Int. J. Simul. Multisci. Des. Optim. 2016, 7, A9

DOI: 10.1051/smdo/2016008

IJSMDO

Available online at:
www.ijsmdo.org

RESEARCH ARTICLE OPEN ACCESS

Road traffic management based on self-load-balancing approach

Ahmed Adnane[1], Mohamed Sadik[1], Saida Talal[1], Hicham Medromi[1], and David Bassir[2,3,*]

[1] Systems Architecture Team, Laboratory of Computer System and Renewable Energy, University Hassan II Casablanca – ENSEM, Casablanca, Morocco
[2] CMLA, CNRS (UMR 8536) – ENS Cachan, France
[3] Institute of Industry Technology, Guangzhou & Chinese Academy of Sciences, China

Received 22 February 2016 / Accepted 27 October 2016

Abstract – Traffic congestion is one of the most challenging problems for nowadays cities. Several contributions mainly based on V2V (Vehicle-to-Vehicle) communication have been published, but most of them have never been applied due to their communication related problems and costs. In this article, a novel cost-effective approach is introduced inspired by social life of insects where direct (V2V) communication does not exist anymore. Vehicles are equipped with devices that perform simple tasks, but their interactions with the environment through RSUs (Road Side Units) allow the creation of an intelligence which notifies drivers about congested road segments to avoid them. We call this emerging behavior self-load balancing. Description of the fundamentals of this approach and its performance are detailed in this work.

Key words: Congestion detection, Traffic management, Traffic data optimization, Emergent intelligence, Density estimation, Collective intelligence.

1 Introduction

Nowadays, congestions have become a daily faced problem which drastically decreases the citizens' quality of life. It has many negative consequences [1] such as longer travel times, lower security level [2] and increased fuel consumption. Choosing a less congested route, even longer, has become an omnipresent concern for each one of us.

In recent years, various researches have been conducted on the issue of density level evaluation. Most of these studies rely on VANET (Vehicular Ad-hoc NETwork) approaches which include two types of communication: V2I (Vehicle-To-Infrastructure) communication and V2V (Vehicle-To-Vehicle) communication. Vehicles and infrastructure units are equipped with devices capable of short-range wireless connectivity. These devices can form a particular mobile ad-hoc network: VANET. Although borrowing WMSNs (Wireless Mobile Sensor Networks) techniques to VANETs, due to their similarities, seems to be attractive, VANETs still suffer from many network issues such as bandwidth saturation and redundant data [3], late and wrong density level evaluation [4], and reliability problems [5].

In this work, a novel cost-effective load balancing approach is presented which aims to distribute vehicles equitably over a road network though notifying drivers about density levels of road segments. In such an approach, V2V communication does not exist anymore, which helps saving investment cost on the one hand, and avoids network related problems on the other hand.

Getting inspired by swarm intelligence, where agents communicate and collaborate through the environment, we developed a model of indirect communication between vehicles. In fact, vehicles gather and share traffic reports through RSUs (Roadside Units), the intelligence which emerges from such simple interactions allows vehicles to draw a traffic map converging to the real-time traffic map of the road network.

Unlike similar contributions in this filed, drivers do not intervene in determining the density levels of visited road segments, otherwise, the proposed model is fully automated and density levels are estimated based on both sensors data and mathematical approximations.

Finally, it should be noted that this work is an extension to our previous work [6] which exposes theoretical background behind our approach. In this article we recall the fundamentals of the approach and we provide proof of concept and performance evaluation. In fact, we have conceived a road traffic simulator based on Repast toolkit [7]. The proposed approach is then simulated and evaluated as described in Section 9.

*e-mail: david.bassir@gmail.com

2 Related work

Interaction between vehicles in terms of communication has been studied by many researchers in order to reduce traffic density. In [8], authors introduced a concept called "platoon" where vehicles have sufficient intelligence to keep a minimum safe distance to the vehicle in front. The advantage of this approach is that the use of the road is improved, however, this approach cannot prevent or unveil high density level segments over the whole network.

