International Journal of Engineering, Mathematical and Physical Sciences
ISSN: 2517-9934
Vol:6, No:1, 2012

A Traffic Simulation Package Based on Travel Demand

Tuong Huan Nguyen, Quoc Bao Vo, and Hai L. Vu

***Abstract*—**In this paper we propose a new traffic simulation package, TDMSim, which supports both macroscopic and microscopic simulation on free-flowing and regulated traffic systems. Both simulators are based on travel demands, which specify the numbers of vehicles departing from origins to arrive at different destinations. The microscopic simulator implements the car-following model given the pre-defined routes of the vehicles but also supports the rerouting of vehicles. We also propose a macroscopic simulator which is built in integration with the microscopic simulator to allow the simulation to be scaled for larger networks without sacrificing the precision achievable through the microscopic simulator. The macroscopic simulator also enables the reuse of previous simulation results when simulating traffic on the same networks at later time. Validations have been conducted to show the correctness of both simulators.

***Keywords*—**Macroscopic, Microscopic, Simulation, Traffic, Travel demand, Fundamental diagrams.

I. Introduction

TRAFFIC simulation models have been extensively used in various application areas including traffic engineering, driver behavior modeling, studies in public transport, impact of traffic on environment and emission, infrastructure analysis and development. They provide an important tool for modeling traffic behaviors in dynamic traffic systems, testing ideas and solutions to traffic problems before deploying them into reality. Currently, traffic simulators are mainly categorized into macroscopic, mesoscopic and microscopic simulators. Macroscopic simulators, in general, use mathematical models for describing densities and flows of vehicles throughout a traffic network. In other words, they view traffic as flows rather than tracking each individual vehicle. Microscopic simulators, on the other hand, track the behaviors of individual vehicles and govern the interactions between them. Mesoscopic simulators employ an intermediate level of detail, for instance tracking individual vehicles but not their interactions. Due to their natures, microscopic simulators provide more detailed and precise results than macroscopic and mesoscopic simulators. However, they are very computational intensive and usually not suitable for very large networks in comparison to macroscopic simulators which produce results with lesser details and precision but are better in dealing with scalability issue.

In this paper, we propose a new traffic simulation package, TDMSim (Travel-based Demand Modeling Simulator), consisting of a microscopic simulator and a macroscopic simulator, which is built in integration with the microscopic simulator. Both simulators are capable of simulating traffic on free-flowing and regulated traffic systems and based on travel demands, which specify the numbers of vehicles departing from origins to go to different destinations in a certain period of time. With the microscopic simulator, our purposes are to model the behaviors of vehicles, the operation of traffic light systems, the capacity of roads and more importantly, the interaction between vehicles. The microscopic simulator also models the scenarios in which there is communication between vehicles and the central system to simulate the situations when vehicles need to be rerouted due to incidents in the network (i.e. road blocked, accidents or completely jammed roads). Our main objective with the macroscopic simulator is to facilitate the ability of reusing previous results of simulations when simulating traffic on the same networks at later times. That helps shortening the simulation time and minimizes computational resources.

Many existing works have been done on traffic simulation. However most of them have different objectives and do not fully address our concerns. FreeSim [1] shares the same objective of facilitating the communications between vehicles and the central system. However, it is built to simulate free-flowing traffic only. Although FreeSim is an open source project, extending it to enable the traffic simulation on regulated systems is not straightforward. In addition, none of the existing simulators enable the reuse of previous results in macroscopic simulation. Due to these reasons, we build a new traffic simulation package TDMSim to fulfill the objectives.

The remainder of the paper is organized as follows. In section II we present the related works in more details. In section III we describe the microscopic simulation models and present the validation results of the microscopic simulator. The macroscopic simulator and its validation results will be discussed in Section IV. In section V, we conclude our contributions and point out the future work.

II. Related Work

There are many existing traffic simulation packages currently being used and developed. In this section, we choose 7 popular applications [2], [1], [3], [4], [5], [6], [7] for the comparison with TDMSim. This comparison is an extension of the one done in [1] but only includes the 4 aspects of interest: Transportation networks, vehicle models, and the ability to support communications between vehicles and the central system, and the reuse of previous simulation results. The comparison data is presented in Table I.

Authors are with Faculty of ICT, Swinburne University of Technology, Melbourne, Australia (e-mail: {tuongnguyen, bvo, hvu}@swin.edu.au).

TABLE I
COMPARISON BETWEEN THE PROPOSED PACKAGE AND THE EXISTING APPLICATIONS

Package	Transportation networks	Vehicle models	Communications between vehicles and central system	Ability of reusing previous results
CORSIM/TSIS	Free-Flowing, regulated	Micro	No	No
FreeSim	Free-flowing	Macro and Micro	Yes	No
MITSIM	Free-flowing, regulated	Micro	No	No
PARAMICS	Free-flowing, regulated	Micro	No	No
RENAISSANCE	Free-flowing	Macro	No	No
VATSIM	Free-flowing, regulated	Micro	No	No
VISSIM	Free-flowing, regulated	Macro and Micro	No	No
TDMSim	Free-flowing, regulated	Macro and Micro	Yes	Yes

The "Package" field contains all the traffic simulation packages included in this comparison. The "Transportation networks" field shows the type of roadways that the applications support. As presented in the table, apart from FreeSim and RENAISSANCE, all other packages support both free-flowing and regulated systems, which are the traffic systems with light signals and toll booths and so on [1].

The "Vehicle models" fields present the simulation models supported by the applications. Except for RENAISSANCE which only supports macroscopic simulation, all packages allow microscopic modeling of vehicles. However, CORSIM/TSIS, MITSIM, PARAMICS and VATSIM only allow microscopic simulation. FreeSim, VISSIM and TDMSim allow both types of models.

Among the packages, only FreeSim and TDMSim facilitate the communications between vehicles and the central system. In addition, none of the packages, except for TDMSim, allow the reuse of simulation results.

Each of the listed packages has its own strengths and some of them have been widely used. However none of them meets our purposes. FreeSim does support the communications between vehicles and the central system. It only focuses on free-flowing traffic systems however. In addition, one of our objectives is to improve the performance of macroscopic simulation by shortening simulation time and minimizing computational resources, by enabling the reuse of existing simulation results. Those are the main reasons why we build our own simulation package.

