. *CubeView: a system for traffic data visualization*. in *Intelligent Transportation Systems, 2002. Proceedings. The IEEE 5th International Conference on*. 2002: p. 674-678.
- [31].C.-T. Lu, L. N. Sripada, et al., *Transportation data visualization and mining for emergency management*. *Int. J. Crit. Infrastructures*, 2005. **1**(2/3): p. 170-194.