Authors in the following reference [9] have conceived a navigation aid algorithm. Vehicles autonomously optimize their paths to their destinations based on traffic information provided by RSUs. Authors in [1] discuss the vehicle scheduling problem, they try to schedule vehicles trips so that the average travel time is reduced. However, both approaches are not really adapted to high traffic density environment.

In [2], cognitive management functionalities based on V2V communication using WMSNs mechanisms was introduced by the authors. The aim was to issue orientation to drivers as well as to the overall transportation infrastructure to prevent high density situations. As mentioned in the introduction, V2V communication suffers from many network problems such as bandwidth saturation and redundant data problem. In addition to the above difficulties short-range wireless communication technologies make it impossible to detect real-time density levels of remote segments.

It is important to underline that in this reference [10] authors evaluate the COperative Traffic congestion detECtion approach (COTEC) which is based on V2V communication. Although results are interesting in case of high rates of V2V penetration, COTEC suffers from the same V2V communication drawbacks cited above.

3 Vehicle device

The vehicle device is an embedded system connected to four ultrasonic sensors as depicted in Figure 1. The vehicle device gathers traffic reports about the road network and notifies the driver about the high density roads to avoid them.

The ultrasonic sensors allow to automatically calculate the density levels of visited road segments (see Sect. 4). The basic function of an ultrasonic sensor is to measure the distance to the obstacle in front of it, but this information, delivered by the four side sensors, is exploited by the vehicle device to estimate the density level of the visited road segments.

Each vehicle device has two main sources of information: (1) its own knowledge base built during exploration of the network employing the ultrasonic sensors and (2) the knowledge bases of other vehicles shared by visited roadside units (see Sect. 5).

4 Density level estimation

One of the main challenging problems regarding traffic management solutions is to determine the density level of the network roads. Therefore, many solutions prefer assigning this task to the human being intelligence, i.e. the driver. In this work, we propose an automated way to estimate the density levels without the intervention of the driver.

Figure 1. Vehicle's ultrasonic sensors.

Ultrasonic sensors are attributed weights depending on their installation location, and thus on their influence in estimating the density levels. For example, the front side sensor unveils congestion situations better than the back side one. So, it is attributed a weight higher than the back one. For more examples, refer to our previous work [6].

Distances returned by ultrasonic sensors are passed to a function to calculate the average distance to encountered obstacles during a period of time called DT (Damping Time). The density level estimation formula is defined as follows:

$$\mathrm{DL}(t) = \frac{1}{\sum_{i=0}^{n} \propto_i} \sum_{i=0}^{n} \propto_i d_{DT}(t,i) \qquad (1)$$

Where $d_{DT}(t, i)$ represents the average distance returned by the ultrasonic sensor US_i during last DT seconds and $\propto_i$ represents the weight attributed to US_i. Note that d_{DT} function has an upper limit $d_{\max}$ and a lower limit $d_{\min} > 0$. The d_{DT} parameters are defined with regard to road network category: highway scenarios versus downtown scenarios.

In the remainder of this article, a location is considered to be relatively congested if the density level exceeds a predefined threshold called DLT (Density Level Threshold).

5 Roadside units (RSUs)

RSUs are computing devices located on the roadside, such as traffic signal controllers, which provide connectivity support to passing vehicles as depicted in Figure 2. They serve as a communication medium to vehicle devices. Direct communication between vehicles has been discarded due to technical and cost related problems.

Figure 2. Vehicle/RSU connection.