III. MICROSCOPIC SIMULATOR

In the current traffic simulation studies, there are 2 main approaches of microscopic traffic models: one is cellular automata and the other is car-following. In cellular automata approach, which is described in [8, 9, 10], each lane are divided into cells that can either be empty or occupied by only one vehicle. The vehicle movement is controlled by taking account of the rules governing the occupation or liberation of these cells. This approach generally considers simple microscopic models and does not require very intensive computation [11]. The other approach, car-following models [12], [13], [14], treat lanes as continuous sets of points and vehicles' positions can be at any of them. This approach provides more realistic modeling of driver and vehicular behavior as it allows better control of vehicle movement [11]. However, the disadvantage of car-following approach is that, it usually requires more computational resources in comparison to cellular automata.

We choose car-following approach for the microscopic simulator to achieve greater accuracy. In addition, the disadvantage regarding high computational resource consumption can be overcome in the proposed macroscopic simulator, which only uses the microscopic simulator in small sections of the road networks (this will be described in more details in the section of Macroscopic Simulator).

The remainder of this section is structured as follows. We will first describe the model for traffic environment. The traffic light regulation model will be covered in the next sub-section. The model for vehicle behaviors will also be presented. In the last sub-section, we describe our microscopic simulator and the results of validation.

A. Traffic Environment Model

For the traffic simulation to be performed, the geometric information and traffic demand of the network being simulated need to be known in advance. The traffic environment is modeled by the following elements:

1. Links

Represent one-way streets between pairs of intersections. A link is modeled by the number of lanes it has, the dedicated lanes for turning right, left and going straight, the maximum velocity allowed, the link's length and Green phase and Red phase's lengths of the traffic light at the end of the link (this will be described in more details in the traffic light regulation model sub-section).

2. Nodes

Nodes represent intersections in a traffic network.

3. Origin-Destination Pairs (OD-pairs)

An OD-pair represents a pair of origin and destination of vehicles in a traffic network. In other words, each vehicle, when joining a network, must have an origin where it departs and a destination where it aims to reach, the destination of that vehicle never changes during the time it travels through the network.

4. Routes

Each OD-pair is associated with a number of different routes connecting the origin to the destination. Each route is represented by a set of consecutive links. The length of a route is calculated as the total length of the links it contains.

5. Travel Demand

Each OD-pair has a travel demand, which is represented by the number of vehicles wanting to travel from an origin to a particular destination in one minute.

B. Traffic Light Regulation Model

Each node on the network has a traffic light controller, which regulates the exit of vehicles on the relevant links (links having this node as their ending points). In the model, there are only 2 traffic light statuses, Green and Red. Instead of using amber light, we implement a period, called buffered time, which lies between Green and Red phases of a traffic light. In other words, when the Green period of a traffic light at the end of a link is over, it switches to the buffered period (which is a few seconds long). During this period, no vehicles are permitted to exit the link (as in Red period) and the traffic lights on other relevant links remain unchanged. When the buffered period is over, it switches to the Red period. At this time, traffic lights on other relevant links, which are currently in Red periods, switch to their Green phases. The idea of using this buffered time is to enforce a period in which no vehicle on any of the relevant links is permitted to exit, to avoid accidents for drivers who exited just before the end of Green phases (this has been being used in many traffic systems in reality). It can be seen that from the perspective of a link, the buffered time is part of its Red period (the traffic light is also red in the buffered time). The only difference is that, in buffered time, the traffic lights on other links are also red while in Red phase, at least one of the other links has green light.

Given a node, it can be established that:

$$t_i = t_{Gi} + t_{Ri} + n.t_B$$

Where:

t_i is the cycle time at the end of link i, it is also the total cycle time of all the traffic lights at the node;

t_{Gi}, t_{Ri} is the length of Green and Red periods of the traffic light at the end of link i respectively;

t_B is the length of the buffered period of the traffic light at the end of each link (buffered times of traffic lights at the same node are always the same);

n is the number of streets associated with the node.

Depending on the number of streets passing through a node, the traffic light controller has a different number of phases. A streetpassing through a node can contains 1 or 2 links associated with that node. If 2 links share the same street, they share the same Green, Red and Buffered period. A phase is defined as the period that whenever the traffic light controller switches from one phase to another phase, there is at least one of those relevant links which has its traffic light status changed. Fig. 1 presents an example of how the traffic controller works.

According to Fig. 1, there are 2 streets and 4 links associated with node$N$. Links Lk1 and Lk2 belong to street S1 and Lk3 and Lk4 belong to street S2. Therefore, the traffic light controller has 4 phases (the number of phases is always double the number of streets). The phases and changes of the traffic light on each link are presented in Table II.

At two-street intersections, Green period of a street is Red period of another and vice versa. In buffered time, all traffic lights are red. At more-than-two-street intersections, when traffic lights on one street is green, all other streets have red lights. Then the lights on that street switch to buffered period, then to Red phase, the next street has Green lights. This process is repeated forever.

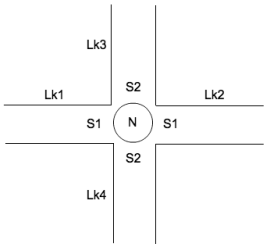


Fig. 1 Example of how traffic light controller works

TABLE II
PHASES OF TRAFFIC LIGHT CONTROLLER

Phase	Street S1 Link Lk1 & Link Lk2 Traffic light	Street S2 Link Lk3 & Link Lk4 Traffic light
1	Green	Red
2	Red (buffered time)	Red
3	Red	Green
4	Red	Red (buffered time)

TABLE III
VEHICLE PARAMETERS

Parameter	Value
Vehicle length	4 meters
Acceleration rate	$\frac{25}{9}\ m/s^2$ (Accelerate from 0km/h to 100km/h in 10 seconds)
Maximum deceleration rate	$\frac{250}{27}\ m/s^2$ (Decelerate from 100km/h to 0 in 3 seconds)
Normal deceleration rate	$\frac{sp}{2}\ m/s^2$ (sp is the current speed of the vehicle, measured in m/s) (Vehicles would stop after 2 seconds since it started to break)
Safety distance when stopping in a queue (S_{stop})	1 meter
Safety distance to the last vehicle on a lane for another vehicle to enter (S_{enter})	3 meters
Safety distance to	3 meters

C. Vehicle Model

The vehicle model defines the behaviors of vehicles when they travel through road networks including acceleration, deceleration, lane switching, queuing and so on. The behaviors of a vehicle are decided by the vehicle model based on the traffic conditions surrounding the vehicle and the capacity and restrictions of the link it is traveling on. The following sub-sections describe the parameters associated with the vehicles. The vehicle behavioral criteria will also be presented.