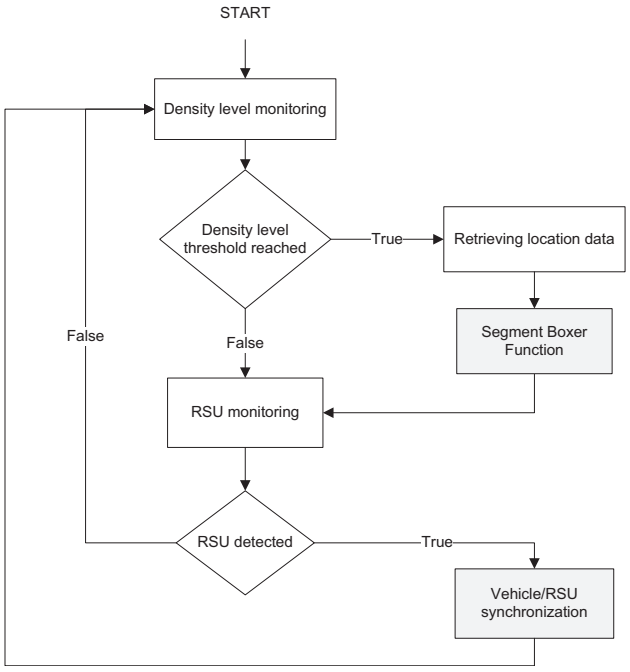

Figure 3. Vehicle density manager flowchart.

This work has been inspired by the swarm intelligence where collaboration between agents is a result of multiple interactions with the environments. The whole behavior of the system is seen as an emergent intelligence which has not been planned at the beginning by any agent. As a result, vehicles of the road network are notified about the real-time density map. Therefore, prioritizing low density roads drastically reduces the average density level of the studied area.

6 Vehicle device algorithm

The vehicle device performs operations regarding three main functions [6]: (1) gathering traffic reports about high density level road segments, (2) exchanging its knowledge base with other vehicles through RSUs (emerging intelligence) and (3) notifying the driver about the less congested roads to its destination; see Figure 3.

When the density level of the vehicle's location exceeds the DLT, the VSB (Vehicle Segment Boxer) function (see Algorithm 1) consists in retrieving the current GPS position of the vehicle, generating a polygon box of the congested road segment and updating the VKB (Vehicle Knowledge Base). In this way, the VKB is kept updated about the vehicle experience; and each other vehicle within the road network of the studied area gets notified about the event. Therefore the system tends to guide other vehicles to less congested roads.

```
1  Let VKB be the vehicle knowledge base, and p be the
   current position of the vehicle
2  ibox ← Generate initial box (polygon) including p
3  for each vbox in the VKB
4  |  if vbox is obsolete
5  |  |  delete vbox from VKB
6  |  else
7  |  |  if vbox and ibox can be merged together
8  |  |  |  Update the VKB by replacing vbox with the
   |  |  |  merging result
9  |  |  |  Break the current loop
10 |  |  end if
11 |  end if
12 end for
13 if no merging possibilities for ibox
14 |  update the VKB by adding ibox
15 end if
```

Algorithm 1. VSB.

One of the main challenging points with regard to road traffic management solutions is redundant data. In fact, one same piece of information can be gathered by the same vehicle many times as it moves multiple times in the same congested road segment, or gathered by other vehicles in the same location. This challenging issue may rapidly cause bandwidth saturation if it is not carefully investigated.

In this work, we have come up with a new principle in this field which helps managing redundant data in an efficient way. It is what we called the merging technique. Initially, in a high density level place, the gathered location is not stored as a GPS coordinates, instead, a rectangular polygon, i.e. a box, surrounding the location is generated and stored as illustrated in Algorithm 1. The main benefit behind is to reduce the number of stored locations as one box may contains infinite locations. So storing boxes is much more space saving than storing locations.

Boxes are stored in the VKB; they are updated as the vehicle moves through new places. The merging technique consists in updating existing boxes by merging them with new ones. So, instead of creating and storing a new box, an