1. Vehicle Parameters

There is only one type of vehicles currently covered by the model. The parameters are presented in Table III.

2. Vehicle Model Behavioral Criteria

This section presents how a particular vehicle acts in response to different traffic conditions when it travels through a road network. We use the term *controlled vehicle* to refer to the vehicle on which the model is currently takes decisions.

- *Behaviors when traveling on links*

When moving on a link, a vehicle needs to adapt to the link's restrictions. When decided the next link to join after exiting the current link, the vehicle must switch to the appropriate lanes (i.e. the most left lane is reserved for vehicles which want to turn left after exiting the link).

In general, a vehicle always tries to travel as fast as possible while considering the maximum speed allowed on the link.However, if there is a preceding vehicle and the distance to that vehicle is less than the safety distance (S_{tvl}), the controlled vehicle first looks for an appropriate adjacent lane to switch to if the trafficcondition on that lane allows it to maintain the current speed. If there is no such lane, the vehicle decelerates to maintain the safety distance to the preceding vehicle. However, the controlled vehicle would continuously tries to change lane if there is a chance. The conditions for the controlled vehicle to switch to another lane include:

1) The lane the controlled vehicle is about to switch to must be the one that allows it to join the link that it decided to join after exiting the current link (i.e. if it currently decides to go straight, it wouldn't choose the left most lane to switch to if that lane is reserved for vehicles that turn left).
2) The potential position of the controlled vehicle on the new lane is "safety", which means it doesn't violate the safety distance requirement with any of the existing vehicles on that lane. This is illustrated in Fig. 2.

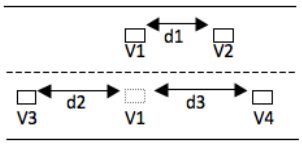


Fig. 2 Lane Switching

In Fig. 2 assume that the distance between vehicles V1 and V2 (d1) is less than the safety distance (S_{tvl}). V1 checks the adjacent lane to determine if it can switch to in order to avoid decelerating. On the adjacent lane there are 2 vehicles, V3 and V4. d2 and d3 are the distance between the potential positionof V1 if switching to that lane and vehicles V2 and V3 respectively. V1 only switches to the lane if:

$$\begin{cases} d2 > S_{tvl3} \\ d3 > S_{tvl1} \end{cases}$$

Where S_{tvl3}, S_{tvl1} is the safety distance between V3, V1 and any vehicle in front of them respectively (with their current speeds).

In cases when the potential position of V1 is in front or behind all the existing vehicles on the lane, obviously one of the conditions can be omitted.

- *When joining a queue*

If a vehicle encounters a queue, it starts decelerating as soon as the distance to the last vehicle in the queue drops to the safety distance (S_{tvl}). When the controlled vehicle stops in the queue, the distance to the preceding vehicle is S_{stop}.

- *When leaving a queue*

If the controlled vehicle is not at the head of a queue, it can only start leaving the queue if the distance to the preceding vehicle reaches S_{leave}. Fig. 3 illustrates this situation:

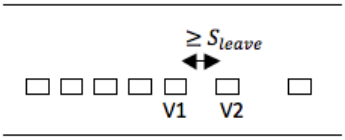


Fig. 3 Vehicle leaving queue

In Fig. 3, vehicles V1 and V2 are in a queue at the beginning. V2 starts leaving and increases the distance to V1. Until when the distance reaches S_{leave}, V1 can start leaving.

The case when the controlled vehicle is at the head of a queue is described in the next sub-section.

- *When vehicle is about to join a link*

If the controlled vehicle is at the head of a queue and allowed to exit (i.e. traffic light turns to green), it needs to determine if the available space on that link is enough for it to join "safely", which means there is at least one lane on which the distance from the last vehicle to the beginning of the link is not less thanS_{join}. If there is not enough space, it keeps staying at the head of the queue and continuously checks for an opportunity to join the link.

If the controlled vehicle is currently moving, the check for available space on the next link is done when the vehicle reaches the end of the current link. If there is not enough space, the vehicle stops there and waits until when the space is cleared out. Otherwise, it joins the next link and adapts its velocity to the maximum speed allowed on the link.

If a vehicle departs from its origin, it starts with the speed of zero and accelerates to travel as fast as possible (while considering the surrounding traffic and the link's restrictions).

D.Microscopic Simulator and Validation Results

We developed a microscopic simulator (MicroSim) based on the described models. MicroSim is a real-time simulator adopting the car-following approach. In MicroSim, each vehicle is treated as an individual object and controlled by the vehicle model. Vehicles are created at their origins and put into links based on travel demands, using Poisson random process. They can communicate with the central system about their current positions and speeds. The central system, with the data provided by vehicles, can broadcast alert messages to all vehicles in the network in cases of incidents (i.e. road blocked, accidents…). The central system can also provide the information regarding the better routes for each individual vehicle, based on the data it holds, if the vehicle requests.

We validate the microscopic simulation models by running MicroSim to obtain traffic fundamental diagrams, which are graphical representations of the relationships between speed

and flow, flow and density and density and speed. They are widely used by transportation professionals to gain insights into the three traffic fundamental characteristics and also provide a method for validating the correctness of traffic modeling [15]. The accuracy of MicroSim is validated by the comparisons between the obtained fundamental diagrams and the theoretical fundamental diagrams (TFDs) described by Greenberg [16] and Newell [17] (known as Greenberg's and Newell's fundamental diagrams respectively). The simulation's results are considered accurate if they obtained diagrams share the same patterns with TFDs.

In order to generate fundamental diagrams, we record density-flow mappings using a validation program. The validation program runs MicroSim with different configurations of the network described in Fig. 4. Each configuration is defined by the combination of the number of lanes on each of the four links. For instance, the setting in which Lk1 and Lk2 has 2 lanes each, Lk3 has 1 lane and Lk4 has 3 lanes is one configuration.

The densities and flows recording are done on Lk1 (with density is the density of Lk1 and flow is the total flows of vehicles from Lk1 to Lk2, Lk3 and Lk4). With a particular configuration, we define a set of targeted density values to record the corresponding flows (5%, 10%, 15%, 20%, 30%, 40%, 50%, 60% and 70% of the maximum density on a link[1]). For each targeted density of Lk1, there are a number of different scenarios, which are defined as the combinations of targeted density on each of the downstream links (Lk2, Lk3, Lk4). The targeted densities in each scenario need to be reasonable and achievable given the targeted density on Lk1. The density-flow we collect is really the mapping between each targeted density of Lk1 and a range of flows recorded from all the associated scenarios.

- Each link is 500 meters long and has a 70km/h speed limit
- OD-pairs: O-D1, O-D2 and O-D3
- Traffic light cycles and travel demands are set by the Validation Program, depending on the targeted densities and scenarios

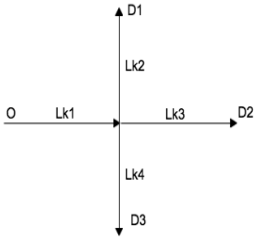


Fig. 4 The network used for MicroSim validation

In order to record data for a particular configuration, the validation program invokes MicroSim for each targeted density of Lk1 and each of the associated scenarios. Depending on the targeted density and scenario, an initial set of travel demands for the three OD-pairs is selected. The initial traffic light cycles are also set. The validation program is run to build up the density on each link to its targeted density and maintain that density for a period of time for recording. Densities on links are built and maintained by adjusting travel demands, traffic light cycles' lengths and exit rates at the destinations. Once the program detects that the density of each link is stable, it starts record the vehicle flows for 5 continuous traffic light cycles. With the collected data, the density-flow fundamental diagram can be generated. The speed-flow and density-speed diagrams are produced based on the density-flow data[2].

We conducted the validation on a number of different configurations. In all cases, the obtained fundamental diagrams show the similar patterns to TFDs'. Fig. 5 (a) and (b) present the fundamental diagrams that we collected for the configurations of (3,2,3,2) and (2,1,2,1) (the configuration $(a_1, a_2, a_3, a_4,)$ means that Lk1, Lk2, Lk3, Lk4 has a_1, a_2, a_3, a_4 lanes respectively).

IV. Macroscopic Simulator

Macroscopic simulation models share some sub-models with the microscopic models, including traffic environment and traffic light regulation models. However, while microscopic models capture the behaviors of individual vehicles in details, the macroscopic models view traffic as a continuous flow. The proposed macroscopic simulator (MacroSim) aims to provide the estimation of densities and traffic flows on links and travel time of routes at any point in time. In addition, it facilitates the ability of reusing existing simulation results on simulated networks at later times with the help of a database. In this section, we firstly describe the density-flow model. We then discuss the architecture of MacroSim. The section ends with the validation results.

A. Density-Flow Model

The main concerns of macroscopic models are densities and traffic flows on the links of a traffic network, which are modeled by the following elements:

1. Density

The term *density* is referred to as the distribution of vehicles on a link. Depending on a vehicle's status and position, it is categorized as *queuing* or *moving*. The categorization is based on a parameter called *queue position. Queue position* is defined as the position of the first vehicle (count from the starting point to the ending point of a link) which is stopping at an intersection during Red light cycle or waiting for the vehicle in front of it to move for it to be able to move. If a vehicle's position is behind the *queue position*, it is categorized as *moving* (even when it is not stopping but moving to leave the link); it is classified as *queuing* otherwise. In the remainder of the paper, we denote $a$is the number of *queuing* vehicles, b is the number of *moving* vehicles and q is the *queue position* on a lane. Therefore, $a + b$ is the total number of vehicles on that lane.