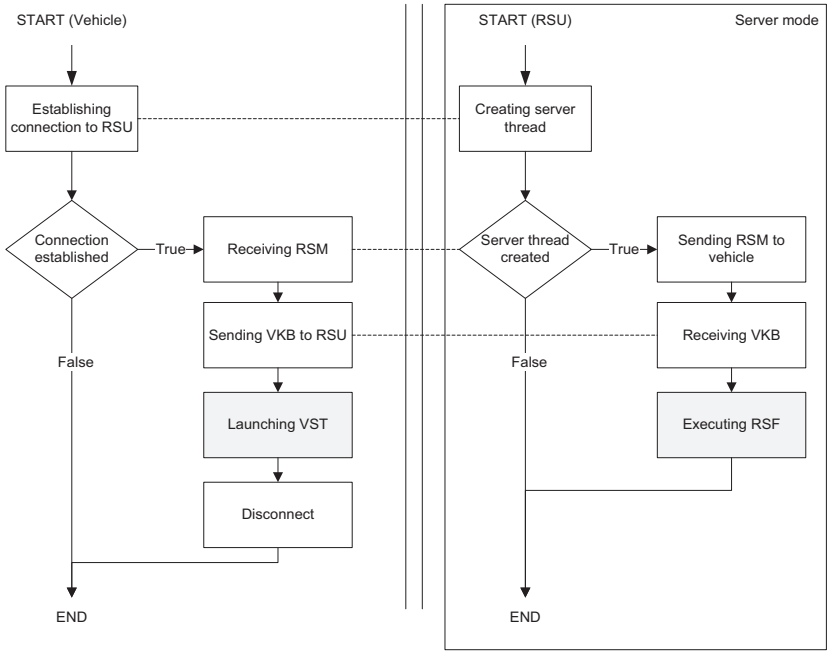

Figure 4. Vehicle/RSU synchronization flowchart.

existing one in the same vicinity may grow up to cover the new one and so on. Therefore, through multiple merges, we may have a box covering a whole boulevard with detailed information about its density levels.

This model helps managing gathered geographical data in an efficient way which saves both storage space and processing time.

While moving within the transmission range of a RSU, a vehicle establishes a connection to it and synchronizes its VKB with the knowledge base of the RSU [6], see Figure 4.

The RSM (RSU Shared Memory) and the VKB (Vehicle KB) are databases with the same structures. Each RSU within the studied area has its own RSM as each vehicle device has its own VKB. A VKB summarizes the experience and learning of a vehicle. As it moves, it collects data about density levels of visited road segments. When it is in the vicinity of a RSU, it shares with it its VKB as it retrieves in turn its RSM. This work is done by launching a synchronization process which consists in exchanging the databases then updating each one based on the received content in an asynchronous way. The goal behind the asynchronous model is to make the connection time between the vehicle device and the RSU as short as possible in order to exchange the maximum amount of data while being within the transmission range of the RSU (see Figure 4).

The updating mechanism consists for the vehicle device in running the VST (Vehicle Synchronization Thread) as described in Algorithm 2, and for the RSU in running the RSF (RSU Synchronization Function) as described in Algorithm 3. The VST runs in background with minimum priority.

```
1  Let VKB be the vehicle knowledge base, and RSM be
   the received RSU shared memory
2  Lock the VKB for writing
3  for each rbox in the RSM
4  |  for each vbox in the VKB
5  |  |  if rbox and vbox can be merged together
6  |  |  |  Update the VKB by replacing vbox with the
   |  |  |  merging result
7  |  |  |  Break the current loop
8  |  |  end if
9  |  end for
10 |  if no merging possibilities for rbox
11 |  |  update the VKB by adding rbox
12 |  end if
13 end for
14 Release the VKB lock
```

Algorithm 2 VST.

Before starting the update process, the VST locks the VKB for writing to ensure data integrity. The key point of the proposed algorithm is the merging technique through which information about new road segments are merged with existing ones helping to maintain small-sized and optimal database.

7 RSU algorithm

RSUs serve as servers to vehicles' devices. They allow vehicles to share and exchange traffic reports about different road segments of the network. When a vehicle encounters a RSU, the latter launches a server thread to fulfill its requests. The vehicle device transfers its KB to the RSU to be processed

and integrated into its shared memory as described in Algorithm 3. Thus, other vehicles in its vicinity will be notified about the vehicle experience with regard to density levels of visited road segments.

```
1  Let RSM be the RSU shared memory, and VKB be the
   received vehicle knowledge base
2  Lock the RSM for writing
3  for each vbox in the VKB
4  |  for each rbox in the RSM
5  |  |  if vbox and rbox can be merged together
6  |  |  |  Update the RSM by replacing rbox with the
   |  |  |  merging result
7  |  |  |  Break the current loop
8  |  |  end if
9  |  end for
10 |  if no merging possibilities for vbox
11 |  |  update the RSM by adding vbox
12 |  end if
13 end for
14  Release the RSM lock
```

Algorithm 3. RSF.

So as to ensure data integrity while concurrent accesses to the RSU, the RSM is locked for writing. Therefore, other vehicles can send request to the RSU without being able to change the content of its shared memory until the lock is released. The synchronization algorithm does its best to update the content of the RSM without allocating new storage space thanks to the merging technique. In this way, the RSM is kept small and optimal as much as possible.

If the number of simultaneously connected vehicles reaches the limit, the RSU blocks new connections until resources are released. If there are enough resources and no new connections are pending, the RSU launches the BMT (Box Merging Thread) in background as depicted in Figure 5.

The BMT maintains and optimizes the RSM by grouping boxes fulfilling merging conditions and deleting boxes which exceed obsolescence time threshold. Therefore, browsing and processing time of the RSM is enhanced during synchronization events.

8 Traffic simulator

To prove the efficiency of our model, we have conceived a traffic simulator based on Repast toolkit [7] and java programming as shown in Figure 6.

Repast Symphony [11] is a widely used, free and open source agent-based modeling environment that has collectively been under continuous development for over 14 years. Agent-based modeling provides a mechanism for modeling Complex Adaptive Systems (CASs) [12–15] in many disciplines such as economics, military and biology. Repast was originally developed by David Sallach, Nick Collier, Tom Howe, Michael North and others at the University of Chicago [15].

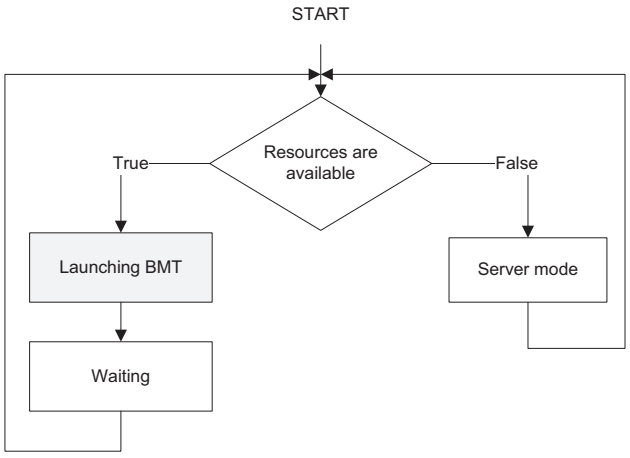

Figure 5. RSU density manager flowchart.

The simulator consists of a grid road network. Each intersection of the grid is fitted with traffic signals. Cars are initially generated at the bottom-left point of the grid and they are programmed to achieve the top-right corner using an available road segment. They move forward if they are not at a red light or behind another stopped car. All cars drive at a constant same speed. In intersections, cars can change direction if blocking cars are in front of them. If a car reaches its destination, it is dropped out of the simulation freeing space to others. Every path to the destination point is of equal distance.

9 Performance evaluation

The simulation consists in generating 100 cars at the bottom-left point of the grid and waiting until the last car achieves the top-right point of the grid. Since all paths are of equal distances and cars drive at the same constant speed, the key parameter to measure the efficiency of the traffic management is the average time taken by all cars to achieve their destination.

In the first scenario, the traffic management will be assured only by traffic signals which will be all synchronized: signal colors alternate from the N/S direction to the E/W direction but are the same in each direction (see Figure 7). Signal colors alternate each 30 ticks of the simulation.

After simulation, we notice that the network is not well exploited; cars are concentrated in the middle of the network which causes traffic congestions slowing down the overall speed. Figure 7 shows the state of the roads at tick 136.