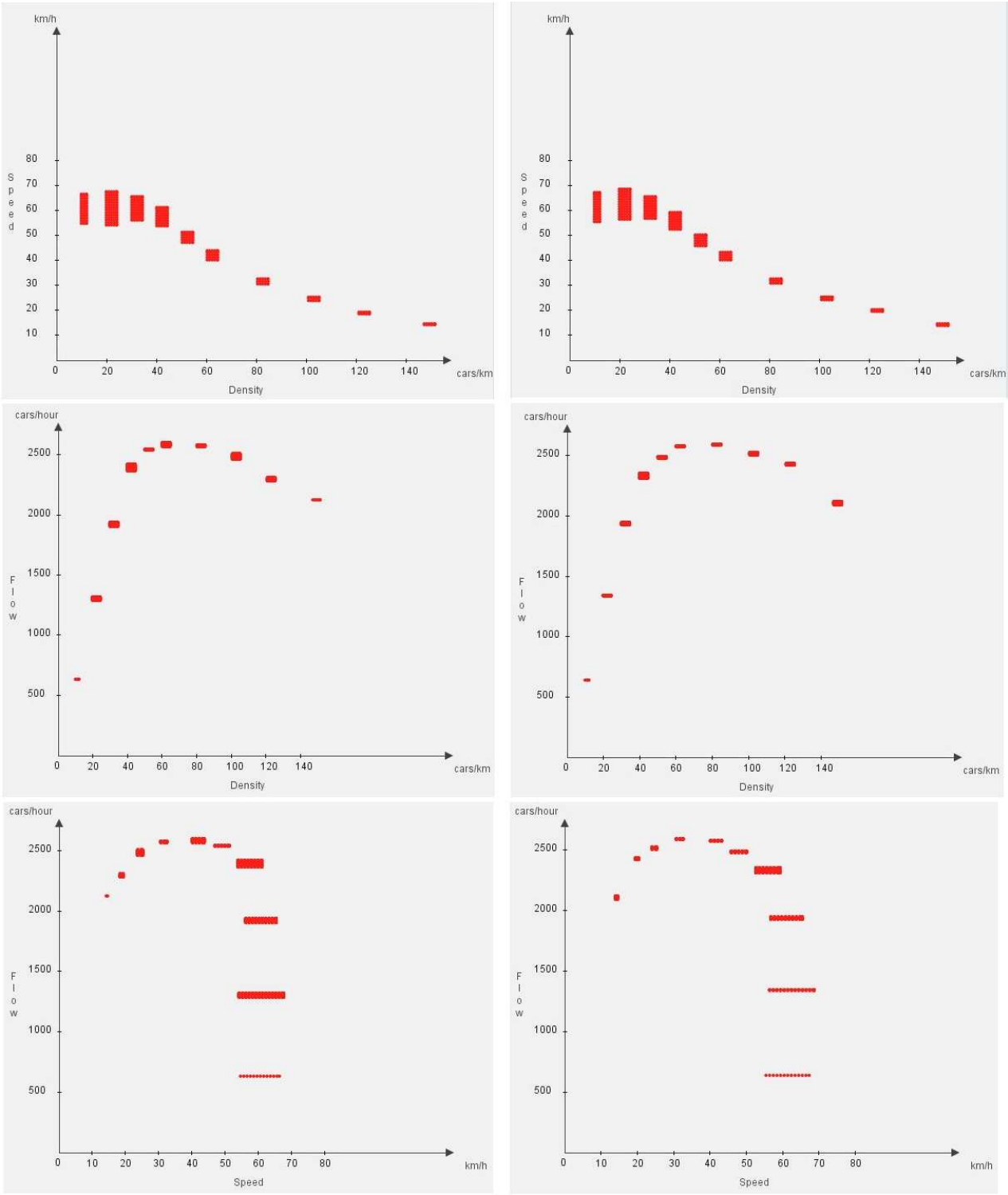
Fig. 6 provides the examples to illustrate the concept of density. In Fig. 6 (a), vehicle 1, 2, 3 are stopping at the intersection while the vehicles behind are moving. Therefore, $a = 3, b = p - 3$ and q is the position of vehicle 3. In Fig. 6 (b), vehicles 4, 5, 6 are stopping while vehicles 1, 2, 3 are

[1]Densities higher than 70% are generally unrealistically to be generated as in such cases the link is so dense and nearly completely jam. The variation allowed here is 2% (i.e. if the targeted density is 30%, the validation program doesn't need to generate the density of the upstream link at exactly 30%, but any value from 28% to 32% is acceptable).

[2] Using the formula $flow = density.speed$

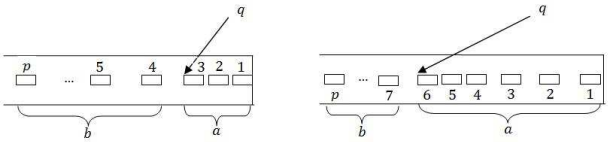
moving to leave the link. They are all considered *queuing* vehicles. Therefore $a = 6, b = p - 6$ and q is the position of vehicle 6 (with p is the total number of vehicles).

We represent the *density* of a particular link by a set of parameters $(\{a_1, b_1, q_1\}, \{a_1, b_1, q_1\}, \ldots, \{a_k, b_k, q_k\})$ with k is the number of lanes on the link and a_i, b_i, q_i are the number of *queuing* vehicles, number of *moving* vehicles and the position of the queue on lane i respectively.



(a) For configuration (3,2,3,2) (b) For configuration (2,1,2,1)

Fig. 5 Fundamental diagrams obtained from MicroSim's validation



(a) (b)

Fig. 6 Examples of vehicle distribution

2. Vehicle Flows

The traffic flows between consecutive links, together with *densities*, is at the heart of the model. For a particular link, traffic flows are categorized into two types: inbound flows and outbound flows. The inbound flows and outbound flows, as the names suggest, are about the numbers of vehicles joining and exiting a link respectively.

B. MacroSim Architecture

In this section, we present the architecture of the macroscopic simulator. Before going into details, we start with the definitions of some relevant terminologies.

1. Terminologies

a. Traffic Light Event

The macroscopic models share the same traffic light regulation model as described in the Microscopic Simulator section. A traffic light event is defined as the point in time that the traffic light controller at a node (intersection) switches from one phase to another (which causes traffic light status changed on one or more of the associated links, which has this intersection as their ending point). From the perspective of a link, traffic light events are categorized as No-Effect if the event doesn't make any status change to the traffic light at the end of the link or Green-to-Red or Red-to-Green events depending on the status change it makes.

b. Sub-network

In the macroscopic models, a traffic network is viewed as a collection of inter-related sub-networks, each contains a link, called upstream link, and a number of links, called downstream links, to which vehicles from the upstream link can be flowed. In other words, a sub-network consists of a set of links in which there is one and only one link having its ending point as the starting points of all other links. Two sub-networks are considered related if a downstream link of a sub-network is the upstream link of another. Fig. 7 illustrates a simple traffic network consisting of 3 sub-networks – M1, M2 and M3. M1 and M2, M2 and M3 are related sub-networks.

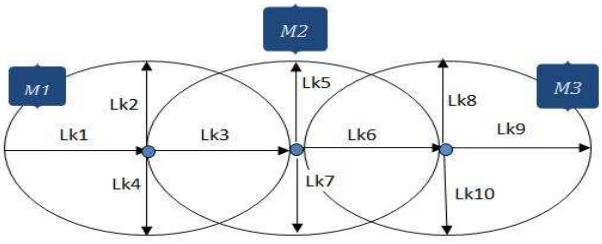


Fig. 7 An example of how a network is divided into sub-networks

c. Traffic Light Alignment

The term *traffic light alignment* is used to refer to the alignment between the traffic light at the end of the upstream link and all of the downstream links in a sub-network. For instance, during the Green phase of the upstream link's traffic light, the traffic light on one of the downstream link has 20 seconds of its Green phase and 30 seconds of its Red and buffered phases, the traffic light on another downstream link has 25 seconds of its Green phase and 20 seconds of Red and buffered phases and so on.

2. The Architecture

The MacroSim consists of different components which are integrated to produce simulation services. Fig. 8 presents the core components of this integration model.

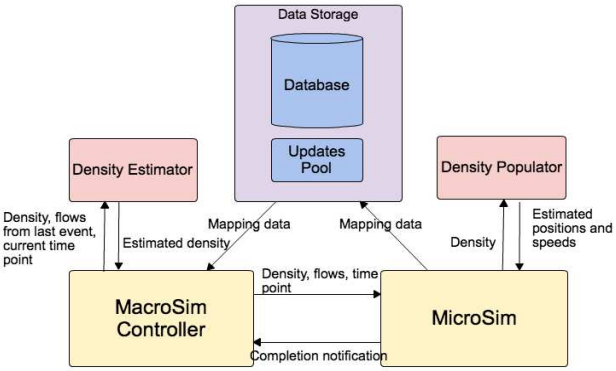


Fig. 8 Integration model

As mentioned before, MacroSim views a traffic network as a collection of inter-related sub-networks. In this integration model, each sub-network is handled by MicroSim to produce data (i.e. densities, flows) requested by the MacroSim controller. In other words, MicroSim plays the roles of a data generator and provider.In the following part, we describe each individual component of the model.

a. MacroSim Controller

MacroSim is an event-based simulator. The events here are the traffic light events as described earlier (in the rest of the paper, we use the term *event* to refer to traffic light event for simplicity). The aim of MacroSim is to keep track of the densities and flows on all links of a traffic network being simulated. The updates for each link (density and flows) are made whenever an event at its ending point is processed. MacroSim Controller (MacroSimController) is the heart of MacroSim, with the roles of processing events and interacting with other components to produce simulation results. MacroSimController maintains a nonempty event pool and executes the events one by one. When an event occurs, the following tasks are conducted to get the updates before the next event is processed:

- MacroSim Controller checks if there is any link affected by the event. If there is not, it ignores the event and processes the next one.
- For each of the affected link. There are two situations:

– If the event is Red-to-Green, MacroSimController invokes

the Density Estimator to get the density update of the link (in this case, the outbound flows are not a concern because it is certainlythatthe outbound flows are 0 during the last period when the link is in Red light phase). If the link has some inbound flows or/and a non-zero density, then MacroSimController checks in the database if it contains any record for this case, if it does MacroSimController retrieves the results and places them into the Updates Pool (database and Updates Pool will be described in more details in later parts), if there is no record, a MicroSim instance is invoked to simulate the associated sub-network (the sub-network which has this link as its upstream link).

- If the event is Green-to-Red, MacroSimController checks if there is a MicroSim instance invoked to produce results for this link (at the last Red-to-Green event). If the MicroSim instance is running, MacroSimController suspends at this event until the results are returned from it. If the MicroSim instance has returned the results, MacroSimController looks for them in Updates Pool. If there is no such MicroSim instance, MacroSimController verifies if the update data was retrieved from the database and stored in Updates Pool. If it is the case, MacroSimController gets the data from Updates Pool. Otherwise, MacroSimController updates the density and flows with the old values (as no MicroSim invoked for this link means that it has no inbound flows and empty. Thus, there is no change on this link).

b. MicroSim

In this integration model, MicroSim is used only on sub-networks to produce simulation results (that's why the problem regarding high computational resource consumption of car-following approach can be overcome). As a traffic network consists of multiple sub-networks, it would be the case that multiple MicroSim instances are invoked and running at the same time. To minimize the computational costs, MicroSim is only called at Red-to-Green events. In other words, MicroSim is only invoked to simulate the traffic during the Green phases of a link's traffic light. For Green-to-Red event of a link, as it is certainly that there is no outbound flow during Red phases of traffic light cycles, the updates of density is done by the Density Estimator.

When invoked, the MicroSim instance is provided by MacroSimController with the geometric information (links' lengths, number of lanes, speed limits…), traffic light data (Red, Green phases' lengths…), current density of each link in the sub-network, the inbound flows of the upstream link and the current time point of the event. Based on the data received, MicroSim constructs the sub-network (links, nodes and traffic lights) for simulation. If the links are non-empty, MicroSim calls up the Density Populator, which is responsible for generating position and speed of each individual vehicle given the density provided, to populate vehicles into the links.

When the MicroSim instance stops (at the end of the Green phase of the upstream link), it stores the result of simulation into the database in the form of the mapping between original conditions (including original densities, flows, traffic light alignment) and the resultant conditions (density of the upstream link and outbound flows from the upstream link into each downstream link). The MicroSim instance sends the completion notification to MacroSimController to notify that the results have been generated. If the MacroSimController is waiting for the results from this MicroSim instance, then the results are sent directly to MacroSimController, otherwise a copy of the results is stored in the Updates Pool, which is a temporary storage of updates, for the MacroSimController to retrieve at a later time.

c. Data Storages

This component includes 2 elements: the database and the Updates Pool. The difference between them is that, database is a permanent storage of the mappings between the original conditions and the resultant conditions of a particular link while Updates Pool is the temporary storage of updates information which is generated by MicroSim instances. The data in Updates Pool is only used at runtime and is cleared immediately when the MacroSimController retrieves the updates. The purpose of Updates Pool is to minimize unnecessary database accesses, which could increase processing cost and slow down the performance.

The implementation of the database conveys the idea of reusing previous simulation results that we propose. If we run the simulation of a certain traffic network for a substantially long period of time, we would get a large number of mappings in the database. Then at a later time we need to simulate the same network with different traffic demands, the existing mapping data help us minimize the number of calls to MicroSim (as described in the MacroSimController section, whenever the match is found between an entry in the database with the current conditions of the links, the results from the database are retrieved instead of invoking a MicroSim instance). The more number of mappings stored in the database, the higher probability that we can minimize the number of calls to MicroSim, the faster MacroSim is (in the sense that, MacroSim is faster if it can produce the results which require MicroSim to take longer to generate).

d. Density Populator

Density Populator plays the important roleof an interface between MacroSim and MicroSim. It is responsible for transforming the aggregated information provided by MacroSim into "microscopic" information required for MicroSim to operate. Specifically, Density Populator produces the estimation about the position and speed of each vehicle given the density of a link. It is called whenever a MicroSim instance is invoked. The MicroSim instance will send the densities of links in its sub-network to Density Populator and receive back the speeds and positions of vehicles on each link. Then it generates vehicles on the sub-network according to the data and starts the simulation.

e. Density Estimator

As mentioned earlier, the Density Estimator component is used to calculate the updates of a link's density in cases when

MicroSim hasn't been invoked. The inputs into this component include the Red phase' duration, the original density, the inbound flows (outbound flows are not matters as no vehicle can exit the link during Red phases), and the traffic light alignment. These parameters are provided by MacroSimController whenever Density Estimator is consulted. The output of Density Estimator is the link's density (*a*, *b* and *q* on each individual lane).