The first simulation has required 430 ticks until the last car achieved its destination, which is far from being optimal.

The second simulation will run under the same conditions except for the fact that cars and traffic signals will implement our approach of self load-balancing. We define a segment as the road which links between two consecutive intersections. Each segment has its own coordinates (see Figure 8).

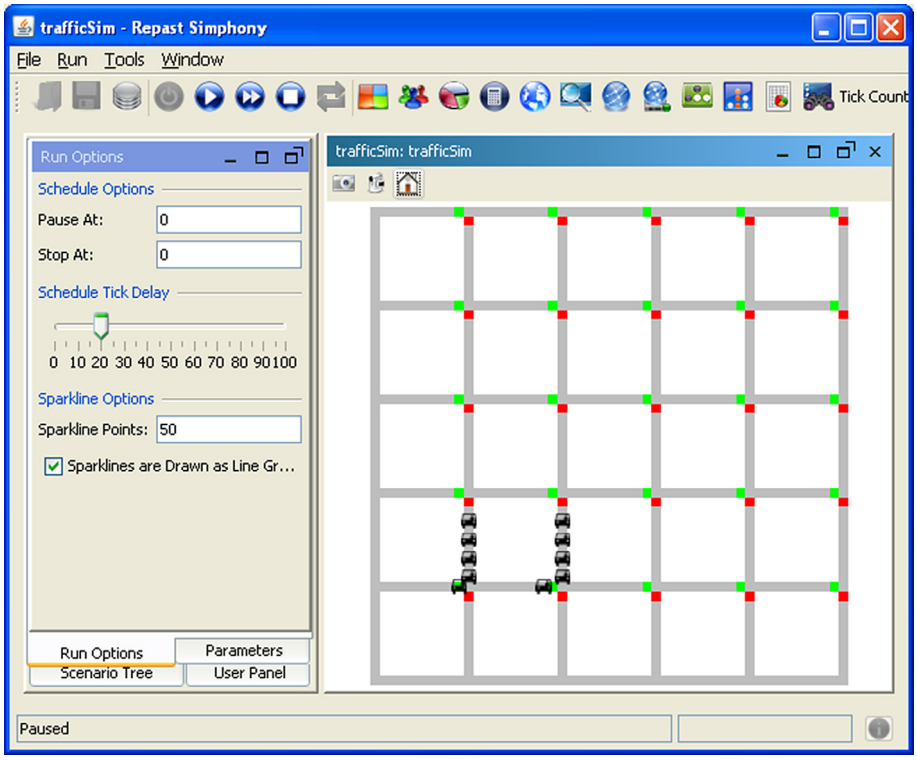

Figure 6. Repast traffic simulator.

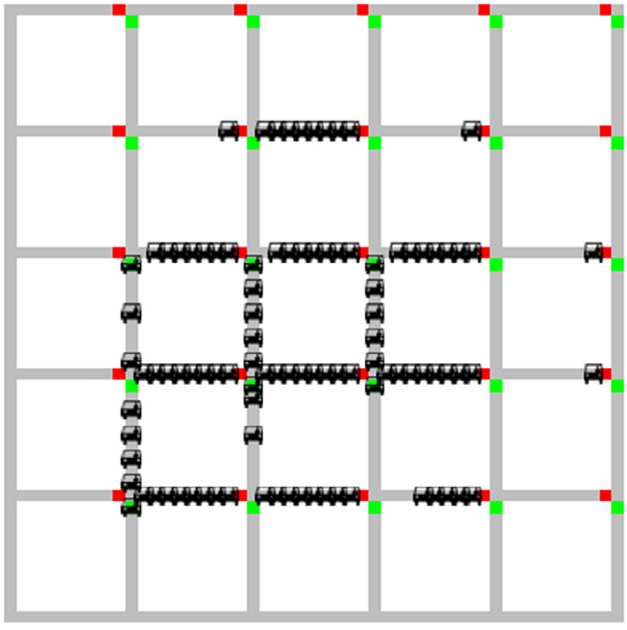

Figure 7. Road network at tick 136, traffic signals management.