C. Validation Results and Discussions

We validated MacroSim by matching the results it produces with the results coming from MicroSim, whose correctness has been proven. The validation is conducted by runningthe two simulators concurrently on the same network and comparing the two sets of results. Since the concerns of the MacroSim are about densities, flows and travel time, we collected those data on two simulators and did the analysis on the variances between them.

We choose Melbourne CBD as the scenario for this validation. The network is presented in Fig. 9.

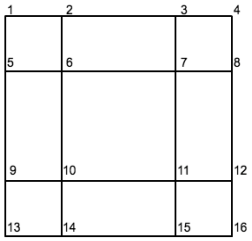


Fig. 9 Scenario for the validation

The settings of the network are as follows:

- Each short link is 250 meters long.
- Each longer link is 500 meters long.
- Every line represents two opposite links; each has a 40km/h speed limit.
- Every link whose ending point is 1, 6, 11 or 16 has 59 seconds for both Green and Red phase and 1 second for each buffered phase (total cycle length is 120 seconds).
- Every link whose ending point is 2, 7, 12, 5, 10 or 15 has 49 seconds for both Green and Red phases and 1 second for each buffered phase (total cycle length is 100 seconds).
- Every link whose ending point is 3, 8, 9 or 14 has 44 seconds for both Green and Red phase and 1 second for each buffered phase (total cycle length is 90 seconds).
- Every link whose ending point is 4 or13 has 39 seconds for both Green and Red phase and 1 second for each buffered phase (total cycle length is 80 seconds).
- Two OD-pairs are: 1 and 16, 4 and 13, each with the travel demand is 50 vehicles/ minute.

For the purpose of this validation, in both simulators, we record the density (including *a* and *b*) of each link at every event of that link (Green-to-Red and Red-to-Green). We also record the total outbound flows at Red-to-Green events (the outbound flows at Green-to-Red events are always 0 as no vehicle exits during Red phases).

Due to space limitation, we only choose to present some typical results of the validation. For the data produced on each link, the differences between densities and outbound flows at the same events (from two simulators) are taken into account. For each route, the comparison is on the average travel times.

The following tables present the analysis on the results. The comparisons are shown for the first 4 hours of the simulation and we assume the simulation time starts from 0:00.

1. Links

For each one-hour period, we show the frequency of the variances encountered when comparing *a*, *b*, *a*+*b* or outbound flows in specific ranges (i.e. how many percent of the variances fall into the range 0%-10% or 10%-20%). We later compare the average values of *a*+*b* and outbound flow over the same period.Tables IV and V show the validation results on two selected links. Other results are in a very similar pattern. In general, 70% of the variances encountered when comparing *a*, *b* or outbound flows are from 0% to 20%. The variances of *a*+*b* are usually small, from 0% to 10%. That means the densities produced by MacroSim are close to that of MicroSim even when there are big variances with *a* and*b*(individually)since they actually compensate to each other.

The variances of the averages of densities and outbound flows over each period are below 15% in most cases, which means the two result sets are close in overall.

2. Routes

The differences between results of travel time collected from the two simulators are not considerable. Table VI shows the comparisons between travel times collected from MacroSim and MicroSim on 4 selected routes in each period. In most cases (the routes shown in the table and others which are not presented), the variances are below 15%.

In Table VI:

- Route 1 is made up of the nodes: 1-2-3-4-8-12-16
- Route 2 is made up of the nodes: 1-5-9-13-14-15-16
- Route 3 is made up of the nodes: 4-3-2-1-5-9-13
- Route 4 is made up of the nodes: 4-8-12-16-15-14-13

With the concerns on densities, flows and travel time, the collected validation results, in general, shown that MacroSim produces results which are not much different from MicroSim. The big variances we got with *a* and*b* come from some simplifications made in the models of Density Populator and Density Estimator. However, the results show that although the differences are big if we compare the values of *a* and*b* individually, the differences between densities (*a*+*b*) produced by the two simulators, which is the main concerns, are not considerable. In addition, the results also show that outbound flows and travel time data collected from the MacroSim are reliable with the precision of 15%.

V. Conclusion And Future Work

In this paper, we have proposed a traffic simulation package, TDMSim, which contains the macroscopic simulator (MacroSim) and a microscopic simulator (MicroSim). Both of them support the simulation on free-flowing traffic as well as regulated traffic systems. MicroSim supports the simulation of traffic light systems, vehicle behaviors, road capacity…In

addition, it allows the simulation of route changing of vehicles, e.g. in cases of incidents, byfacilitating the communications between vehicles and the central system. MacroSim implements our new approach of integrating macroscopic and microscopic models to take advantages and minimize the disadvantages of existing models. The proposed macroscopic simulator is also able to reuse the existing simulation results for later running times with the help of a database.

TABLE IV
VALIDATION RESULTS ON LINK 1-2

Period	Range (%)	*a* *(%)*	*b* *(%)*	*a+b* *(%)*	*Outbound flows (%)*	Avg *a+b (%)*	*Average Outbound flow (%)*
0:00 – 1:00	***0 - 10***	**36**	**38**	**69**	**63**	**14**	**15**
	10 - 20	**42**	**35**	**21**	**24**		
	20 - 30	18	21	8	11		
	30 - 40	4	6		2		
1:00 – 2:00	***0 - 10***	**31**	**33**	**69**	**68**	**10**	**12**
	10 - 20	**38**	**38**	**26**	**25**		
	20 - 30	25	24	5	7		
	30 - 40	6	5				
2:00 – 3:00	***0 - 10***	**35**	**30**	**71**	**75**	**12**	**12**
	10 - 20	**43**	**42**	**27**	**16**		
	20 - 30	19	23	2	9		

TABLE V
VALIDATION RESULTS ON LINK 14-15

Period	Range (%)	*a* *(%)*	*b* *(%)*	*a+b* *(%)*	*Outbound flows (%)*	Avg *a+b (%)*	*Average Outbound flow (%)*
0:00 – 1:00	***0 - 10***	**39**	**33**	**65**	**62**	**15**	**11**
	10 - 20	**44**	**39**	**26**	**24**		
	20 - 30	17	23	7	14		
	30 - 40		6	2			
1:00 – 2:00	***0 - 10***	**34**	**33**	**68**	**68**	**13**	**14**
	10 - 20	**45**	**42**	**23**	**23**		
	20 - 30	18	19	8	9		
	30 - 40	3	4	1			
2:00 – 3:00	***0 - 10***	**45**	**40**	**74**	**72**	**13**	**9**
	10 - 20	**38**	**43**	**22**	**21**		
	20 - 30	17	15	4	7		

TABLE VI
TRAVEL TIME VALIDATION RESULTS

Period	Route 1	Route 2	Route 3	Route 4
0:00 – 1:00	14%	16%	12%	10%
1:00 – 2:00	10%	14%	8%	11%
2:00 – 3:00	15%	16%	10%	9%
3:00 – 4:00	12%	11%	10%	12%

The preliminary validation results for simple networks with low and medium traffic are promising. However we encountered problems in situations with heavy traffic, where we got big variances with densities and flows when comparing results produced by MicroSim and MacroSim. The problem comes from the fact that we calculate the inbound flow into a link by taking into account the outbound flows of each of its upstream links, which are resulted from applying MicroSim on related sub-networks in the previous events. However, in heavy traffic situations, the actual inbound flow into the link could be a very different value. Therefore, that affects the results on that link and also the other subsequence links. We consider solving this problem as one of our future works. In addition, more complicated scenarios will be involved in the validation process. Furthermore, the models of density population and density estimator will also be reviewed and improved in order to reduce the variances of *a* and *b*. Finally, we would like to conduct an experiment on how efficient the database of simulation results is in helping reducing the number of calls to MicroSim and fastening the simulator.

REFERENCES

[1] J. Miller, and E. Horowitz, "FreeSim – A Free Real-Time Freeway Traffic Simulator", Proc. IEEE Conference on Intelligent Transportation Systems, ITSC, pp. 18-23, 2007.
[2] Owen, E. Larry, Y. Zhang, L. Rao, and G. McHale, "Traffic Flow Simulation Using CORSIM." Proceedings of the 2000 Winter Simulation Conference, December 10-13, 2000.
[3] Q. Yang, and H. N. Koutsopoulos, "A Microscopic Traffic Simulator for Evaluation of Dynamic Traffic Management Systems." Transpn Res., Part C, Volume 4, Number 3, 1996.
[4] Cameron, Gordon, J.N. Wylie, and D. McArthur, "Paramics: Moving Vehicles on the Connection Machine." Proceedings of the 1994 ACM/IEEE Conference on Supercomputing, November 14-19, 1994.
[5] Wang, Yibing, MarkosPapageorgiou, Albert Messmer,"RENAISSANCE: A Real-Time Freeway Network Traffic Surveillance Tool." IEEE 9th International Intelligent Transportation Systems Conference, September 17-20, 2006.
[6] Redmill, A. Keith, and U.Ozguner, "VATSIM: A Vehicle and Traffic Simulator." IEEE 2nd Inetrnational Intelligent Transportation Systems Conference, October 5-8, 2001.
[7] Fellendorf, Martin, "VISSIM: A Microscopic Simulation Tool to Evaluate Actuated Signal Control Including Bus Priority." 64th Institute of Transportation Engineers Annual Meeting, October 1994.
[8] Ch. L. Barrett, S. Eubank, K. Nagel, S. Rasmussen, J. Riordan, and M. Wolinsky, "Issues in the Representation of Traffic Using Multi-Resolution Cellular Automata", Los Alamos National Laboratory Technical Report: LA/UR 95-2658.
[9] P. Wagner, "Traffic Simulations Using Cellular Automata: Comparison with Reality", Traffic and Granular Flow, 1996.
[10] J. Esser, M. Schreckenberg, "Microscopic simulation of urban traffic based on cellular automata", International Journal of Modern Physics C, vol. 8, no. 5, pp. 1025–1036, 1997.
[11] J. Maroto, E. Delso, J. Flez, and J. M, Cabanellas, "Real-time traffic simulation with a microscopic model", IEEE Transactions on Intelligent Transportation Systems, vol. 7, no. 4, pp. 513-527, 2006.
[12] Q. Yang, and H. N. Koutsopoulos, "A microscopic traffic simulator for evaluation of dynamic traffic management systems", Transp. Res. Part C, vol. 4, no. 3, pp. 113–129, 1996.
[13] G. I. Duncan, "PARAMICS wide area microscopic simulation of ATT and traffic management", Proc. 28th ISATA Conf., Stuttgart, Germany, pp. 475–484, 1995.
[14] J. Barcelo, J. F. Ferrer, D. Garcia, M. Florian, and L. E.Saux, "The parallelization of AIMSUN2 microscopic simulator for ITS applications", Proc. 3rd World Congr. Intell. Transp. Syst., Orlando, FL, 1996.
[15] J. Li, andY. Xu, "Dependence patterns associated with the fundamental diagram: a copula function approach", Journal of Zhejiang University-Science A, vol. 11, no. 1, pp. 18-24, 2010.
[16] H. Greenberg, "An analysis of traffic flow". Operations Research, vol. 7, pp. 79–85, 1959.
[17] G. F. Newell, "Instability in Dense Highway Traffic", Proceedings of 2nd International Symposium on the Theory of Road Traffic Flow, pp. 73-83, 1965.