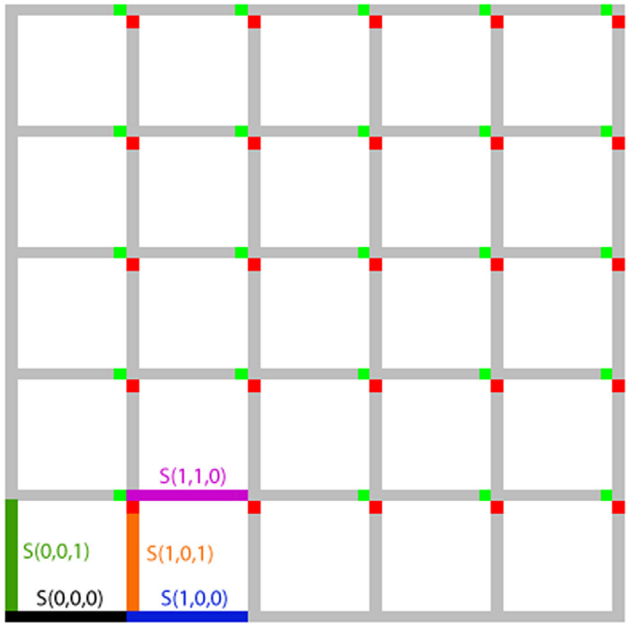

Figure 8. Marking grid segments.

Cars are fitted with front and back ultrasonic sensors. The car business logic consists in calculating and storing the density levels of visited roads. At intersections, cars exchange information about density levels with RSUs, i.e. traffic signals. In this way, cars get notified about segments of the network with high level density to avoid them.

We notice that cars exploit the road network more efficiently. Figure 9 highlights cars behavior at tick 136. When a road segment begins to get congested, it is reported to other cars through RSUs to avoid it. The simulation has lasted only 300 ticks until the last car achieves its destination.

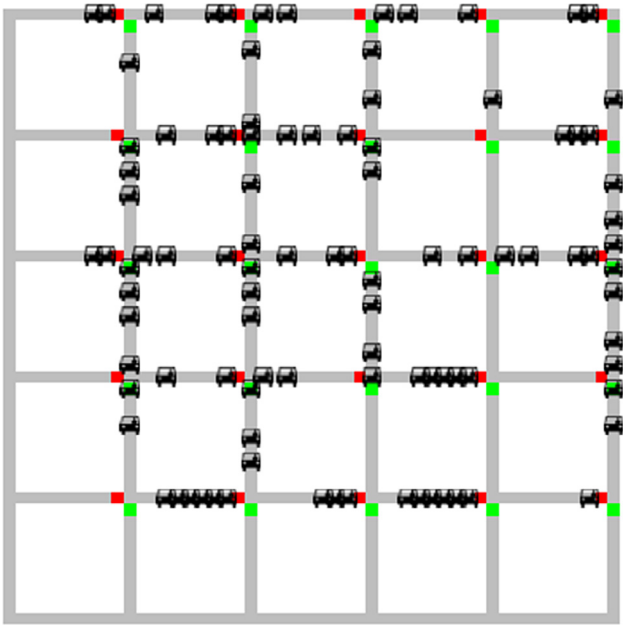

Figure 9. Road network at tick 136, self-load balancing management.

10 Conclusion

As result of this innovative approach, we have demonstrated the efficiency of the self-load-balancing approach applied to the traffic density problem. Indeed, through stigmergy concept, i.e. indirect communication, borrowed from social insects, vehicles load-balance themselves by notifying drivers about congested road segments to be avoided. With this idea the whole road network is efficiently exploited preventing high level density scenarios.

V2V approaches are impractical and challenging since they still suffer from many network problems with regard to the mobility of vehicles and the overall results are not that relevant, i.e. late and not accurate density evaluations.

This work proposes a new approach based on indirect communication between vehicles. In fact, RSUs serve as a medium through which vehicles exchange traffic reports about their experiences. Computational geometry algorithms for data storage and exchange are proposed and evaluated. These algorithms allow reducing processing time and thus enhancing the overall performance.

The load balancing algorithm is not explicitly programmed, but it emerges as a result of several interactions between vehicles and the environment, and this is what we described as emergent load balancing or self-load balancing.

The proposed approach is implemented and simulated under Repast toolkit. Results of the performance evaluation were very encouraging. Further work on the approach aiming to apply it in a real test field is under study.

References

1. Shah N, Kumar S, Bastani F, Yen I-L. 2012. Optimization models for assessing the peak capacity utilization of intelligent transportation systems. European Journal of Operational Research, 216(1), 239–251.
2. Dimitrakopoulos G, Demestichas P. 2010. Intelligent transportation systems. IEEE Vehicular Technology Magazine, 5(1), 77–84.
3. Schunemann B, Wedel J, Radusch I. 2010. V2x-based traffic congestion recognition and avoidance. Tamkang Journal of Science and Engineering, 13(1), 63–70.
4. Zhong T, Xu B, Wolfson O. 2008. Disseminating real-time traffic information in vehicular ad-hoc networks, in IEEE Intelligent Vehicles Symposium. p. 1056–1061.
5. Korkmaz G, Ekici E, Özgüner F, Özgüner Ü. 2004. Urban multi-hop broadcast protocol for inter-vehicle communication systems, in Proceedings of the 1st ACM International Workshop on Vehicular Ad Hoc Networks. p. 76–85
6. Adnane A, Lmimouni MS, Rezzai M, Medromi H. 2016. Traffic congestion manager, a cost-effective approach, Advances in Ubiquitous Networking, Proceedings of UNET'15, Lecture Notes in Electrical Engineering, Vol. 366, Springer. p. 237–248. DOI: 10.1007/978-981-287-990-5_19.
7. North MJ, Howe TR, Collier NT, Vos RJ. 2005. The Repast simphony runtime system, in Proceedings of the Agent 2005 Conference on Generative Social Processes, Models, and Mechanisms. p. 151–158.
8. Varaiya P. 1993. Smart cars on smart roads: problems of control. IEEE Transactions on Automatic Control, 38(2), 195–207.
9. Mamei M, Zambonelli F, Leonardi L. 2003. Distributed motion coordination with Co-fields: a case study in urban traffic management. IEEE Computer Society, 63–70.
10. Bauza R, Gozalvez J. 2013. Traffic congestion detection in large-scale scenarios using vehicle-to-vehicle communications. Journal of Network and Computer Applications, 36(5), 1295–1307.
11. North MJ, Collier NT, Vos RJ. 2006. Traffic reports creating three implementations of the repast agent modeling toolkit. ACM Transactions on Modelling and Computer Simulation, 16(1), 1–25.
12. Bonabeau E. 2002. Agent-based modeling: methods and techniques for simulating human systems, in Proceedings of the National Academy of Sciences 99 (3), National Academy of Sciences Press: Washington, DC, USA. p. 7280–7287.
13. Macal CM. 2009. Agent-based modeling and artificial life, in Encyclopedia of complexity and system science. Meyers RA, Editor. Springer: New York. p. 112–131. ISBN 978-0-387-75888-6.
14. Macal CM, North MJ. 2010. Tutorial on agent-based modeling and simulation. Journal of Simulation, 4, 151–162. DOI: 10.1057/jos.2010.3.
15. North MJ, Macal CM, St. Aubin J, Thimmapuram P, Bragen M, Hahn J, Karr J, Brigham N, Lacy ME, Hampton D. 2010. Multiscale agent-based consumer market modeling. Complexity, 15(5), 37–47. DOI: 10.1002/cplx.20304.

Cite this article as: Adnane A, Sadik M, Talal S, Medromi H & Bassir D: Road traffic management based on self-load-balancing approach. Int. J. Simul. Multisci. Des. Optim., 2016, **7**, A9.